Inverters



# 8400

E84AVTCxxxxx...

## Inverter Drives 8400 TopLine C _ _ _ _ _ _ _ _ _

Reference manual EN

13507302

Lenze

# Overview of technical documentation for Inverter Drives 8400

| Project planning, selection & ordering |
|---|
| ☐ 8400 hardware manual |
| 🖹 Catalogue |

| Mounting & wiring |
|---|
| 🖹 MA 8400 BaseLine/StateLine/HighLine/TopLine |
| 🖹 MA for the communication module |
| 🖹 MA for the extension module |
| 🖹 MA for the safety module |
| 🖹 MA for the accessories |

| Parameter setting |
|---|
| 🖹 BA keypad |
| ☐ SW 8400 BaseLine |
| ☐ SW 8400 StateLine |
| ☐ SW 8400 HighLine |
| ☐ SW 8400 TopLine |
| ☐ KHB for the communication module |

← **This documentation**

| Drive commissioning |
|---|
| ☐ SW 8400 BaseLine/StateLine/HighLine/TopLine |
| → chapter "Commissioning" |
| → chapter "Diagnostics & error management" |
| ☐ Remote maintenance manual |

| Networking |
|---|
| ☐ KHB for the communication medium used |

Legend:

- 🖹 Printed documentation
- ☐ Online documentation (PDF/Engineer online help)

Abbreviations used:

| | |
|---|---|
| BA | Operating instructions |
| KHB | Communication manual |
| MA | Mounting instructions |
| SW | Software/reference manual |

# Contents


**1 About this documentation** 25
1.1 Document history 25
1.2 Conventions used 27
1.3 Terminology used 28
1.4 Definition of the notes used 30

**2 Introduction: Parameterising the inverter** 31
2.1 Integrated technology applications 33
2.1.1 Purpose of the technology applications 34
2.1.2 Application cases for a technology application 34
2.1.3 Technology application = function block interconnection 35
2.2 Selection of the appropriate commissioning tool 36
2.2.1 Overview: Accessories for commissioning 37
2.3 General notes on parameters 38
2.3.1 Changing the parameterisation with the keypad 39
2.3.2 Change parameter settings with PC and Lenze software 42
2.3.3 Save parameter settings in the memory module safe against mains failure 43
2.3.4 User menu for quick access to frequently used parameters 45
2.4 Device access protection 46
2.4.1 Password protection 47
2.4.2 Individual password protection for single communication channels 49
2.4.3 Device personalisation 51
2.4.4 Unlocking the inverter with a MasterPin 53
2.5 Device identification 54
2.5.1 Automatic acceptance of the device name in the »Engineer« 54
2.5.2 Extended item designation 54

# Contents


**3 Commissioning** 55
3.1 Safety instructions with regard to commissioning 56
3.2 Notes on motor control 57
3.3 Preconditions for commissioning with the »Engineer« 58
3.4 Trouble-shooting during commissioning 59
3.5 Commissioning wizard 8400 59
3.6 Manual motor direction of rotation check (manual control) 61
3.7 Commissioning of the "Actuating drive speed" technology application 62
3.7.1 Prepare inverter for commissioning 63
3.7.2 Creating an »Engineer« project & going online 64
3.7.3 Parameterising the motor control 65
3.7.4 Parameterise application 66
3.7.5 Save parameter settings safe against mains failure 68
3.7.6 Enable inverter and test application 68
3.8 Commissioning of the "Table positioning" technology application 70
3.8.1 Prepare inverter for commissioning 72
3.8.2 Creating an »Engineer« project & going online 73
3.8.3 Parameterising the motor control 74
3.8.4 Parameterise application 75
3.8.4.1 Set machine parameters 76
3.8.4.2 Set the position encoder 77
3.8.4.3 Set homing 79
3.8.4.4 Enter one or more profiles 81
3.8.5 Save parameter settings safe against mains failure 83
3.8.6 Enable inverter and test application 83
3.9 Commissioning of the "Switch-off positioning" technology application 85
3.9.1 Prepare inverter for commissioning 87
3.9.2 Creating an »Engineer« project & going online 88
3.9.3 Parameterising the motor control 89
3.9.4 Parameterise application 90
3.9.5 Save parameter settings safe against mains failure 92
3.9.6 Enable inverter and test application 92
3.10 PC manual control 93
3.10.1 Activate PC manual control 93
3.10.2 Speed control 96
3.10.3 Set/reset home position 98
3.10.4 Manual jog 99
3.10.5 Positioning (relative or absolute) 100

# Contents


**4 Device control (DCTRL) 102**
4.1 Device commands (C00002/x) 104
4.1.1 Load Lenze setting 107
4.1.2 Load all parameter sets 108
4.1.3 Save all parameter sets 109
4.1.4 Enable/inhibit inverter 110
4.1.5 Activate/deactivate quick stop 111
4.1.5.1 Optional settings 112
4.1.5.2 Flow diagram 113
4.1.6 Reset error 114
4.1.7 Delete logbook 115
4.1.8 Device search function 115
4.2 Device state machine and device states 116
4.2.1 FirmwareUpdate 117
4.2.2 Init 118
4.2.3 Ident 119
4.2.4 SafeTorqueOff 120
4.2.5 ReadyToSwitchOn 121
4.2.6 SwitchedOn 122
4.2.7 OperationEnabled 123
4.2.8 TroubleQSP 124
4.2.9 Trouble 125
4.2.10 Fault 126
4.2.11 SystemFault 126
4.3 Automatic restart after mains connection/fault... 127
4.3.1 "Inhibit at power-on" auto-start option 127
4.3.2 Auto-start option "Inhibit at Lenze setting" 129
4.4 Internal interfaces | "LS_DriveInterface" system block 130
4.4.1 wCANControl/wMCIControl control words 133
4.4.2 wDeviceStatusWord status word 135

# Contents


**5 Motor control (MCTRL)** 138
5.1 Motor selection/Motor data 139
5.1.1 Selecting a motor from the motor catalogue in the »Engineer« 143
5.1.2 Automatic motor data identification 145
5.1.3 Application notes for asynchronous motors with high slip speed 152
5.2 Selecting the control mode 153
5.2.1 Selection help 157
5.3 Defining current and speed limits 158
5.4 V/f characteristic control (VFCplus) 161
5.4.1 Parameterisation dialog/signal flow 162
5.4.2 Basic settings 164
5.4.2.1 Defining the V/f characteristic shape 165
5.4.2.2 Defining current limits (Imax controller) 166
5.4.3 Optimising the control mode 167
5.4.3.1 Adapting the V/f base frequency 168
5.4.3.2 Adapting the Vmin boost 170
5.4.3.3 Optimising the Imax controller 172
5.4.3.4 Optimising the stalling behaviour 173
5.4.3.5 Torque limitation 174
5.4.3.6 Defining a user-defined V/f characteristic 176
5.4.4 Remedies for undesired drive behaviour 179
5.5 V/f characteristic control - energy-saving (VFCplusEco) 180
5.5.1 Parameterisation dialog/signal flow 181
5.5.2 Comparison of VFCplusEco - VFCplus 183
5.5.3 Basic settings 184
5.5.4 Optimising the control mode 185
5.5.4.1 Improving the behaviour at high dynamic load changes 186
5.5.4.2 Adapting the slope limitation for lowering the Eco function 187
5.5.4.3 Optimising the cos/phi controller 187
5.5.5 Remedies for undesired drive behaviour 188
5.6 V/f control (VFCplus + encoder) 190
5.6.1 Parameterisation dialog/signal flow 191
5.6.2 Basic settings 193
5.6.2.1 Parameterising the slip regulator 194
5.7 Sensorless vector control (SLVC) 197
5.7.1 Parameterisation dialog/signal flow 198
5.7.2 Types of control 200
5.7.2.1 Speed control with torque limitation 200
5.7.2.2 Torque control with speed limitation 201
5.7.3 Basic settings 202
5.7.4 Optimising the control mode 203
5.7.4.1 Optimising the starting performance after a controller enable 203
5.7.4.2 Optimise speed controller 204
5.7.4.3 Optimising dynamic performance and field weakening behaviour 205
5.7.4.4 Optimising the stalling behaviour 206
5.7.4.5 Optimise response to setpoint changes and determine mass inertia 207
5.7.4.6 Slip calculation from motor equivalent circuit diagram data 210
5.7.4.7 Optimising field feedforward control and torque feedforward control 211
5.7.5 Remedies for undesired drive behaviour 212
5.8 Sensorless control for synchronous motors (SLPSM) 213
5.8.1 Parameterisation dialog/signal flow 215
5.8.2 Types of control 219
5.8.3 Basic settings 220

# Contents


5.8.4 Optimising the control mode 222
5.8.4.1 Optimise current controller 223
5.8.4.2 Optimise speed controller 224
5.8.4.3 Optimise response to setpoint changes and determine mass inertia 227
5.9 Servo control (SC) 231
5.9.1 Parameterisation dialog/signal flow 232
5.9.2 Types of control 239
5.9.2.1 Speed control with torque limitation 239
5.9.2.2 Torque control with speed limitation 241
5.9.3 Basic settings 242
5.9.4 Optimising the control mode 243
5.9.4.1 Optimise current controller 244
5.9.4.2 Optimise speed controller 245
5.9.4.3 Optimise response to setpoint changes and determine mass inertia 248
5.9.4.4 Setting the current setpoint filter (band-stop filter) 251
5.9.4.5 Adapting the max. acceleration change (jerk limitation) 252
5.9.4.6 Slip calculation from motor equivalent circuit diagram data 253
5.9.4.7 Temperature compensation within the motor control 255
5.9.4.8 Optimising the behaviour of the asynchronous motor in the field weakening range 257
5.9.4.9 Feedforward control of the current controller 262
5.10 Parameterisable additional functions 263
5.10.1 Selection of switching frequency 263
5.10.2 Operation with increased rated power 267
5.10.3 Current-dependent stator leakage inductance Ppp(I) 269
5.10.4 Flying restart function 271
5.10.5 DC-injection braking 274
5.10.5.1 Manual DC-injection braking (DCB) 275
5.10.5.2 Automatic DC-injection braking (Auto-DCB) 275
5.10.6 Slip compensation 278
5.10.7 Oscillation damping 279
5.10.7.1 Oscillation damping voltage range 280
5.10.7.2 Oscillation damping in the field weakening range 281
5.10.8 Phase sequence reversal for correcting misconnected UVW motor phases 282
5.10.9 Field weakening for synchronous motors 283
5.11 Position control/additive speed specification 287
5.12 Braking operation/brake energy management 288
5.12.1 Setting the voltage source for braking operation 291
5.12.2 Selecting the response to an increase of the DC-bus voltage 291
5.12.2.1 Inverter motor brake 293
5.12.3 Avoiding thermal overload of the brake resistor 296
5.12.4 Control of multiple internal brake choppers in the DC-bus system 296
5.13 Monitoring 299
5.13.1 Device overload monitoring (Ixt) 300
5.13.2 Motor load monitoring (I2xt) 301
5.13.3 Motor overcurrent monitoring 303
5.13.4 Motor temperature monitoring (PTC) 304
5.13.5 Brake resistor monitoring (I2xt) 305
5.13.6 Motor phase failure monitoring 307
5.13.7 Motor phase error monitoring before operation 308
5.13.8 Mains phase failure monitoring 310
5.13.9 Current monitoring for overload 310
5.13.10 Maximum current monitoring 311
5.13.11 Maximum torque monitoring 311
5.13.12 Motor speed monitoring 311

# Contents


5.14 Internal interfaces | System block "LS_MotorInterface" 312
5.15 Internal status signals | System block "LS_DeviceMonitor" 318

**6 Encoder/feedback system** 321
6.1 Encoder selection 322
6.1.1 Signal flow - encoder interface 324
6.2 Resolver (X7) 326
6.2.1 Parameterising the resolver 326
6.2.2 Optimising resolver behaviour 328
6.2.3 Internal interfaces | "LS_Resolver" system block 331
6.3 Multi-Encoder (X8) 332
6.3.1 Incremental TTL and SinCos encoder 333
6.3.2 SinCos absolute value encoder with HIPERFACE® protocol 335
6.3.2.1 Initialisation 335
6.3.2.2 Parameterising the encoder 337
6.3.2.3 Display of the current position 340
6.3.2.4 Notes on evaluating absolute value encoder of SEL37 and SELxx types by SICK 341
6.3.2.5 Configuration example 1: Absolute wire draw encoder "XKS09-HTBM0527" by SICK 342
6.3.2.6 Configuration example 2: Distance sensor "DME5000-117" by Sick 343
6.3.3 SSI absolute value encoder with Stegmann-SSI protocol 344
6.3.3.1 Parameterising the encoder 345
6.3.3.2 Important notes for using an SSI laser position encoder 349
6.3.3.3 Configuration example 1: Optical multi-turn rotary encoder "G0M2H Z04" by Baumer 350
6.3.3.4 Configuration example 2: Distance sensor "DME5000-111" by Sick 351
6.3.4 Digital frequency coupling 353
6.3.4.1 Problem description - speed variations 355
6.3.4.2 Internal interfaces | System block "LS_DFOut" 356
6.3.5 Internal interfaces | "LS_MultiEncoder" system block 357
6.3.5.1 Scaling of position value 359
6.3.5.2 Setting of a static position offset 359
6.4 Frequency HTL encoder (DI1/DI2, DI6/DI7) 360
6.4.1 Parameterising digital inputs as encoder inputs 361
6.4.2 HTL encoder at DI1/DI2 363
6.4.3 HTL encoder at DI6/DI7 365
6.5 Pole position identification (PPI) 366
6.5.1 Pole position identification 360° 368
6.5.2 Pole position identification without motion 372
6.6 Mounting 375
6.7 Monitoring 376
6.7.1 Resolver/multi-encoder open-circuit monitoring 377
6.7.2 Open-circuit monitoring HTL encoder 378
6.7.3 Encoder angular drift monitoring 379
6.7.4 Encoder value monitoring 380
6.7.5 Motor temperature monitoring (KTY) 381

# Contents


**7 I/O terminals** 384
7.1 Digital input terminals 385
7.1.1 Change function assignment 387
7.1.1.1 Using DI1(6) and DI2(7) as digital inputs 388
7.1.1.2 Using DI1(6) and DI2(7) as frequency inputs 389
7.1.1.3 Using DI1(6) as counting input 394
7.1.2 Internal interfaces | System block "LS_DigitalInput" 398
7.1.2.1 Output of the encoder position of the DI1/DI2 frequency input 401
7.1.2.2 Switching status of the motor holding brake at the high current output 405
7.2 Digital output terminals 406
7.2.1 Internal interfaces | System block "LS_DigitalOutput" 408
7.3 Analog terminals 409
7.3.1 Parameterising analog input 412
7.3.1.1 Signal adaptation by means of characteristic 414
7.3.2 Parameterising analog output 416
7.3.3 Internal interfaces | System block "LS_AnalogInput" 417
7.3.4 Internal interfaces | System block "LS_AnalogOutput" 418
7.4 Touch probe detection 419
7.4.1 Parameter setting 421
7.4.1.1 Z-track encoder as a touch probe signal source 424
7.4.1.2 Z-track resolver as a touch probe signal source 424
7.4.2 Internal interfaces | System block "LS_TouchProbe" 425
7.4.2.1 Application example: "Position measurement" 427
7.5 Configuring exception handling of the output terminals 428
7.6 User-defined terminal assignment 429
7.6.1 Source-destination principle 430
7.6.2 Changing the terminal assignment with the keypad 431
7.6.3 Changing the terminal assignment with the »Engineer« 433

# Contents


**8 Technology applications** 436
8.1 Selection of the technology application and the control mode 437
8.2 TA "Actuating drive speed" 438
8.2.1 Basic signal flow 439
8.2.2 Internal interfaces | application block "LA_NCtrl" 441
8.2.3 Terminal assignment of the control modes 449
8.2.3.1 Terminals 0 450
8.2.3.2 Terminals 2 451
8.2.3.3 Terminals 11 452
8.2.3.4 Terminal 16 453
8.2.3.5 Keypad 454
8.2.3.6 PC 455
8.2.3.7 CAN 456
8.2.3.8 MCI 457
8.2.4 Process data assignment for fieldbus communication 458
8.2.5 Setting parameters (short overview) 460
8.2.6 Configuration parameters 462
8.3 TA "actuating drive speed (AC Drive Profile)" 465
8.3.1 I/O assemblies 466
8.3.2 Basic signal flow 467
8.3.3 Internal interfaces | application block "LA_NCtrl" 470
8.3.4 Terminal assignment of the control modes 479
8.3.4.1 Terminals 0 480
8.3.4.2 Terminals 2 481
8.3.4.3 Terminals 11 482
8.3.4.4 Terminal 16 483
8.3.4.5 Keypad 484
8.3.4.6 PC 485
8.3.4.7 CAN 486
8.3.4.8 MCI 487
8.3.5 Process data assignment for fieldbus communication 488
8.3.5.1 Run/Stop event 489
8.3.5.2 Scaling of speed and torque values 490
8.3.6 AC Drive Profile diagnostic parameters 491
8.3.7 Setting parameters (short overview) 491
8.3.8 Configuration parameters 493
8.4 TA "Table positioning" 498
8.4.1 Basic signal flow 499
8.4.1.1 Possibilities for the position selection 500
8.4.2 Internal interfaces | application block "LA_TabPos" 502
8.4.3 Terminal assignment of the control modes 512
8.4.3.1 Terminals 0 513
8.4.3.2 Terminals 2 514
8.4.3.3 Terminals 11 515
8.4.3.4 Terminal 16 516
8.4.3.5 Keypad 517
8.4.3.6 PC 518
8.4.3.7 CAN 519
8.4.3.8 MCI 520
8.4.4 Process data assignment for fieldbus communication 521
8.4.5 Setting parameters (short overview) 523
8.4.6 Configuration parameters 524
8.5 TA "Switch-off positioning" 527
8.5.1 Basic signal flow 529

# Contents


8.5.2 Internal interfaces | application block "LA_SwitchPos" 530
8.5.2.1 Truth table for activating the pre-switch off 537
8.5.3 Terminal assignment of the control modes 538
8.5.3.1 Terminals 0 539
8.5.3.2 Terminals 2 540
8.5.3.3 Terminals 11 541
8.5.3.4 Terminal 16 542
8.5.3.5 Keypad 543
8.5.3.6 PC 544
8.5.3.7 CAN 545
8.5.3.8 MCI 546
8.5.4 Process data assignment for fieldbus communication 547
8.5.5 Setting parameters (short overview) 549
8.5.6 Configuration parameters 551
8.6 "GeneralPurpose" functions 554
8.6.1 Analog switch 554
8.6.2 Arithmetic 555
8.6.3 Multiplication/Division 555
8.6.4 Binary delay element 556
8.6.5 Binary logic 556
8.6.6 Analog comparison 557
8.6.7 Binary signal monitor 557
8.6.8 Analog signal monitor 558
8.6.9 D-FlipFlop 558
8.6.10 Counter 559

**9 Basic drive functions (MCK)** 560
9.1 Basic signal flow 561
9.2 Internal interfaces | System block "LS_MotionControlKernel" 562
9.2.1 MCK control word 569
9.2.2 MCK status word 571
9.2.2.1 Status bit "HomPosAvailable" 573
9.2.2.2 Status bit "S_ShapingActive" 574
9.2.2.3 Status bit "bReadyToOperate" 575
9.2.3 MCK diagnostic word 576
9.2.4 MCK state machine 577
9.2.4.1 "StandBy" operating mode 578
9.2.5 Interface to safety system 579
9.2.6 Consideration of residual value in case of external profile calculation 580
9.3 MCKInterface 581
9.3.1 Control inputs | "L_MckCtrlInterface" function block 583
9.3.1.1 Alternative functions for control bit "PosExecute" 588
9.3.1.2 Operating mode change with profile number 589
9.3.2 Status outputs | FB "L_MckStateInterface" 590
9.3.2.1 Configurable status display *bOperationState* *592*
9.4 Basic settings 593
9.4.1 Machine parameters 593
9.4.1.1 Gearbox ratio 595
9.4.1.2 Feed constant 596
9.4.1.3 Activation of the modulo measuring system 597
9.4.2 Min/Max speed 600
9.4.3 Limit position monitoring 601
9.4.3.1 Software limit positions 601
9.4.3.2 Hardware limit switches 604
9.4.4 Target position monitoring (status "drive in target") 606

# Contents


9.4.5 Monitoring of the maximum travel distance 608
9.4.6 Following error monitoring system 609
9.4.6.1 Monitoring of the following error in case of controller inhibit 610
9.4.7 Setpoint holdback for bus runtime compensation 611
9.4.8 Setting or activation of maximum jerk for traversing profiles 615
9.5 Speed follower 616
9.5.1 Parameter setting 616
9.5.1.1 Functional settings 616
9.5.2 Requesting the operating mode 617
9.5.3 Setpoint selection 617
9.6 Homing 618
9.6.1 Parameter setting 619
9.6.1.1 Functional settings 621
9.6.1.2 Homing mode 622
9.6.1.3 Home position & home value offset 634
9.6.1.4 Traversing a sequence profile after completion of homing 634
9.6.1.5 Measurement of the distance between pre-stop mark/limit switch and TP/MP 634
9.6.1.6 Option "Actual MCTRL position received at mains switch-off" 635
9.6.2 Requesting the operating mode 635
9.6.3 Execute homing 636
9.6.3.1 Homing on the fly 637
9.7 Manual jog 638
9.7.1 Parameter setting 639
9.7.1.1 Functional settings 640
9.7.1.2 Smooth start and quick stop of the drive 641
9.7.1.3 Second speed 641
9.7.1.4 Time-based start of second speed 642
9.7.2 Requesting the operating mode 643
9.7.3 Executing manual jogging 643
9.7.3.1 Manual jog to limit position 644
9.7.3.2 Retracting of an operated limit switch 645
9.8 Positioning 646
9.8.1 Possible motion profiles 647
9.8.2 Parameter setting 648
9.8.2.1 Functional settings 649
9.8.2.2 Profile entry 651
9.8.2.3 Positioning modes 657
9.8.2.4 Touch probe positioning 658
9.8.2.5 S-ramp time for jerk limitation 659
9.8.3 Requesting the operating mode 661
9.8.4 Carrying out positioning 661
9.8.4.1 Stipulation of the profile to be executed 661
9.8.4.2 Starting/cancelling a traversing task 662
9.8.4.3 Override of the parameterised positioning mode 663
9.8.4.4 Position teaching 664
9.9 Stop 665
9.9.1 Parameter setting 665
9.9.2 Requesting the operating mode 665
9.10 Position follower 666
9.10.1 Parameter setting 667
9.10.1.1 Functional settings 667
9.10.2 Requesting the operating mode 668
9.10.3 Setpoint selection 668
9.11 Override 669
9.11.1 Speed override 670

# Contents


9.11.2 Acceleration override ____ 671
9.11.3 S-ramp smoothing override ____ 672
9.12 Holding brake control ____ 673
9.12.1 Internal interfaces ____ 674
9.12.2 Parameter setting ____ 676
9.12.2.1 Operating mode ____ 677
9.12.2.2 Functional settings ____ 679
9.12.2.3 Switching thresholds ____ 680
9.12.2.4 Application and release time ____ 681
9.12.2.5 Ramp time for approaching the setpoint speed ____ 683
9.12.2.6 Motor magnetising time (only with asynchronous motor) ____ 684
9.12.2.7 Actual value monitoring ____ 684
9.12.3 Process when brake is released ____ 685
9.12.4 Process when brake is closed ____ 686
9.12.5 Behaviour in case of pulse inhibit ____ 688
9.12.6 Feedforward control of the motor before release ____ 689

# Contents


**10 Diagnostics & error management** 691
10.1 Basics on error handling in the inverter 691
10.2 LED status displays 692
10.2.1 LED status displays of the device status 693
10.3 Drive diagnostics with the »Engineer« 694
10.3.1 Display details of the error 696
10.4 Drive diagnostics via keypad/bus system 697
10.5 Logbook 700
10.5.1 Functional description 700
10.5.2 Filtering logbook entries 701
10.5.3 Automatic recording of device-internal signals at the time the error occurs 701
10.5.4 Reading out logbook entries 703
10.5.5 Exporting logbook entries to a file 704
10.5.6 Storing the logbook in the project 705
10.5.7 Reading out the logbook from an external control/visualisation 706
10.6 Monitoring 708
10.6.1 Monitoring configuration 709
10.6.2 Setting the error response 710
10.6.3 AutoFailReset function 711
10.7 Maloperation of the drive 712
10.8 Operation without mains supply 714
10.9 Error messages of the operating system 715
10.9.1 Structure of the 32-bit error number (bit coding) 715
10.9.2 Structure of the 16 bit error number (bit coding) 718
10.9.3 Reset error message 719
10.9.4 Export error texts 720
10.9.5 Short overview (A-Z) 721
10.9.6 Cause & possible remedies 724
10.10 "LS_SetError_1" system block 752
10.11 "LS_SetError_2" system block 753

**11 Oscilloscope function** 754
11.1 Technical data 755
11.2 Operation 756
11.2.1 User interface 757
11.2.2 Selecting signal sources to be recorded 759
11.2.3 Selecting the recording time/sample rate 760
11.2.4 Defining the trigger condition 761
11.2.5 Starting recording 762
11.2.6 Cyclic recording 763
11.2.7 Adjusting the representation 763
11.2.8 Cursor function: Reading individual measured values 765
11.2.9 Last settings 765
11.3 Managing oscillograms 767
11.3.1 Commenting the oscillogram 767
11.3.2 Saving an oscillogram in a file 768
11.3.3 Loading the oscillogram file 769
11.3.4 Overlay function 770
11.3.5 Deleting an oscillogram file saved in the project 771

# Contents


**12 System bus "CAN on board"** 772
12.1 General information 773
12.1.1 General data and application conditions 773
12.1.2 Supported protocols 774
12.1.3 Communication time 775
12.2 Possible settings via DIP switch 776
12.2.1 Activating the bus terminating resistor 776
12.2.2 Setting the baud rate 777
12.2.3 Setting the node address 777
12.3 LED status displays for the system bus 778
12.4 Going online via the system bus 779
12.5 Reinitialising the CANopen interface 779
12.6 Structure of the CAN data telegram 780
12.6.1 Identifier 780
12.6.2 User data 782
12.7 Communication phases/network management 783
12.7.1 Status transitions 784
12.7.2 Network management telegram (NMT) 785
12.7.3 Parameterising the inverter as CAN master 786
12.8 Process data transfer 787
12.8.1 Available process data objects 788
12.8.1.1 RPDO1 | Port block "LP_CanIn1" 789
12.8.1.2 RPDO2 | "LP_CanIn2" port block 791
12.8.1.3 RPDO3 | "LP_CanIn3" port block 793
12.8.1.4 RPDO4 | "LP_CanIn4" port block 795
12.8.1.5 TPDO1 | "LP_CanOut1" port block 797
12.8.1.6 TPDO2 | "LP_CanOut2" port block 798
12.8.1.7 TPDO3 | "LP_CanOut3" port block 799
12.8.1.8 TPDO4 | "LP_CanOut4" port block 800
12.8.2 Identifiers of the process data objects 801
12.8.3 Transmission type 803
12.8.4 Synchronisation of PDOs via sync telegram 805
12.8.5 Monitoring of the RPDOs for data reception 806
12.8.6 Configuring exception handling of the CAN PDOs 806
12.9 Parameter data transfer 808
12.9.1 Identifiers of the parameter data objects 809
12.9.2 User data 809
12.9.2.1 Command 810
12.9.2.2 Addressing by means of index and subindex 811
12.9.2.3 Data 1 ... Data 4 812
12.9.2.4 Error messages 813
12.9.3 Parameter data telegram examples 815
12.9.3.1 Read parameters 815
12.9.3.2 Write parameters 816
12.9.3.3 Read block parameters 817
12.10 Monitoring 820
12.10.1 Integrated error detection 820
12.10.2 Heartbeat protocol 821
12.10.2.1 Telegram structure 821
12.10.2.2 Parameter setting 822
12.10.2.3 Commissioning example 823
12.10.3 Emergency telegram 824
12.11 Implemented CANopen objects 825
12.12 Internal interfaces | System block "LS_CANManagement" 851

# Contents


**13 Fieldbus interface (MCI)** 852
13.1 Process data transfer 853
13.2 Control mode "MCI" 855
13.2.1 Port block "LP_MciIn" 856
13.2.2 Port block "LP_MciOut" 857
13.3 CAN gateway 858

**14 Axis bus** 859
14.1 LED status displays of the axis bus 860
14.2 Data transfer axis bus 861
14.2.1 Topologies 861
14.2.2 Transfer mechanisms 861
14.2.3 Activating the bus terminating resistor 863
14.2.4 Parameter setting 864
14.2.4.1 Required settings at the master 865
14.2.4.2 Required settings for the slaves 865
14.2.4.3 Boot-up phase 866
14.2.4.4 Disconnect nodes from the axis bus 866
14.2.4.5 Configuring exception handling of the received data 867
14.2.4.6 Integrated error detection 868
14.2.4.7 Axis bus data error monitoring 869
14.2.5 Internal interfaces | System block "LS_AxisBusOut" 870
14.2.6 Internal interfaces | System block "LS_AxisBusIn" 871
14.2.7 Internal interfaces | System block "LS_AxisBusAux" 872
14.3 IO axis bus 874
14.3.1 "Master/slave" function 875
14.3.2 "IO" function 878
14.3.3 Internal interfaces | System block "LS_AxisBusIO" 879

**15 Synchronisation of the internal time base** 880
15.1 Internal interfaces | System block "LS_SyncManagement" 881

**16 Parameter change-over** 882
16.1 Configuring parameter change-over via the »Engineer« parameterisation dialog 883
16.1.1 Configuring the parameter list(s) 884
16.1.2 Configuring control inputs 887
16.1.3 Functional settings 888
16.1.4 Error message 888
16.2 Configuring the definable parameter list by means of parameterisation 889
16.3 Configuring the motor data parameter list by means of parameterisation 890
16.4 Internal interfaces | System block "LS_WriteParamList" 893

**17 Parameter reference** 895
17.1 Structure of the parameter descriptions 896
17.1.1 Data type 897
17.1.2 Parameters with read-only access 897
17.1.3 Parameters with write access 898
17.1.3.1 Parameters with setting range 898
17.1.3.2 Parameters with selection list 898
17.1.3.3 Parameters with bit-coded setting 899
17.1.3.4 Parameters with subcodes 900
17.1.4 Parameter attributes 900
17.2 Parameter list 901
17.3 Overview of all parameters with power-related Lenze setting 1335
17.4 Selection list - analog signals 1337

# Contents


17.5 Selection list - digital signals 1343
17.6 Selection list - angle signals 1351
17.7 Table of attributes 1353

**18 Working with the FB Editor** 1373
18.1 Basics 1373
18.1.1 Basic components of a drive solution 1374
18.1.1.1 What is a function block? 1375
18.1.1.2 Parameterisable function blocks 1376
18.1.1.3 What is a system block? 1376
18.1.1.4 What is a port block? 1377
18.1.1.5 What is an application block? 1377
18.1.2 Conventions used for input/output identifiers 1378
18.1.3 Scaling of physical units 1379
18.2 User interface 1380
18.2.1 Toolbar 1381
18.2.2 Search function 1382
18.2.3 Level selection 1383
18.2.4 Editor view/overview 1385
18.2.5 Context menu 1386
18.2.6 Status bar 1386
18.2.7 Overview 1387
18.3 Using the FB Editor as "Viewer" 1389
18.3.1 Following connections of inputs and outputs 1390
18.3.2 Keyboard commands for navigation 1391
18.3.3 Change online display format 1392
18.4 Reconfiguring the predefined interconnection 1394
18.4.1 Inserting/Deleting objects 1394
18.4.1.1 Inserting a function block 1395
18.4.1.2 Inserting a system block 1397
18.4.1.3 Insert port block 1399
18.4.1.4 Inserting a comment 1401
18.4.1.5 Deleting objects that are no longer required 1403
18.4.2 Changing connector visibilities 1404
18.4.3 Arranging objects in the drawing area 1405
18.4.4 Creating/deleting connections 1406
18.4.4.1 Creating a connection using the connection line 1408
18.4.4.2 Creating a connection using port identifiers 1409
18.4.4.3 Creating a connection via connection dialog 1410
18.4.4.4 Deleting connections that are no longer required 1411
18.4.5 Changing the processing order 1412
18.4.6 Copying interconnection elements (across all devices) 1414
18.4.6.1 Insert options for copied elements 1416
18.4.7 Resetting changed interconnection 1417
18.5 Adjusting online and offline interconnection 1418
18.6 Printing the interconnection 1419
18.7 Comparing interconnections 1420
18.8 Copying an interconnection 1423
18.9 Exporting/Importing an interconnection 1424

# Contents


**19 Function library** 1425
19.1 Function blocks 1425
19.1.1 L_Absolut_1 1430
19.1.2 L_Absolut_2 1430
19.1.3 L_AddSub_1 1431
19.1.4 L_AnalogSwitch_1 1432
19.1.5 L_AnalogSwitch_2 1433
19.1.6 L_AnalogSwitch_3 1434
19.1.7 L_AnalogSwitch_4 1435
19.1.8 L_AnalogSwitch_5 1436
19.1.9 L_And_1 1437
19.1.10 L_And_2 1438
19.1.11 L_And_3 1439
19.1.12 L_And5_1 1440
19.1.13 L_And5_2 1441
19.1.14 L_Arithmetik_1 1442
19.1.15 L_Arithmetik_2 1443
19.1.16 L_Arithmetik_3 1444
19.1.17 L_Arithmetik_4 1445
19.1.18 L_Arithmetik_5 1446
19.1.19 L_ArithmetikPhi_1 1447
19.1.20 L_ArithmetikPhi_2 1448
19.1.21 L_ArithmetikPhi_3 1449
19.1.22 L_ArithmetikPhi_4 1450
19.1.23 L_ArithmetikPhi_5 1451
19.1.24 L_ArithmetikPhi_6 1452
19.1.25 L_CalcDiameter_1 1453
19.1.25.1 Set initial value 1456
19.1.25.2 Calculate diameter 1456
19.1.25.3 Select change direction, web break monitoring 1456
19.1.25.4 Holding the current value 1456
19.1.25.5 Limit value monitoring 1457
19.1.25.6 Converting diameter in 1/D 1457
19.1.25.7 Physical state variables 1457
19.1.26 L_Compare_1 1458
19.1.26.1 Function 1: nIn1 = nIn2 1459
19.1.26.2 Function 2: nIn1 > nIn2 1460
19.1.26.3 Function 3: nIn1 < nIn2 1461
19.1.26.4 Function 4: |nIn1| = |nIn2| 1462
19.1.26.5 Function 5: |nIn1| > |nIn2| 1462
19.1.26.6 Function 6: |nIn1| < |nIn2| 1462
19.1.27 L_Compare_2 1463
19.1.28 L_Compare_3 1464
19.1.29 L_Compare_4 1465
19.1.30 L_Compare_5 1466
19.1.31 L_ComparePhi_1 1467
19.1.31.1 Function 1: dnIn1 = dnIn2 1468
19.1.31.2 Function 2: dnIn1 > dnIn2 1469
19.1.31.3 Function 3: dnIn1 < dnIn2 1470
19.1.31.4 Function 4: |dnIn1| = |dnIn2| 1471
19.1.31.5 Function 5: |dnIn1| > |dnIn2| 1471
19.1.31.6 Function 6: |dnIn1| < |dnIn2| 1471
19.1.32 L_ComparePhi_2 1472
19.1.33 L_ComparePhi_3 1473
19.1.34 L_ComparePhi_4 1474

# Contents


19.1.35 L_ComparePhi_5 1475
19.1.36 L_ConvActPos_1 1476
19.1.36.1 Determining limit positions 1478
19.1.36.2 Monitoring of the dancer position 1478
19.1.36.3 Circumferential speed of the reel 1478
19.1.36.4 Additional torque for damping the dancer control 1478
19.1.37 L_ConvAP_1 1479
19.1.38 L_ConvAP_2 1480
19.1.39 L_ConvAP_3 1481
19.1.40 L_ConvBitsToWord_1 1482
19.1.41 L_ConvBitsToWord_2 1483
19.1.42 L_ConvBitsToWord_3 1484
19.1.43 L_ConvDIntToWords_1 1485
19.1.44 L_ConvDIntToWords_2 1486
19.1.45 L_ConvDIntToWords_3 1487
19.1.46 L_ConvPA_1 1488
19.1.47 L_ConvPA_2 1489
19.1.48 L_ConvPA_3 1490
19.1.49 L_ConvPP_1 1491
19.1.50 L_ConvPP_2 1492
19.1.51 L_ConvPP_3 1493
19.1.52 L_ConvUnitsToIncr_1 1494
19.1.53 L_ConvUnitsToIncr_2 1496
19.1.54 L_ConvUnitsToIncr_3 1498
19.1.55 L_ConvW_1 1500
19.1.56 L_ConvW_2 1502
19.1.57 L_ConvW_3 1503
19.1.58 L_ConvW_4 1504
19.1.59 L_ConvWordsToDInt_1 1505
19.1.60 L_ConvWordsToDInt_2 1506
19.1.61 L_ConvWordsToDInt_3 1507
19.1.62 L_ConvWordToBits_1 1508
19.1.63 L_ConvWordToBits_2 1509
19.1.64 L_ConvWordToBits_3 1510
19.1.65 L_ConvX_1 1511
19.1.66 L_ConvX_2 1512
19.1.67 L_ConvX_3 1513
19.1.68 L_Counter_1 1514
19.1.69 L_Counter_2 1516
19.1.70 L_Counter_3 1518
19.1.71 L_Curve_1 1520
19.1.71.1 Function 3: nOut_a = f(table) 1521
19.1.72 L_Curve_2 1524
19.1.73 L_Curve_3 1526
19.1.73.1 Use of the L_Curve_3 for tensile force characteristic 1528
19.1.74 L_DFlipFlop_1 1531
19.1.75 L_DFlipFlop_2 1532
19.1.76 L_DFRFG_1 1533
19.1.76.1 Ramp function generator (profile generator) 1535
19.1.76.2 Quick stop 1537
19.1.76.3 Ramp function generator stop 1538
19.1.76.4 Reset angle setpoint 1538
19.1.76.5 Detecting the angular difference 1538
19.1.76.6 Offset setting 1539

# Contents


19.1.77 L_DFSET_1 1542
19.1.77.1 Master value rail/slave cascade 1547
19.1.77.2 Setpoint conditioning with stretch factor and gearbox factor 1547
19.1.77.3 Processing correction values 1548
19.1.77.4 Synchronising slave drive to master drive 1550
19.1.77.5 Masking out touch probe signals 1553
19.1.77.6 Process monitoring functions 1553
19.1.78 L_DigitalDelay_1 1554
19.1.79 L_DigitalDelay_2 1556
19.1.80 L_DigitalDelay_3 1557
19.1.81 L_DigitalLogic_1 1558
19.1.82 L_DigitalLogic_2 1560
19.1.83 L_DigitalLogic_3 1562
19.1.84 L_DigitalLogic5_1 1564
19.1.85 L_DigitalLogic5_2 1566
19.1.86 L_DT1_1 1568
19.1.87 L_FixSet_a_1 1569
19.1.88 L_FixSet_w_1 1570
19.1.89 L_FixSet_w_2 1571
19.1.90 L_GainOffset_1 1572
19.1.91 L_GainOffset_2 1573
19.1.92 L_GainOffset_3 1574
19.1.93 L_GainOffsetP_1 1575
19.1.94 L_GainOffsetP_2 1576
19.1.95 L_GainOffsetP_3 1577
19.1.96 L_GainOffsetPhiP_1 1578
19.1.97 L_GainOffsetPhiP_2 1579
19.1.98 L_GearComp_1 1580
19.1.99 L_Interpolator_1 1582
19.1.99.1 Signal interpolation 1583
19.1.99.2 Signal monitoring 1584
19.1.99.3 Angle correction in case of transmission errors 1585
19.1.100 L_JogCtrlExtension_1 1586
19.1.101 L_Limit_1 1588
19.1.102 L_Limit_2 1589
19.1.103 L_LimitPhi_1 1590
19.1.104 L_LimitPhi_2 1591
19.1.105 L_LimitPhi_3 1592
19.1.106 L_MFail_1 1593
19.1.106.1 Procedure of the mains failure control 1595
19.1.106.2 Signal flow for mains failure control 1597
19.1.106.3 Activating the mains failure control 1599
19.1.106.4 Parameterising the mains failure control 1600
19.1.106.5 Commissioning the mains failure control 1601
19.1.106.6 Fine adjustment of the mains failure control 1602
19.1.106.7 Reset mains failure control 1603
19.1.106.8 Restart protection 1603
19.1.106.9 Quick mains recovery (KU) 1603
19.1.107 L_MPot_1 1604
19.1.107.1 Activate & control motor potentiometer 1606
19.1.107.2 Deactivate motor potentiometer 1607
19.1.108 L_MulDiv_1 1608
19.1.109 L_MulDiv_2 1609
19.1.110 L_Mux_1 1610
19.1.111 L_Negation_1 1611

# Contents


19.1.112 L_Negation_2 1612
19.1.113 L_NLim_1 1613
19.1.114 L_NLim_2 1616
19.1.115 L_Not_1 1618
19.1.116 L_Not_2 1618
19.1.117 L_Not_3 1619
19.1.118 L_Not_4 1619
19.1.119 L_Not_5 1620
19.1.120 L_Not_6 1620
19.1.121 L_Not_7 1621
19.1.122 L_NSet_1 1622
19.1.122.1 Main setpoint path 1626
19.1.122.2 JOG setpoints 1626
19.1.122.3 Setpoint inversion 1627
19.1.122.4 Value range of the input signal 1627
19.1.122.5 Skip frequency function 1627
19.1.122.6 Ramp function generator for the main setpoint 1629
19.1.122.7 S-shaped ramp 1631
19.1.122.8 Additional setpoint 1632
19.1.122.9 Application example for the additional load function 1632
19.1.123 L_Odometer_1 1633
19.1.124 L_OffsetGain_1 1635
19.1.125 L_OffsetGain_2 1636
19.1.126 L_OffsetGain_3 1637
19.1.127 L_OffsetGainP_1 1638
19.1.128 L_OffsetGainP_2 1639
19.1.129 L_OffsetGainP_3 1640
19.1.130 L_OffsetGainPhiP_1 1641
19.1.131 L_OffsetGainPhiP_2 1642
19.1.132 L_Or_1 1643
19.1.133 L_Or_2 1644
19.1.134 L_Or_3 1645
19.1.135 L_Or_4 1646
19.1.136 L_Or5_1 1647
19.1.137 L_Or5_2 1648
19.1.138 L_PCTRL_1 1649
19.1.138.1 Control characteristic 1653
19.1.138.2 Ramp function generator 1654
19.1.138.3 Operating range of the PID process controller 1654
19.1.138.4 Evaluation of the output signal 1655
19.1.138.5 Comparison function "Actual value = setpoint" 1655
19.1.138.6 Control functions 1656
19.1.139 L_PhaseDiff_1 1657
19.1.140 L_PhaseDiff_2 1658
19.1.141 L_PhaseIntK_1 1659
19.1.141.1 Function at constant input value 1660
19.1.141.2 Function at input value with sign reversal 1661
19.1.141.3 Calculation of the output signal 1662
19.1.142 L_PhaseIntK_2 1663
19.1.143 L_PhiAdd_1 1665
19.1.144 L_PhiDiv_1 1666
19.1.145 L_PhiIntegrator_1 1667
19.1.145.1 Function 1669
19.1.145.2 Example 1670
19.1.146 L_PosCtrlLin_1 1671

# Contents


19.1.147 L_PosCtrlLin_2 1674
19.1.148 L_PosiShaftCtrlInterface_1 1677
19.1.149 L_ProcessCtrl_1 1678
19.1.149.1 Control characteristic 1680
19.1.150 L_PT1_1 1682
19.1.151 L_PT1_2 1683
19.1.152 L_PT1_3 1684
19.1.153 L_RLQ_1 1685
19.1.154 L_RSFlipFlop_1 1686
19.1.155 L_RSFlipFlop_2 1687
19.1.156 L_SampleHold_1 1688
19.1.157 L_SampleHold_2 1689
19.1.158 L_Sequencer_1 1690
19.1.158.1 Connection to the basic drive functions (MCK) 1695
19.1.158.2 Priorities of the control signals 1696
19.1.158.3 States of the sequencer 1697
19.1.158.4 Overview: Available action types 1698
19.1.158.5 Action type "Program end" 1699
19.1.158.6 Action type "positioning" 1700
19.1.158.7 Action type "switching" 1701
19.1.158.8 Action type "Branching" 1702
19.1.158.9 Action type "Variable branching" 1703
19.1.158.10 Action type "Homing" 1704
19.1.158.11 Action type "Time waiting" 1705
19.1.158.12 Action type "Counter setting" 1706
19.1.158.13 Action type "Counting" 1707
19.1.158.14 "Stand-by" action type 1708
19.1.158.15 Example: Sequence table 1709
19.1.159 L_SignalMonitor_a 1710
19.1.160 L_SignalMonitor_b 1711
19.1.161 L_SignalSwitch_1 1712
19.1.162 L_SignalSwitch_2 1713
19.1.163 L_SignalSwitch_3 1714
19.1.164 L_SignalSwitch_4 1715
19.1.165 L_SignalSwitch32_1 1716
19.1.166 L_SignalSwitch32_2 1717
19.1.167 L_SignalSwitch32_3 1718
19.1.168 L_SQrt_1 1719
19.1.169 L_SRFG_1 1720
19.1.170 L_SRFG_2 1722
19.1.171 L_SwitchPoint_1 1724
19.1.171.1 Definition of the switching range 1726
19.1.171.2 Dead time compensation 1727
19.1.171.3 Switching hysteresis 1727
19.1.171.4 Position/time-based cams 1728
19.1.172 L_SwitchPointPar_1 1729
19.1.173 L_Transient_1 1732
19.1.173.1 Function 0: Evaluate rising signal edges 1733
19.1.173.2 Function 1: Evaluate falling signal edges 1733
19.1.173.3 Function 2: Evaluate rising and falling signal edges 1734
19.1.174 L_Transient_2 1735
19.1.175 L_Transient_3 1736
19.1.176 L_Transient_4 1737
19.1.177 L_Transient_5 1738
19.1.178 L_Transient_6 1739

# Contents


19.1.179 L_Transient_7 ____ 1740
19.1.180 L_Transient_8 ____ 1741
19.2 System blocks ____ 1742
19.2.1 LS_AnalogInput ____ 1744
19.2.2 LS_AnalogOutput ____ 1744
19.2.3 LS_AxisBusAux ____ 1744
19.2.4 LS_AxisBusIn ____ 1744
19.2.5 LS_AxisBusIO ____ 1744
19.2.6 LS_AxisBusOut ____ 1745
19.2.7 LS_CANManagement ____ 1745
19.2.8 LS_DataAccess ____ 1745
19.2.9 LS_DeviceMonitor ____ 1745
19.2.10 LS_DFOut ____ 1745
19.2.11 LS_DigitalInput ____ 1746
19.2.12 LS_DigitalOutput ____ 1746
19.2.13 LS_DisFree ____ 1746
19.2.14 LS_DisFree_a ____ 1747
19.2.14.1 Display of internal process variables in application units ____ 1748
19.2.15 LS_DisFree_b ____ 1749
19.2.16 LS_DisFree_p ____ 1750
19.2.17 LS_DriveInterface ____ 1750
19.2.18 LS_Keypad ____ 1751
19.2.19 LS_MotionControlKernel ____ 1752
19.2.20 LS_MotorInterface ____ 1752
19.2.21 LS_MultiEncoder ____ 1752
19.2.22 LS_ParFix ____ 1753
19.2.23 LS_ParFree ____ 1754
19.2.24 LS_ParFree_2 ____ 1755
19.2.25 LS_ParFree_a ____ 1756
19.2.26 LS_ParFree_a_2 ____ 1757
19.2.27 LS_ParFree_b ____ 1758
19.2.28 LS_ParFree_p ____ 1759
19.2.29 LS_ParFree_v ____ 1760
19.2.30 LS_ParFree_v_2 ____ 1761
19.2.31 LS_ParFree32 ____ 1762
19.2.32 LS_ParFreeUnit ____ 1763
19.2.33 LS_ParFreeUnit_2 ____ 1764
19.2.34 LS_ParReadWrite_1-6 ____ 1765
19.2.34.1 Arithmetic function ____ 1768
19.2.35 LS_PulseGenerator ____ 1769
19.2.36 LS_Resolver ____ 1770
19.2.37 LS_RetainData ____ 1771
19.2.38 LS_SetError_1 ____ 1773
19.2.39 LS_SetError_2 ____ 1773
19.2.40 LS_SyncManagement ____ 1773
19.2.41 LS_TouchProbe ____ 1773
19.2.42 LS_WriteParamList ____ 1773
19.3 Application blocks ____ 1774
19.3.1 Technology application "Actuating drive speed" ____ 1774
19.3.2 Technology application "actuating drive speed (AC Drive profile)" ____ 1774
19.3.3 Technology application "Table positioning" ____ 1774
19.3.4 "Switch-off positioning" technology application ____ 1774
19.4 Port blocks ____ 1775
19.4.1 Port block "LP_CanIn1" ____ 1775
19.4.2 Port block "LP_CanIn2" ____ 1775

# Contents


19.4.3 Port block "LP_CanIn3" _____ 1775
19.4.4 Port block "LP_CanIn4" _____ 1775
19.4.5 Port block "LP_CanOut1" _____ 1775
19.4.6 Port block "LP_MciIn" _____ 1776
19.4.7 Port block "LP_MciOut" _____ 1776
19.4.8 Port block "LP_CanOut2" _____ 1776
19.4.9 Port block "LP_CanOut3" _____ 1776
19.4.10 Port block "LP_CanOut4" _____ 1776

**Index** _____ 1777

Your opinion is important to us _____ 1806

# 1 About this documentation

**Danger!**

The inverter is a source of danger which may lead to death or the severe injury of persons.

To protect yourself and others against these dangers, observe the safety instructions before switching on the inverter.

Please read the safety instructions provided in the **8400 mounting instructions** and in the **8400 hardware manual**. Both documents are supplied with the inverter.

### Target group

This documentation is intended for all persons who would like to parameterise, configure and diagnose the 8400 TopLine inverter with the Lenze engineering software »Engineer« and the X400 keypad.

### Information regarding the validity

The information in this documentation are valid for the following standard devices:

| Product series | Type designation | From software version |
|---|---|---|
| 8400 TopLine C | E84AVTCxxxxx | 01.00 |

### Screenshots/application examples

All screenshots provided in this documentation are application examples. Depending on the software version of the inverter and the version of the »Engineer« software installed, the screenshots in this documentation may differ from the representation in the »Engineer«.

**Tip!**

Information and tools for Lenze products are provided in the Download area at

http://www.lenze.com → Download

## 1.1 Document history

| Version | | | Description |
|---|---|---|---|
| 8.0 | 02/2016 | TD06 | Error corrections & supplements |
| 7.1 | 01/2016 | TD06 | Oscilloscope function (📖 754) revised |
| 7.0 | 12/2015 | TD06 | Extended by new functions for 8400 TopLine C V16.00.00 |
| 6.0 | 06/2014 | TD05 | Extended by new functions for 8400 TopLine C V15.00.00 |
| 5.0 | 11/2013 | TD05 | Extended by new functions for 8400 TopLine C V14.00.00 |
| 4.1 | 01/2013 | TD05 | Error corrections & supplements |
| 4.0 | 11/2012 | TD05 | Extended by new functions for 8400 TopLine C V13.00.00 and converted to new layout. |
| 3.1 | 09/2012 | TD05 | Error corrections & supplements |

| Version | | | Description |
|---|---|---|---|
| 3.0 | 07/2012 | TD05 | Extended by new functions for 8400 TopLine C V12.00.00 |
| 2.0 | 08/2011 | TD05 | Extended by new functions for 8400 TopLine C V02.00.00 |

## 1.2 Conventions used

This documentation uses the following conventions to distinguish between different types of information:

| Type of information | Highlighting | Examples/notes |
|---|---|---|
| Spelling of numbers | | |
| Decimal separator | Point | The decimal point is always used.<br>Example: 1234.56 |
| Hexadecimal number | 0x | For hexadecimal numbers, the "0x" prefix is used.<br>Example: 0x60F4 |
| Binary number | 0b | For binary numbers, the "0b" prefix is used.<br>Example: 0b00010111 |
| Text | | |
| Version information | Blue text colour | All information that only applies to a certain inverter software version or higher is identified accordingly in this documentation.<br>Example: This function extension is available from software version V3.0 onwards! |
| Program name | » « | The Lenze PC software »Engineer«... |
| Window | *italics* | The *Message window*... / The dialog box *Options*... |
| Variable names | | By setting *bEnable* to TRUE... |
| Control element | **bold** | The **OK** button ... / The **Copy** command ... / The **Properties** tab ... / The **Name** input field ... |
| Sequence of menu commands | | If several commands must be used in sequence to carry out a function, the individual commands are separated by an arrow: Select **File→Open** to... |
| Shortcut | **<bold>** | Use **<F1>** to open the online help. |
| | | If a shortcut is required for a command to be executed, a "+" has been put between the key identifiers: With **<Shift>+<ESC>** ... |
| Hyperlink | Underlined | Optically highlighted reference to another topic. It is activated with a mouse-click in this online documentation. |
| Icons | | |
| Page reference | (📖 27) | Optically highlighted reference to another page. In this online documentation activated via mouse-click. |
| Step-by-step instructions |  | Step-by-step instructions are indicated by a pictograph. |

All information that only applies to a certain inverter software version or higher is identified accordingly in this documentation.

## 1.3 Terminology used

| Term | Meaning |
|---|---|
| Engineering tools | Software solutions for easy engineering in all project stages |
| |  »EASY Navigator« – ensures easy operator guidance<br>• All convenient Lenze engineering tools at a glance<br>• Tools can be quickly selected<br>• The clear structure simplifies the engineering process from the start |
| |  »EASY Starter« – simple tool for service technicians<br>• Specially designed for the commissioning and maintenance of Lenze devices<br>• Graphical user interface with only a few buttons<br>• Simple online diagnostics, parameterisation, and commissioning<br>• No risk of an unintended change in applications<br>• Loading of ready-to-use applications to the device |
| |  »Engineer« – multi-device engineering<br>• For all products in our L-force portfolio<br>• Practical user interface<br>• Easy handling by graphical user interfaces<br>• Can be applied in every phase of a project (project planning, commissioning, production)<br>• Parameter setting and configuration |
| Application block | Block for a technology application (e.g. actuating drive speed)<br>A technology application is a drive solution based on the experience and know-how of Lenze in which function blocks interconnected to a signal flow form the basis for implementing typical drive tasks. |
| ASM | Abbreviation for asynchronous motor |
| Service brake | The service brake serves to shutdown rotary or translatory masses in motion in a controlled manner. The energy to be dissipated in this process is produced in the form of friction energy. Unlike emergency braking, this process is a regular and recurring operating mode. |
| CAN | Abbreviation for Controller Area Network. CAN is an asynchronous, serial fieldbus system. |
| CANopen | CANopen® is a communication protocol based on CAN. The Lenze system bus (CAN on board) operates with a subset of this communication protocol.<br>CANopen® is a registered community trademark of the CAN user organisation CiA® (CAN in Automation e. V.).<br>▸ System bus "CAN on board" |
| Code | Parameter used for inverter parameterisation or monitoring.<br>Is usually referred to as "index". |
| Display code | Parameter that displays the current status or value of an input/output of a system block. |
| EtherCAT | EtherCAT® is a real-time capable Ethernet system with top performance.<br>EtherCAT® is a registered trademark and patented technology, licenced by Beckhoff Automation GmbH, Germany. |
| ETHERNET POWERLINK | Ethernet POWERLINK is a real-time capable fieldbus system based on Ethernet. For user data exchange, Ethernet POWERLINK specifies a communication protocol based on CANopen.<br>Ethernet POWERLINK is a patented technology licensed by the Ethernet POWERLINK Standardization Group (EPSG), Germany. |
| EtherNet/IP | EtherNet/IP™ (EtherNet Industrial Protocol) is a fieldbus system based on Ethernet which uses the Common Industrial Protocol™ (CIP™) for data exchange.<br>EtherNet/IP™ and Common Industrial Protocol™ (CIP™) are trademarks and patented technologies licensed by the user organisation ODVA (Open DeviceNet Vendor Association), USA. |
| FB Editor | Abbreviation for "function block editor". Graphic interconnection tool which is available in the »Engineer« for function block interconnections on the **FB Editor**.<br>▸ Working with the FB Editor |

| Term | Meaning |
|---|---|
| Function block | General designation of a function block for free interconnection in the FB Editor.<br>A function block (short: FB) can be compared with an integrated circuit that contains a specific control logic and delivers one or several values when being executed. Example: "L_Arithmetik_1" (FB for arithmetic operations)<br>Many function blocks are available several times (e.g. L_And_1, L_And_2, and L_And_3).<br>▸ Function blocks |
| Holding brake | The holding brake serves to statically hold e.g. a position during the downtimes of a robot/ travelling/synchronous/hoist drive. |
| HIPERFACE® | HIPERFACE® stands for High Performance Interface and is a universal interface between motor feedback system and inverter.<br>HIPERFACE® is a registered trademark of the SICK STEGMANN GmbH. |
| INTERBUS | INTERBUS was developed as sensor/actuator/bus system for transmitting process data.<br>Today, maintenance activities for the INTERBUS technologies are carried out by the user organisation PROFIBUS & PROFINET International (PI). |
| Keypad | The keypad is an alternative to the PC for the local operation, parameterisation, and diagnostics in a simple manner. |
| LA | Abbreviation for **L**enze **A**pplication block<br>Example: "LA_NCtrl" (block for the "Actuating drive speed" application) |
| Lenze setting | This setting is the default factory setting of the device. |
| LP | Abbreviation for **L**enze **P**ort block<br>Example: "LP_CanIn1" (CAN1 port block) |
| LS | Abbreviation for **L**enze **S**ystem block<br>Example: "LS_DigitalInput" (system block for digital input signals) |
| MCI | Abbreviation for **M**otionbus **C**ommunication **I**nterface (fieldbus interface)<br>The Inverter Drives 8400 can accommodate plug-in communication modules and can therefore take part in the data transfer of an existing fieldbus system.<br>▸ Fieldbus interface (MCI) |
| Emergency brake | The emergency brake serves to shutdown rotary or translatory masses in motion in emergency situations. Emergency situations are exceptional situations that only occur sporadically. |
| Port block | Block for implementing the process data transfer via a fieldbus |
| PROFIBUS | PROFIBUS® (Process Field Bus) is fieldbus system used worldwide for automating machines and production plants.<br>PROFIBUS® is a registered trademark and patented technology licensed by the PROFIBUS & PROFINET International (PI) user organisation. |
| PROFINET | PROFINET® (Process Field Network) is a real-time capable fieldbus system based on Ethernet.<br>PROFINET® is a registered trademark and patented technology licensed by the PROFIBUS & PROFINET International user organisation (PI). |
| PSM | Abbreviation for permanently excited synchronous motor |
| QSP | Abbreviation for quick stop |
| SC | Abbreviation for **S**ervo **C**ontrol |
| SLPSM | Abbreviation for sensorless control of synchronous motors |
| SLVC | Abbreviation for **S**ensor**L**ess **V**ector **C**ontrol |
| SSI | Abbreviation for **S**ynchronous **S**erial **I**nterface<br>Interface for absolute angle encoders developed by SICK STEGMANN GmbH. |
| Subcode | If a code contains several parameters, they are stored in "subcodes".<br>This manual uses a slash "/" as a separator between code and subcode (e.g. "C00118/3").<br>Is usually referred to as "subindex". |
| System block | In the function block editor of the »Engineer«, system blocks provide interfaces to basic functions, "free codes", and to the hardware of the inverter (e.g. to the digital inputs). Each system block is available only once.<br>▸ System blocks |

| Term | Meaning |
|---|---|
| USB diagnostic adapter | The USB diagnostic adapter is used for the operation, parameterisation, and diagnostics of the inverter. Data are exchanged between the PC (USB connection) and the inverter (diagnostic interface on the front) via the diagnostic adapter. Order designation: E94AZCUS |
| VFCplus | Abbreviation for **V**oltage **F**requency **C**ontrol |

## 1.4 Definition of the notes used

The following signal words and symbols are used in this documentation to indicate dangers and important information:

### Safety instructions

Layout of the safety instructions:



**Pictograph and signal word!**

(characterise the type and severity of danger)

**Note**

(describes the danger and gives information about how to prevent dangerous situations)

| Pictograph | Signal word | Meaning |
|---|---|---|
|  | **Danger!** | **Danger of personal injury through dangerous electrical voltage**<br>Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken. |
|  | **Danger!** | **Danger of personal injury through a general source of danger**<br>Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken. |
|  | **Stop!** | **Danger of property damage**<br>Reference to a possible danger that may result in property damage if the corresponding measures are not taken. |

### Application notes

| Pictograph | Signal word | Meaning |
|---|---|---|
|  | **Note!** | Important note to ensure trouble-free operation |
|  | **Tip!** | Useful tip for simple handling |

# 2 Introduction: Parameterising the inverter

Being a component of a machine which includes a speed-variable drive system, the inverter needs to be adjusted to its drive task. The inverter is adjusted by changing parameters which are saved in the memory module. The parameters can be accessed by keypad, by the »EASY Starter« or by the »Engineer«. Access is also possible by a master control via fieldbus communication. For this purpose, the "CAN on board" CAN interface and the MCI interface for using a communication module (e.g. PROFIBUS) are available.

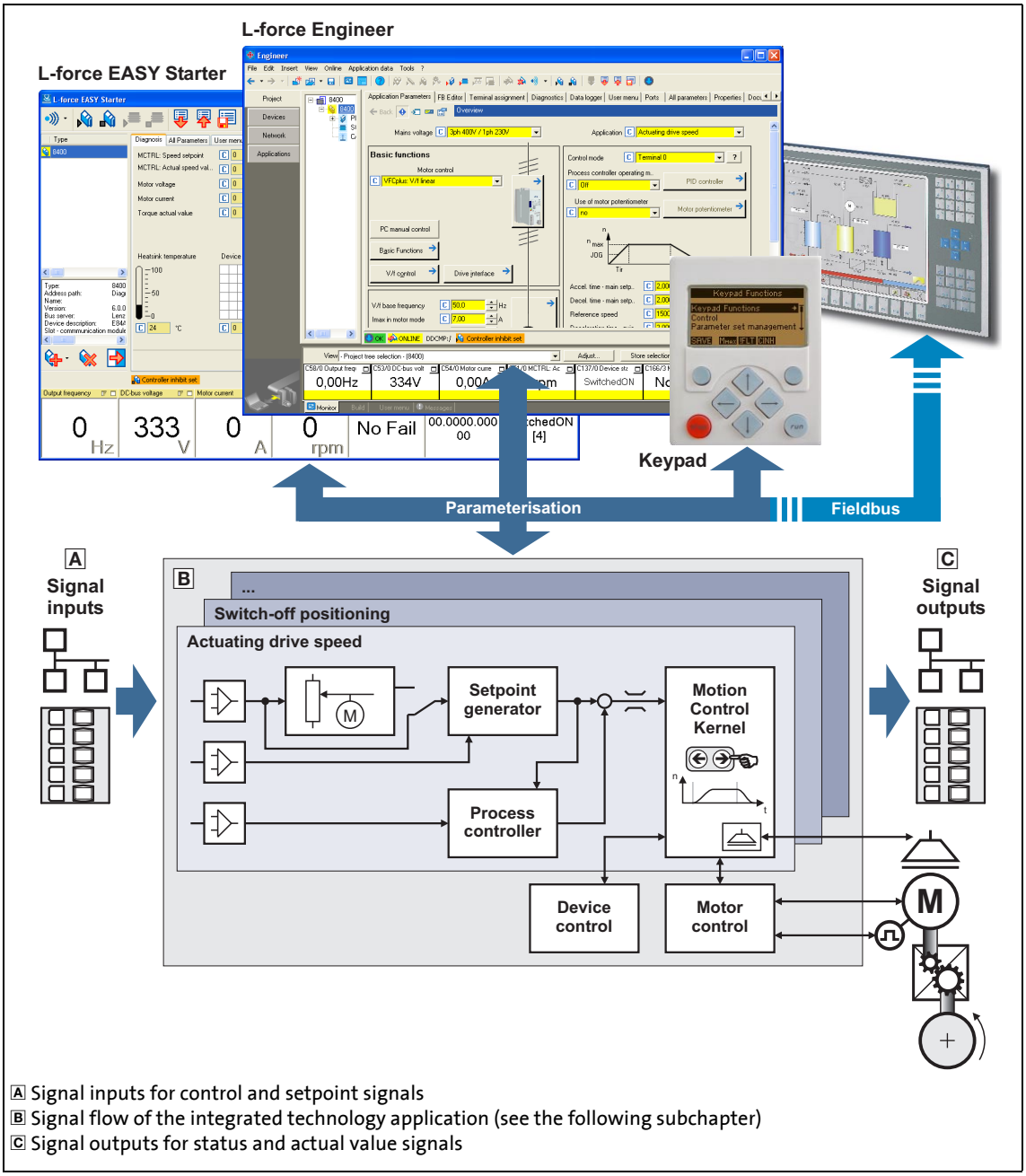


**Danger!**

In general, changing a parameter causes an immediate response in the inverter!

An enabled inverter can cause an unwanted behaviour at the motor shaft! For instance, setpoint sources can change over abruptly (e.g. when the signal source is configured for the main setpoint).

Certain device commands or settings which may cause critical states of drive behaviour constitute exceptions. Such parameter changes are only possible if the inverter is inhibited. Otherwise, a corresponding error message will be issued.




A Signal inputs for control and setpoint signals
B Signal flow of the integrated technology application (see the following subchapter)
C Signal outputs for status and actual value signals

[2-1] Adaptation of the drive solution via parameter setting

## 2.1 Integrated technology applications

The following technology applications integrated in the inverter 8400 TopLine provide the main signal flow for the implementation of a general or a special drive solution:



**Technology application "Actuating drive speed"**

This preset technology application serves to solve speed-controlled drive tasks, e.g. conveyor drives (interconnected), extruders, test benches, vibrators, travelling drives, presses, machining systems, metering units.



**Technology application "actuating drive speed (AC Drive profile)"**

This technology application available from version 13.00.00 provides a speed and torque control by means of "AC Drive Profile". The fieldbuses EtherNet/IP™ and system bus (CANopen) are supported.



**Technology application "Table positioning"**

This technology application serves to solve position-controlled drive tasks which are normally controlled by a higher-level control via a fieldbus, e.g. transport facilities, rotary tables, storage and retrieval units, feed drives, metering units, hoists.

**Note:** This TA requires an external sequence control!
From version 12.00.00, the FB L_Sequencer_1 can also be used as internal sequence control. This FB processes a positioning program based on a sequence table.



**"Switch-off positioning" technology application**

This technology application is used to solve speed-controlled drive tasks which require a pre-switch off or stopping at certain positions, e.g. roller conveyors and conveying belts. The pre-switch off is implemented by connecting switch-off sensors.



**Note!**

Please note that the "StateLine", "HighLine" and "TopLine" device types differ with regard to the number, functional range, and flexibility of the technology applications offered.



Detailed information on each technology application can be found in the main chapter entitled "Technology applications". (📖 436)

### 2.1.1 Purpose of the technology applications

The stepped Inverter Drives 8400 series provides solutions for simple to complex applications – depending on the user's experience and knowledge about the handling of drives and drive tasks.

On the one hand, a great scope of standard drive tasks for frequency inverters is covered by the technology applications offered by Lenze, and on the other hand, the user is relieved from time-consuming programming activities. In practice, some drive tasks are alike so that minor modifications of the corresponding technology applications lead to quick results.

Other important features of technology applications are:

- Direct implementation of drive tasks without recreating a function block interconnection inside the device
- Operation via keypad and/or operation via convenient operator dialogs in the «Engineer».
- Commissioning via few operating and diagnosing parameters (local keypad operation).
- Achieving a transparency as high as possible via the integrated functionality of the device by representing signal flow diagrams.
- Provision of a basic functionality suitable and often sufficient for many applications.

### 2.1.2 Application cases for a technology application

You should use a technology application if

- the task can be solved completely or to a great extent by the basic functionality of the technology application.
- the end customer does not want to create the comprehensive core functions of the corresponding technology on his own.
- the creation time for a project is to be reduced by using the ready-made technology application
- the end customer wants to build upon the know-how of Lenze.

**Tip!**

If the end customer of the machine does not want to use ready-made Lenze functions, it is also possible to implement individual drive solutions with the "StateLine C", "HighLine C" and "TopLine C" versions in the form of "free interconnections". (In case of "StateLine C", free interconnection is only possible from version 12.00.00 and »Engineer« V2.17.)

Here, a technology application can be used as starting basis, which has to be adapted to the requirement by a change or extension via function block editor (see the following chapter).

### 2.1.3 Technology application = function block interconnection

In case of the 8400 device version, each technology application is connected to a "function block interconnection" ("FB interconnection"). This FB interconnections serve to implement signal interconnections. Various FBs are available for digital signal processing, signal conversion and logic modules.

For special drive tasks it has proved of value to use the integrated technology applications as a basis for modifications or extensions of the available FB interconnections.

#### I/O level & application level

The interconnection of the interfaces is shown in the I/O level of the function block editor according to the selected control mode. In the "deeper" application level, the main signal flow is realised in the form of an interconnection of various function and system blocks.

For the device versions "StateLine C", "HighLine C" and "TopLine C" the following applies:

- The preconfigured signal links can be reconfigured in the I/O level if required.
- Moreover, the applications integrated in the inverter can be reconfigured and extended by individual functions.
- Moreover, experienced users are offered the opportunity to implement their own drive solutions independent of the predefined technology applications by using so-called "free interconnections". (In case of "StateLine C", free interconnection is only possible from version 12.00.00 and »Engineer« V2.17.)

#### Motion control kernel

Important basic (drive) functions as well as further basic functionalities are implemented in the firmware of the inverter in the so-called **Motion Control Kernel** (MCK) which can be accessed by the active technology application via defined internal interfaces. By this means the expensive creation of single function block interconnections is omitted so that the expenditure and the complexity for the realisation of standard functions is minimised.

The **Motion Control Kernel** is integrated in the main setpoint path and, depending on the set operating mode, it creates the required control and setpoint signals for the motor control and the drive interface.



More information:

- A detailed description of the basic functions implemented in the **Motion Control Kernel** can be found in the main chapter "Basic drive functions (MCK)". (📖 560)
- Detailed information on the creation or change of interconnections by means of the function block editor can be found in the main chapter "Working with the FB Editor". (📖 1373)
- All available function and system blocks are described in the main chapter "Function library". (📖 1425)

## 2.2 Selection of the appropriate commissioning tool

There are several possibilities for commissioning the 8400 TopLine inverter:



**Commissioning via keypad X400 (or diagnosis terminal X400)**

The keypad is an alternative to the PC for the local operation, parameterisation, and diagnostics in a simple manner. The keypad is especially suited for test and demonstration purposes and for the case that only few parameters have to be adapted.



**Commissioning with PC and »EASY Starter«**

The »EASY Starter« is a Lenze tool for easy online diagnostics, parameter setting and commissioning of the inverter.



**Commissioning with PC and »Engineer«**

The »Engineer« is a Lenze engineering software for parameter setting across all devices, configuring and diagnosing individual components (as for instance inverters, industrial PCs, motors, I/O systems) and machine control systems.



**Tip!**

The Engineering tools »EASY Starter« and »Engineer StateLevel« are provided free of charge in the internet:

http://www.lenze.com → Download → Software Downloads

For communication between PC and inverter, the USB diagnostic adapter can be used for instance (see the following subchapter).

### 2.2.1 Overview: Accessories for commissioning

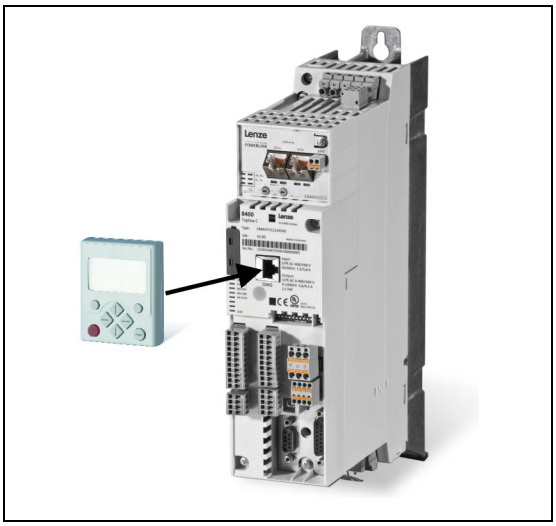
| Version | Features | Product key |
|---|---|---|
| Keypad X400  | Quick access to parameters and operating data<br>• Supports hot plugging<br>• Graphic display with plain texts<br>• Backlighting<br>• Easy user guidance<br>• 4 navigation keys, 2 context-sensitive keys<br>• Adjustable RUN/STOP function<br>• Can be used for L-force Inverter Drives 8400 and Servo Drives 9400 | EZAEBK1001 |
| Diagnosis terminal X400  | Keypad X400 in a robust housing, also suitable for installation into the control cabinet door.<br>• Supports hot plugging<br>• Graphic display with plain texts<br>• Backlighting<br>• Easy user guidance<br>• 4 navigation keys, 2 context-sensitive keys<br>• Adjustable RUN/STOP function<br>• Incl. 2.5 m cable<br>• Enclosure IP20; in case of front installation in control cabinet IP65<br>• Can be used for L-force Inverter Drives 8400 and Servo Drives 9400 | EZAEBK2001 |
| USB diagnostic adapter  | For electrical isolation of your PC and the inverter.<br>• Supports hot plugging<br>• Diagnostic LED for data transfer display<br>• plug and play<br>• Input-side voltage supply via USB connection from PC<br>• Output-side voltage supply via the diagnostic interface of the inverter<br>• Connecting cables can be selected in various lengths: | E94AZCUS |
| Connecting cable for USB diagnostic adapter | 2.5 m length | EWL0070 |
| | 5 m length | EWL0071 |
| | 10 m length | EWL0072 |

## 2.3 General notes on parameters

All parameters for inverter parameterising or monitoring are saved as so-called "codes".

- The codes are numbered and indicated by the prefix "C" before the code, e.g. "C00002".
- Moreover, each code has a name and specific attributes, as for example access type (reading, writing), data type, limit values and default setting ("Lenze setting").
- For the sake of clarity, some codes contain "subcodes" for saving parameters. This Manual uses a slash "/" as a separator between code and subcode, e.g. C00118/3".
- According to their functionality, the parameters are divided into three groups:
  - Setting parameters: For specifying setpoints and for setting device / monitoring functions.
  - Configuration parameters: For configuring signal connections and terminal assignments.
  - Diagnostic/display parameters: For displaying device-internal process factors, current actual values and status messages. These are read-only parameters.

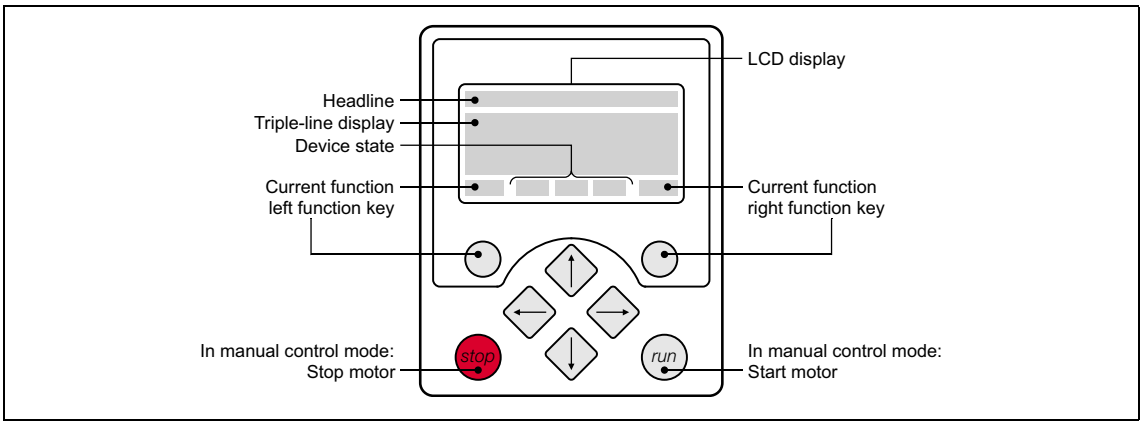
### 2.3.1 Changing the parameterisation with the keypad



The keypad is simply plugged on the diagnostic interface X6 ("DIAG") at the front of the standard device.

Plugging and unplugging the keypad is possible during operation.

**Keypad display and control elements**




| LCD display | | | |
|---|---|---|---|
| **Headline** | | | |
| In the menu level: Menu name<br>In the parameter level: Parameter name | | | |
| **Three-part display** | | | |
| In the menu level: List of available menus<br>In the parameter level: Code/subcode and setting or actual value | | | |
| **Device status** | | | |
| RDY | Inverter is switched on | IMP | Pulse inhibit active |
| RUN | Inverter is enabled | !SFLT | System fault active |
| CINH | Inverter is inhibited | !FLT | "Fault" device status is active |
| QSP | Quick stop active | !TRB | "Trouble" device status is active |
| Imax | Current limit exceeded | !TQSP | "TroubleQSP" device status is active |
| Mmax | Speed controller 1 in the limitation | WRN | A warning is indicated |

| LCD display | | | |
|---|---|---|---|
| **Function - left function key** | | **Function - right function key** | |
| EDIT | Change parameter setting<br>(change to editing mode) | OK | Accept change in the inverter<br>(no saving with mains failure protection → SAVE) |
| ☰ | Back to main menu | ESC | Abort (discard change) |
| CINH!! | Parameter can only be changed when the inverter is inhibited. | | |
| SAVE | Save all parameter settings in the memory module safe against mains failure | | |

| Control elements | |
|---|---|
| ○ | Execute the function assigned to the function key (see LCD display) |
| stop | Execute the stop function set in C00469 (Lenze setting: Inhibit inverter) |
| run | Deactivate stop function again (Lenze setting: Enable inverter again) |
| ↑<br>↓ | In the menu level: Select menu/submenu<br>In the parameter level: Select parameter<br>In the editing mode: Change marked digits or select list entry |
| CINH | In the menu level: Select submenu/change to parameter level<br>In the editing mode: Cursor to the right |
| RDY | In the menu level: One menu level higher (if available)<br>In the parameter level: Back to the menu level<br>In the editing mode: Cursor to the left |

### Menu structure

In the keypad, the parameters are classified into various menus and submenus.

- The **USER menu** includes a selection of frequently used parameters.
- The **Code list** contains all parameters.
- The **Go to param** function enables you to reach the corresponding parameter directly.
- The **Logbook** logs all errors and their chronological history.
- The **Diagnostics** menu contains diagnostic/display parameters for displaying device-internal process factors, current actual values and status messages.

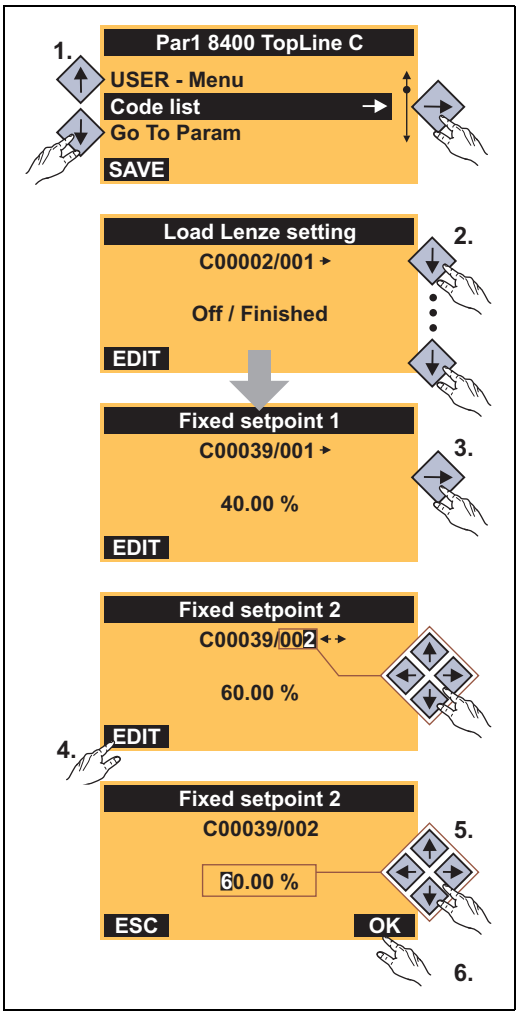
### User level

From version 12.00.00 onwards, the extent of menus, submenus an codes shown in the keypad can be adapted by selecting the "Userlevel" in C00001/1:

- **Standard** user level (Lenze setting): Only the most important menus and codes are displayed in the keypad.
- **Expert** user level: All menus and codes are displayed in the keypad.
- **Service** user level: Only for the purpose of service (Lenze Service).

After changing the user level, the menus in the keypad are restructured according to the selected user level. The parameters of plugged-in communication module are always shown completely independent of the set user level.

### General operation



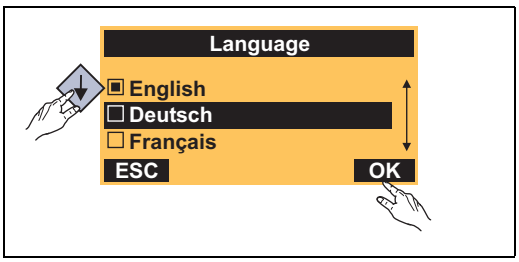

[2-2] Example: Changing parameters with the keypad

1. Use the ↑/↓ navigation keys to select the desired menu.
   - Use the ←/→ navigation keys to reach a higher/lower menu level.
   - Use the ☰ function key to return to the main menu.
2. Use the ↑/↓ navigation keys to select the parameter to be set within a submenu.
3. In order to select another subcode in case of a parameter with subcodes:
   - Press the navigation key → to change to the editing mode for the subcode.
   - Use the navigation keys to set the desired subcode.
4. Use the EDIT function key to switch over to the editing mode.
5. Use the navigation keys to set the desired value.
6. Use the OK function key to accept the change and to leave the editing mode.
   - Use the ESC function key to leave the editing mode without accepting the change.

### Multilingualism

All texts displayed in the keypad are in English.

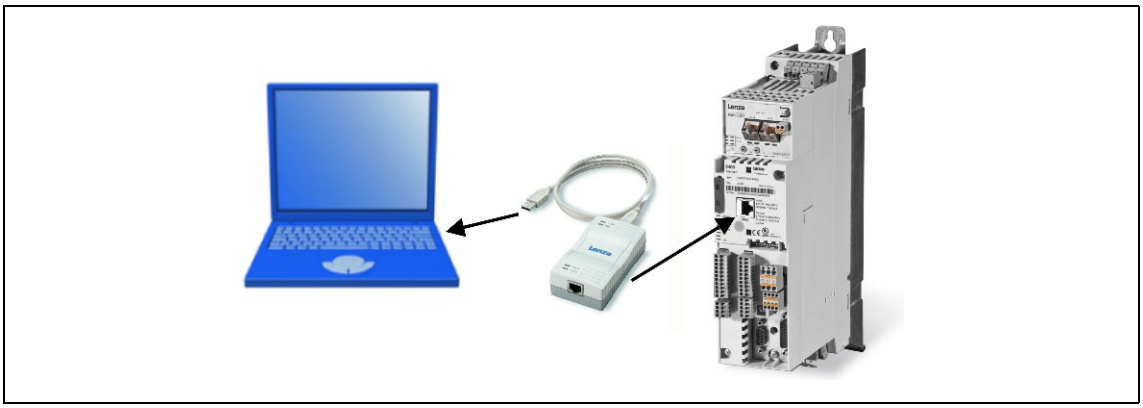
From version 02.00.00 onwards, the most important menus as well as diagnostic and configuration parameters can are also available in German and French. To set a different language, select the **Language selection** menu item in the main menu of the keypad.




- The multilingual texts are stored in the inverter and do not have to be loaded into the device.
- For reasons of disc space, only the most important menus and parameters as well as the error messages are available in several languages.

### 2.3.2 Change parameter settings with PC and Lenze software

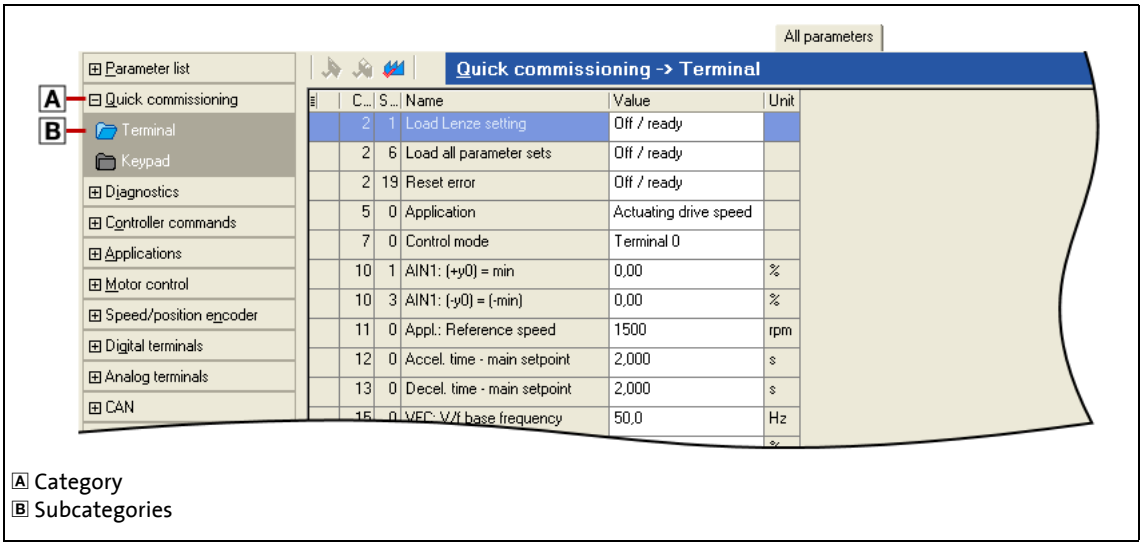
The USB diagnostic adapter, for instance, can be used for the communication between the PC (including the »EASY Starter« or » Engineer« software) and the inverter, see the following illustration. The USB diagnostic adapter is the connection between the PC (free USB port) and the inverter (X6 diagnostic interface).



[2-3] Exemplary constellation for parameterising the inverter

The **All parameters** tab in the »EASY Starter« and the »Engineer« provides a quick access to all parameters of the inverter.

The given categories and subcategories correspond 1:1 to the menus and submenus of the keypad:




A Category
B Subcategories

[2-4] **All parameters** tab in the »Engineer«

Moreover, the »Engineer« provides a commissioning interface on the **Application parameters** tab where you can commission the application in a few steps.

Detailed information on how to handle the »Engineer« can be found in the integrated online help that you can call with the **[F1]** function key.

### 2.3.3 Save parameter settings in the memory module safe against mains failure

Inverter parameter changes via the EASY Starter /»Engineer«, the keypad, or a master control via fieldbus communication will be lost after mains switching of the controller unless the settings have been explicitly saved to the integrated memory module.

**General information**

- In the delivery state, the Lenze setting of the parameters has been saved to the integrated memory module. These parameters are
  - the parameters of the inverter
  - the parameters of the communication module plugged into the MCI interface
  - the parameters of the possibly existing safety module (device variant)
- When the device or the external 24 V voltage supply is switched on, all parameters are automatically loaded from the memory module into the main memory of the inverter.
- Full functionality of the memory module is even provided if the power supply has been switched off and only the electronic components of the inverter are externally supplied by a 24 V DC voltage, e.g. via the X4/24E terminal.
- The memory module can be preconfigured with customised data.
- The memory module is available as a spare part - without any data.

**During operation**

- Parameter sets can be saved and loaded manually.
- Using the keypad, you can press function key SAVE to save the parameter settings.
- The »EASY Starter«/»Engineer« serves to execute the saving via the icon in the *toolbar* or via the device command "Save all parameter sets" (C00002/11 = "1: On / start").
  - The storage process may take a couple of seconds. After the device command has been called in C00002/11, dynamic status information ("Work in progress 20%" → "Work in progress 40%" → "Work in progress 60%", etc.) is returned.

**Note!**

In order to prevent data inconsistencies during the saving process:
- Do not switch off the supply voltage!
- Do not remove the memory module from the device!

Automatic saving of changed parameter settings is explicitly not supported because this significantly reduces the service life of the memory module.

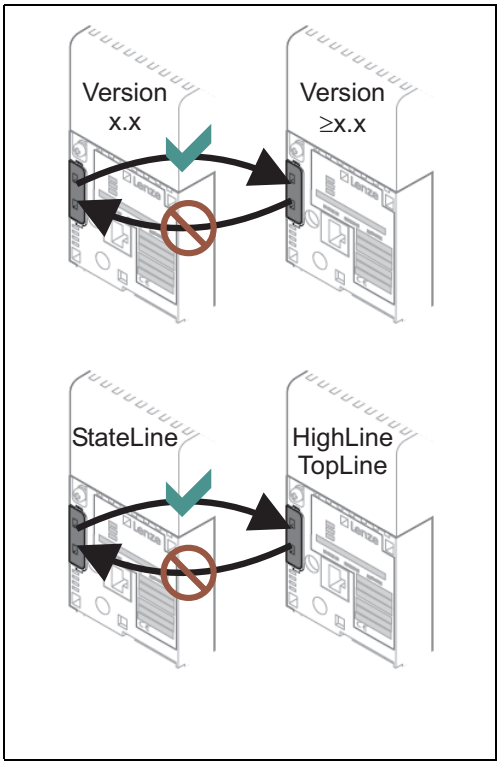
### Unplugging the memory module

The memory module is hot-pluggable. A removal during operation causes a warning "PS01: No memory module" and should thus be avoided. The behaviour of the device, however, remains unchanged as all parameters are available in the RAM memory after the device has been started. The device can also be parameterised when the memory module has been unplugged. In this case, the parameter sets cannot be saved in the memory module.

### Replacement of the inverter

In the event of a device replacement, the entire parameter data of an axis can be copied to the replacement device by "taking along" the memory module, so that additional PC or keypad operations are not required.

When replacing the inverter, the versions of the old device and the new device are of importance. Before data are actually transferred, the versions are internally checked. Basically, the following applies:




- Parameter sets of old devices can only be processed on new devices with the same or higher device version (downward compatibility).
- Parameter sets of devices with versions that have less functions (e.g. 8400 StateLine) can be loaded into and executed on devices with versions that have more functions (e.g. 8400 HighLine). The reverse is not possible!
- If the parameter set stored in the memory module is incompatible with the standard device, the "PS03: Par.set device invalid" error message is output and the inverter changes to the "Fault" device state.
- If the parameter set stored in the memory module is compatible with the standard device but has a different (lower) version, the "PS03: Par.set device invalid" message is only output as "Information". The message can be eliminated by storing the parameter set again.
  **Note:** If you save the parameter set in a higher device version, you can no longer load this parameter set to a lower device version.

Example of parameter set compatibilities:

| Parameter version on memory module | Device version | | | |
|---|---|---|---|---|
| | StateLine V12 | HighLine V12 | HighLine V13 | TopLine V12 |
| StateLine V12 | OK | PS03 (Information) | PS03 (Information) | PS03 (Information) |
| HighLine V12 | PS03 (Fault) | OK | PS03 (Information) | PS03 (Information) |
| HighLine V13 | PS03 (Fault) | PS03 (Fault) | OK | PS03 (Fault) |
| TopLine V12 | PS03 (Fault) | PS03 (Fault) | PS03 (Fault) | OK |

### 2.3.4 User menu for quick access to frequently used parameters

When a system is installed, parameters must be changed time and again until the system runs satisfactorily. The user menu of the inverter contains a selection of frequently used parameters to be able to access and change these parameters quickly:

| Parameters | Name | Lenze setting |
|---|---|---|
| C00051 | MCTRL: Actual speed value | - |
| C00053 | DC-bus voltage | - |
| C00054 | Motor current | - |
| C00061 | Heatsink temperature | - |
| C00137 | Device status | - |
| C00166/3 | Mess. - status det. error | - |
| C00011 | Appl.: Reference speed | 1500 rpm |
| C00039/1 | Fixed setpoint 1 | 40.00 % |
| C00039/2 | Fixed setpoint 2 | 60.00 % |
| C00012 | Acceleration time - main setpoint | 2.000 s |
| C00013 | Deceleration time - main setpoint | 2.000 s |
| C00015 | VFC: V/f base frequency | 50 Hz |
| C00016 | VFC: Vmin boost | 1.60 % |
| C00022 | Imax in motor mode | depending on the device power |
| C00120 | Setting of motor overload ($I^2xt$) | 100.00 % |
| C00087 | Rated motor speed | 1460 rpm |
| C00099 | Firmware version | - |
| C00200 | Firmware product type | - |
| C00105 | Decel. time - quick stop | 2.000 s |
| C00173 | Mains voltage | 0: "3ph 400V / 1ph 230V" |
| Greyed out = display parameter | | |

**Tip!**

The user menu can be freely configured in C00517.

In the »Engineer«, you can configure the user menu comfortably via the **User menu** tab (see »Engineer« online help).

The password protection serves to restrict the access to parameters of the user menu. Then, all other parameters cannot be accessed without knowing the password and are thus protected against unwanted changes.

## 2.4 Device access protection

Various tasks can be executed via the functions of the device access protection:

- Password protection
  - Only authorised persons (with password knowledge) may read/change all parameters of the inverter.
  - Non-authorised persons (without password knowledge) can only access the max. 32 parameters of the user menu.
- Device personalisation
  - Only inverters and memory modules personalised with a specific binding ID can be used in the system.

**Note!**

If password protection/device personalisation is used:

- Inform the end customer that Lenze can only provide restricted service for the devices with access protection.
- It is not possible for Lenze to modify a replacement device via special accesses in such a way that it cooperates with a personalised memory module.
- From firmware version 4.2 onwards, the X400 keypad supports the alphanumeric entry of a password. Keypads with lower firmware versions cannot be used for the entry.

### 2.4.1 Password protection

When the password protection is active, only write/read access to the parameters of the user menu is possible.

- From version 15.00.00, the password protection can be configured individually for every single communication channel. ▸ Individual password protection for single communication channels (📖 49)
- The following describes how to set/check/delete a password by means of the parameters relevant for these functions.
- From »Engineer« V2.14 onwards. these functions can also be executed via dialog (menu command **Online → Set/check/delete password**).

**Note!**

The password protection serves to only limit the access to parameters of the standard device. The access to parameters of a plugged-in communication module is not concerned.

**Short overview of the relevant parameters for password protection:**

| Parameters | Info | | Lenze setting |
|---|---|---|---|
| C00505/3 | Password<br>• The password must have a length of 4 ... 16 characters.<br>• The password may consist of any character. However, this is not sensible. Recommended characters are: lower case letters (a - z), upper case letters (A - Z) and digits (0 - 9).<br>**Note:**<br>After the execution of one of the device commands listed below, this parameter provides the current password status: | | |
| | off | No password is set, password protection is not active (Lenze delivery status). | |
| | on | Password is set, password protection is active.<br>• This status is also displayed if checking/deleting the password has not been successful due to an invalid entry. | |
| | ok | Password is set, password protection is not active.<br>• The password protection is temporarily deactivated. | |
| **Device commands**<br>Before the following device commands are executed, enter the corresponding password in C00505/3. | | | |
| C00002/31 | Set password<br>▸ Activate the password protection | | 0: Off / ready |
| C00002/32 | Check password<br>▸ Temporarily deactivate the password protection | | 0: Off / ready |
| C00002/33 | Delete password<br>▸ Deactivate password protection/change password | | 0: Off / ready |
| Greyed out = display parameter | | | |

| Parameters | Info | | Lenze setting |
|---|---|---|---|
| **Status displays** | | | |
| C00003 | Status of the last device command | | - |
| C00507/1 | Password protection - all communication channels<br>• Bit coded display of the active protective functions: | | - |
| | Bit 0 | Only access to user menu | |
| | Bit 1 | Parameter write protection | |
| | Bit 2 | Parameter read protection | |
| | Bit 3 ... 14 | Reserved | |
| | Bit 15 | Memory module binding on | |
| | **Note:**<br>As the password protection can be configured individually for each single communication channel from version 15.00.00 bit 1 and bit 2 indicate the active protection always with regard to the communication channel used. | | |
| Greyed out = display parameter | | | |

### Activate the password protection

The password protection is activated by setting a password.

**How to set a password:**

1. Enter the desired password in C00505/3.
   - The password must have a length of 4 ... 16 characters.
   - The password may consist of any character. However, this is not sensible. Recommended characters are: lower case letters (a - z), upper case letters (A - Z) and digits (0 - 9).
2. Execute "Set password" device command: C00002/31 = "1: On / start"
   - After successful execution, password status ON is displayed in C00505/3 and password protection takes immediate effect.

### Temporarily deactivate the password protection

The "Check password" device command serves to temporarily deactivate the password protection in order to execute password-protected functions.

- The password protection remains deactivated until
  - an invalid password will be entered and checked
    - or -
  - the external 24-V supply of the control electronics is switched off (< 19 V).

**How to temporarily deactivate the active password protection:**

1. Enter the set password in C00505/3.
2. Execute "Check password" device command C00002/32 = "1: On / start"
   - After a successful check, password status OK is displayed in C00505/3.

**Deactivate password protection/change password**

The password protection is simply activated by deleting the set password. If you want to change the set password, first delete the set password as well. Then set the new password.

**How to delete the set password:**

1. Enter the set password in C00505/3.
2. Execute "Delete password" device command C00002/33 = "1: On / start"
   - After a successful deletion, password status OFF is displayed in C00505/3.

## 2.4.2 Individual password protection for single communication channels

This function extension is available from version 15.00.00!

The 8400 TopLine can be parameterised via the following "communication channels":

- Internally via system blocks (LS_ParReadWrite1-6 and LS_WriteParamList)
- externally via
  - Diagnostics interface X6 ("DIAG")
  - CANopen interface X1 ("CAN on board")
  - Fieldbus interface (MCI)

As in some applications, only certain communication channels have to be protected, the parameters described in the following serve to configure individual access restrictions for each communication channel.

In the Lenze setting, all access restrictions are active for all communication channels when the password is set. This corresponds to the previous behaviour with a set password, i.e. only one write/read access to the parameters of the user menu is possible.

**Note!**

The configuration can only be changed if no password has been set!

- This means that even if the password has been deactivated temporarily by the "check password" device command, the configuration cannot be changed.
- As long as a password is set, the "Load Lenze setting" device command does not change the configuration either.

The password protection is activated by setting a password.

▸ Activate the password protection (📖 48)
▸ Deactivate password protection/change password (📖 49)

**Short overview of the relevant parameters for configuring the password protection:**

When the password is set, the respective restriction is active if the corresponding bit has been set:

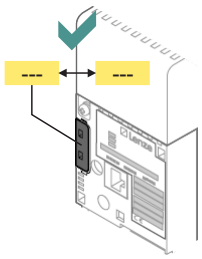
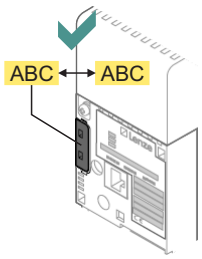
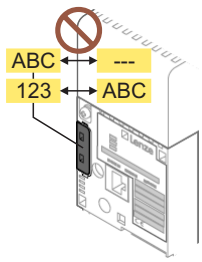
| Parameters | Info | | Lenze setting |
|---|---|---|---|
| C00506/1 | PW protection internal config | | 0x0007 |
| | Bit 0 | Only access to user menu<br>• The menus in the keypad are reduced.<br>• Only the user menu can be used to change parameters. | |
| | Bit 1 | Parameter write protection<br>• System blocks cannot be used to write into write/read parameters. | |
| | Bit 2 | Parameter read protection<br>• System blocks cannot be used to read write/read parameters. | |
| | Bit 3 ... 15 | Reserved | |
| C00508/1 | PW protection config. X6 (DIAG) | | 0x0006 |
| | Bit 0 | Reserved | |
| | Bit 1 | Parameter write protection<br>• The diagnostics interface X6 cannot be used to write into write/read parameters. | |
| | Bit 2 | Parameter read protection<br>• The diagnostics interface X6 cannot be used to read write/read parameters. | |
| | Bit 3 ... 15 | Reserved | |
| C00509/1 | PW protection config. X1 (CAN) | | 0x0006 |
| | Bit 0 | Reserved | |
| | Bit 1 | Parameter write protection<br>• The CANopen interface X1 cannot be used to write into write/read parameters. | |
| | Bit 2 | Parameter read protection<br>• The CANopen interface X1 cannot be used to read write/read parameters. | |
| | Bit 3 ... 15 | Reserved | |
| C00510/1 | PW protection config. MCI | | 0x0006 |
| | Bit 0 | Reserved | |
| | Bit 1 | Parameter write protection<br>• The fieldbus interface (MCI) cannot be used to write into write/read parameters. | |
| | Bit 2 | Parameter read protection<br>• The fieldbus interface (MCI) cannot be used to read write/read parameters. | |
| | Bit 3 ... 15 | Reserved | |

The parameter set must be saved to the device safe against mains failure to prevent password protection configured in the device from getting lost due to mains switching.

▸ Save parameter settings in the memory module safe against mains failure (📖 43)

### 2.4.3 Device personalisation

The inverter and the memory module are married via the device personalisation by means of a binding ID. When the device personalisation is active, all write/read actions between inverter and memory module are only executed if both components have the same binding ID.

| Lenze delivery status: | Procedure carried out by the customer: | Impermissible replacement by the end user: |
|---|---|---|
|  |  |  |
| No binding ID is set. | Customer sets binding ID for device personalisation. | When device personalisation is active:<br>The replacement of the inverter or memory module causes an error message if the binding ID is incorrect or not available. |

If, for instance, a parameter set has been loaded from the memory module with an active device personalisation, saving of this parameter set on another memory module with a different or non-existent binding ID is not possible.

- Thus, copying the parameter set from a personalised memory module to a non-personalised memory module is not possible!

Two types of checks are distinguished:

- If a differing binding ID is detected when the inverter is switched on (during device initialisation):
  - the "Fault" error response is returned.
  - The "PS10: Invalid memory module binding" error message is entered into the logbook.
- If a differing binding ID is detected while a device command for loading/saving the parameter set is executed:
  - Loading/saving is not executed.
  - A corresponding status for the device command is output in C00003.

**Short overview of the relevant parameters for device personalisation:**

| Parameters | Info | | Lenze setting |
|---|---|---|---|
| C00505/2 | Binding ID<br>• The binding ID must have a length of 4 ... 16 characters.<br>• The binding ID may consist of any character. However, this is not sensible. Recommended characters are: lower case letters (a - z), upper case letters (A - Z) and digits (0 - 9).<br>**Note:**<br>After the execution of one of the device commands listed below, this parameter provides the current binding ID status: | | |
| | off | No binding ID is set. | |
| | on | Binding ID is set. | |
| Greyed out = display parameter | | | |

| Parameters | Info | Lenze setting |
|---|---|---|
| **Device commands**<br>Before the following device commands are executed, enter the corresponding binding ID in C00505/2. | | |
| C00002/29 | Set binding ID<br>▸ Activate device personalisation | 0: Off / ready |
| C00002/30 | Delete binding ID<br>▸ Deactivate device personalisation/change binding ID | 0: Off / ready |
| **Status displays** | | |
| C00003 | Status of the last device command | - |
| C00507/1 | Password protection - all communication channels<br>• Bit coded display of the active protective functions: | - |
| | Bit 0: Only access to user menu | |
| | Bit 1: Parameter write protection | |
| | Bit 2: Parameter read protection | |
| | Bit 3 ... 14: Reserved | |
| | Bit 16: Memory module binding on | |
| Greyed out = display parameter | | |

### Activate device personalisation

The device personalisation is activated by setting a binding ID.



**How to set the binding ID:**

1. Enter the desired binding ID in C00505/2.
   - The binding ID must have a length of 4 ... 16 characters.
   - The binding ID may consist of any character. However, this is not sensible. Recommended characters are: lower case letters (a - z), upper case letters (A - Z) and digits (0 - 9).
2. Execute "Set binding ID" device command C00002/29 = "1: On / start"
   - After successful execution, status ON is displayed in C00505/2.

### Deactivate device personalisation/change binding ID

The device personalisation is simply deactivated by deleting the set binding ID. If you want to change the set binding ID, first delete the set binding ID as well. Then set the new binding ID.



**How to delete the binding ID:**

1. Enter the set binding ID in C00505/2.
   - If inverter and memory module do not have the same binding ID, enter the binding ID of the memory module to delete the binding ID of both components.
2. Execute "Delete binding ID" device command C00002/30 = "1: On / start"
   - After a successful deletion, status OFF is displayed in C00505/2.

### 2.4.4 Unlocking the inverter with a MasterPin

Every inverter has an individual master password called "MasterPin". By entering the MasterPin, an inverter inhibited by the password mechanisms can be reset to the delivery status.



**Stop!**

When the MasterPin is used, the parameter set is reset to the Lenze setting both in the inverter and in the memory module!

- This results in a permanent loss of the customised parameterisation that must be recreated!
- A reset to the Lenze setting can result in unforeseen level changes at the I/O terminals (e.g. brake control)!



**How to restore the delivery status:**

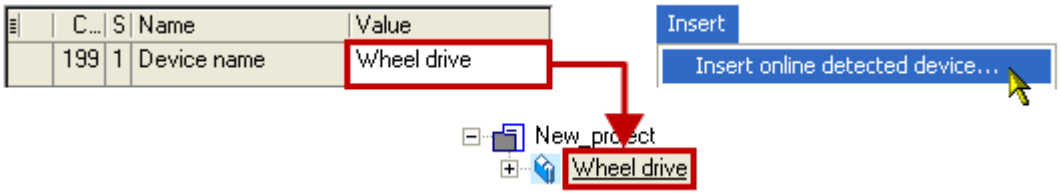
1. Inhibit the inverter if it is enabled, e.g. via the C00002/16 device command.
2. Enter the MasterPin in C00505/1.
   - The MasterPin comprises the last 6 digits of the serial number of the memory module.
3. Execute "Check MasterPin" device command C00002/28 = "1: On / start"

## 2.5 Device identification

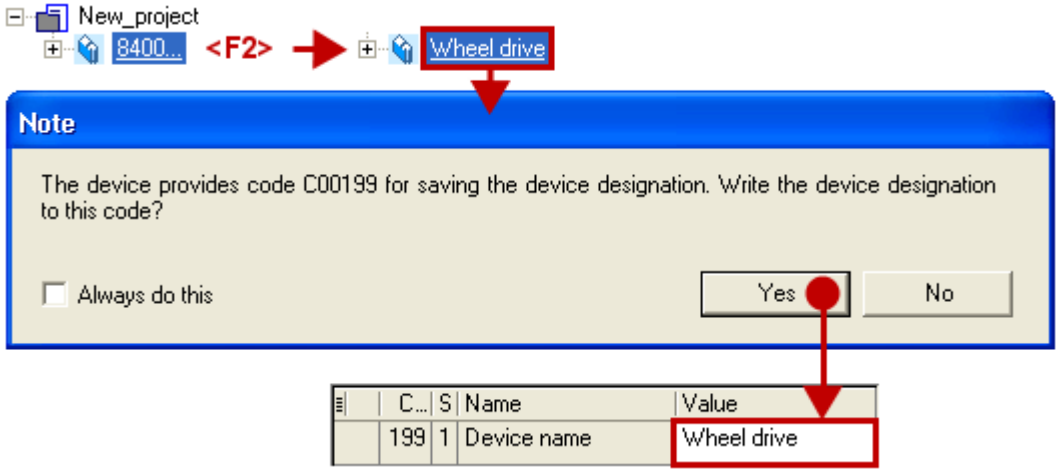
For device identification, any device name (e.g. wheel drive) with max 32 characters can be set in C00199/1 for the inverter and saved in the memory module with mains failure protection.

### 2.5.1 Automatic acceptance of the device name in the »Engineer«

If a device name is assigned in C00199/1 and the inverter in the »Engineer« is added to the project via the **Insert → Insert device detected online...** function, the device name stored in C00199/1 (here: wheel drive) is used as device designation in the *Project view* instead of the type (8400 TopLine):




This mechanism also functions in reverse direction:
If you rename the inverter in the *project view* via **<F2>**, you will be asked afterwards if you want to take over the changed name in C00199/1:




### 2.5.2 Extended item designation

This function extension is available from version 12.00.00!

In C00199/2...5, four texts à max. 32 characters can be set for item designation and saved in the memory module safe against mains failure. These additional subcodes are not dependent on the previously described text acceptance mechanism in the »Engineer« for the device name (C00199/1).

# 3 Commissioning

**Danger!**

**Unexpected motor movements can occur**

Under certain conditions the motor may rotate after mains connection.

**Possible consequences:**

- Persons in the vicinity of the machine or plant risk getting hurt.
- Unexpected starting action may damage the machine or plant.

**Protective measures:**

- Commissioning with external 24 V supply and without mains voltage. In this case, the inverter can only be parameterised and diagnosed during commissioning.
- Ensure that setpoints are not active.

**Tip!**

- Information on some of the operating statuses can quickly be obtained via the LED status displays on the front of the inverter. (📖 692)
- **Check firmware:** Particularly with regard to the use of an older inverter (e.g. if the customer is using one from stock) it makes sense to check the software (firmware) version. The software version of the inverter can be seen on the nameplate in the "SW" line and can be determined by reading out code C00099.
- **Restore delivery status:** Set code C00002/1 to "1: On / start" to reset all parameter settings of the device to the Lenze setting. This leaves you with a defined device configuration. ▸ Load Lenze setting (📖 107)

The following chapters describe the commissioning of the available technology applications with the »Engineer«.

Information on how to commission using the keypad (or diagnosis terminal) can be found in the **8400 hardware manual**. The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 inverter.

## 3.1 Safety instructions with regard to commissioning

### General safety instructions

In order to prevent injury to persons or damage to material assets

- check before connecting the mains voltage
  - the wiring for completeness, short circuit, and earth fault
  - the "emergency stop" function of the entire system
  - that the motor circuit configuration (star/delta) is adapted to the output voltage of the inverter
  - the in-phase connection of the motor
- check the setting of the most important drive parameters before enabling the controller:
  - the V/f rated frequency must be adapted to the motor circuit configuration!
  - the drive parameters relevant for your application must be set correctly!
  - the configuration of the I/O terminals must be adapted to the wiring!
- ensure that there are no active speed setpoints before enabling the controller.

### Safety instructions with regard to motor operation

**Danger!**

- For thermal reasons, continuous operation of self-ventilated motors at a low field frequency and rated motor current is not permissible!
  - In the Lenze setting, the Motor temperature monitoring (PTC) is activated. (📖 304)
  - Activate the Brake resistor monitoring (I2xt) if necessary. (📖 305)
- C00015 must be used to select 87 Hz operation if a delta-connected asynchronous motor (nameplate data: 400 V ⅄ /230 V △) is to be operated in conjunction with an inverter for a mains voltage of 400 V.

## 3.2 Notes on motor control

In the Lenze setting, the V/f characteristic control (VFCplus) as motor control is set in C00006 with a linear characteristic.

- V/f characteristic control (VFCplus) is a motor control mode for classic frequency inverter applications on the basis of a simple and robust control procedure for the operation of machines with a linear or quadratic load torque characteristic (e.g. fans).
- The presettings of the parameters ensure that the inverter is immediately ready for operation and the motor works adequately without further parameterisation if an inverter and a 50 Hz asynchronous machine with matching performances are assigned to each other.

**Note!**

Check the nameplate data against the motor data set in the inverter. Further information is provided in the chapter "Motor selection/Motor data". (📖 139)

**Recommendations for the following application cases:**

- If the inverter and motor differ greatly in terms of performance:
  Set the Imax limit (in motor mode) in C00022 to 2x rated motor current.
- If a high starting torque is required:
  When the motor is idling, set a value for Vmin boost in C00016 which ensures that the rated motor current flows at a field frequency of f = 3 Hz (display in C00058).
- For noise optimisation:
  In C00018, set a switching frequency of "16 kHz var./drive-opt.".
- If a high torque must be provided at small speeds without feedback:
  Select "Sensorless vector control (SLVC) as motor control mode in C00006.

**Related topics:**

▸ Motor control (MCTRL) (📖 138)

## 3.3 Preconditions for commissioning with the »Engineer«

For commissioning, you need

- a PC that satisfies the following requirements:
  - processor with 1.4 GHz or higher
  - at least 512 MB RAM and 650 MB free hard disc space
  - Microsoft® Windows® 2000 operating system (from service pack 2 onwards) or Windows® XP
- the Lenze »Engineer« PC software
- a connection to the inverter, e.g. via a USB diagnostic adapter:
  - connect the USB diagnostic adapter to the X6 diagnostic interface.
  - establish a connection between the USB diagnostic adapter and the PC via a free USB port.



**Tip!**

How to obtain/update the »Engineer« software:

- **Download from the internet:** The full version of the »Engineer StateLevel« is provided free of charge in the internet:
  http://www.Lenze.com → Download → Software downloads
- **Requesting the CD**You can also request the »Engineer« separately on CD free of charge at your Lenze representative. See the "About Lenze" area on our homepage for e.g. the corresponding German address.

## 3.4 Trouble-shooting during commissioning

When the »Engineer« is used, trouble during commissioning can be detected and eliminated conveniently. Proceed as follows:

- Check whether error messages appear in the »Engineer«.
  - On the **Diagnostics** tab, relevant actual states of the inverter and pending error messages are displayed in a well-arranged visualisation.
- Check the input terminals for their corresponding setpoints.
  - The **Terminal assignment** tab displays the current input/output signals.
- Check the signal flow of the application.
  - For this purpose, click the **Signal flow** button on the **Application parameter** tab. The displayed signal flow shows active setpoints and their further processing.

**Related topics:**

- Diagnostics & error management (📖 691)
- LED status displays (📖 692)
- Error messages of the operating system (📖 715)

## 3.5 Commissioning wizard 8400

This function extension is supported by the »Engineer« from version 2.15 onwards!

The **commissioning wizard 8400** serves to carry out a guided commissioning of the inverter based on the Lenze setting of the parameters. The set parameters can then be saved in the inverter with mains failure protection.



**Note!**

Take all the necessary safety precautions before you carry out the following commissioning steps and switch the device on!

- Safety instructions with regard to commissioning (📖 56)

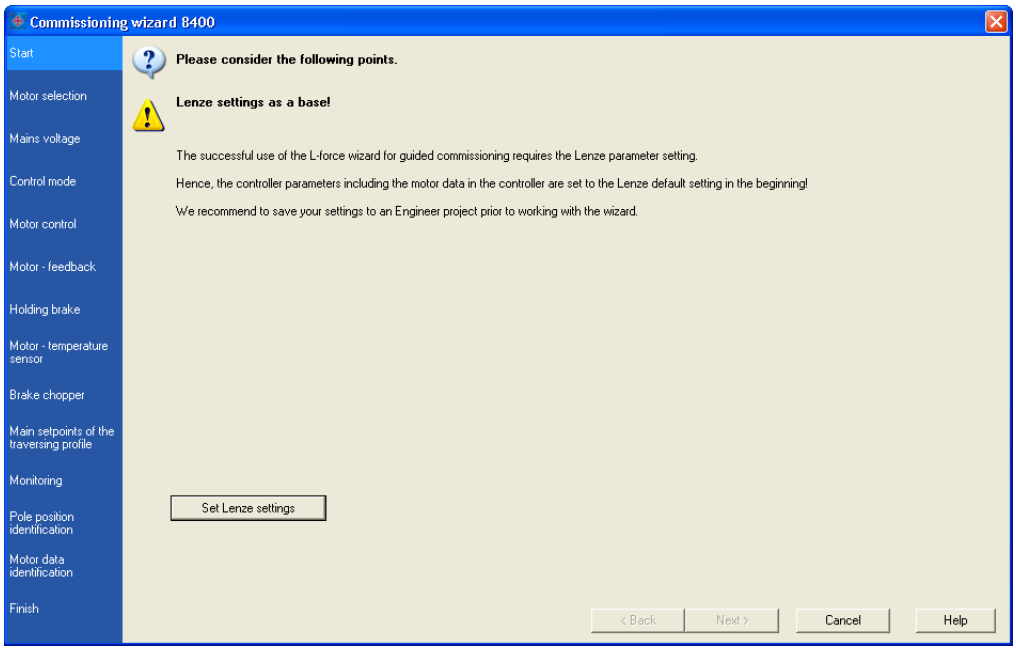
**How to carry out a guided commissioning using the »Engineer«:**

1. Go to the *Project view* and select the 8400 TopLine inverter.
2. Go online.

   After a connection to the inverter has been established, the following status is displayed in the *Status line*:

   ONLINE

3. Click the icon to open the *commissioning wizard 8400* dialog box.
   - Now the commissioning wizard guides you step by step through the setting of the important parameters for a quick commissioning.
   - The **Next** button can only be activated again after all parameter settings in the device have been reset via the **Load Lenze setting** button.




**Related topics:**

- Commissioning of the "Actuating drive speed" technology application (62)

## 3.6 Manual motor direction of rotation check (manual control)

The »Engineer« function "Manual control" easily allows for controlling the direction of rotation of the motor. The manual control serves to let the motor connected to the inverter with an adjustable speed for an adjustable time.

**Note!**

Take all the necessary safety precautions before you carry out the following commissioning steps and switch the device on!

▸ Safety instructions with regard to commissioning (📖 56)

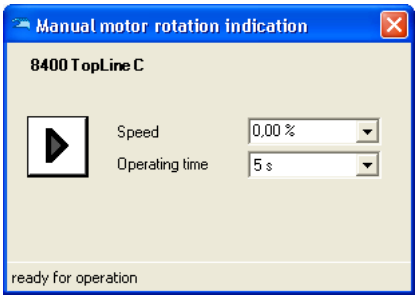
**How to carry out a manual motor direction of rotation check using the »Engineer«:**

1. Go to the *Project view* and select the 8400 TopLine inverter.
2. Go online.

   After a connection to the inverter has been established, the following status is displayed in the *Status line*:

   
   
3. Click the symbol to inhibit the inverter via device command.
4. Ensure that the following conditions are met:
   - The mains voltage is switched on.
   - No trouble is active.
   - Safe torque off (STO) is not active.
5. Enable inverter via terminal: Set terminal X5/RFR to HIGH level.
6. Click the icon to open the *Manual motor rotation indication* dialog box.

   **Note:** If the "not ready for operation" status is displayed, check whether all conditions mentioned before (see steps 3 ... 5) have been met.

   
   
7. Set the desired speed and runtime.
   (The speed refers to the reference speed set in C00011.)
8. Click the button to let the motor rotate with the set speed for the set runtime.

   By clicking the button, the function can be aborted.

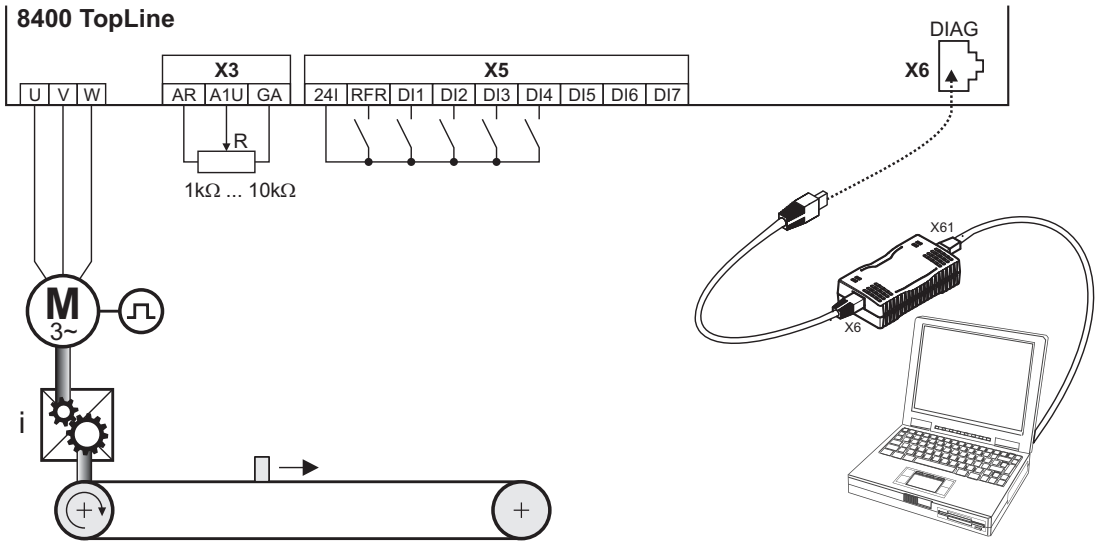
## 3.7 Commissioning of the "Actuating drive speed" technology application

**Note!**

Take all the necessary safety precautions before you carry out the following commissioning steps and switch the device on!

▸ Safety instructions with regard to commissioning (📖 56)

**System constellation**




| | |
|---|---|
| X3/AR | Reference voltage (10 V) for analog signals |
| X3/A1U | Speed setpoint selection (slider of setpoint potentiometer R)<br>• Scaling: 10 V ≡ 100 % ≡ 1500 rpm[1] (for a 4-pole motor) |
| X3/GA | Ground potential (GND) for analog signals |
| X5/RFR | Controller enable |
| X6 | Diagnostic interface for connecting the USB diagnostic adapter |

[3-1] Block diagram for wiring the commissioning example for the "Actuating drive speed" application

### Commissioning steps

Find a description of the commissioning steps of the "Actuating drive speed" technology application below.

Please observe the sequence of the steps in the following chapters and follow them through carefully. This will help you to commission your inverter quickly and as safely as possible:

▸ Prepare inverter for commissioning (📖 63)
▸ Creating an »Engineer« project & going online (📖 64)
▸ Parameterising the motor control (📖 65)
▸ Parameterise application (📖 66)

- Save parameter settings safe against mains failure (📖 68)
- Enable inverter and test application (📖 68)

### 3.7.1 Prepare inverter for commissioning

1. Power terminal wiring.
   Refer to the mounting instructions supplied with the inverter to find help on how to correctly design the power connections to match the requirements of your device.
2. Control terminal wiring.
   The assignment for your digital inputs should correspond to one of the preconfigured control modes (C00007) for terminal control:

| | Assignment of the digital terminals | | | |
|---|---|---|---|---|
| **Control mode** | **DI1** | **DI2** | **DI3** | **DI4** |
| Terminals 0 | JOG 1/3 | JOG 2/3 | DCB | Cw/Ccw |
| Terminals 2 | JOG 1/3 | JOG 2/3 | QSP | Cw/Ccw |
| Terminals 11 | Cw/Ccw | DCB | MPotUp | MPotDown |
| Terminal 16 | JOG 1/3 | JOG 2/3 | Cw/QSP | Ccw/QSP |

**Abbreviations used:**

| | |
|---|---|
| JOG | Selection of fixed setpoints 1 ... 3 parameterised in C00039/1...3 |
| DCB | Manual DC-injection braking |
| Cw/Ccw | CW/CCW rotation |
| QSP | Quick stop |
| MPotUp | Motor potentiometer: Increase speed |
| MPotDown | Motor potentiometer: Reduce speed |
| Cw/QSP | Fail-safe selection of the direction of rotation in connection with quick stop |
| Ccw/QSP | |

3. Inhibit inverter: Set terminal X5/RFR to LOW level or open contact.
4. Connect USB diagnostic adapter.
5. Switch on voltage supply of the inverter.
   - Without motor operation: Connect external 24 V supply.
   - With motor operation: Connect mains voltage.

If the green "DRV-RDY" LED is blinking and the red "DRV-ERR" LED is off, the inverter is ready for operation and commissioning can proceed.

**Related topics:**

- Automatic restart after mains connection/fault... (📖 127)
- LED status displays (📖 692)

### 3.7.2 Creating an »Engineer« project & going online

You can find detailed information on the general use of the »Engineer« in the online help which you can call with **[F1]**.
- The chapter "Working with projects" describes, among other things, all options of the *Start-up wizard* which are available to create a new »Engineer« project.

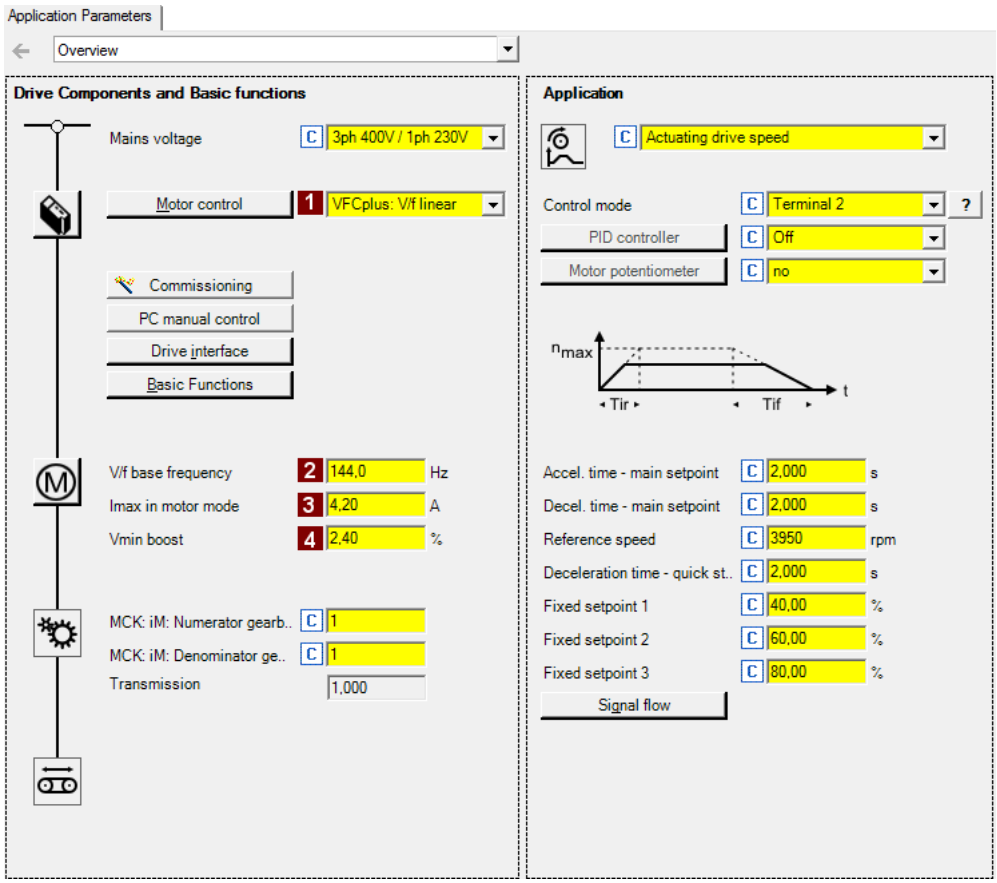
The following steps serve to describe a general method for creating a project with the **Select component from catalogue** option. For this purpose, individual components ( inverter, motor, etc.) are selected from selection lists.

1. Start »Engineer«.
2. Create a new project with the *Start-up wizard* and the **Select component from catalogue** option:
   - In the **Component** step, select the 8400 TopLine inverter.
   - In the **Device modules** step, select the available communication module.
   - In the **Application** step, select the "Actuating drive speed" application. (The application can also be selected any time afterwards via the **Application parameter** tab or C00005.)
   - Select the other components (motor/gearbox) to be added to the project in the **Other components** dialog step.
3. Go online.
   After a connection to the inverter has been established, the following status is displayed in the *Status line*:

   
   
4. Transfer parameter set to the device.
   This command serves to overwrite the current parameter settings in the inverter by parameter settings of the »Engineer« project.

### 3.7.3 Parameterising the motor control

1. Select the **Application parameters** tab from the *Workspace*.
   The motor control parameters, among other things, can be found on the left:




2. In the **1 Motor control** list field (C00006), select the desired motor control.
3. Adapt the motor control parameters:

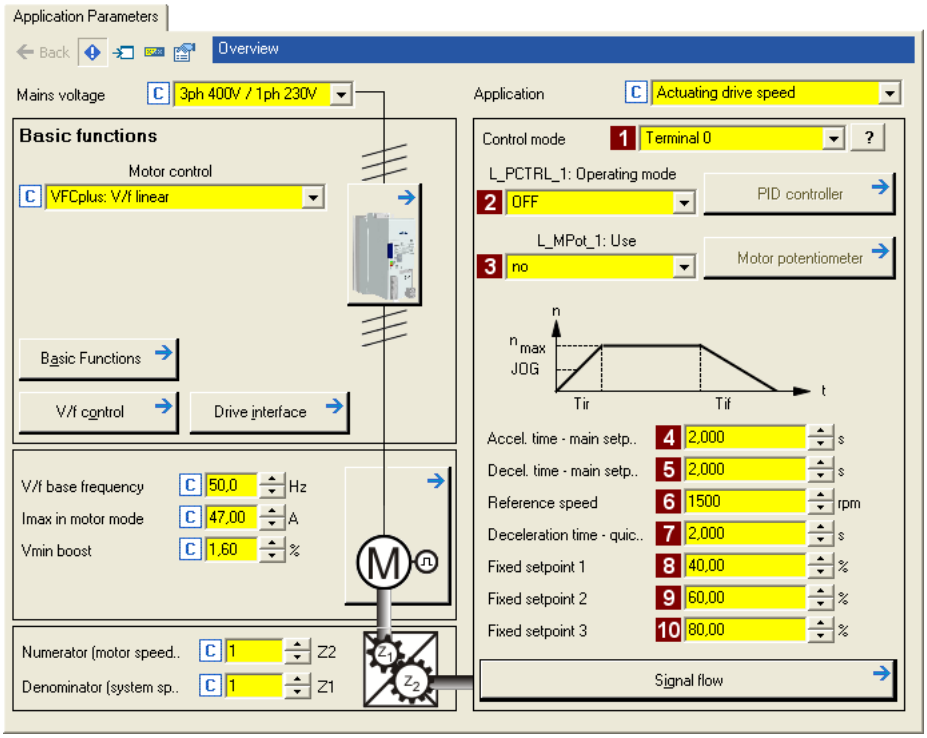
| Parameters | | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 2 | **V/f base frequency** (C00015) | 50.0 | Hz | ▸ Adapting the V/f base frequency (📖 168) |
| 3 | **Imax in motor mode** (C00022) | 47.00 | A | ▸ Optimising the Imax controller (📖 172) |
| 4 | **Vmin boost** (C00016) | 1.60 | % | ▸ Adapting the Vmin boost (📖 170) |

**Related topics:**

- Notes on motor control (📖 57)
- Motor control (MCTRL) (📖 138)

### 3.7.4 Parameterise application

The application parameters can be found on the right side of the **Application parameter** tab:




1. In the **1 Control mode** list field (C00007), select the control mode suitable for the wiring of the terminals.
   - The corresponding wiring diagram is displayed in a pop-up window if you click the ? button right to the list field.
   - For a detailed description, see the chapter "Terminal assignment of the control modes". (📖 449)
2. Optional: Use process controller.
   - For this purpose, select the desired operating mode in the **2 L_PCTRL_1: Operating mode** list field (C00242).
   - For a detailed description see the L_PCTRL_1 function block. (📖 1649)
   - Go to the parameterisation dialog of the process controller via the **Process controller** button.
3. Optional: Use motor potentiometer.
   - For this purpose, select "1: On" in the **3 L_MPot_1: Use** list field (C00806).
   - For a detailed description see the L_MPot_1 function block. (📖 1604)
   - Go to the parameterisation dialog of the motor potentiometer via the **Motor potentiometer** button.

4. Adapt the application parameters:

| | Parameters | Lenze setting Value | Unit | Info |
|---|---|---|---|---|
| 4 | **Accel. time - main setpoint** (C00012) | 2.000 | s | The setpoint is led via a ramp function generator with linear characteristic. The ramp function generator converts setpoint step-changes at the input into a ramp.<br>▸ L_NSet_1 (📖 1622) |
| 5 | **Decel. time - main setpoint** (C00013) | 2.000 | s | |
| 6 | **Reference speed** (C00011) | 1500 | rpm | All speed setpoint selections are provided in % and always refer to the reference speed set in C00011.<br>The motor reference speed is indicated on the motor nameplate. |
| 7 | **Decel. time - quick stop** (C00105) | 2.000 | s | If quick stop is requested, motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0).<br>▸ Activate/deactivate quick stop (📖 111) |
| 8 | **Fixed setpoint 1** (C00039/1) | 40.00 | % | A fixed setpoint for the setpoint generator can be activated instead of the main setpoint via the digital DI1 and DI2 inputs.<br>• Fixed setpoints are selected in [%] based on the reference speed (C00011).<br>▸ L_NSet_1 (📖 1622) |
| 9 | **Fixed setpoint 2** (C00039/2) | 60.00 | % | |
| 10 | **Fixed setpoint 3** (C00039/3) | 80.00 | % | |

**Tip!**

- Click the **Signal flow** button to go down one dialog level to the signal flow of the application with further possible parameter settings. See chapter "Basic signal flow". (📖 439)
- The preconfigured I/O connection in the selected control mode can be changed via configuration parameters. See chapter "User-defined terminal assignment". (📖 429)

**More detailed information on the technology application:**

- TA "Actuating drive speed" (📖 438)
- Internal interfaces | application block "LA_NCtrl" (📖 441)
- Process data assignment for fieldbus communication (📖 458)
- Terminal assignment of the control modes (📖 449)
- Setting parameters (short overview) (📖 460)
- Configuration parameters (📖 462)

### 3.7.5 Save parameter settings safe against mains failure

In order to prevent parameter settings carried out in the device from being lost by mains switching, you have to explicitly save the parameter set with mains failure protection in the device.

Save parameter set.

### 3.7.6 Enable inverter and test application

**Stop!**

Before stipulating a speed setpoint, check whether the brake in the form of a holding brake on the motor shaft has been released!

**Note!**

If the controller is enabled and the "Inhibit at power-on" auto-start option is activated in C00142 (Lenze setting) when the mains is connected, the inverter remains in the "ReadyToSwitchOn" state.

To be able to change to the "SwitchedOn" status, the controller enable must be deactivated first: set terminal X5/RFR to LOW level.

If the inverter is in the "SwitchedOn" status:

1. Enable inverter: Set terminal X5/RFR to HIGH level or close contact.
   - If there is no other active source for the controller inhibit, the inverter changes from the "SwitchedOn" status to the "OperationEnabled" status.
   - The **Diagnostics** tab and C00158 display all active sources for the controller inhibit.
2. Select the speed setpoint.
   In the "Terminal 0" control mode by selecting a voltage at the analog input via the setpoint potentiometer or by selecting a fixed setpoint via the digital DI1/DI2 inputs:

| DI1 | DI2 | Speed selection |
|---|---|---|
| **LOW** | **LOW** | The setpoint speed is selected via analog input 1<br>• Scaling: 10 V ≡ 100 % ≡ reference speed (C00011) |
| HIGH | LOW | Fixed setpoint 1 (C00039/1) is used as setpoint speed.<br>• Lenze setting: 40 % of the reference speed (C00011) |
| LOW | HIGH | Fixed setpoint 2 (C00039/2) is used as setpoint speed.<br>• Lenze setting: 60 % of the reference speed (C00011) |
| HIGH | HIGH | Fixed setpoint 3 (C00039/3) is used as setpoint speed.<br>• Lenze setting: 80 % of the reference speed (C00011) |

**Note!**

Observe the actual speed value (display in C00051) as well as the LED status displays. (📖 692)



**Tip!**

Other control functions in the "Terminal 0" control mode:

- DI3: HIGH level ≡ Request DC-injection braking
- DI4: HIGH level ≡ Request a change of direction of rotation

**Related topics:**

- "Inhibit at power-on" auto-start option (📖 127)
- Trouble-shooting during commissioning (📖 59)
- Diagnostics & error management (📖 691)

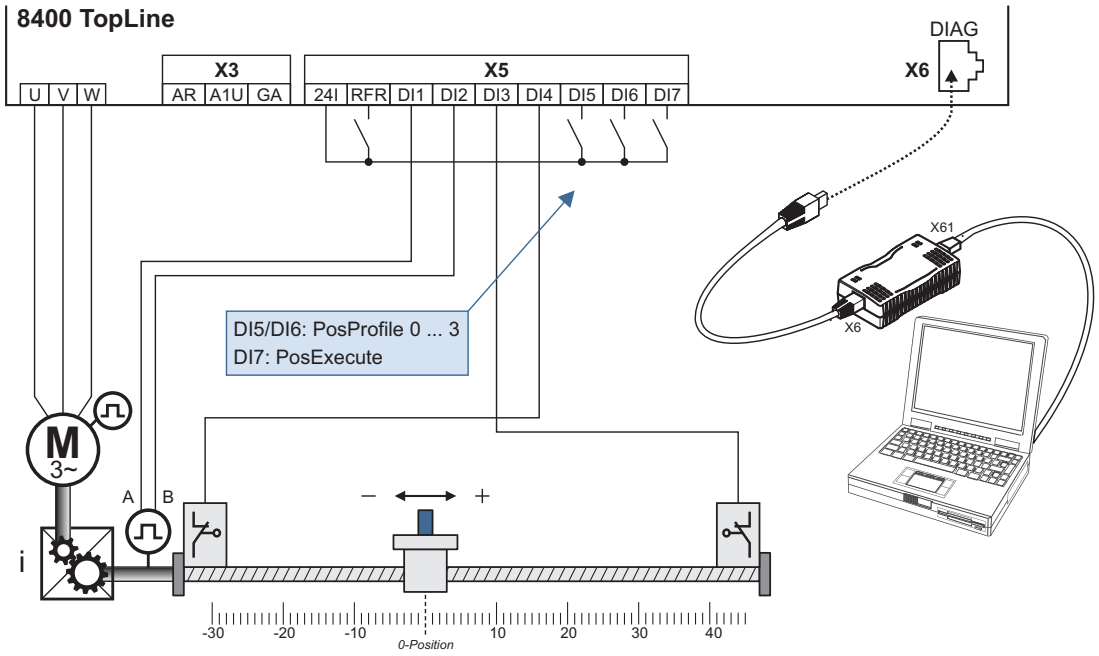
## 3.8 Commissioning of the "Table positioning" technology application

**Note!**

Take all the necessary safety precautions before you carry out the following commissioning steps and switch the device on!

▸ Safety instructions with regard to commissioning (📖 56)

**System constellation**




| | |
|---|---|
| X5/RFR | Controller enable |
| X5/DI1 | Position encoder connection, track A |
| X5/DI2 | Position encoder connection, track B |
| X5/DI3 | Hardware limit switch connection in positive direction |
| X5/DI4 | Hardware limit switch connection in negative direction |
| X5/DI5/DI6 | Profile selection and change of operating mode at the same time:<br>0 \| 0 = Profile selection 0 and request for "Speed follower" operating mode<br>0 \| 1 = Profile selection 1 and request for "Homing" operating mode<br>1 \| 0 = Profile selection 2 and request for "Manual jog" operating mode<br>1 \| 1 = Profile selection 3 and request for "Positioning" operating mode |
| X5/DI7 | Start of profile generation |
| X6 | Diagnostic interface for connecting the USB diagnostic adapter |

[3-2] Block diagram for wiring the commissioning example for the "Table positioning" application

**Commissioning steps**

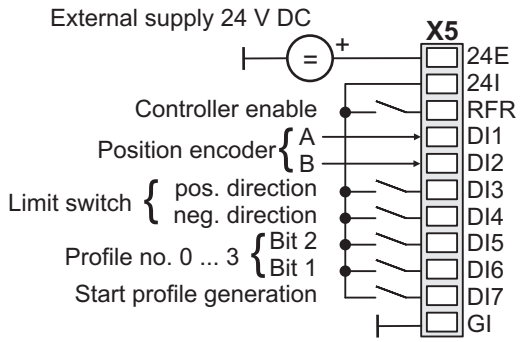
Below find a description of the commissioning steps of the "Table positioning" application shown in illustration [3-2].

Please observe the sequence of the steps in the following chapters and follow them through carefully. This will help you to commission your inverter quickly and as safely as possible:

- Prepare inverter for commissioning (📖 72)
- Creating an »Engineer« project & going online (📖 73)
- Parameterising the motor control (📖 74)
- Parameterise application (📖 75)
- Save parameter settings safe against mains failure (📖 83)
- Enable inverter and test application (📖 83)

### 3.8.1 Prepare inverter for commissioning

1. Power terminal wiring.
   Refer to the mounting instructions supplied with the inverter to find help on how to correctly design the power connections to match the requirements of your device.
2. Control terminal wiring.
   The system constellation including position encoder and hardware limit switches shown in illustration [3-2] requires wiring according to control mode "Terminals 0":




3. Inhibit inverter: Set terminal X5/RFR to LOW level or open contact.
4. Connect USB diagnostic adapter.
5. Switch on voltage supply of the inverter.
   - Without motor operation: Connect external 24 V supply.
   - With motor operation: Connect mains voltage.

If the green "DRV-RDY" LED is blinking and the red "DRV-ERR" LED is off, the inverter is ready for operation and commissioning can proceed.

**Related topics:**

- Automatic restart after mains connection/fault... (📖 127)
- LED status displays (📖 692)

### 3.8.2 Creating an »Engineer« project & going online

You can find detailed information on the general use of the »Engineer« in the online help which you can call with **[F1]**.
- The chapter "Working with projects" describes, among other things, all options of the *Start-up wizard* which are available to create a new »Engineer« project.

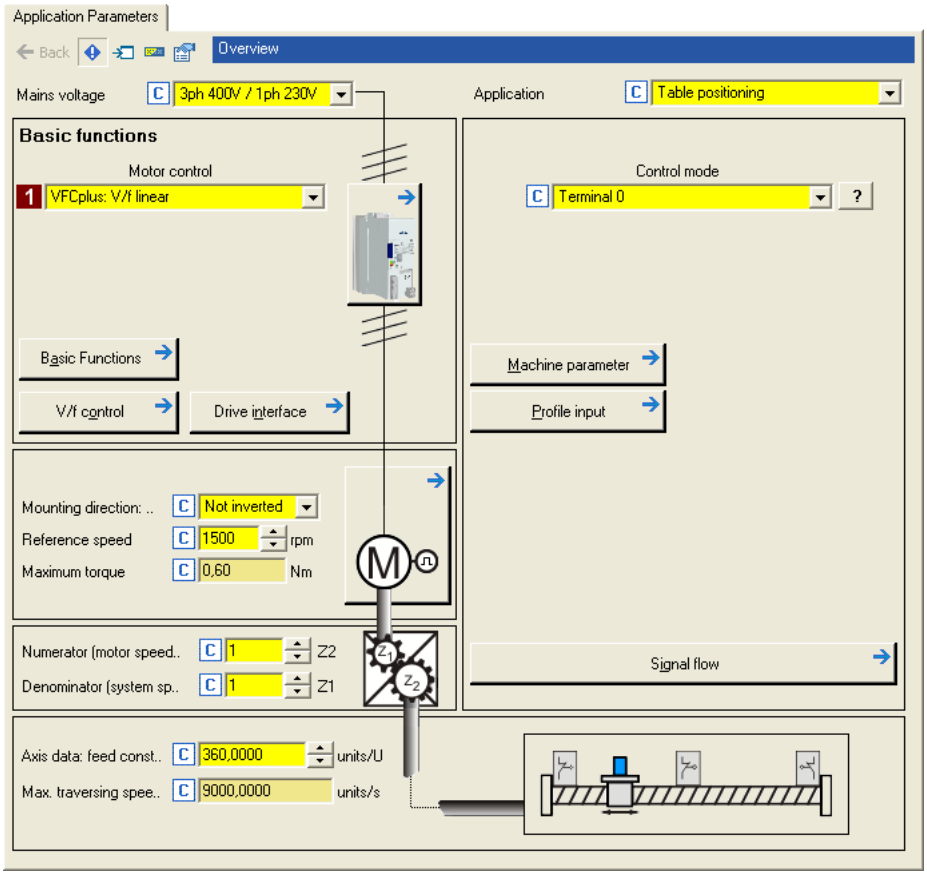
The following steps serve to describe a general method for creating a project with the **Select component from catalogue** option. For this purpose, individual components ( inverter, motor, etc.) are selected from selection lists.

1. Start »Engineer«.
2. Create a new project with the *Start-up wizard* and the **Select component from catalogue** option:
   - In the **Component** step, select the 8400 TopLine inverter.
   - In the **Device modules** step, select the available communication module.
   - In the **Application** step, select the "Table positioning" application. (The application can also be selected any time afterwards via the **Application parameter** tab or C00005.)
   - Select the other components (motor/gearbox) to be added to the project in the **Other components** dialog step.
3. Go online.
   After a connection to the inverter has been established, the following status is displayed in the *Status line*:

   
   
4. Transfer parameter set to the device.
   This command serves to overwrite the current parameter settings in the inverter by parameter settings of the »Engineer« project.

### 3.8.3 Parameterising the motor control

1. Select the **Application parameters** tab from the *Workspace*.
   The motor control parameters, among other things, can be found on the left:



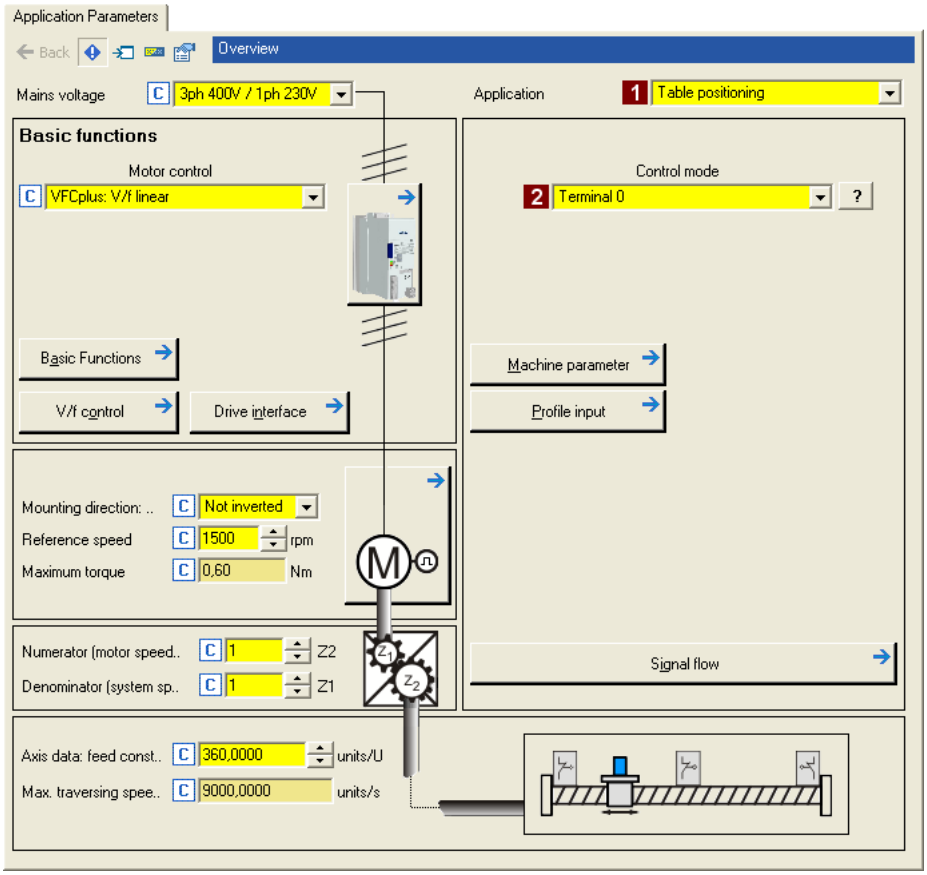

2. In the 1 **Motor control** list field (C00006), select the desired motor control.

**Related topics:**

- Notes on motor control (📖 57)
- Motor control (MCTRL) (📖 138)

### 3.8.4 Parameterise application

The application parameters can be found on the right side of the **Application parameter** tab:




1. Select "Table positioning" in the 1 **Application** list field (C00005) (if you have not already done so while creating the project).
   After the "Table positioning" application is selected, the contents of the tab change, e.g. the **Machine parameters** and **Profile entry** buttons are shown now.
2. In the 2 **Control mode** list field (C00007), the preset "Terminals 0" control mode is already suitable for the system constellation shown in illustration [3-2] and need not be changed.
   - The corresponding wiring diagram is displayed in a pop-up window if you click the ? button right to the list field.
   - For a detailed description, see the chapter "Terminal assignment of the control modes". (📖 449)

### 3.8.4.1 Set machine parameters

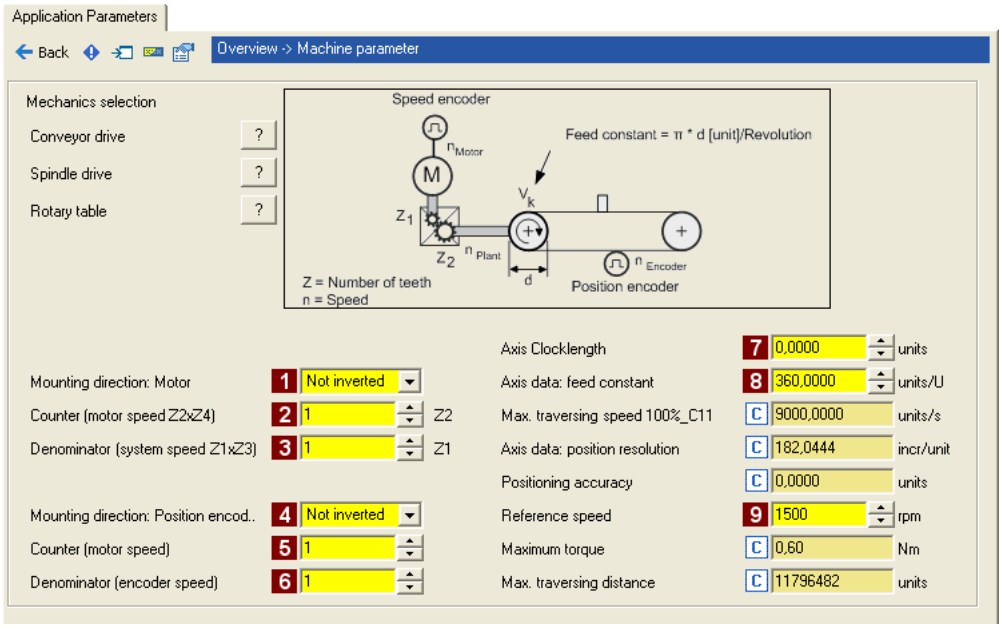
**Note!**

Setting the machine parameters is a basic prerequisite for the operating modes "Homing", "Manual jog" and "Positioning".

The more precisely the machine parameters are set, the better the results of positioning!

Detailed information on the machine parameters is provided in chapter "Basic drive functions" in subchapter "Machine parameters". (📖 593)

1. Go to the right side of the **Application parameter** tab and click the **Machine parameter** button to change to the *Overview* ➞ *Machine parameter* dialog level:




2. Set the machine parameters according to the system constellation at hand.

| Parameters | | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 1 | **Mounting direction: Motor** (C01206/1) | not inverted | | Inversion if motor mounting is mirrored. |
| 2 | **Numerator (Z2)** (C01202/1) | 1 | | Gearbox factor - motor<br>• Entry of the gearbox factor as numerator/denominator ratio (numerator = motor speed and denominator = output speed of gearbox) or from the number of teeth of the gearbox arrangement. |
| 3 | **Denominator (Z1)** (C01202/2) | 1 | | |
| 4 | **Mounting direction: Position encoder** (C01206/2) | not inverted | | Inversion if position encoder mounting is mirrored. |
| 5 | **Counter** (C01203/1) | 1 | | Gearbox factor - position encoder<br>• Entry of the gearbox factor as numerator/denominator ratio, with numerator = motor speed and denominator = position encoder speed. |
| 6 | **Denominator** (C01203/2) | 1 | | |

| Parameters | | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 7 | **Axis cycle** (C01201/1) | 0.0000 | units | Cycle for Modulo measuring system<br>• The Modulo system is activated by setting a cycle (C01201/1) > 0 units.<br>• When the cycle (C01201/1) is set to 0 units (Lenze setting), the traversing range is unlimited (classical measuring system). |
| 8 | **Feed constant** (C01204) | 360.0000 | units/rev. | The feed constant corresponds to the movement of the machine during one revolution of the gearbox output shaft.<br>• The value is entered in application units referred to one revolution. |
| 9 | **Reference speed** (C00011) | 1500 | rpm | All speed setpoint selections are provided in % and always refer to the reference speed set in C00011.<br>The motor reference speed is indicated on the motor nameplate. |

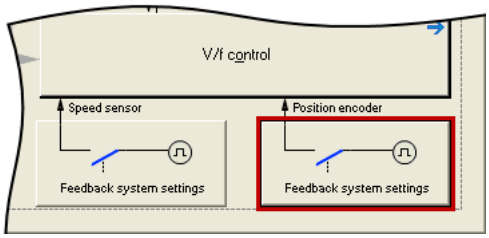
3. After setting the machine parameters, click the **Back** button to change to the *Overview* dialog level.

### 3.8.4.2 Set the position encoder

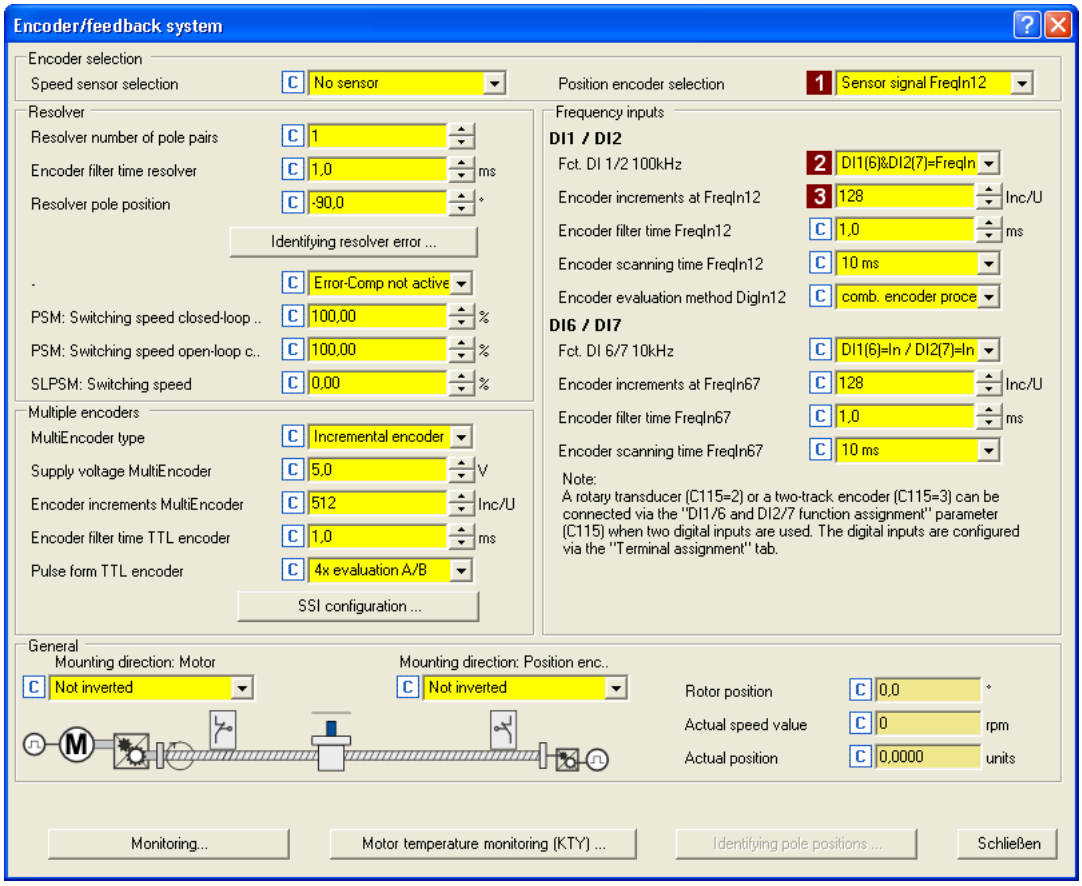
In the Lenze setting, the digital input terminals are configured as "normal" digital inputs. Since in this system constellation DI1 and DI2 are used to connect a two-track position encoder, the function assignment of these two inputs must be changed accordingly.

Detailed information on how to use a resolver/encoder as motor or position encoder can be found in the main chapter "Encoder/feedback system". (📖 321)

1. Go to the right side of the **Application parameter** tab and click the **Signal flow** button to change to the *Overview* → *Signal flow* dialog level.
2. Click the **Feedback system settings** button in the signal flow:



The *Encoder/Feedback system* parameterisation dialog will be shown:




3. In the **1** **Position encoder selection** list field (C00490), select "Encoder signal FreqIn12".
4. In the **2** **Fct. DI 1/2 100kHz** list field (C00115/1), select "DI1(6)&DI2(7)=FreqIn (2-track)".
   This selection ensures that the digital DI1 and DI2 input terminals are configured as frequency inputs.
5. Set the number of position encoder increments in the **3** **Number of encoder increments at FreqIn12** input field (C00420/1).
6. Click the **Close** button to close the parameterisation dialog again.

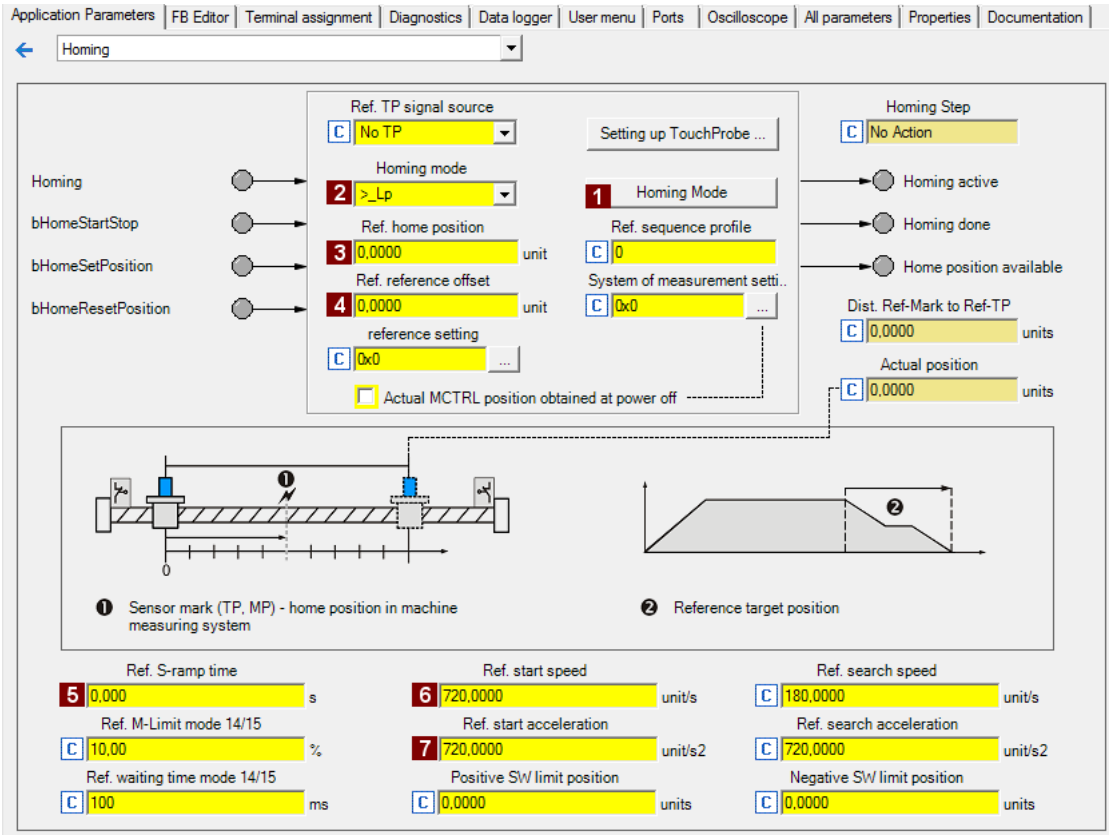
**Related topics:**

- Digital input terminals (📖 385)
- Using DI1(6) and DI2(7) as frequency inputs (📖 389)
- Encoder/feedback system (📖 321)

### 3.8.4.3 Set homing

Detailed information on the "Homing" operating mode is provided in chapter "Basic drive functions" in subchapter "Homing". (📖 618)

1. Click the  button (in the "MotionControlKernel" block) in the signal flow to change to the *Overview* → *Signal flow* → *Homing* dialog level:




2. Select the homing mode and hence the type of homing via the 1 **Homing mode** button or directly in the 2 **Homing mode** list field (C01221).
   - A graphic display of all possible homing modes can be seen via the **Homing mode** button.
   - In this system constellation without touch probe, the preset ">_Lp" mode can be used: movement in positive direction, positive edge of the limit switch sets reference.
3. Set the position to be used for loading the set position and the actual position after homing has finished in the 3 **Homing home position** input field (C01227/2).

4. Recommendation: Set an offset in the **4** **Homing home offset** input field (C01227/1) to prevent the machine from stopping at an activated limit switch.
   Background: The load machine can also leave the travel range limit switch. There follows a return to the home position that was set with the positive edge of the travel range limit switch.
5. Set suitable profile parameters for the homing:

| Parameters | | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 5 | **S-ramp time** (C01226/1) | 0.000 | s | S-ramp time for reference search/homing.<br>• Setting "0 s" ≡ no rounding |
| 6 | **Start speed** (C01224/1) | 720.0000 | unit/s | Starting speed for approaching the pre-stop signals. |
| 7 | **Start acceleration** (C01225/1) | 720.0000 | unit/s$^2$ | Starting acceleration for the starting speed ramps. |

6. After setting the parameters for the homing, click the **Back** button to change back to the *Overview* ➞ *Signal flow* dialog level.

### 3.8.4.4 Enter one or more profiles

You can find detailed information on entering profiles in the chapter entitled "Basic drive functions", subchapter "Profile entry". (📖 651)

1. Click the **Profile entry** button (in the "MotionControlKernel" block) in the signal flow to change to the *Overview* ➞ *Signal flow* ➞ *Profile entry* dialog level:




**Note!**

In the Lenze setting, the profiles are assigned to certain operating modes, i.e. the selection of a profile also results in a change of operating mode:

- If profile 0 is selected: Activation of "Speed follower" operating mode
- If profile 1 is selected: Activation of "Homing" operating mode
- If profile 2 is selected: Activation of "Manual jog" operating mode
- If profile 3 ... 15 is selected: Activation of "Positioning" operating mode

2. Select number 3 in the 1 **Profile number** list field to enter a positioning profile which can be selected via digital inputs DI5 and DI6.

3. Set the profile parameters:

| | Parameters | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 2 | **Profile x: Mode** (C01300/x) | absolute (shortest way)t | | Selection of the way in which positioning is to be carried out. ▸ Positioning modes (📖 657) |
| 3 | **Profile x: Position** (C01301/x) | 360.0000 | unit | Target position or distance to be traversed. |
| 4 | **Profile x: Speed** (C01302/x) | 360.0000 | unit/s | Maximum velocity at which the target is to be approached. |
| 5 | **Profile x: Acceleration** (C01303/x) | 720.0000 | unit/s$^2$ | Specification of the maximum velocity variation which is to be used for acceleration. |
| 6 | **Profile x: Deceleration** (C01304/x) | 720.0000 | unit/s$^2$ | Specification of the velocity variation which is to be used for maximum deceleration to standstill. |
| 7 | **Profile x: S-ramp time** (C01306/x) | 0.000 | s | Due to stipulation of an S-ramp time for a profile, the profile is executed with S-shaped ramps, i.e. acceleration and braking processes are initiated smoothly in order to reduce jerk and thus the stress on the drive components. |
| 8 | **Profile x: Final speed** (C01305/x) | 0.0000 | unit/s | This specifies the velocity at which the drive is to start the next profile after reaching the target position. |
| 9 | **Profile x: Sequence profile** (C01307/x) | 0 | | Optional: Sequence profile for profile linkage / following block control.<br>After execution of the profile (target position reached), the set following (subsequent) profile is started automatically. In this way, profile chains can be stipulated without additional control processes. |
| 10 | **Profile x: TP profile** (C01308/x) | 0 | | Optional: Profile number of the profile (1 ... 15) that is to be executed after a touch probe has been detected.<br>• If "0" is set, there will be no profile stepping through touch probe.<br>• Only relevant for positioning modes with touch-probe. |
| 11 | **Profile x: TP source** (C01308/x) | TP-DigIn3 | | Optional: Selection of the signal source for touch probe detection.<br>• Only relevant for positioning modes with touch-probe. |

4. After entering the profile, click the **Back** button to change back to the *Overview* → *Signal flow* dialog level.

**Related topics:**

- Touch probe detection (📖 419)
- Operating mode change with profile number (📖 589)
- Positioning (📖 646)

**More detailed information on the technology application:**

- TA "Table positioning" (📖 498)
- Basic signal flow (📖 499)
- Internal interfaces | application block "LA_TabPos" (📖 502)
- Process data assignment for fieldbus communication (📖 521)
- Terminal assignment of the control modes (📖 512)
- Setting parameters (short overview) (📖 523)
- Configuration parameters (📖 524)

### 3.8.5 Save parameter settings safe against mains failure

In order to prevent parameter settings carried out in the device from being lost by mains switching, you have to explicitly save the parameter set with mains failure protection in the device.

 Save parameter set.

### 3.8.6 Enable inverter and test application



**Stop!**

Before stipulating a speed setpoint, check whether the brake in the form of a holding brake on the motor shaft has been released!



**Note!**

If the controller is enabled and the "Inhibit at power-on" auto-start option is activated in C00142 (Lenze setting) when the mains is connected, the inverter remains in the "ReadyToSwitchOn" state.

To be able to change to the "SwitchedOn" status, the controller enable must be deactivated first: set terminal X5/RFR to LOW level.

If the inverter is in the "SwitchedOn" status:

1. Enable inverter: Set terminal X5/RFR to HIGH level or close contact.
   - If there is no other active source for the controller inhibit, the inverter changes from the "SwitchedOn" status to the "OperationEnabled" status.
   - The **Diagnostics** tab and C00158 display all active sources for the controller inhibit.
2. Select the respective control signals via the digital inputs (see sections below).



**Note!**

Observe the actual speed value (display in C00051) as well as the LED status displays. (📖 692)

**Homing**

1. Request "homing" mode:
   Set digital input DI5 to LOW level and digital input DI6 to HIGH level in order to select the profile 1 and request the "homing" mode at the same time.
2. Start homing:
   Set digital input DI7 to HIGH level ("alternative function PosExecute").
3. Complete homing when home position has been reached:
   Reset digital input DI7 to LOW level.

**Positioning**

1. Request "positioning" mode:
   Set both digital inputs DI5 and DI6 to HIGH level to select the profile 3 and request the "positioning" mode at the same time.
2. Start positioning:
   Set digital input DI7 to HIGH level ("alternative function PosExecute").
3. Complete positioning if target position has been reached:
   Reset digital input DI7 to LOW level.

**Tip!**

The active operating mode (C01243) in the signal flow can also be controlled by means of the setting of the switch in the "Motion Control Kernel" block.

**Related topics:**

- "Inhibit at power-on" auto-start option (📖 127)
- Trouble-shooting during commissioning (📖 59)
- Diagnostics & error management (📖 691)

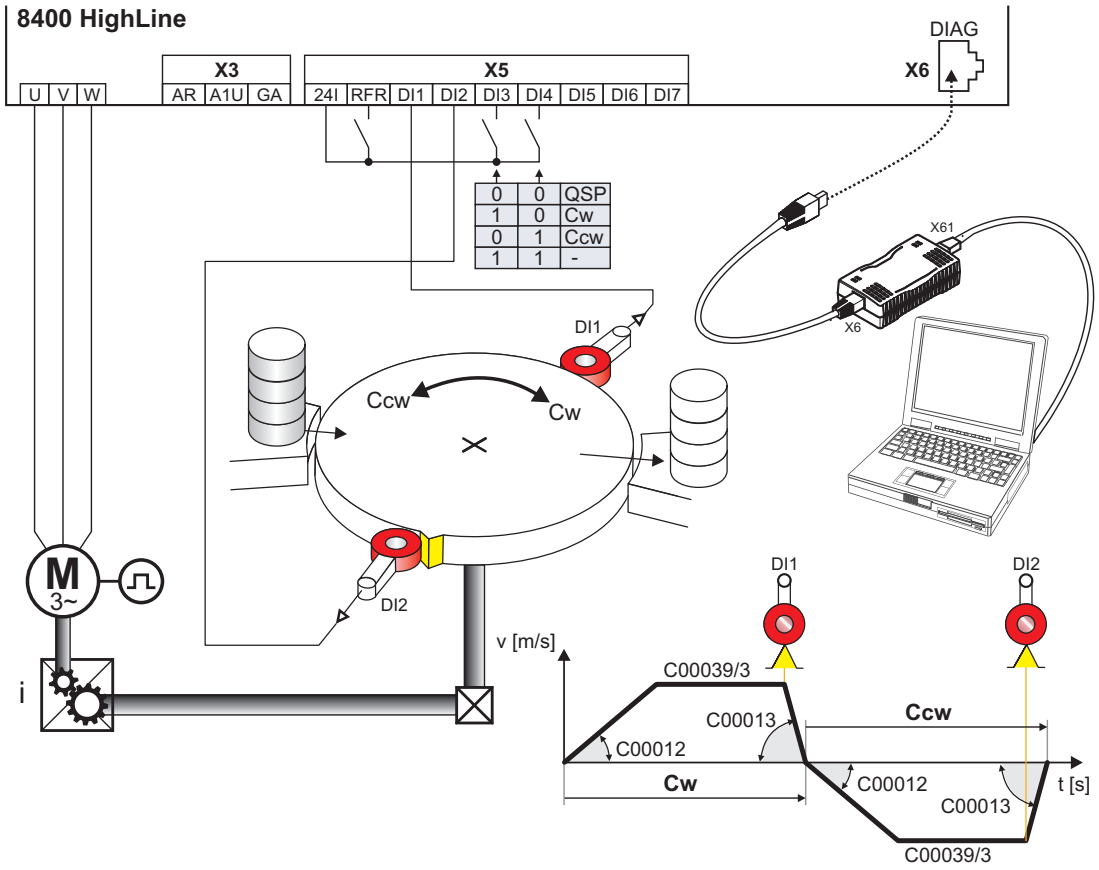
## 3.9 Commissioning of the "Switch-off positioning" technology application

**Note!**

Take all the necessary safety precautions before you carry out the following commissioning steps and switch the device on!

▸ Safety instructions with regard to commissioning (📖 56)

**System constellation**




| | |
|---|---|
| X4/RFR | Controller enable |
| X5/DI1 | Initiator connection for stop function 1 |
| X5/DI2 | Initiator connection for stop function 2 |
| X5/DI3 | Quick stop for CW rotation / selection of switch-off position 1 |
| X5/DI4 | Quick stop for CCW rotation / selection of switch-off position 2 |
| X6 | Diagnostic interface for connecting the USB diagnostic adapter |

[3-3] Block diagram for wiring of the commissioning example for the "Switch-off positioning" application

**Functional principle of a switch-off positioning without pre-switch off**

In case of the switch-off positioning without pre-switch off shown above, it makes sense to use the "Terminals 2" control mode:

1. Set DI3 to HIGH level to activate CW rotation.
2. The drive accelerates along the acceleration ramp (C00012) up to the traversing speed set in C00039/3.
3. After reaching the DI1 contact, the drive comes to a stop with quick stop (QSP) in the target position.
4. Reset DI3 to LOW level and set DI4 to HIGH level to activate CCW rotation now.
5. The drive is accelerated along the acceleration ramp (C00012) up to the traversing speed set in C00039/3.
6. After the DI2 contact has been reached, the drive is braked to standstill with quick stop (QSP) in the initial position.

**Tip!**

- In order to avoid positioning inaccuracy due to signal propagation delays, the initiators can be directly evaluated by the inverter. Limit switch evaluation can be configured in the inverter. In code C00488/x you can change the method of detecting position signals from level evaluation to edge evaluation.
- In order to prevent unintended movements of the load in the target position, the use of a holding brake is recommended as an alternative to DC-injection braking (limited torque).
- The device terminals and their function assignment do not appear in the FB Editor. The assignment of (hardware) terminals to (software) functions is explained in the chapter "Terminal assignment of the control modes". (📖 538)

**Commissioning steps**

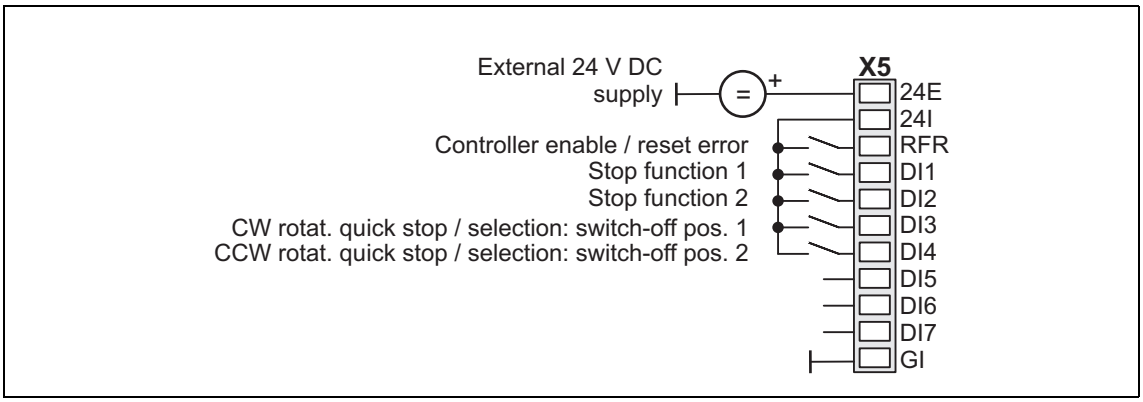
As shown in illustration [3-3], below find a description of the commissioning steps of the "Switch-off positioning" application without pre-switch off.

Please observe the sequence of the steps in the following chapters and follow them through carefully. This will help you to commission your inverter quickly and as safely as possible:

▸ Prepare inverter for commissioning (📖 87)
▸ Creating an »Engineer« project & going online (📖 88)
▸ Parameterising the motor control (📖 89)
▸ Parameterise application (📖 90)
▸ Save parameter settings safe against mains failure (📖 92)
▸ Enable inverter and test application (📖 92)

### 3.9.1 Prepare inverter for commissioning

1. Power terminal wiring.
   Refer to the mounting instructions supplied with the inverter to find help on how to correctly design the power connections to match the requirements of your device.
2. Control terminal wiring.
   In case of the application shown in illustration [3-3], switch-off positioning without pre-switch off, wiring according to the "Terminals 2" control mode makes sense:




3. Inhibit inverter: Set terminal X5/RFR to LOW level or open contact.
4. Connect USB diagnostic adapter.
5. Switch on voltage supply of the inverter.
   - Without motor operation: Connect external 24 V supply.
   - With motor operation: Connect mains voltage.

If the green "DRV-RDY" LED is blinking and the red "DRV-ERR" LED is off, the inverter is ready for operation and commissioning can proceed.

**Related topics:**

- Automatic restart after mains connection/fault... (📖 127)
- LED status displays (📖 692)

### 3.9.2 Creating an »Engineer« project & going online



You can find detailed information on the general use of the »Engineer« in the online help which you can call with **[F1]**.

- The chapter "Working with projects" describes, among other things, all options of the *Start-up wizard* which are available to create a new »Engineer« project.

The following steps serve to describe a general method for creating a project with the **Select component from catalogue** option. For this purpose, individual components ( inverter, motor, etc.) are selected from selection lists.

1. Start »Engineer«.
2. Create a new project with the *Start-up wizard* and the **Select component from catalogue** option:
   - In the **Component** step, select the 8400 TopLine inverter.
   - In the **Device modules** step, select the available communication module.
   - In the **Application** step, select the "Switch-off positioning" application. (The application can also be selected any time afterwards via the **Application parameter** tab or C00005.)
   - Select the other components (motor/gearbox) to be added to the project in the **Other components** dialog step.
3. Go online.
   After a connection to the inverter has been established, the following status is displayed in the *Status line*:

4. Transfer parameter set to the device.
   This command serves to overwrite the current parameter settings in the inverter by parameter settings of the »Engineer« project.

### 3.9.3 Parameterising the motor control

1. Select the **Application parameters** tab from the *Workspace*.
   The motor control parameters, among other things, can be found on the left:




2. In the **1 Motor control** list field (C00006), select the desired motor control.
3. Adapt the motor control parameters:

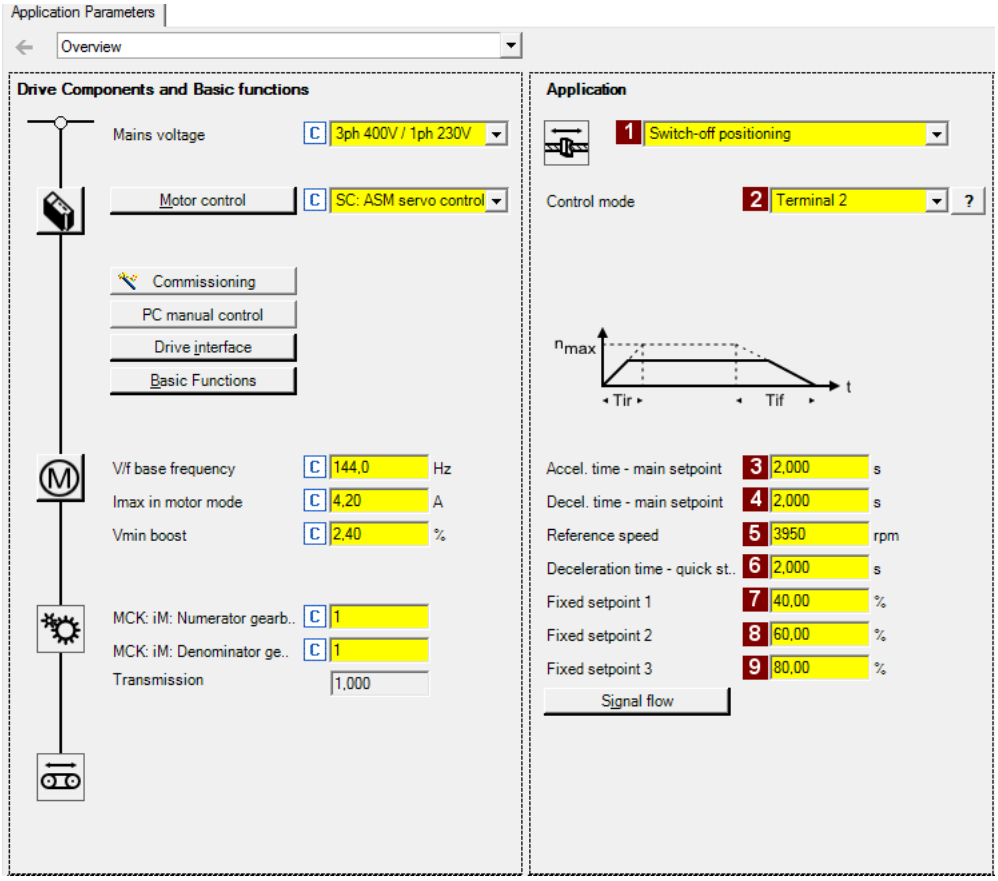
| Parameters | | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 2 | **V/f base frequency** (C00015) | 50.0 | Hz | ▸ Adapting the V/f base frequency (📖 168) |
| 3 | **Imax in motor mode** (C00022) | 47.00 | A | ▸ Optimising the Imax controller (📖 172) |
| 4 | **Vmin boost** (C00016) | 1.60 | % | ▸ Adapting the Vmin boost (📖 170) |

**Related topics:**

- Notes on motor control (📖 57)
- Motor control (MCTRL) (📖 138)

### 3.9.4 Parameterise application

The application parameters can be found on the right side of the **Application parameter** tab:




1. In the **1 Application** list field (C00005), select the "Switch-off positioning" application (if you have not already done so while creating the project).
   After the "Switch-off positioning" application is selected, the contents of the tab change, e.g. the **Process controller** and **Motor potentiometer** buttons are not shown any more.
2. In the **2 Control mode** list field (C00007) and in case of illustration [3-3], for the shown switch-off positioning without pre-switch off the "Terminals 2" control mode must be selected.
   - The corresponding wiring diagram is displayed in a pop-up window if you click the ? button right to the list field.
   - For a detailed description, see the chapter "Terminal assignment of the control modes". (📖 449)

3. Adapt the application parameters:

| | Parameters | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 3 | **Accel. time - main setpoint** (C00012) | 2.000 | s | The setpoint is led via a ramp function generator with linear characteristic. The ramp function generator converts setpoint step-changes at the input into a ramp. Note: These settings only apply if no other ramp times have been selected at the L_NSet FB! |
| 4 | **Decel. time - main setpoint** (C00013) | 2.000 | s | |
| 5 | **Reference speed** (C00011) | 1500 | rpm | All speed setpoint selections are provided in % and always refer to the reference speed set in C00011. The motor reference speed is indicated on the motor nameplate. |
| 6 | **Decel. time - quick stop** (C00105) | 2.000 | s | If quick stop is requested, motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0). ▸ Activate/deactivate quick stop (📖 111) |
| 7 | **Fixed setpoint 1** (C00039/1) | 40.00 | % | Fixed setpoints are selected in [%] based on the reference speed (C00011). **Fixed setpoint 2 must be smaller than fixed setpoint 3!** Otherwise, the drive will be started with a low speed and accelerated after the pre-switch off. |
| 8 | **Fixed setpoint 2** (C00039/2) | 60.00 | % | |
| 9 | **Fixed setpoint 3** (C00039/3) | 80.00 | % | |

**Tip!**

- Click the **Signal flow** button to go down one dialog level to the signal flow of the application with further possible parameter settings. See chapter "Basic signal flow". (📖 529)
- The preconfigured I/O connection in the selected control mode can be changed via configuration parameters. See chapter "User-defined terminal assignment". (📖 429)
- Low-jerk traversing profiles can be implemented by means of S-shaped ramps.
- In the case of high breakaway torques combined with horizontal motion sequences, "Sensorless vector control (SLVC)" can be used as motor control (C00006).
- For reversal of rotation direction (bidirectional motion), comprehensive configuration options are available in the inverter (e.g. by means of the L_DFlipFlop function block).

**More detailed information on the technology application:**

- TA "Switch-off positioning" (📖 527)
- Internal interfaces | application block "LA_SwitchPos" (📖 530)
- Process data assignment for fieldbus communication (📖 547)
- Terminal assignment of the control modes (📖 538)
- Setting parameters (short overview) (📖 549)
- Configuration parameters (📖 551)

### 3.9.5 Save parameter settings safe against mains failure

In order to prevent parameter settings carried out in the device from being lost by mains switching, you have to explicitly save the parameter set with mains failure protection in the device.

Save parameter set.

### 3.9.6 Enable inverter and test application

**Stop!**

Before stipulating a speed setpoint, check whether the brake in the form of a holding brake on the motor shaft has been released!

**Note!**

If the controller is enabled and the "Inhibit at power-on" auto-start option is activated in C00142 (Lenze setting) when the mains is connected, the inverter remains in the "ReadyToSwitchOn" state.

To be able to change to the "SwitchedOn" status, the controller enable must be deactivated first: set terminal X5/RFR to LOW level.

If the inverter is in the "SwitchedOn" status:

1. Enable inverter: Set terminal X5/RFR to HIGH level or close contact.
   - If there is no other active source for the controller inhibit, the inverter changes from the "SwitchedOn" status to the "OperationEnabled" status.
   - The **Diagnostics** tab and C00158 display all active sources for the controller inhibit.
2. Select the respective control signals via the digital inputs.

**Note!**

Observe the actual speed value (display in C00051) as well as the LED status displays. (📖 692)

**Related topics:**

- "Inhibit at power-on" auto-start option (📖 127)
- Trouble-shooting during commissioning (📖 59)
- Diagnostics & error management (📖 691)

## 3.10 PC manual control

For the purpose of testing and demonstration and when an online connection has been established, the PC manual control enables the manual control of various drive functions from the »Engineer«.

**Supported drive functions:**

- Speed control (follow speed setpoint)
- Activate/deactivate quick stop
- Set/reset home position
- Manual jog
- Positioning (relative or absolute)

**More control functions:**

- Reset error message
- Set digital/analog outputs (in preparation)

**Diagnostic functions:**

- Display of the actual speed value and motor current (as time characteristic)
- Display of the current device status
- Display of the status determining error
- Display of the status of the digital/analog inputs (in preparation)

### 3.10.1 Activate PC manual control




**Stop!**

PC manual control must be explicitly activated by the user.

If PC manual control is activated, the inverter is inhibited via device command (C00002/16) first.

## Note!

**With active PC manual control:**

The online connection between PC and controller is monitored by the inverter.

- If the online connection is interrupted longer than the timeout time set (Lenze setting: 2 s):
  - The error response "Fault" takes place, i.e. the motor becomes torqueless and coasts, if it has not yet been in standstill.
  - The "Ck16: Time overflow manual control" error message is entered into the logbook.

PC manual control provides the **Motion Control Kernel** and the motor interface with all required control signals and setpoint signals.

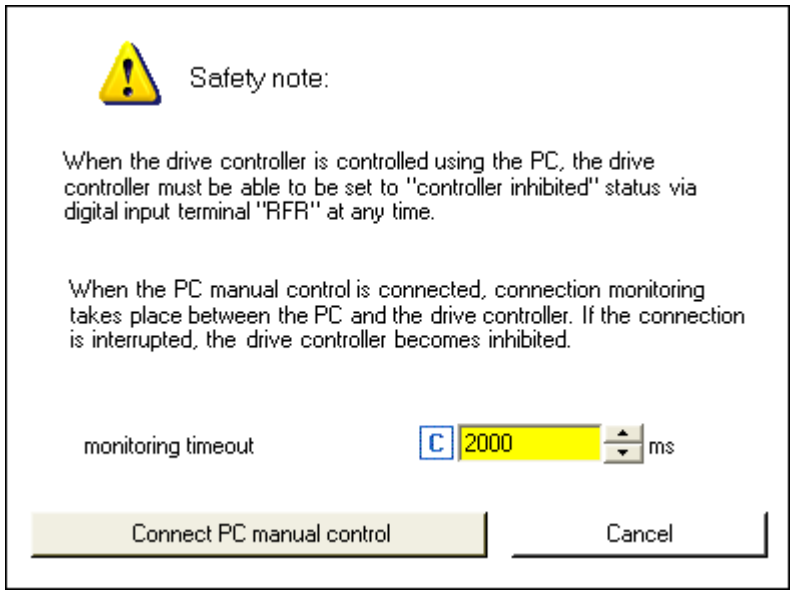
- The available application (function block interconnection) is now decoupled from these interfaces, but is continued to be processed and remains unchanged.
- It does not matter what type of motor control is set in C00006.

**How to activate the PC manual control:**

1. If an online connection to the inverter has not been established yet:

   Go online.

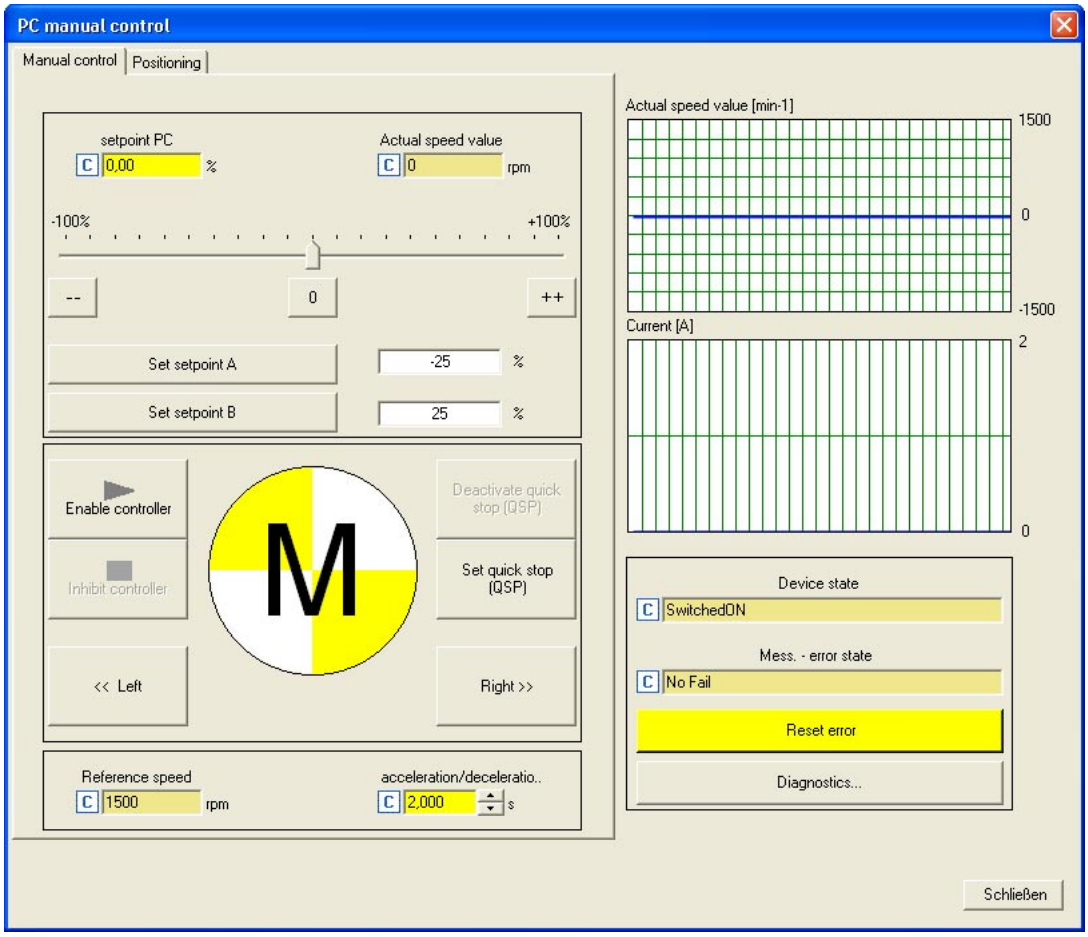
2. Select the **Application parameters** tab from the *Workspace*.
3. Go to the *Overview* dialog level and click the "**PC manual control**" button.
   - First, the following safety note is displayed:




   - Click the **Cancel** button to abort the action and close the dialog box.
   - The **Timeout monitoring** input field serves to adapt the timeout time for the connection monitoring between PC and inverter.
4. To acknowledge the note and activate PC manual control:
   Click the **Activate PC manual control** button.
   - The inverter is inhibited via device command (C00002/16).
   - The *PC manual control* operator dialog is displayed.

### PC manual control - operator dialog

The left side of the *PC manual control* operator dialog contains several tabs for selecting the various control functions and the right side contains setpoint and status displays for diagnostic purposes:




**Note!**

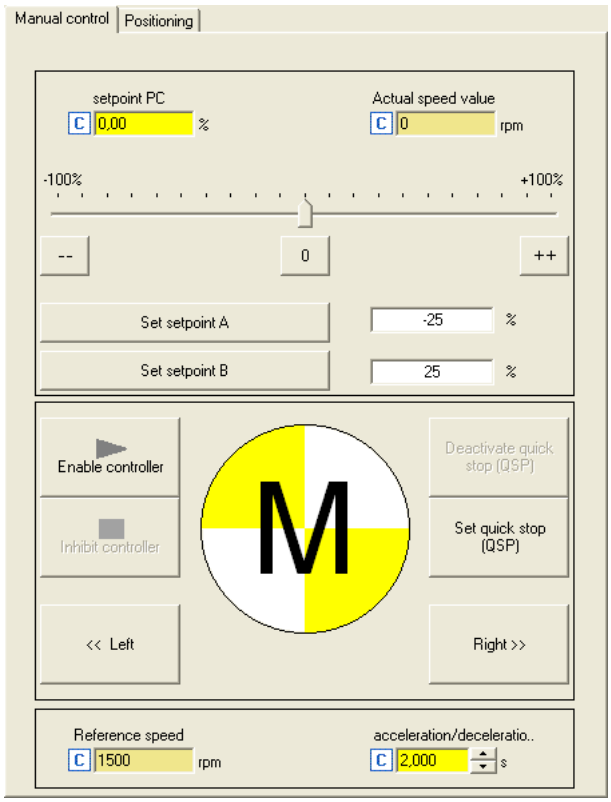
PC manual control can be exited any time by clicking the **Close** button.

If you exit PC manual control or change to another tab, the inverter is inhibited via device command (C00002/16), i.e. the motor becomes torqueless and is coasting unless it already is at standstill.

The different versions are described in the following chapters.

### 3.10.2 Speed control

The **Speed control** serves to easily rotate the drive without setting control parameters or feedback systems in the "Speed follower" mode:

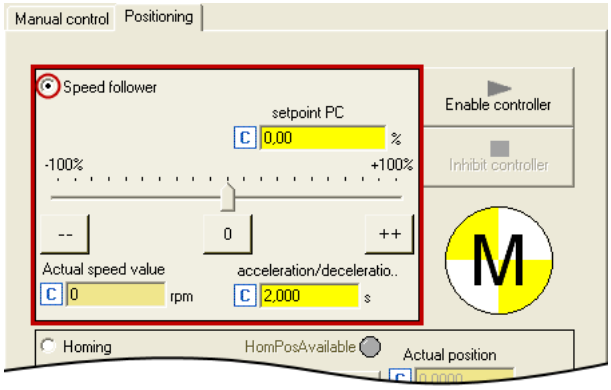



**How to easily rotate the motor:**

1. Set the desired speed setpoint in [%] based on the reference speed, e.g. directly in the **Setpoint PC** input field or via the slider.
   - Via the **--** / **0** / **++** buttons, the currently set speed setpoint can be reduced/increased in steps of 10 percent or set to zero.
   - Via the **Set setpoint A/B** buttons, the speed setpoint can be set to a previously set constant value A/B.
2. To start the speed follower:
   Enable the inverter via the **Enable controller** button.
   - Please observe that the inverter will not be enabled if other sources of controller inhibit (e.g. terminal RFR) are active.
   - The enabled drive now follows the defined speed setpoint.
   - In order to prevent shocks or overload at higher setpoint changes, the speed setpoint is lead via a linear ramp generator with adjustable acceleration/deceleration time.
   - Via the **Inhibit controller** button, the inverter can be inhibited again, i.e. the motor becomes torqueless and is coasting unless it already is at standstill.

**Further functions:**

- If the **Set quick stop (QSP)** button is clicked, the motor is braked to a standstill within the deceleration time parameterised in C00105.
  - Via the **Deactivate quick stop (QSP)** button, the quick stop can be deactivated.
- Via the **<< CCW** and **CW >>** buttons, the direction of rotation can be changed.
- The "speed follower" operating mode can be activated via the second **Positioning** tab without the functions described before:

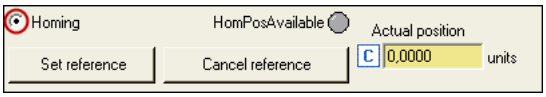


### 3.10.3 Set/reset home position

The measuring system in the machine is selected by means of homing and the 0 position is set within the possible physical travel range.

Detailed information on the "Homing" operating mode is provided in chapter "Basic drive functions" in subchapter "Homing". (📖 618)

A change to the "Homing" operating mode is made by selecting the correspondent option on the **Positioning** tab:




**How to set the reference manually with standing drive:**

Click the **Set reference** button.

The current actual position in the machine measuring system now corresponds to the home position set in C01227/2 (Lenze setting: 0.0000 units).

**Stop!**

If you set the reference with PC manual control, the drive is referenced with it.

**If another home position is used for normal operation:**

Click **Cancel reference** and reset the status signals *HomPosAvailable* and *HomePosDone* in the MCK status word in order that no mechanical problems occur in normal operation.

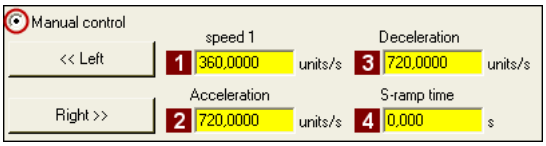
- Setpoints and actual positions remain untouched until a renewed reference setting or homing.

### 3.10.4 Manual jog

In the "Manual jog" mode, the drive can be traversed manually in a clockwise or anticlockwise direction.

You can find detailed information on the "Manual jog" mode in the chapter entitled "Basic drive functions" in the subchapter "Manual jog". (📖 638)

A change to the "manual jog" operating mode is made by selecting the correspondent option on the **Positioning** tab:




- The profile parameters stored in the parameter set for the basic "manual jog" function are used:

| Parameters | | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 1 | **Speed 1** (C01231/1) | 360.0000 | units/s | Manual jog speed |
| 2 | **Acceleration** (C01232/1) | 720.0000 | units/s$^2$ | For accelerating and decelerating, different values can be set so that smooth starting and quick stopping of the drive can be implemented. |
| 3 | **Delay** (C01232/2) | 720.0000 | units/s$^2$ | |
| 4 | **S-ramp time** (C01233/1) | 0.000 | s | In order to reduce jerking, the two ramps can be set in such a way that they are s-shaped. This is done by entering a relative S-ramp time. |

**How to traverse the drive manually in inching mode:**

1. If the inverter is still inhibited, enable the inverter via the **Enable controller** button.
   - Please observe that the inverter will not be enabled if other sources of controller inhibit (e.g. terminal RFR) are active.

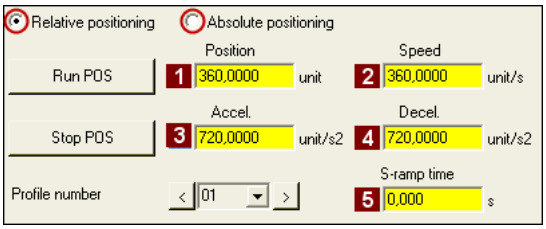
   If the inverter is in the "OperationEnabled" device status:
2. Press the **<< Left** or **Right >>** button (and keep it pressed) to traverse the drive with the corresponding direction of rotation.

### 3.10.5 Positioning (relative or absolute)

Positioning means that a workpiece/tool or material is moved from a starting position to a defined destination.

You can find detailed information on the "Positioning" mode in the chapter entitled "Basic drive functions" in the subchapter "Positioning". (📖 646)

A change to the "Positioning" mode and a simultaneous selection of the positioning mode (relative or absolute) is made by selecting the corresponding option on the **Positioning** tab:




- The profile parameters stored in the parameter set for the basic "Positioning" function are used:

| Parameters | | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 1 | **Position** (C01301/1...15) | 360.0000 | units | Target position or distance to be traversed.<br>• With "Relative positioning": Distance to starting position (current position).<br>• With "Absolute positioning": Distance to defined zero position. |
| 2 | **Speed** (C01302/1...15) | 360.0000 | units/s | Maximum velocity at which the target is to be approached. |
| 3 | **Acceleration** (C01303/1...15) | 720.0000 | units/s$^2$ | Specification of the maximum velocity variation which is to be used for acceleration. |
| 4 | **Delay** (C01304/1...15) | 720.0000 | units/s$^2$ | Specification of the velocity variation which is to be used for maximum deceleration to standstill. |
| 5 | **S-ramp time** (C01306/1...15) | 0.000 | s | A profile is traversed with S-shaped ramps by defining an S-ramp time for it. |

**Note!**

In order that no complex travel movements occur, no switching to a sequence profile takes place that has been set in the profile data!

You can find detailed information on entering profiles in the chapter entitled "Basic drive functions", subchapter "Profile entry". (📖 651)

**How to carry out positioning:**

1. Select the profile to be executed (1 ... 15) in the **Profile number** list field.
2. If the inverter is still inhibited, enable the inverter via the **Enable controller** button.
   - Please observe that the inverter will not be enabled if other sources of controller inhibit (e.g. terminal RFR) are active.

   If the inverter is in the "OperationEnabled" device status:
3. Press the **Run POS** button to start positioning.
   - The **Stop POS** serves to stop active positioning at any time.

# 4 Device control (DCTRL)

This chapter provides information on internal device control as well as the device commands which can be executed via the subcodes of C00002.

- The device control causes the inverter to take defined device statuses.
- The device control provides a multitude of status information in many ways:
  - Visually via the LED status displays on the front of the inverter. (📖 693)
  - As text messages in the Logbook. (📖 700)
  - As process signals via the outputs of the LS_DriveInterface system block. (📖 130)
  - Via diagnostic / display parameters which are included in the »Engineer« parameter list as well as in the **Diagnostics** category in the keypad.



**Note!**

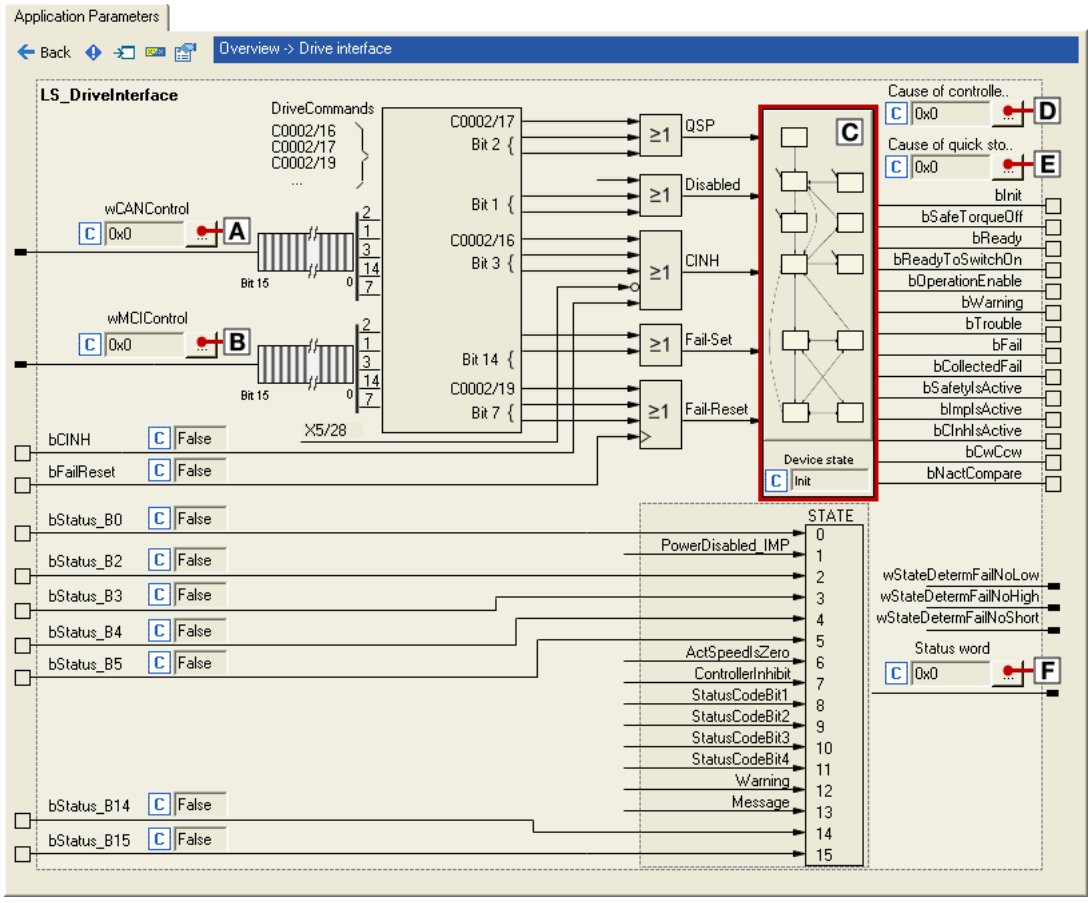
The device states of the inverter are based on the operating states of the CiA402 standard. ▸ Device state machine and device states (📖 116)

**How to get to the parameterisation dialog of the device control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Go to the *Overview* dialog level and click the **Drive interface** button.

### Parameterisation dialog in the »Engineer«

The parameterisation dialog shows the input / output signals and the internal signal flow of the LS_DriveInterface system block which displays the device control in the function block editor:




| Range / Meaning | | Display parameter |
|---|---|---|
| A | Display of the control word via system bus (CAN) | C00136/2 |
| B | Display of the control word via communication module (e.g. PROFIBUS) | C00136/1 |
| C | Display of the internal state machine and the current device status | C00137 |
| D | Display of all active sources of a controller inhibit | C00158 |
| E | Display of all active sources of a quick stop | C00159 |
| F | Display of the status word of the device control | C00150 |

## 4.1 Device commands (C00002/x)

This chapter describes the device commands of the inverter which are provided in the subcodes of C00002 and can be carried out using the keypad or, alternatively, the »Engineer« when an online connection has been established.

The device commands serve, among other things, to directly control the inverter to organise parameter sets, and to call diagnostic services.

Regarding the execution of the device commands, a distinction is drawn between:

- Device commands which have an immediate effect on control (e.g. "Activate quick stop")
  - After being called in C00002/x, these device commands provide static status information ("On" or "Off").
- Device commands with longer execution duration (several seconds)
  - After being called in C00002/x, these device commands provide dynamic status information ("Work in progress 20%" → "Work in progress 40%", etc.).
  - The execution of the device command has not finished successfully until the "Off / ready" status information is provided in C00002/x.
  - In the event of an error, the "Action cancelled" status information is provided in C00002/x . In this case, further details can be obtained from the status of the device command executed last which is displayed in C00003.




### Stop!

Before the supply voltage is switched off after a device command has been transmitted via C00002/x, the device command must be checked for successful completion on the basis of the status information provided in C00002/x!

- This is of particular importance for device commands which save data to the memory module of the device. Incomplete storage processes may lead to data inconsistencies in the memory module.



### Note!

- Before activating device commands by a master control, wait for the "Ready" signal of the inverter.
- The device will reject a write process to C00002/x if the value is >1 and issue an error message.
- C00003 displays the status of the device command that was executed last.

### Activate device command

When an online connection has been established, simply use the »Engineer« to activate a device command by selecting the corresponding option from the **Parameters** tab in C00002/x ("0: off" or "1: On / start").

- Alternatively, the device command can also be activated via e.g. keypad or through a master control by writing to C00002/x.
- Some of the frequently used device commands (such as "Save parameter set") can also be executed via the *Toolbar* icons of the »Engineer« when an online connection has been established:

| Symbol | Function |
|---|---|
|  | Enable inverter |
|  | Inhibit inverter |
|  | Save parameter set (for 8400: Save all parameter sets) |



**Note!**

Device commands that can be executed via the *Toolbar* of the »Engineer« always affect the element currently selected in the *Project view* including all subelements!

- If no inverter but a system module is selected in the *Project view*, the corresponding device command will be activated in all lower-level inverters having an online connection with the »Engineer«.

Before the desired action is carried out, a confirmation prompt appears first, asking whether the action is really to be carried out.

### Short overview of device commands

Device commands described in this chapter:

| C00002 Subcode: | Device command | Controller inhibit required | Status information |
|---|---|---|---|
| 1 | Load Lenze setting | ● | dynamic |
| 6 | Load all parameter sets | ● | dynamic |
| 11 | Save all parameter sets | | dynamic |
| 16 | Enable/inhibit inverter | | static |
| 17 | Activate/deactivate quick stop | | static |
| 19 | Reset error | | static |
| 21 | Delete logbook | | static |
| 27 | Device search function | | static |

Device commands described in other chapters:

| C00002 Subcode: | Device command | Controller inhibit required | Status information |
|---|---|---|---|
| 23 | Identify motor parameters<br>▸ Automatic motor data identification | ● | dynamic |
| 25 | Identify resolver error<br>▸ Optimising resolver behaviour | ● | dynamic |
| 26 | CAN reset node<br>▸ Reinitialising the CANopen interface | | static |
| 28 | Check MasterPin<br>▸ Unlocking the inverter with a MasterPin | ● | static |
| 29 | Set binding ID<br>▸ Device personalisation | | static |
| 30 | Delete binding ID<br>▸ Device personalisation | | static |
| 31 | Set password<br>▸ Password protection | | static |
| 32 | Check password<br>▸ Password protection | | static |
| 33 | Delete password<br>▸ Password protection | | static |
| 34 | Identify pole position (360°)<br>▸ Pole position identification (PPI) | ● | dynamic |

### 4.1.1 Load Lenze setting

The C00002/1 = "1: On / start" device command resets the parameters to the Lenze setting which are saved in the inverter firmware.

- Can only be executed if the controller is inhibited; otherwise, the feedback C00002/1 = "6: No access - controller inhibit" will be returned.
- All parameter changes made since the last saving of the parameter set will get lost!
- This device command has an effect on the settings of the parameters of the operating system, application and module.



**How to load the Lenze setting:**

1. If the inverter is enabled, it must be inhibited, e.g. by executing the "Enable/Inhibit inverter" device command "(C00002/16 = "0: Off / ready").
2. Execute the "Load Lenze setting" device command:
   C00002/1 = "1: On / start"

   The loading process may take a couple of seconds. After the device command has been called, C00002/1 returns dynamic status information ("Work in progress 20 %" → "Work in progress 40 %" → "Work in progress 60 %", etc.).

**Notes on execution time**

The execution time of this device command usually is 2.5 s, independent of the communication module used. If - in addition to the device-internal time monitoring - a monitoring of the execution time is to be implemented for this device command in a higher-level control, the monitoring time has to be 2 ... 3 times higher.

### 4.1.2 Load all parameter sets

The C00002/6 = "1: On / start" device command reloads all parameter settings from the memory module to the inverter.

- Can only be executed if the controller is inhibited; otherwise, the feedback C00002/6 = "6: No access - controller inhibit" will be returned.
- All parameter changes made since the last saving of the parameter set will get lost!
- This device command has an effect on the settings of the parameters of the operating system, application and module.

**Note!**

The inverter is currently provided with one data record for all parameters, i.e. every parameter has a value. Several data records per inverter are in preparation.

The basic function Parameter change-over provides a change-over between four sets with different parameter values for up to 32 freely selectable parameters. (📖 882)

**How to load the parameter settings from the memory module:**

1. If the inverter is enabled, it must be inhibited, e.g. by executing the "Enable/Inhibit inverter" device command "(C00002/16 = "0: Off / ready").
2. Execute the "Load all parameter sets" device command:
   C00002/6 = "1: On / start"

   The loading process may take a couple of seconds. After the device command has been called, C00002/6 returns dynamic status information ("Work in progress 20 %" → "Work in progress 40 %" → "Work in progress 60 %", etc.).

#### Notes on execution time

The execution time of this device command depends on the communication module used. If - in addition to the device-internal time monitoring - a monitoring of the execution time is to be implemented for this device command in a higher-level control, the monitoring time has to be 2 ... 3 times higher than the times specified in the following table:

| Communication module used | Typical execution time | | |
|---|---|---|---|
| | StateLine C | HighLine C | TopLine C |
| Without communication module | 3.5 s | 3.5 s | 3.5 s |
| PROFIBUS® or INTERBUS | 3.5 s | 3.5 s | 3.5 s |
| EtherCAT®, PROFINET® or EtherNet/IP™ | 4.0 s | 3.5 s | 4.0 s |
| Ethernet POWERLINK | 15.0 s | 15.0 s | 15.0 s |

### 4.1.3 Save all parameter sets

If parameter settings are changed in the inverter, those changes will be lost after mains switching of the inverter unless the settings have been saved explicitly.

The C00002/11 = "1: On / start" device command saves the current parameter settings safe against mains failure to the memory module of the inverter.

**Note!**

When the device is switched on, all parameters are automatically loaded from the memory module to the main memory of the inverter.

Observe the following to avoid data inconsistencies which cause errors when the parameters are loaded from the memory module:

During the storage process:

- Do not switch off the supply voltage!
- Do not remove the memory module from the device!

The inverter is currently provided with one data record for all parameters, i.e. every parameter has a value. Several data records per inverter are in preparation.

**How to save the parameter settings to the memory module:**

Execute the "Save all parameter sets" device command:
C00002/11 = "1: On / start"

The storage process may take a couple of seconds. After the device command has been called in C00002/11, dynamic status information ("Work in progress 20%" → "Work in progress 40%" → "Work in progress 60%", etc.) is returned.

**Tip!**

- This device command can also be activated via the icon in the *Toolbar*.
- The "Load Lenze setting" device command (C00002/1 = "1: On / start") resets the parameter settings to the delivery status of the device.

#### Notes on execution time

The execution time of this device command depends on the communication module used. If - in addition to the device-internal time monitoring - a monitoring of the execution time is to be implemented for this device command in a higher-level control, the monitoring time has to be 2 ... 3 times higher than the times specified in the following table:

| Communication module used | Typical execution time | | |
|---|---|---|---|
| | StateLine C | HighLine C | TopLine C |
| Without communication module | 2.0 s | 2.5 s | 2.5 s |
| PROFIBUS® or INTERBUS | 3.5 s | 3.5 s | 4.0 s |
| EtherCAT®, PROFINET® or EtherNet/IP™ | 4.5 s | 4.5 s | 4.5 s |
| Ethernet POWERLINK | 11.0 s | 11.0 s | 11.0 s |

### 4.1.4 Enable/inhibit inverter

The C00002/16 = "1: On / start" device command enables the inverter, provided that no other source of an inverter inhibit is active.

The C00002/16 = "0: Off / ready" device command inhibits the inverter again, i.e. the power output stages in the inverter are inhibited and the speed/current controllers of the motor control are reset.

- The motor becomes torqueless and coasts, if it has not yet been in standstill.
- When the controller is inhibited, the status output *bCInhActive* of the LS_DriveInterface system block is set to TRUE.
- When the controller inhibit request is reset, the drive synchronises to the actual speed. For this purpose,
  - If the flying restart circuit is activated in C00990, the flying restart function parameterised in C00991 is used for the synchronisation to the rotary or standing drive. ▸ Flying restart function (📖 271)
  - In the case of an operation with feedback, the actual speed is read out by the encoder system.
  - In the case of a sensorless vector control (SLVC), the actual speed from the motor model of the motor control is used for the synchronisation.
- C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit:

| Bit | Cause/Source of controller inhibit |
|---|---|
| Bit 0 | Terminal controller enable |
| Bit 1 | CAN control word |
| Bit 2 | MCI control word |
| Bit 3 | SwitchOn |
| Bit 4 | Application (LS_DriveInterface system block: *bCInh* input) |
| **Bit 5** | **Device command** (C00002/16) |
| Bit 6 | Error with a "Fault"/"Trouble" error response or system error, respectively |
| Bit 7 | Internal signal |
| Bit 8 | Reserved |
| Bit 9 | Reserved |
| Bit 10 | AutoStartLock |
| Bit 11 | Motor parameter identification |
| Bit 12 | Automatic brake operation |
| Bit 13 | DCB-IMP |
| Bit 14 | Reserved |
| Bit 15 | Reserved |

**Tip!**

The inverter can also be enabled or inhibited via the and *toolbar* icons.

### 4.1.5 Activate/deactivate quick stop

The C00002/17 = "1: On / start" device command activates the quick stop function, i.e. the motor control is separated from the setpoint selection, and within the deceleration time parameterised inC00105 the motor is brought to a standstill ($n_{act}$ = 0).

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00105 | Decel. time - quick stop | 2.000 | s |

- The motor is kept at a standstill during closed-loop operation.
- A pulse inhibit is set if the auto-DCB function has been activated via C00019.
- From version 02.00.00, further options for the quick stop function can be activated in C00104/1.
  ▸ Optional settings (📖 112)
- C00159 provides a bit coded representation of all active sources/triggers of a quick stop:

| Bit | Cause/source for quick stop |
|---|---|
| Bit 0 | Reserved |
| Bit 1 | CAN control word (bit 2) |
| Bit 2 | MCI control word (bit 2) |
| Bit 3 | Reserved |
| Bit 4 | Application (LS_MotorInterface system block: *bQspOn* input) |
| **Bit 5** | **Device command** (C00002/17) |
| Bit 6 | Device error with "TroubleQSP" error response |
| Bit 7 | Internal signal |
| Bit 8 | Reserved |
| Bit 9 | Reserved |
| Bit 10 | Operating system of |
| Bit 11 | Reserved |
| Bit 12 | MCK (System block LS_MotionControlKernel: Input *bQspOn*) |
| Bit 13 | Reserved |
| Bit 14 | Reserved |
| Bit 15 | Reserved |

#### Deactivate quick stop again

The C00002/17 = "0: Off / ready" device command deactivates the quick stop again, provided that no other source of a quick stop is active.

#### 4.1.5.1 Optional settings

From version 02.00.00, the options described in the following for the quick stop function can be activated in C00104/1. In the Lenze setting, no option is activated.

**Standstill position-controlled**

When bit 0 is set in C00104/1, the position control is activated at standstill.

- The basic conditions are shown in the Flow diagram for the quick stop function.
- More conditions for a proper function:
  The inputs *nPosCtrlOutLimit_a* and *nPosCtrlPAdapt_a* at the LS_MotorInterface system block have to be set to 100 % (default setting for all technology applications).

**Ramp position-controlled**

If bit 1 is set in C00104/1, the position control is activated while ramping down.

- The basic conditions are shown in the Flow diagram for the quick stop function.
- More conditions for a proper function:
  The inputs *nPosCtrlOutLimit_a* and *nPosCtrlPAdapt_a* at the LS_MotorInterface system block have to be set to 100 % (default setting for all technology applications).

**Continue following error**

(from version 15.00.00)

If bit 2 is set in C00104/1, the current following error is continued if the quick stop with position control is activated while ramping down.

**Starting value is the setpoint speed**

(from version 15.00.00)

If bit 3 is set in C00104/1, the starting value for quick stop is the setpoint speed. If the bit is not set, the starting value for quick stop is the actual speed if this does not derive more than 5 rpm from the setpoint. This serves to prevent a jump to a setpoint with a high deviation.

**Use position encoder**

(from version 15.00.00)

If bit 4 is set in C00104/1, the position encoder is used instead of the speed encoder for quick stop with position control.

- The basic conditions are shown in the Flow diagram for the quick stop function.

**Standstill depending on the actual speed**

(from version 16.00.00)

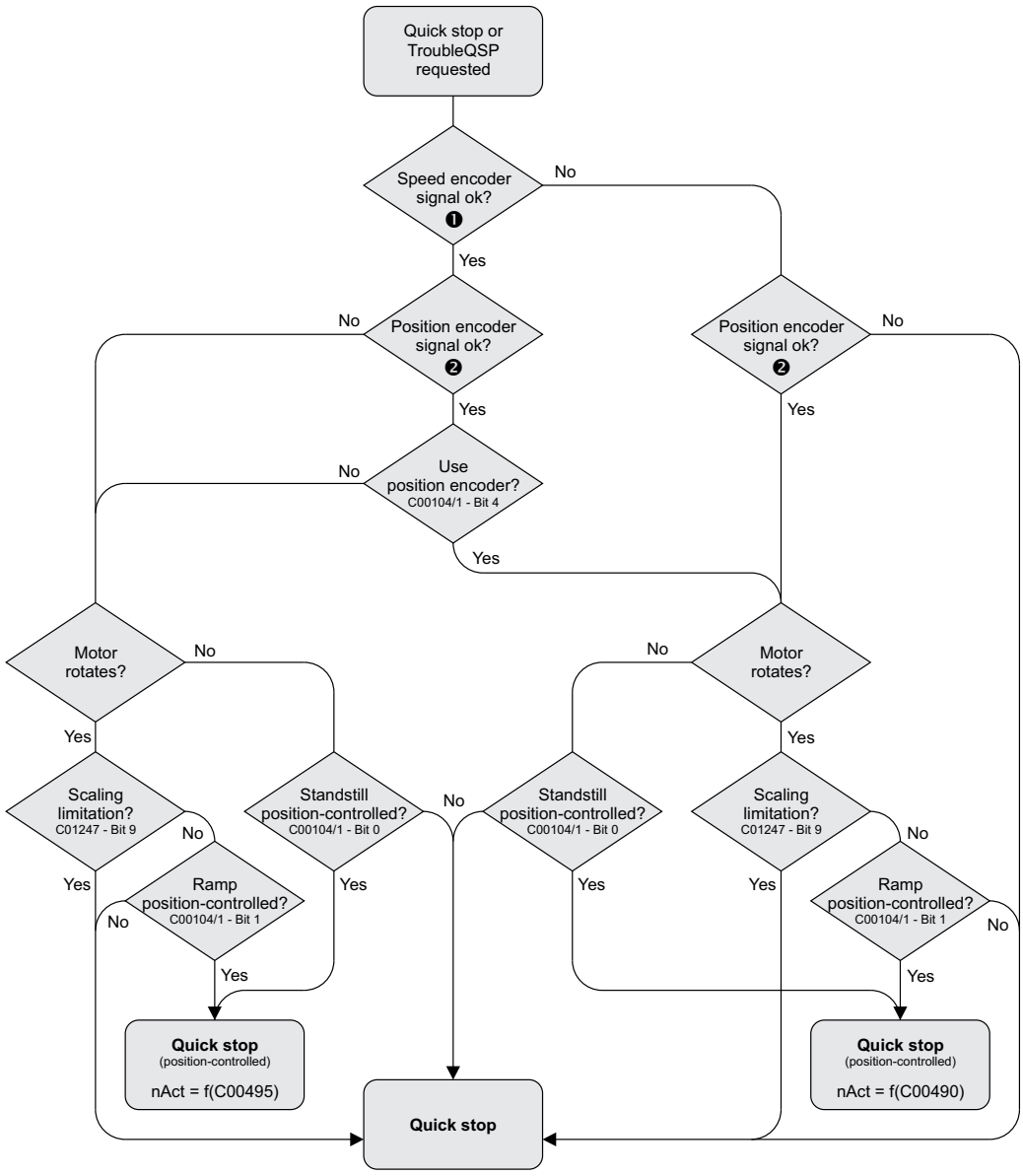
Extended option for quick stop with position control at standstill by setting C00104/1, Bit5 . When this setting is activated, the standstill of the motor shaft is detected under consideration of the speed setpoint and the actual speed.

The standstill of the motor shaft is reached when

- Bit5 = FALSE (Lenze setting)
  - Speed setpoint = 0
- Bit5 = TRUE
  - speed setpoint = 0 AND 0 ≤ actual speed value ≤ 1 rpm

### 4.1.5.2 Flow diagram

**Flow diagram for quick stop function** (from version 15.00.00)




| | |
|---|---|
| ❶ | The "speed encoder signal ok" status calls for the following requirements:<br>• Speed sensor is set in C00495<br>• Speed sensor is running (no open circuit) |
| ❷ | The "position encoder signal ok" status calls for the following requirements:<br>• Position encoder is set in C00490<br>• Position encoder is running (no open circuit) |

Although the use of SSI absolute value encoders and 2-track HTL encoders is possible, the error detection system of both encoders always returns "OK" as error status !

### 4.1.6 Reset error

The C00002/19 = "1: On / start" device command acknowledges an existing error message if the error cause has been eliminated and thus the error is no longer pending.

- After the reset (acknowledgement) of the current error, further errors may be pending which must also be reset.
- The status determining error is displayed in C00168.
- The current error is displayed in C00170.



**Tip!**

An error message can also be acknowledged by activating the **Reset error** button in the **Diagnostics** tab.

Detailed information on error messages can be found in the "Diagnostics & error management" chapter. (📖 691)

### 4.1.7 Delete logbook

The C00002/21 = "1: On / start" device command deletes all logbook entries.

**Tip!**

To display the logbook in the »Engineer«, click the **Logbook** button on the **Diagnostics** tab.

In the *Logbook* dialog box, it is also possible to delete all logbook entries by clicking the **Delete** button.

Detailed information on the logbook can be found in the "Diagnostics & error management" chapter. (📖 691)

### 4.1.8 Device search function

In some applications where inverters are installed in control cabinets or are positioned in a spacious plant, it is often difficult to locate a device connected online for e.g. maintenance work. There is an established online connection with the device but you do not know where the inverter is located physically.

The C00002/27 = "1: On / start" device command serves to carry out an "optical location":

- For the time set in C00181/1, all four status LEDs at the front of the inverter flash. Afterwards, the function is turned off automatically.
- If the device command is executed again within the set time period, the duration is extended accordingly.
- The setting C00002/27 = "0: Off / ready" serves to abort or switch off the function.
- Adjustable time period: 0 ... 6000 s (Lenze setting: 5 s)

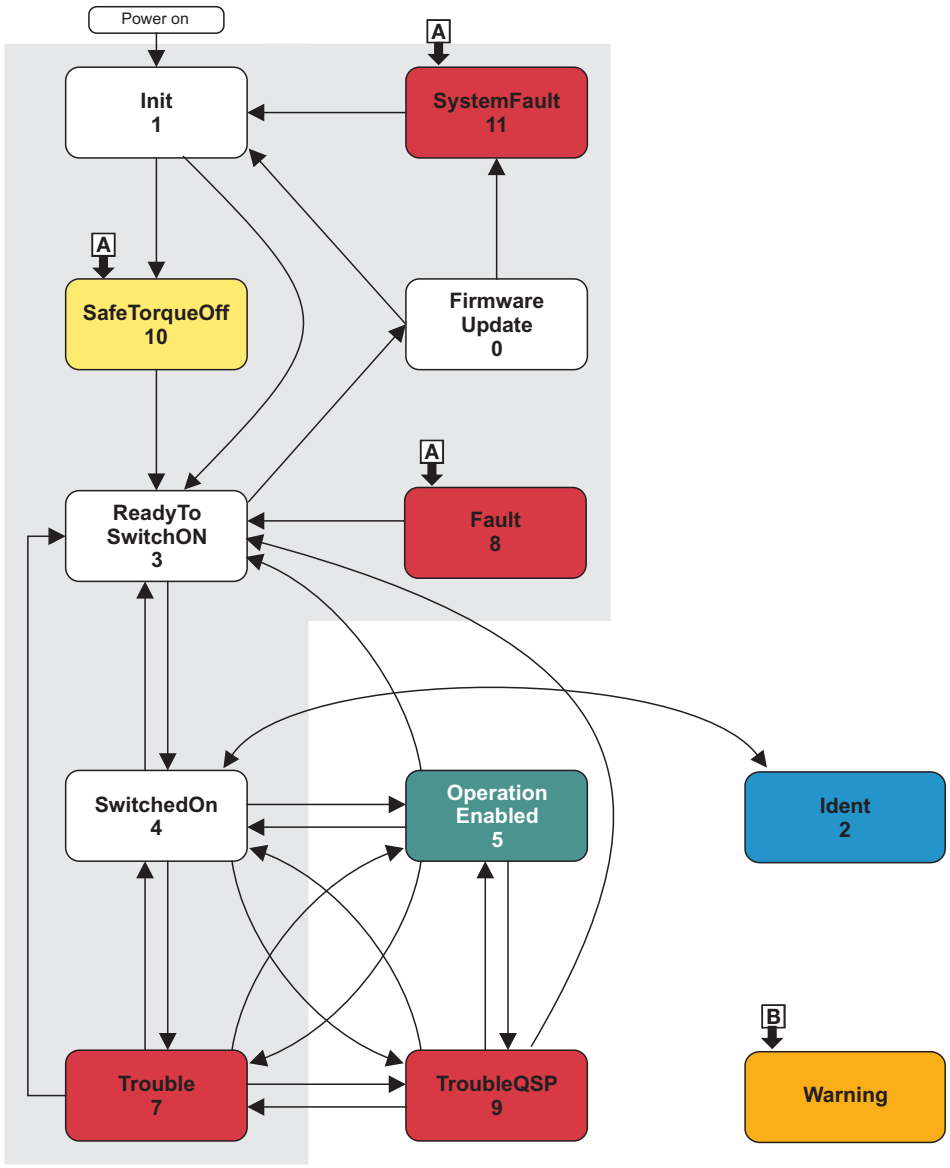
**Tip!**

The device search function can also be activated via the *toolbar* icon.

## 4.2 Device state machine and device states

The behaviour of the inverter is mainly determined by the current device status within the device state machine. Which device status is active and which device status is next depends on certain control signals (e.g. for controller inhibit and quick stop) and status parameters.

**Device state machine**




Grey field: Pulse inhibit

| | |
|---|---|
| A | Can be reached from all statuses. |
| B | "Warning" contradicts the definition of a device status. In fact, it is a message which is to call attention to the device status the warning exists for.<br>"Warning" can occur in parallel to other device states. |

- The arrows between the device states mark possible state changes.
- The digits stand for the status ID (see table below).

- The change from one status to the other is carried out within a 1-ms cycle. If within this time there are several requests for status changes, the status with the higher priority is processed first (see table below).
- The C00137 displays the current device status.
- C00150 (status word) provides a bit coded representation of the current device status via bits 8 ... 11 (see table below).

| ID | Device status (Display in C00137) | Priority | Status bits (Display in C00150) | | | | Meaning |
|---|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 | |
| **0** | FirmwareUpdate | - | 0 | 0 | 0 | 0 | Firmware update function is active |
| **1** | Init | - | 0 | 0 | 0 | 1 | Initialisation active |
| **2** | Ident | - | 0 | 0 | 1 | 0 | Identification active |
| **3** | ReadyToSwitchOn | Prio 5 | 0 | 0 | 1 | 1 | Device is ready to start |
| **4** | SwitchedOn | Prio 4 | 0 | 1 | 0 | 0 | Device is switched on |
| **5** | OperationEnabled | Prio 1 | 0 | 1 | 0 | 1 | Operation |
| **6** | - | - | 0 | 1 | 1 | 0 | - |
| **7** | Trouble | Prio 3 | 0 | 1 | 1 | 1 | Trouble active |
| **8** | Fault | Prio 7 | 1 | 0 | 0 | 0 | Error active |
| **9** | TroubleQSP | Prio 2 | 1 | 0 | 0 | 1 | TroubleQSP is active |
| **10** | SafeTorqueOff | Prio 6 | 1 | 0 | 1 | 0 | Safe torque off is active |
| **11** | SystemFault | Prio 8 | 1 | 0 | 1 | 1 | System fault active |

[4-1] Device statuses, priorities, and meaning of the status bits in the status word

### 4.2.1 FirmwareUpdate

**Note!**

This function may only be executed by qualified Lenze personnel!

### 4.2.2 Init

| "DRV-RDY" LED | LED "DRV-ERR" | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| OFF | OFF | Init | 0 | 0 | 0 | 1 |

The inverter is in this status immediately after switching on its 24 V supply voltage.

In the "Init" status, the operating system is initialised and all device components (communication module, memory module, power section, etc.) are identified. When identifying the power section, it is checked first if it is switched on or if the required voltage lies within the tolerance zone, respectively.

- The inverter is inhibited, i.e. the motor terminals (U, V, W) of the inverter are deenergised.
- The digital and analog inputs are not yet evaluated at this time.
- The bus systems (CAN, PROFIBUS etc.) do not work yet, i.e. communication is not possible.
- The application is not yet processed.
- The monitoring functions are not active yet.
- The inverter cannot be parameterised yet and no device commands can be carried out yet.

**Note!**

If the 24V voltage supply is in the valid range (>19V) and the initialisation is finished, the device changes automatically to the "ReadyToSwitchOn" status.

If only the 24V voltage supply is available during the mains connection, the error message "LU: Undervoltage in the DC bus" is also entered into the logbook of the inverter.

### 4.2.3 Ident

| "DRV-RDY" LED | LED "DRV-ERR" | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | OFF | Ident | 0 | 0 | 1 | 0 |

The inverter has been provided with functions for automatically detecting the motor parameters, the resolver error and the pole position of a synchronous motor. If such an "identification" is active, the inverter is in the "Ident" status.

The "Ident" device status can only be reached from the "SwitchedOn" device status, i.e. the inverter must be inhibited first so that identification can be started afterwards via the corresponding device command:

| Device command | Function | Detailed information |
|---|---|---|
| C00002/23 | Identify motor parameters | ▸ Automatic motor parameter identification |
| C00002/25 | Identify resolver error | e ▸ Optimise resolver behaviour |
| C00002/34 | Identify pole position (360°) | ▸ Pole position identification |

**Stop!**

During the identification of the motor parameters or the pole position, the inverter does not respond to setpoint changes or control processes (e.g. speed setpoints, quick stop, torque limitations).

While an identification is being executed,

- the application remains active,
- all system interfaces (IO, bus systems, etc.) remain active,
- error monitoring remains active,
- the inverter is controlled independently of the setpoint sources.

After the identification is completed, the status changes back to "SwitchedOn".

### 4.2.4 SafeTorqueOff

| "DRV-RDY" LED | LED "DRV-ERR" | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
|  | OFF | SafeTorqueOff | 1 | 0 | 1 | 0 |

**Note!**

Up to and including version 13.xx.xx the following applies:

This device status is only possible in connection with an integrated safety system and if a power section supply is available!

From version 14.00.00 the following applies:

This device status is only possible in connection with an integrated safety system!

**Integrated safety systems with Inverter Drives 8400**

Inverters of the 8400 series can be equipped with the integrated "Safe torque off (STO)" safety system.

The integrated safety system is applicable on machines for the protection of persons.

The drive function is still carried out by the inverter. The safety system provides safe inputs. If the safety system is activated, it executes control functions according to EN 60204-1 directly in the inverter in case of errors.

**Safety state**

If the inverter is switched off by the safety system, the device changes to the "SafeTorqueOff" status.

If the safety system deactivates the "Safe torque off (STO)" request, the device changes to the "ReadyToSwitchOn" status.

Detailed information on the integrated safety system can be found in the hardware manual!

**The hardware manual contains important notes on the safety system which must be observed!**

The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 inverter.

### 4.2.5 ReadyToSwitchOn

| "DRV-RDY" LED | LED "DRV-ERR" | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
|  | OFF | ReadyToSwitchOn | 0 | 0 | 1 | 1 |

The inverter is in this device status directly after the initialisation has been completed!

- The bus systems are running and the terminals and encoders are evaluated.
- The monitoring functions are active.
- The inverter can be parameterised.
- The application is basically executable.

**Note!**

- The "ReadyToSwitchOn" state is not only activated after the mains connection, but also after the deactivation of "Trouble", "Fault" or "SafeTorqueOff".
- If C00142 activates the autostart option "Inhibit at power-on" (Lenze setting), explicit deactivation of the controller inhibit after mains connection is always required for the inverter to change from the "ReadyToSwitchOn" status to the "SwitchedOn" status.
- If only the 24V voltage supply is available during the mains connection, the error message "LU: Undervoltage in the DC bus" is entered into the logbook of the inverter and the inverter remains in the "ReadyToSwitchOn" status.

**Danger!**

If the "Inhibit at power-on" auto-start option has been deactivated in C00142, the "ReadyToSwitchOn status switches directly to the "SwitchedOn" status after mains connection.

▸ Automatic restart after mains connection/fault... (📖 127)

### 4.2.6 SwitchedOn

| "DRV-RDY" LED | LED "DRV-ERR" | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
|  | OFF | SwitchedON | 0 | 1 | 0 | 0 |

The drive is in this device status if the DC bus voltage is applied and the inverter is still inhibited by the user (controller inhibit).

- The bus systems are running and the terminals and encoders are evaluated.
- The monitoring functions are active.
- The application is basically executable.

If the controller inhibit is deactivated, the devices changes to the "OperationEnabled" status and the motor follows the setpoint defined by the active application.

**Tip!**

C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit.

Depending on certain conditions, a status change takes place based on the "SwitchedOn" device status:

| Change condition | Changeover to the device status |
|---|---|
| Control bit "EnableOperation" of all control channels = "1" **AND** terminal RFR = HIGH level (controller enable) | OperationEnabled |
| Control bit "SwitchOn" of a control channel = "0". | ReadyToSwitchOn |
| Identification requested. | Ident |
| Undervoltage in the DC bus. | Trouble/Fault (depending on C00600/1) |
| Error with error response "Trouble" occurs. | Trouble |
| Error with error response "TroubleQSP" occurs. | TroubleQSP |

**Related topics:**

- wCANControl/wMCIControl control words (📖 133)

### 4.2.7 OperationEnabled

| "DRV-RDY" LED | LED "DRV-ERR" | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | OFF | OperationEnabled | 0 | 1 | 0 | 1 |

The inverter is in this device status if the controller inhibit is deactivated and no error ("Trouble" or "TroubleQSP") has occurred.

If the operation is enabled and the magnetisation in case of servo control (SC) and sensorless vector control (SLVC) has been completed, the motor follows the setpoint determined by the active application.

Depending on certain conditions, a status change takes place based on the "OperationEnabled" device status.

| Change condition | Changeover to the device status |
|---|---|
| Control bit "EnableOperation" of a control channel = "0"<br>**OR** terminal RFR = LOW level (controller inhibit). | SwitchedOn |
| Control bit "SwitchOn" of a control channel = "0". | ReadyToSwitchOn |
| Undervoltage in the DC bus. | Trouble/Fault (depending on C00600/1) |
| Error with error response "Trouble" occurs. | Trouble |
| Error with error response "TroubleQSP" occurs. | TroubleQSP |

**Related topics:**

- wCANControl/wMCIControl control words (📖 133)

### 4.2.8 TroubleQSP

| "DRV-RDY" LED | LED "DRV-ERR" | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
|  |  | TroubleQSP | 1 | 0 | 0 | 1 |

This device status will be active as soon as a monitoring mode responds the error response "TroubleQSP" has been parameterised for.

- The drive is decelerated to standstill with torque within the deceleration time parameterised for quick stop independently of the defined setpoint and can be kept there.
- The device status can only be abandoned by acknowledging the error if the error cause is removed.
- When the controller is inhibited, it is possible to jump to the "SwitchedOn" status even during the error status since the controller inhibit function has a higher priority. As long as the error is pending and has not been acknowledged, the status is changed back to the "TroubleQSP" status when the controller is enabled afterwards.

Depending on certain conditions a status change takes place based on the "TroubleQSP" device status.

| Change condition | Changeover to the device status |
|---|---|
| Control bit "SwitchOn" of a control channel = "0". | ReadyToSwitchOn |
| Control bit "EnableOperation" of all control channels = "1"<br>**AND** terminal RFR = HIGH level (controller enable)<br>**AND** error is reset by the control bit "ResetFault"<br>**AND** no more errors are pending. | OperationEnabled |
| Control bit "EnableOperation" of a control channel = "0"<br>**OR** terminal RFR = LOW level (controller inhibit)<br>**AND** error is reset by the control bit "ResetFault"<br>**AND** no more errors are pending. | SwitchedOn |
| A message is active in the system. | Trouble |

**Related topics:**

- wCANControl/wMCIControl control words (📖 133)
- Basics on error handling in the inverter (📖 691)
- Error messages of the operating system (📖 715)

### 4.2.9 Trouble

| "DRV-RDY" LED | LED "DRV-ERR" | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| OFF |  | Trouble | 0 | 1 | 1 | 1 |

This device status becomes active as soon as a monitoring mode responds for which the error response "Trouble" has been parameterised.

- The motor has no torque (is coasting) when the inverter is inhibited.
- The "Trouble" device status is automatically exited if the error cause has been removed.

**Note!**

If in C00142 the "Inhibit at trouble" is activated, explicit deactivation of the controller inhibit is required before this status can be abandoned.

Depending on certain conditions a status change takes place based on the "Trouble" device status.

| Change condition | Changeover to the device status |
|---|---|
| The error cause is no longer active. | ReadyToSwitchOn |
| Control bit "EnableOperation" of all control channels = "1"<br>**AND** terminal RFR = HIGH level (controller enable)<br>**AND** the message has been cancelled. | OperationEnabled |
| Control bit "EnableOperation" of a control channel = "0"<br>**OR** terminal RFR = LOW level (controller inhibit)<br>**AND** the message has been cancelled. | SwitchedOn |
| In the system, there is an error configured on "TroubleQSP".<br>**AND** the message has been cancelled. | TroubleQSP |

**Related topics:**

- wCANControl/wMCIControl control words (📖 133)
- Basics on error handling in the inverter (📖 691)
- Error messages of the operating system (📖 715)

### 4.2.10 Fault

| "DRV-RDY" LED | LED "DRV-ERR" | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| OFF |  | Fault | 1 | 0 | 0 | 0 |

This device status will be active as soon as a monitoring mode responds the error response "Fault" has been parameterised for.

- The motor has no torque (is coasting) when the inverter is inhibited.
- The error must explicitly be reset ("acknowledged") in order to exit the device status, e.g. by the device command "Reset error" or via the control bit "ResetFault" in the control word *wCanControl* or *wMCIControl*.

**Note!**

If an undervoltage in the DC bus of the inverter occurs (error message "LU"), the device changes to the "Trouble" status.

An additional error of higher priority leads the device into the "Fault" status.

According to the Device state machine, the device changes to the "ReadyToSwitchOn" status after acknowledging the error although the undervoltage is still available!

If the "Inhibit at fault" auto-start option has been activated in C00142, explicit deactivation of the controller inhibit is required before the status can be abandoned.

**Related topics:**

- wCANControl/wMCIControl control words (📖 133)
- Basics on error handling in the inverter (📖 691)
- Error messages of the operating system (📖 715)

### 4.2.11 SystemFault

| "DRV-RDY" LED | LED "DRV-ERR" | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| OFF |  | SystemFault | 1 | 0 | 1 | 1 |

This device status becomes active if a system fault occurs.

- The device status can only be abandoned by
  - mains switching or
  - a system restart *(in preparation)*.

## 4.3 Automatic restart after mains connection/fault...

### .../Error/undervoltage/loading of the Lenze setting

In C00142, the starting performance of the inverter after mains connection, undervoltage, loading of the Lenze setting as well as a "Trouble" or a "Fault" reset can be parameterised individually:

| Auto-start option (C00142) | | Lenze setting |
|---|---|---|
| Bit 0 | "Inhibit at power-on" auto-start option | 1 ≡ Inhibit is active |
| Bit 1 | Inhibit at trouble | 0 ≡ Inhibit is not active |
| Bit 2 | Inhibit at fault | 0 ≡ Inhibit is not active |
| Bit 3 | Inhibit at undervoltage | 1 ≡ Inhibit is active |
| Bit 4 | Auto-start option "Inhibit at Lenze setting" | 1 ≡ Inhibit is active |
| Bit 5 | Reserved | 0 |
| Bit 6 | | |
| Bit 7 | | |

**Note!**

In the Lenze setting , automatic restart after mains connection, undervoltage, and loading of the Lenze setting is inhibited.

### 4.3.1 "Inhibit at power-on" auto-start option

The auto-start option "Inhibit at power-on" prevents the change to the "SwitchedOn" status after mains connection if the controller is already enabled at mains connection.

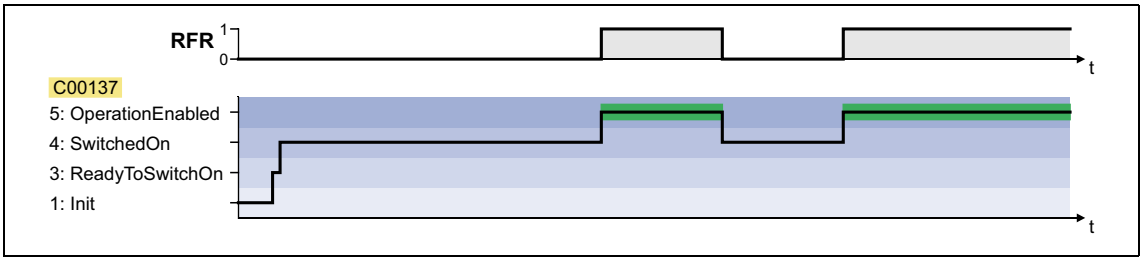
**Danger!**

If the "Inhibit at power-on" auto-start option has been deactivated in C00142, (bit 0 = 0), the motor can directly start to run if the controller is enabled after mains connection!

The following three cases describe the behaviour of the inverter after mains connection depending on whether the controller is enabled and the set auto-start option. Here, it is assumed that after mains connection, no errors and trouble occur in the inverter and the "EnableOperation" control bit in the *wDriveControl* is set to "1".

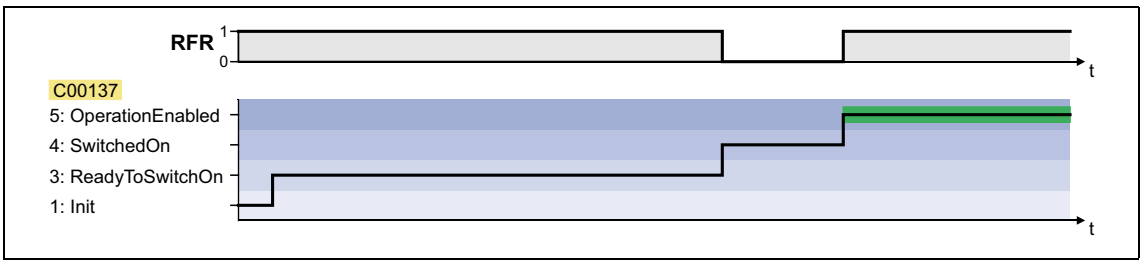
**Case 1: No controller enable at mains connection**

If the controller is not enabled at mains connection, the inverter remains in the "SwitchedOn" status. Only with the controller enable, the device changes to the "OperationEnabled" status, independent of the set auto-start option:



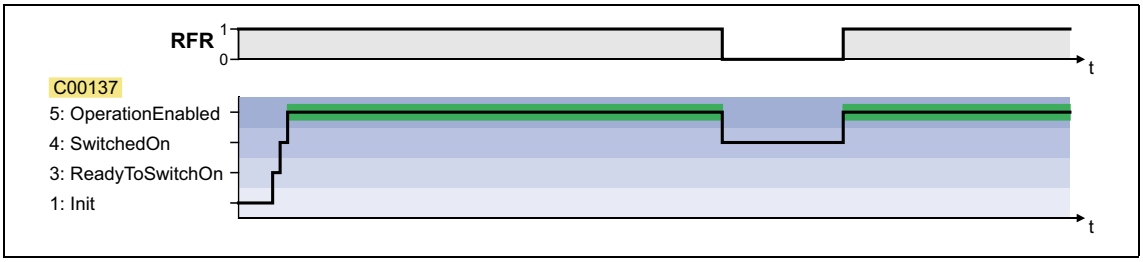

**Case 2: Controller enable at mains connection and "Inhibit at power-on" activated**

If the controller is enabled at mains connection and the auto-start option "Inhibit at power-on" is activated, the inverter remains in the "ReadyToSwitchOn" status. For changing to the "SwitchedOn" status, the controller enable must first be deactivated. Only when the controller is enabled again afterwards, the status changes to "OperationEnabled":




**Case 3: Controller enable at mains connection and "Inhibit at power-on" deactivated**

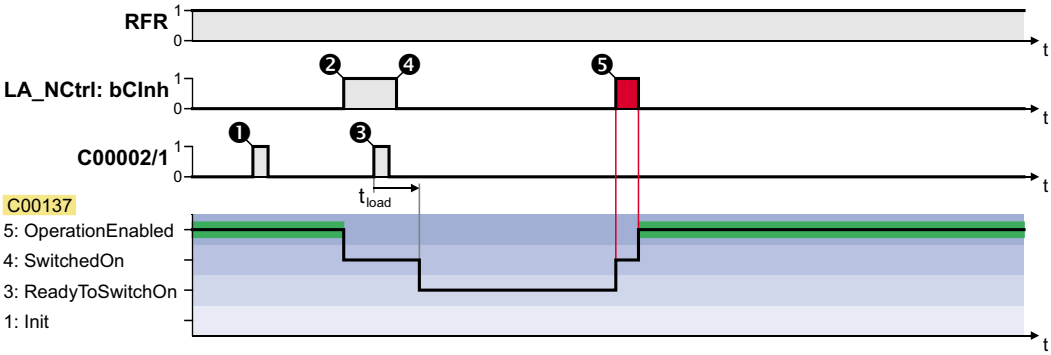
If in C00142 the autostart option "Inhibit at power-on" is deactivated (bit 0 = 0), the status first changes from "ReadyToSwitchOn" to "SwitchedOn" and then to "OperationEnabled" after mains connection with an enabled controller:



### 4.3.2 Auto-start option "Inhibit at Lenze setting"

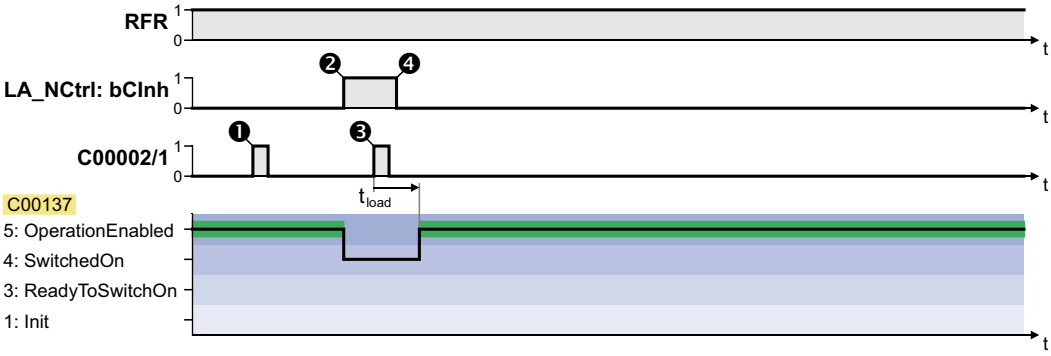
The "Inhibit at Lenze setting" auto-start option configurable via bit 4 of C00142 prevents the change to the "SwitchedOn" status after the Lenze setting has been loaded and the controller is enabled.

For a change to the "SwitchedOn" status, the controller enable must first be deactivated after the Lenze setting has been loaded. Only if the controller is enabled again afterwards, the status changes to "OperationEnabled":




❶ Loading of the Lenze setting not possible since controller inhibit not set.
❷ Controller inhibit via signal change by the user
❸ Loading of the Lenze setting via device command C00002/1 = "1"
❹ Controller enable via signal change by the user
❺ Controller inhibit and subsequent controller enable via signal change by the user
$t_{load}$ = time required for loading the Lenze setting

[4-1] Example 1: Behaviour with activated auto-start option "Inhibit at Lenze setting" (C00142: Bit 4 = "1")

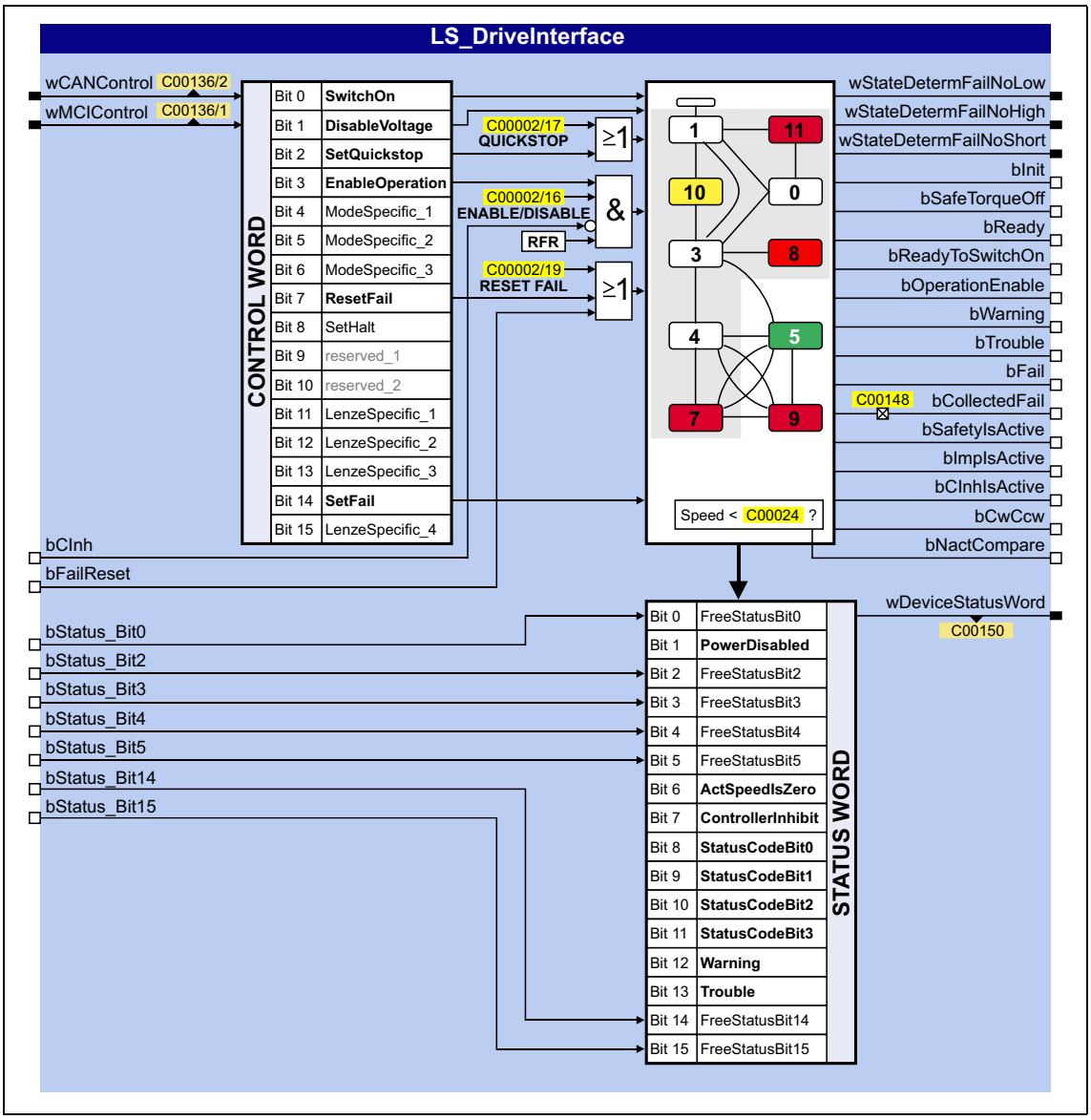



❶ Loading of the Lenze setting not possible since controller inhibit not set.
❷ Controller inhibit via signal change by the user
❸ Loading of the Lenze setting via device command C00002/1 = "1"
❹ Controller enable via signal change by the user
$t_{load}$ = time required for loading the Lenze setting

[4-2] Example 2: Behaviour with deactivated auto-start option "Inhibit at Lenze setting" (C00142: Bit 4 = "0")

## 4.4 Internal interfaces | "LS_DriveInterface" system block

The **LS_DriveInterface** system block displays the device control in the FB Editor.



### Inputs

| Designator<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| wCANControl<br>C00136/2 \| WORD | Control word via system bus (CAN)<br>• The inverter controlled by a master control (e.g. IPC) receives its control word by the CANopen system bus interface. The process data word is provided at this input by the upstream port block LP_CanIn1.<br>• For a detailed description of the individual control bits, see chapter "wCANControl/wMCIControl control words". (📖 133) | |
| wMCIControl<br>C00136/1 \| WORD | Control word via communication module (e.g. PROFIBUS)<br>• The inverter controlled by a master control (e.g. IPC) receives its control word by a plugged-in communication module. The process data word is provided at this input by the upstream port block LP_MciIn1.<br>• For a detailed description of the individual control bits, see chapter "wCANControl/wMCIControl control words". (📖 133) | |
| bCInh<br>C00833/36 \| BOOL | ▸ Enable/inhibit inverter (📖 110) | |
| | FALSE | Enable inverter: The inverter switches to the "OperationEnabled" device status if no other source for controller inhibit is active.<br>• C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit. |
| | TRUE | Inhibit inverter (controller inhibit): The inverter switches to the "SwitchedOn" device status. |
| bFailReset<br>C00833/37 \| BOOL | ▸ Reset error message (📖 719) | |
| | FALSE↗TRUE | The current error is reset. |
| bStatus_Bit0<br>bStatus_Bit2<br>bStatus_Bit3<br>bStatus_Bit4<br>bStatus_Bit5<br>bStatus_Bit14<br>bStatus_Bit15<br>C00833/38 ... 44 \| BOOL | Freely assignable bits in the status word of the inverter<br>• You can use these bits for returning information to the master control (e.g. IPC). | |
| bFree_1<br>bFree_2<br>bFree_3<br>bFree_4<br>C00833/45 ... 48 \| BOOL | Reserved inputs have no function | |

### Outputs

| Designator<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| wDeviceStatusWord<br>C00150 \| WORD | Status word of the inverter (based on DSP-402)<br>• The status word contains all information relevant for controlling the inverter.<br>• The status word is sent as a process data word to the master control via a port block:<br>  • Port block LP_CanOut1 when the CANopen system bus interface is used or<br>  • Port block LP_MciOut when a plugged-in communication module is used (e.g. PROFIBUS).<br>• For a detailed description of each status bit see chapter "wDeviceStatusWord status word". (📖 135) | |
| wStateDetermFailNoLow<br>WORD | Display of the status determining error (32-bit error number, Low-Word)<br>• If the "Use I16BitFailNo" (bit 15 = "1") option is activated in C00148, the short 16-bit error number (*wStateDetermFailNoShort*) is provided via this output as well.<br>  • In this case, the *wStateDetermFailNoHigh* output is "0".<br>  • Advantage: The bus transfer of the error numbers is possible via a data word without changing the interconnection of the technology application. | |
| wStateDetermFailNoHigh<br>WORD | Display of the status determining error (32-bit error number, High-Word) | |
| wStateDetermFailNoShort<br>WORD | Display of the status determining error (16-bit error number) | |
| bInit<br>BOOL | TRUE | "Init" device state is active |
| bSafeTorqueOff<br>BOOL | TRUE | "SafeTorqueOff" device state is active |
| bReady<br>BOOL | TRUE | "SwitchedOn" device state is active |
| bReadyToSwitchOn<br>BOOL | TRUE | "ReadyToSwitchOn" device state is active |
| bOperationEnable<br>BOOL | TRUE | "OperationEnabled" device state is active |
| bWarning<br>BOOL | TRUE | A warning is indicated |
| bMessage<br>BOOL | TRUE | "Trouble" device state is active |
| bFail<br>BOOL | TRUE | "Fault" device state is active |
| bCollectedFail<br>BOOL | TRUE | Group error: A device status according to the group error configuration in C00148 has occurred, the drive is not able to follow the setpoint selection. |
| bSafetyIsActive<br>BOOL | TRUE | In preparation |
| bImpIsActive<br>BOOL | TRUE | Pulse inhibit is active |
| bCInhIsActive<br>BOOL | TRUE | Controller inhibit is active |
| bCwCcw<br>BOOL | FALSE | Motor rotates in CW direction |
| | TRUE | Motor rotates in CCW direction |
| bNactCompare<br>BOOL | TRUE | During open-loop operation:<br>Speed setpoint < comparison value (C00024, C00025/1) |
| | | During closed-loop operation:<br>actual speed value < comparison value (C00024, C00025/1) |

**Option "Lock bFail at TroubleQSP"**

The TroubleQSP device status gets active as soon as a monitoring function responds that has been parameterised for the "TroubleQSP" error response. Since the *bFail* status output is not set in this case, it is not recognisable after pulse inhibit due to e.g. automatic brake operation (as well for a higher-level control), why the drive is standing und does not start when the setpoint is selected. Only after an error reset, a setpoint is accepted again.

From version 02.00.00: If the "Lock bFail at TroubleQSP" option is activated (bit 14 = "1") in C00148, the *bFail* status output is also set to TRUE if the device status is TroubleQSP.

### 4.4.1 wCANControl/wMCIControl control words

The inverter is controlled by a master control (e.g. IPC) via the *wCanControl* or *wMCIControl* control word, respectively.

- *wCANControl*: Control word via system bus (CAN)
  - The process data word is provided at the *wCanControl* input via the upstream LP_CanIn1 port block.
  - Display parameter: C00136/2
- *wMCIControl*: Control word via a plugged-in communication module (e.g. PROFIBUS)
  - The process data word is provided at the *wMCIControl* input via the upstream LP_McIIn1 port block.
  - Display parameter: C00136/1
- The bit assignment for the *wCanControl*/*wMCIControl* control words can be seen from the table below.

**Note!**

The assignment of bits 11 ... 13 and bit 15 depends on the technology application selected in C00005!

- See description of the corresponding technology application.

| Bit | Name | Function |
|---|---|---|
| Bit 0 | SwitchOn | 1 ≡ Change to the "SwitchedOn" device status<br>• This bit has to be set in the CAN **AND** MCI control word in order that the drive changes to the "SwitchedOn" device state. In order to reach the "ReadyToSwitchOn" state, it is sufficient to set the bit to 0 in of the two control words. |
| Bit 1 | DisableVoltage | 1 ≡ Inhibit inverter control (pulse inhibit) |
| Bit 2 | SetQuickStop | Activate quick stop (QSP)<br>▸ Activate/deactivate quick stop (📖 111) |
| Bit 3 | EnableOperation | 1 ≡ Enable inverter (RFR)<br>• This bit must be set in CAN **AND** in the MCI control word, otherwise the controller will be inhibited. |
| Bit 4 | ModeSpecific_1 | Reserved (currently not assigned) |
| Bit 5 | ModeSpecific_2 | |
| Bit 6 | ModeSpecific_3 | |
| Bit 7 | ResetFault | 1 ≡ Reset fault (trip reset)<br>• Acknowledge error message (if the error cause has been eliminated). |
| Bit 8 | SetHalt | 1 ≡ Activate stop function<br>• Stop drive via stopping ramp (in preparation). |
| Bit 9 | reserved_1 | Reserved (currently not assigned) |
| Bit 10 | reserved_2 | |
| Bit 11 | LenzeSpecific_1 | Assignment depends on the selected technology application<br>• See description of the corresponding technology application. |
| Bit 12 | LenzeSpecific_2 | |
| Bit 13 | LenzeSpecific_3 | |
| Bit 14 | SetFail | 1 ≡ Set error (trip set) |
| Bit 15 | LenzeSpecific_4 | Assignment depends on the selected technology application<br>• See description of the corresponding technology application. |

**Tip!**

If a bus control is not wanted (e.g. in case of control via terminals):

Connect both control word inputs with the *wDriveCtrl* output signal of the LS_ParFix system block. This output signal has the fixed value "9", which corresponds to the following assignment:

- Bit 0, SwitchOn = 1
- Bit 3, EnableOperation = 1
- All others: 0

### 4.4.2 wDeviceStatusWord status word

The *wDeviceStatusWord* status word provided by the control system contains all information relevant for controlling the inverter.

- The status word is sent as a process data word to the master control via a port block:
  - The **LP_CanOut1** port block if "CAN on board" is used or
  - the **LP_MciOut1** port block if a plugged-in communication module is used (e.g. PROFIBUS).
- Display parameter: C00150
- The bit assignment of the *wDeviceStatusWord* status word can be seen from the table below.

| Bit | Name | Status |
|---|---|---|
| Bit 0 | FreeStatusBit0 | Free status bit 0 |
| Bit 1 | PowerDisabled | 1 ≡ Inverter control inhibited (pulse inhibit is active) |
| Bit 2 | FreeStatusBit2 | Free status bit 2 (not assigned, freely assignable) |
| Bit 3 | FreeStatusBit3 | Free status bit 3 (not assigned, freely assignable) |
| Bit 4 | FreeStatusBit4 | Free status bit 4 (not assigned, freely assignable) |
| Bit 5 | FreeStatusBit5 | Free status bit 5 (not assigned, freely assignable) |
| Bit 6 | ActSpeedIsZero | During open-loop operation:<br>1 ≡ Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>1 ≡ Actual speed value < Comparison value (C00024) |
| Bit 7 | ControllerInhibit | 1 ≡ Inverter is inhibited (controller inhibit is active) |
| Bit 8 | StatusCodeBit0 | Bit coded display of the active device status<br>▸ Device state machine and device states (see table [4-1]) |
| Bit 9 | StatusCodeBit1 | |
| Bit 10 | StatusCodeBit2 | |
| Bit 11 | StatusCodeBit3 | |
| Bit 12 | Warning | 1 ≡ A warning is indicated |
| Bit 13 | Trouble | 1 ≡ Inverter is in the "Trouble" device status<br>• E.g. if an overvoltage has occurred. |
| Bit 14 | FreeStatusBit14 | Free status bit 14 (not assigned, freely assignable) |
| Bit 15 | FreeStatusBit15 | Free status bit 15 (not assigned, freely assignable) |

*This page has been left blank intentionally,*
*to present the following information more clearly.*

# 5 Motor control (MCTRL)

This chapter provides information on the parameter setting of the inverter's internal motor control. The 8400 TopLine inverter both supports synchronous and asynchronous motors.

**Topics:**

Basic settings:


- Motor selection/Motor data
- Selecting the control mode
- Defining current and speed limits


Description of the motor control types:


- V/f characteristic control (VFCplus)
- V/f characteristic control - energy-saving (VFCplusEco)
- V/f control (VFCplus + encoder)
- Sensorless vector control (SLVC)
- Sensorless control for synchronous motors (SLPSM)
- Servo control (SC)


Parameterisable additional functions:


- Selection of switching frequency
- Operation with increased rated power
- Current-dependent stator leakage inductance Ppp(I)
- Flying restart function
- DC-injection braking
- Slip compensation
- Oscillation damping
- Phase sequence reversal for correcting misconnected UVW motor phases
- Field weakening for synchronous motors


Further topics:


- Position control/additive speed specification
- Braking operation/brake energy management
- Monitoring


Internal interfaces (process signals):

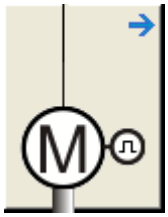

- Internal interfaces | System block "LS_MotorInterface"
- Internal status signals | System block "LS_DeviceMonitor"

## 5.1 Motor selection/Motor data

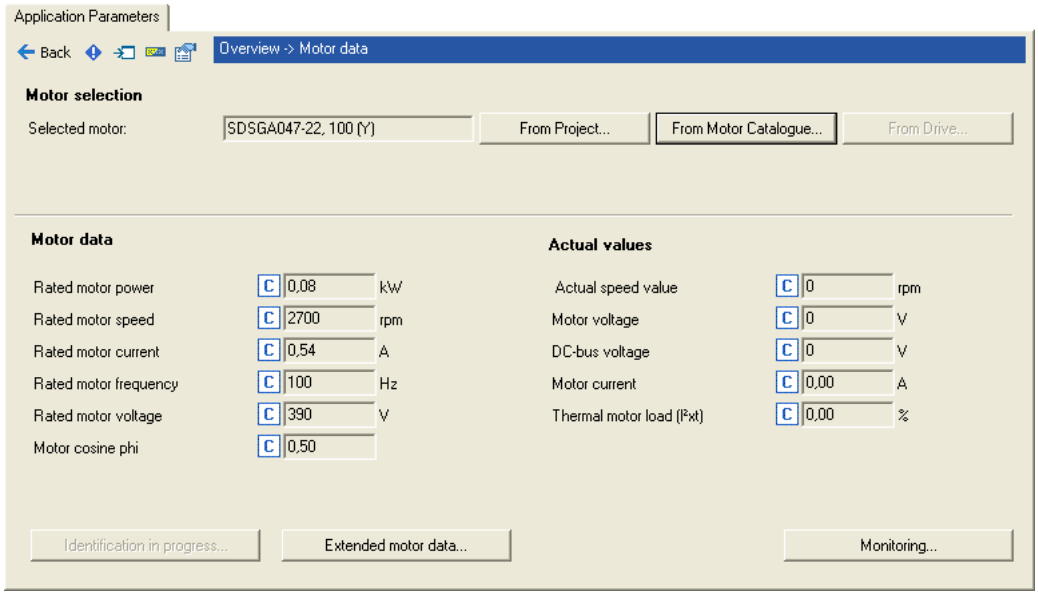
The motor data term comprises all parameters that only depend on the motor and that only characterise the electrical behaviour of the machine. The motor data are independent of the application in which the inverter and the motor are used.

**Proceed as follows to open the dialog for parameterising the motor data:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Go to the *Overview* dialog level and click the following button:



**Parameterisation dialog in the »Engineer«**




- Via the **From Motor Catalogue** button, the motor catalogue can be opened to select another motor. ▸ Selecting a motor from the motor catalogue in the »Engineer« (📖 143)
- Via the **From inverter**... button, the motor data set in the inverter can be copied to the »Engineer« when an online connection has been established.
- When an online connection has been established to the inverter, the **Identification in progress...** button serves to automatically identify different motor data. ▸ Automatic motor data identification (📖 145)
- The **Encoder/feedback system...** button serves to get to the settings for the encoder/feedback system, if available. ▸ Encoder/feedback system (📖 321)

### Stop!

Motors with electronic nameplates (ETS) must not be operated on 8400 inverters!

### Note!

Sensorless vector control (SLVC) and sensorless control for synchronous motors (SLPSM) in particular requires the motor data parameters to be set. The motor data comprise the data of the motor nameplate and the data of the motor equivalent circuit.

If the motor has been selected via the motor catalogue of the »Engineer« or the motor data have been adapted offline using the »Engineer«, all motor data must then be copied to the inverter and saved power-failure-proof to the memory module (device command: C00002/11) when an online connection has been established.

### Note!

**Setting a rated motor frequency with a decimal position**

If the motor has a rated motor frequency with a decimal position (e.g. motor nameplate data "23.7 Hz"):

- Up to and including version 02.xx.xx, the following motor nameplate data must be increased by a factor of 10:
  - C00089: Rated motor frequency
    (a value of "23.7 Hz", for instance, must be increased to 237 Hz.)
  - C00081: Rated motor power
  - C00087: Rated motor speed
  - C00090: Rated motor voltage
- From version 12.00.00, the rated motor frequency must be entered without decimal position to clearly identify the right motor type. (a value of "23.7 Hz", for instance, must be set to 23 Hz in C00089)
  - C01000 displays the set motor type (ASM or PSM).
  - C00969 displays the set number of motor pole pairs.
- From version 14.00.00 onwards, the motor type (ASM or PSM) can also be set manually in C01001/1 for a clear identification.

### Note!

If a rated motor speed is set for an asynchronous motor (ASM) in C00087 which corresponds to a very high slip speed, the number of pole pairs is identified incorrectly. Moreover, a synchronous motor (PSM) is detected instead of an asynchronous motor (ASM) at certain slip speeds. Detailed information and possible remedies can be found here:

▸ Application notes for asynchronous motors with high slip speed (📖 152)

### Motor data

In the parameterisation dialog, the data of the motor nameplate for the selected motor are displayed under "Motor data".

| Parameters | Info |
|---|---|
| C00081 | Rated motor power |
| C00087 | Rated motor speed |
| C00088 | Rated motor current |
| C00089 | Rated motor frequency |
| C00090 | Rated motor voltage |
| C00091 | Motor cos φ |

### Actual values

When an online connection to the inverter has been established, the following actual values are displayed in the parameterisation dialog under "Actual values":

| Parameters | Info |
|---|---|
| C00051 | Actual speed value |
| C00052 | Motor voltage |
| C00053 | DC-bus voltage |
| C00054 | Motor current |
| C00066 | Thermal motor load (I2xt) |
| Greyed out = display parameter | |

### Adapting motor data manually

If a third party manufacturer's motor is used, the displayed motor data can exactly be adapted to the real motor by clicking the **From project**... button and selecting the "Own motor settings" entry from the **Motor selection** dialog box afterwards. For this purpose, the data of the motor nameplate and the equivalent circuit diagram must be available.

**Tip!**

For a better concentricity factor, we recommend to perform motor parameter identification of the third party manufacturer's motor first. The motor parameters can be manually adapted afterwards.

Improving the concentricity factor includes

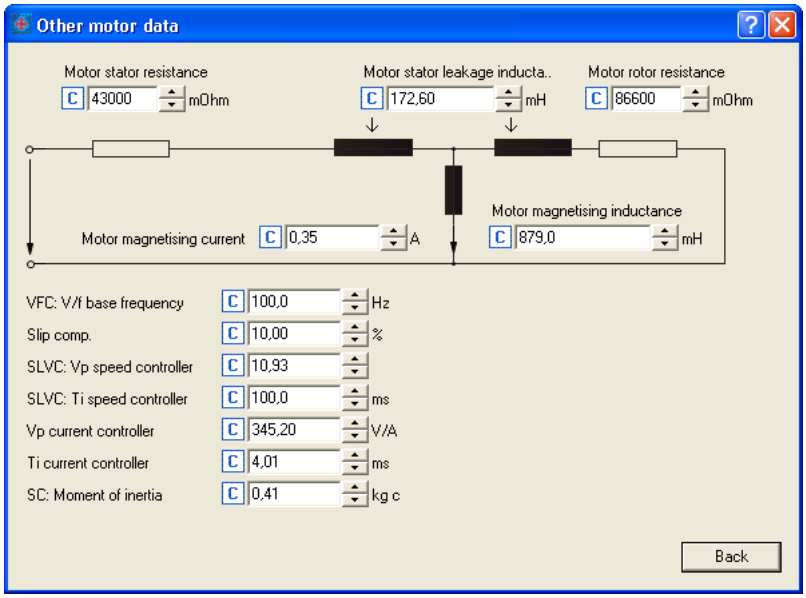
- the adjustment of the inverter error characteristic to the drive system and
- the knowledge of the motor cable resistance.

Both factors are determined in the course of motor parameter identification.

▸ Automatic motor data identification (📖 145)

### Other motor data

Click the **Other motor data...** button and go to the *Other motor data* dialog box including the motor equivalent circuit (in the following for an asynchronous motor):

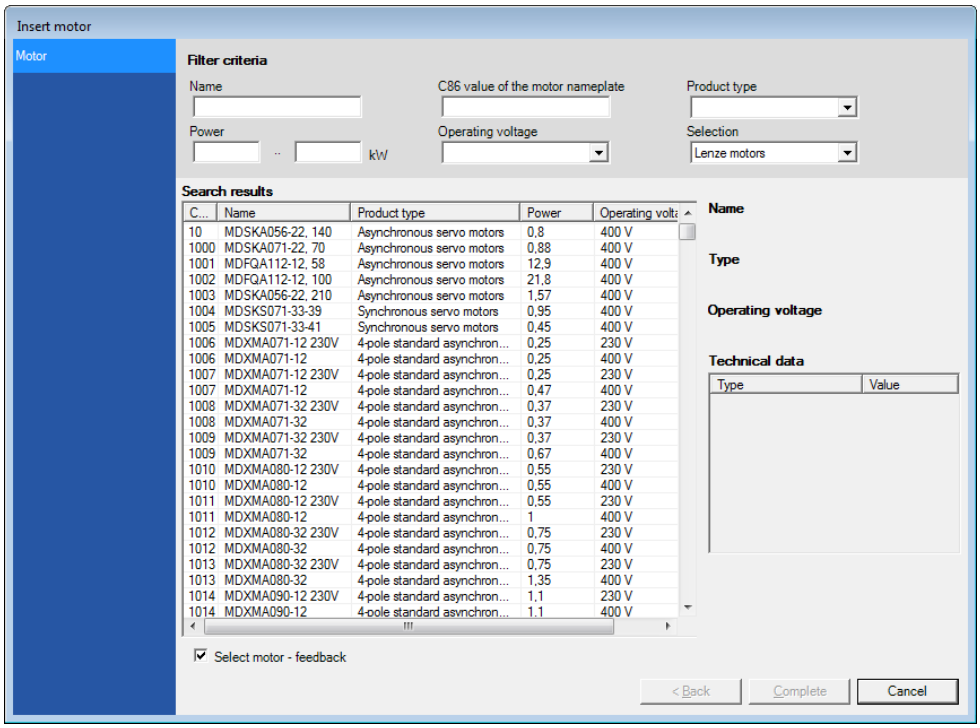



| Parameters | Info | ASM | PSM |
|---|---|---|---|
| C00084 | Motor stator resistance | ● | ● |
| C00085 | Motor stator leakage inductance | ● | ● |
| C00082 | Motor rotor resistance | ● | |
| C00095 | Motor magnetising current | ● | |
| C00092 | Motor magnetising inductance | ● | |
| C00015 | VFC: V/f base frequency | ● | ● |
| C00021 | Slip compensation | ● | |
| C00070/1 | SLVC: Vp speed controller | ● | |
| C00071/1 | SLVC: Ti speed controller | ● | |
| C00075 | Vp current controller | ● | ● |
| C00076 | Ti current controller | ● | ● |
| C00273 | Moment of inertia | ● | ● |
| C00016 | VFC: Vmin boost | ● | ● |
| C00070/2 | SC: Vp speed controller | ● | ● |
| C00071/2 | SC: Ti speed controller | ● | ● |
| C00070/3 | SLPSM: Vp speed controller | ● | ● |
| C00071/4 | SLPSM: Ti speed controller | ● | ● |
| C00072 | SC: Tdn speed controller | ● | ● |
| C00011 | Appl.: Reference speed | ● | ● |
| C00022 | Imax in motor mode | ● | ● |
| C00966 | VFC: Time const. slip comp. | ● | |
| C00982 | VFC-ECO: Voltage reduction ramp | ● | |
| C00073/1 | VFC: Vp Imax controller | ● | ● |
| C00077 | SC: Vp field controller | ● | |
| C00078 | SC: Tn field controller | ● | |

- From version 12.00.00 onwards, C01000 displays the set motor type (ASM or PSM).
- Generally, a synchronous motor without speed feedback can also be operated with the V/f characteristic control (VFCplus) control mode. The parameters for this control mode (e.g. V/f base frequency) thus also have an according influence on synchronous motors.

### 5.1.1 Selecting a motor from the motor catalogue in the »Engineer«

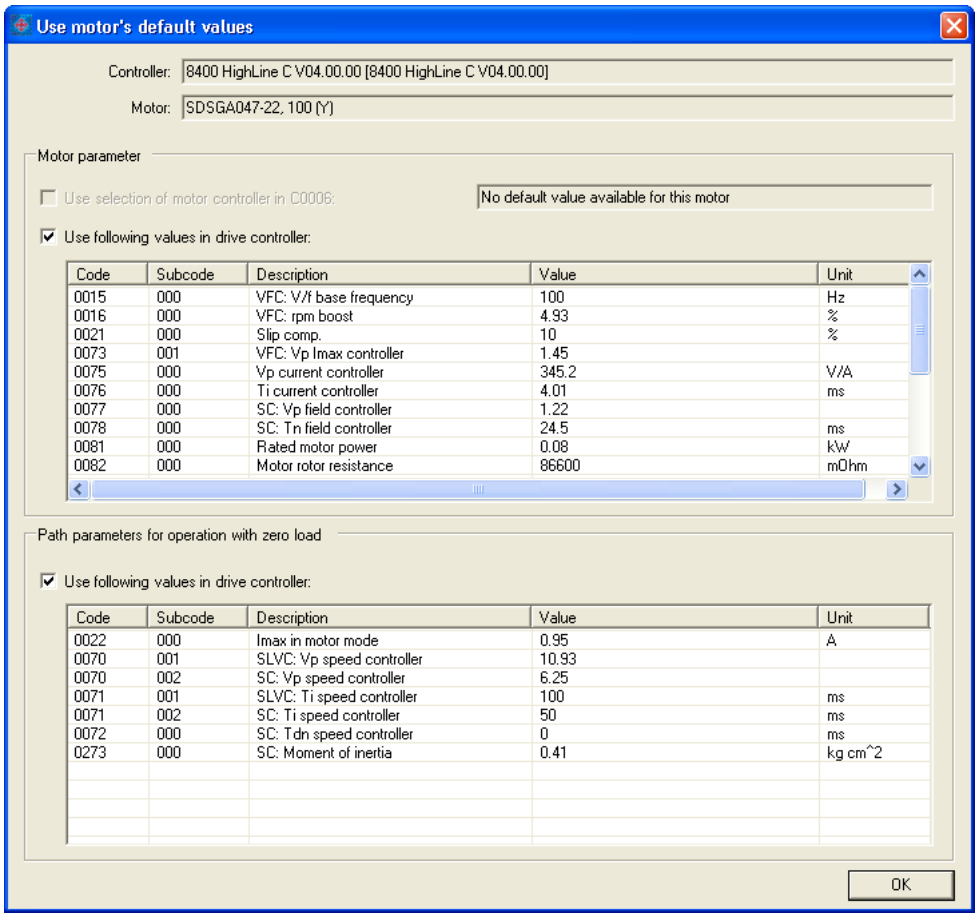
If you, when inserting the inverter into the project in the dialog step "Other components", put a checkmark in the control field **Motor**, you can select as a further dialog step the motor for the inverter from the motor catalogue:




- Alternatively, the motor can be inserted into the project at a later time via the **Insert a component** command.
- Go to the **Application parameters** tab in the *Overview* → *Motor data* dialog level and click the **From motor catalogue...** button to also reach the motor catalogue for the selection of another motor.

### Accepting the default values of the motor

If a motor is selected from the motor catalogue at a later time, the *Use motor's default values* dialog box is displayed afterwards which includes all motor data of the selected motor. Please select here which of the default values are to be copied to the inverter:




- The listed motor parameters are already optimally preset for the selected Lenze motor. An adaptation is not required.
- The "plant parameter" term comprises all parameters that result from the combination of motor and load. These characterise the transfer behaviour of the entire controlled system.
  - The plant parameters depend on the application in which the inverter and motor are used.
  - When a Lenze motor is selected in the »Engineer«, plant parameters are suggested for this motor for a load-free operation.

**Tip!**

If a third party manufacturer's motor is used, select a Lenze motor from the motor catalogue first which is similar in terms of current, voltage and speed rating. Adapt the preselected motor data exactly to the real motor afterwards.

### 5.1.2 Automatic motor data identification

The motor parameter identification serves to automatically identify the inverter characteristic, the influences of the motor cable and various motor parameters.

**Note!**

We strongly recommend motor parameter identification before the initial commissioning of the sensorless vector control (SLVC) and the sensorless control for synchronous motors (SLPSM).

#### Automatically identified motor parameters

| Parameters | Info | ASM | PSM |
|---|---|---|---|
| C00015 | V/f base frequency | ● | ● |
| C00016 | $V_{min}$ boost | ● | ● |
| C00021 | Slip compensation | ● | |
| C00082 | Motor rotor resistance | ● | |
| C00083 | Motor rotor time constant | ● | |
| C00084 | Motor stator resistance | ● | ● |
| C00085 | Motor stator leakage inductance | ● | ● |
| C00092 | Motor magnetising inductance | ● | |
| C00095 | Motor magnetising current | ● | |

#### Automatic calculation of the current controller and field controller parameters

From version 12.00.00 onwards: Following successful motor parameter identification, the current controller and field controller parameters listed in the below table are calculated as well.

- If these parameters are not to be calculated, bit 4 of C02865/1 must be set to "1".

| Parameters | Info | ASM | PSM |
|---|---|---|---|
| C00073/1 | VFC: Vp Imax controller | ● | ● |
| C00075 | Vp current controller | ● | ● |
| C00076 | Ti current controller | ● | ● |

**OPTIONAL: Automatic calculation of the speed controller parameters**

From version 12.00.00 onwards: Following successful motor parameter identification, the speed controller parameters listed in the below table can also be calculated automatically.

- If these parameters are to be calculated, bit 6 of C02865/1 must be set to "5".

| Parameters | Info | ASM | PSM |
|---|---|---|---|
| C00070/1 | SLVC: Vp speed controller | ● | |
| C00071/1 | SLVC: Ti speed controller | ● | |
| C00070/3 | SLPSM: Vp speed controller | ● | ● |
| C00071/3 | SLPSM: Ti speed controller | ● | ● |

- Care must be taken that the constant mass inertia of the drive (mass inertia of motor, gearbox, shaft and constant load) is entered as exactly as possible in C00273 to calculate the speed controller parameters as dynamically as possible.
  - Mass inertias that are not constant (e.g. changing loads of reels or different loads of hoists) must be entered in C00919/1.
  - If the mass inertia set in C00273 is too low, the speed controller parameters are calculated less dynamically.
  - If the mass inertia set in C00273 is too high, speed controller operation is unstable.
- If the mass inertia in C00273 is set to "0", the setting of bit 5 in C02865 has no effect on the speed controller parameter calculation. In this case, the speed controller parameters will not be calculated automatically after motor parameter identification.

**OPTIONAL: Automatic calculation of other controller parameters**

From version 12.00.00 onwards: Following successful motor parameter identification, the controller parameters listed in the below table can also be calculated automatically.

- If these parameters are to be calculated, bit 6 of C02865/1 must be set to "6".

| Parameters | Info | ASM | PSM |
|---|---|---|---|
| C00011 | Appl.: Reference speed | ● | ● |
| C00022 | Imax in motor mode | ● | ● |
| C00497 | Nact filter time constant | ● | ● |
| C00966 | VFC: Time const. slip comp. | ● | |
| C00982 | VFC-ECO: Voltage reduction ramp | ● | |

### Automatic calculation of the field weakening controller parameters for ASM servo control

From version 15.00.00, the field weakening controller parameters are automatically calculated as well after a successful motor parameter identification. Moreover, the special functions of the internal motor control listed in the table are activated.

- If these parameters are not to be calculated/activated, bit 6 has to be set to "1" in C02864/1.

| Parameters | Info | ASM | PSM |
|---|---|---|---|
| C00577 | SC: Vp field weakening controller | ● | |
| C00578 | SC: Tn field weakening controller | ● | |
| Activation of special functions of the internal motor control | | | |
| C02864/1 | Bit 5 is set to "1"<br>→ Field/field weakening controllers are called in a 500-µs cycle.* | ● | |
| | Bit 7 is set to "1"<br>→ In case of servo control for asynchronous motors, the rated motor speed C00087 is assumed for the calculation of the slip from the nameplate (C02879/Bit 1 = "0" for a warm machine.*<br>▸ Slip calculation from motor equivalent circuit diagram data (📖 253) | ● | |
| * If bit 6 is set to "1" in C02864/1, bit 5 and bit 7 are reset to "0". | | | |

### Principal sequence of the motor parameter identification

1. The motor stator resistance (C00084) is measured.
2. The inverter error characteristic is measured.
3. The motor stator leakage inductance (C00085) is measured.
4. The V/f base frequency (C00015) is calculated.
5. The slip compensation (C00021) is calculated.
6. The $V_{min}$ boost (C00016) is detected.
7. Only for asynchronous motors:
   The motor magnetising inductance (C00092) and the motor rotor resistance (C00082) are measured.
   The motor magnetising current (C00095) is measured.

   From version 12.00.00:
8. If C02865/1 - bit 4 = "0" (Lenze setting):
   The current controller and field controller parameters are calculated. (See table.)
9. If C02865/1 - bit 5 = "1":
   The speed controller parameters are calculated. (See table.)
10. If C02865/1 - bit 6 = "1":
    Other controller parameters are calculated. (See table.)

    From version 15.00.00:
11. If C02864/1 - bit 6 = "0" (Lenze setting):
    The field weakening controller parameters are calculated and certain special functions of the internal motor control are activated. (See table.)

**Selection of the identification mode**

From version 10.00.00 onwards, two identification modes are available in C02867/1:

- "1: Basic identification" (previous mode)
  - Only for asynchronous motors
  - Duration approx. 30 s
- "2: extended identification"
  - Stands out due to increased accuracy of the determined motor parameters.
  - Also supports synchronous motors and asynchronous motors with a power of more than 11 kW.
  - Duration approx. 80 s

**Tip!**

In the Lenze setting, a setting of "0: automatic" is selected in C02867/1. This setting ensures that the inverter automatically selects the optimum procedure for motor parameter identification.

**Preconditions for executing the motor parameter identification**

- The motor parameter identification must be carried out when the motor is cold!
- The load machine may remain connected. Holding brakes, if present, may remain in the braking position.
- In case of a synchronous motor:
  - The shaft must be able to rotate freely (must not be locked).
  - An identification is only possible with the extended motor parameter identification.
- For an asynchronous motor from a power of 11 kW:
  An identification is only possible with the extended motor parameter identification.
- The inverter has to be ready for enable (has to be in the "SwitchedOn" state).
  Inhibit the inverter if it is enabled, e.g. via the C00002/16 device command or a LOW signal at the X4/RFR terminal.
- The motor has to be at standstill.

**Required parameter settings before the motor parameter identification is executed**

- For an extended motor parameter identification, a motor control suitable for the motor has to be set in C00006. When the identification starts, this setting determines whether an asynchronous motor or a synchronous motor has to be identified. ▸ Selecting the control mode (📖 153)
- The motor parameters listed in the table below are excluded from automatic identification and must therefore be adapted to the used motor before motor parameter identification is carried out (see motor nameplate).

| Parameters | Info |
|---|---|
| C00081 | Rated motor power |
| C00087 | Rated motor speed |
| C00088 | Rated motor current (according to the connection method ⅄/△) |
| | **Note**The amplitude of the rated motor current (C00088) is injected to identify the stator resistance. If the rated motor current amounts to less than 60 % of the rated inverter current, at least 60 % of the rated inverter current will be injected to ensure sufficient motor parameter identification accuracy. |
| C00089 | Rated motor frequency (according to the connection method ⅄/△) |
| C00090 | Rated motor voltage (according to the connection method ⅄/△) |
| C00091 | Motor cos φ |

- If the motor nameplate data are entered in the »Engineer« motor catalogue instead of selecting a motor, it is recommendable to use the extended motor parameter identification (C02867/1 = "2: extended identification").
- Especially for third-party motors, it is also recommendable to carry out the slip calculation with the motor equivalent circuit diagram data (C02879/1). If the slip calculation is derived from the motor nameplate data, this can negatively affect the stability and accuracy of the drive behaviour because the rated speed specified on the motor nameplate often is rather inaccurate.
- The available motor cable must be specified in terms of length and cross-section:

| Parameters | Info |
|---|---|
| C00915 | Motor cable length |
| C00916 | Motor cable cross-section |

The motor cable resistance resulting from these settings is displayed in C00917.

- For the measurement of the required variables, the motor is energised via the inverter terminals U, V and W during the motor parameter identification. The corresponding current controller is preset in the Lenze setting so that a optimal controller behaviour is achieved with an asynchronous motor power-adapted to the inverter.
  Thanks to optimisation, the current controller can be set via the following parameters:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00075 | Vp current controller | 7.00 | V/A |
| C00076 | Ti current controller | 10.61 | ms |

- Switching frequency for the motor parameter identification:
  - Up to and including version 13.xx.xx, the motor parameter identification is executed with a switching frequency of 4 kHz.
  - From version 14.00.00, the motor parameter identification can also be executed with a switching frequency of 8 kHz instead of 4 kHz. For this purpose, the option "Motor ident.: Switching frequency 8 kHz" (Bit 4 = "1") has to be set in C02864/1.
    Example of how to use this option: Between the output of the inverter and the motor, a sinusoidal filter is connected which may only be operated with a minimum switching frequency of 8 kHz. (See also the section "Preventing a decrease of the switching frequency".) (📖 264)

  Generally, a switching frequency of 4 kHz is recommended for the motor parameter identification as it serves to obtain the most accurate results.

**Premature abort of the motor parameter identification**




**Stop!**

If motor parameter identification is aborted, unstable drive behaviour may be the result!

The motor parameter identification can be aborted in the following cases:

- If a special motor (e.g. mid-frequency motor) or a servo motor is used.
- If there is a large deviation between inverter and motor power.

In case of a simple motor parameter identification, we recommend the following:

- to reduce the P component Vp of the current controller (C00075) e.g. by halving.
- to increase the time constant Ti of the current controller (C00076) e.g. by doubling.

In case of the extended motor parameter identification, the current controller parameters are determined automatically. If the identification is aborted all the same, the current controller parameters set in C00075 and C00076 can be used by parameterising C02866 to "1".

Another cause for the abort of the motor parameter identification could be the implausibility of the entered nameplate data, e.g. the entry P = 0 kW for the motor power.

**Execute automatic motor parameter identification**



**Danger!**

During motor parameter identification, the motor is energised via the outputs U, V and W of the inverter!

- Observe the corresponding safety instructions!
- With an idling motor, a small angular offset may occur at the motor shaft.
- During the motor parameter identification, rotations occur in case of a synchronous motor.



**How to carry out automatic motor parameter identification:**

1. Activate motor parameter identification via the C00002/23 = "1: On / start" device command.
2. Enable inverter.
   - The inverter changes to the "Ident" device status.
   - Motor parameter identification starts.
   - The progress of the identification run can be seen in C00002/23.
   - The identification is completed if the "0: Off / ready" message is displayed in C00002/23.
   - After successful identification, it changes back to the "SwitchedOn" device status.
3. Inhibit inverter again.

### 5.1.3 Application notes for asynchronous motors with high slip speed

**Note!**

If a rated motor speed is set for an asynchronous motor (ASM) in C00087 which corresponds to a very high slip speed, the number of motor pole pairs is identified incorrectly.

Moreover, a synchronous motor is (PSM) is detected instead of an asynchronous motor (ASM). In this case, the "Id7" error message can occur since motor and motor control type do not match.

Examples, possible impacts and remedies are described in the following sections.

**Examples:**

| Number of pole pairs | Mechanical synchronous speed | Rated motor frequency (C00089) | Incorrect number of pole pairs if the rated motor speed (C00087) is lower: |
|---|---|---|---|
| 1 | 3000 rpm | 50 Hz | 1986 rpm |
| 2 | 1500 rpm | 50 Hz | 1195 rpm |
| 3 | 1000 rpm | 50 Hz | 854 rpm |
| 4 | 750 rpm | 50 Hz | 665 rpm |

**Possible impacts depending on the motor control type:**

| Motor control type | Possible impacts on the drive |
|---|---|
| Servo control (SC) - PSM | No impact if the motor data have been entered correctly. |
| Servo control (SC) - ASM | The motor can accelerate up to maximum speed of the motor. |
| V/f control (VFCplus + encoder) | |
| Sensorless vector control (SLVC) | The output speed is considerably too high.<br>Example: p=2, $f_n$=50 Hz, $n_n$=1195 $min^{-1}$ → Speed is 50 % too high |
| V/f characteristic control (VFCplus) | |

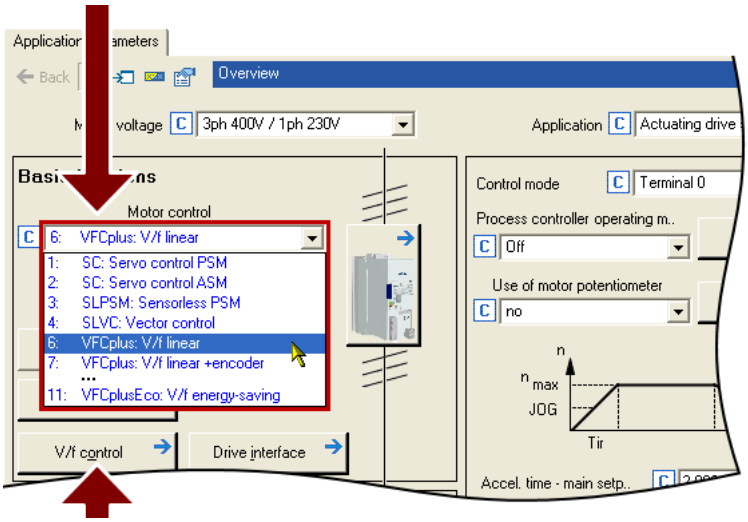
**Remedies:**

- For version 12.xx.xx and version 13.xx.xx the following applies:
  Limit the rated motor speed: C00087 = C00089 * 60 / (number of pole pairs + 0.5)
- From version 14.00.00:
  Set the motor type ("ASM" or "PSM") manually in C01001/1. The rated motor speed does not need to be limited since the number of pole pairs can be detected safely when the motor type has been set manually.

## 5.2 Selecting the control mode

The 8400 TopLine inverter supports various modes for motor control (open loop or closed loop).

- V/f characteristic control (VFCplus) with linear characteristic for asynchronous motors is preset.
- The control mode can be selected in the »Engineer« on the **Application parameter** tab via the **Motor control** (C00006) list field:




- A click on the **Motor control...** button leads you to the parameterisation dialog of the selected motor control. (The button is labelled according to the selected motor control.)

**Tip!**

In order to make the selection of the motor control easier, we provide a selection help with recommendations and alternatives for standard applications in the subchapter entitled "Selection help". (📖 157)

The following section briefly describe the control modes. A reference to more details can be found at the end of each section.

### V/f characteristic control (VFCplus)

The V/f characteristic control (VFCplus) is a motor control mode for standard frequency inverter applications based on a simple and robust control process which is suitable for the operation of asynchronous motors with linear or square-law load torque characteristic (e.g. fans). Furthermore, this motor control mode is also suitable for group drives and special motors. Due to the low parameterisation effort, commissioning of such applications is fast and easy.

The $V_{min}$ boost (C00016) and slip compensation (C00021) required for optimising the drive behaviour are dimensioned for asynchronous motors with power adaptations to the inverter in the Lenze setting.

▸ V/f characteristic control (VFCplus) (📖 161)

### Energy-saving V/f characteristic control (VFCplusEco)

In contrast to the V/f characteristic control mode (VFCplus), this motor control mode uses a cosφ control in partial load operational range to automatically reduce the power loss in the asynchronous motor (energy optimisation).

The motor data required for the cosφ control and the $V_{min}$ boost (C00016) and slip compensation (C00021) required for optimising the drive behaviour are dimensioned for asynchronous motors with power adaptations to the inverter in the Lenze setting.

The required motor data (motor rotor resistance, motor stator resistance, motor stator leakage inductance and mutual motor inductance) only affect the extent of energy optimisation but not the stability.

In case of applications with dynamically very high sudden load variations from the unloaded operation, this motor control mode should not be used since a motor stalling cannot be excluded.

Energy optimisation for dynamic applications is not possible with this motor control mode.

▸ V/f characteristic control - energy-saving (VFCplusEco) (📖 180)

### V/f control (VFCplus + encoder)

The V/f control can be selected for operating asynchronous motors with speed feedback. With this motor control, a slip regulator can be additionally parameterised which adjusts the actual speed value dynamically to the speed setpoint.

▸ V/f control (VFCplus + encoder) (📖 190)

### Sensorless vector control (SLVC)

Sensorless (field-oriented) vector control for asynchronous motors is based on a decoupled, separate control for the torque-producing and the field-producing current component. In addition, the actual speed is reconstructed by means of a motor model so that a speed sensor is not required.

In comparison to the V/f characteristic control without feedback, the following can be achieved by means of sensorless vector control SLVC:

- A higher maximum torque throughout the entire speed range
- A higher speed accuracy
- A higher concentricity factor
- A higher level of efficiency
- The implementation of torque-actuated operation with speed limitation
- The limitation of the maximum torque in motor and generator mode for speed-actuated operation



**Tip!**

If a high torque without feedback is to be provided at small speeds, we recommend the "Sensorless vector control" motor control mode.

▸ Sensorless vector control (SLVC) (📖 197)

### Sensorless control for synchronous motors (SLPSM)

This sensorless control enables an encoderless control of synchronous motors. The process is based on field-oriented control within a higher speed range (e.g. > 10 % of the rated motor speed). The actual speed value and rotor position are reconstructed via a motor model.

Standard applications for this control type are pumps and fans, horizontal materials handling and simple positioning technology.

▸ Sensorless control for synchronous motors (SLPSM) (📖 213)

### Servo control (SC)

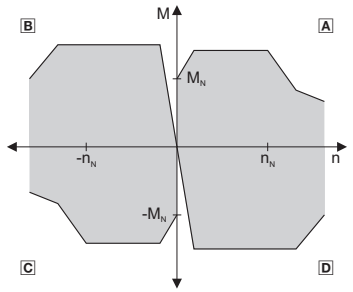
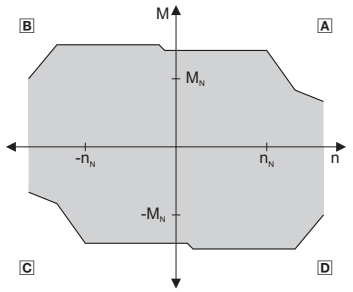
Field-oriented servo control (SC) is based on a decoupled, separate control of the torque-producing and the field-producing current component. The motor control is based on a field-oriented, cascaded controller structure with feedback function and enables dynamic and stable operation in all of the four quadrants.

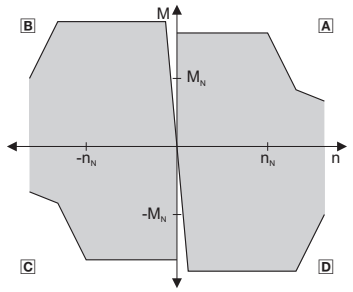
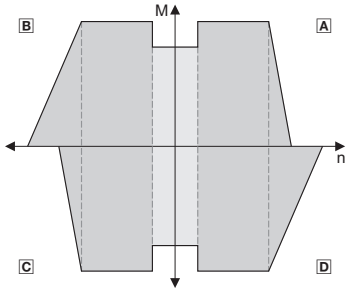
The servo control can be used for synchronous motor (PSM) and asynchronous motors (ASM) and basically offers the same advantages as the sensorless vector control (SLVC).

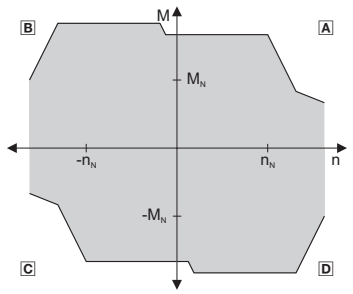
▸ Servo control (SC) (📖 231)

### Speed feedback

As shown in the following graphics, the drive systems with feedback have, independently of the motor control, more advantages than systems without feedback.

| V/f characteristic control (VFCplus) | V/f control (VFCplus + encoder) |
|---|---|
|  |  |
| A Operation in motor mode (CW rotation), B Operation in generator mode (CCW rotation), C Operation in motor mode (CCW rotation), D Operation in generator mode (CW rotation) | |

| Sensorless vector control (SLVC) | Sensorless control for synchronous motors (SLPSM) |
|---|---|
|  |  |
| A Operation in motor mode (CW rotation), B Operation in generator mode (CCW rotation), C Operation in motor mode (CCW rotation), D Operation in generator mode (CW rotation) | |

| Servo control (SC) | |
|---|---|
|  | |
| A Operation in motor mode (CW rotation), B Operation in generator mode (CCW rotation), C Operation in motor mode (CCW rotation), D Operation in generator mode (CW rotation) | |

### 5.2.1 Selection help

To ease the selection of the motor control mode, the two following tables contain recommendations and alternatives to standard applications.

| Application | recommended | Alternatively |
|---|---|---|
| **Single drives** | | |
| With constant load | VFCplus: V/f linear | SLVC or SLPSM |
| With extremely alternating loads | VFCplus: V/f linear | SLVC |
| With high starting duty | SLVC | VFCplus: V/f linear |
| Torque limitation | SLVC | SLPSM |
| With torque limitation (power control) | VFCplus: V/f linear | SLPSM |
| Three-phase reluctance motor | VFCplus: V/f linear | - |
| Three-phase sliding rotor motor | VFCplus: V/f linear | - |
| Three-phase AC motors with permanently assigned frequency/voltage characteristic | VFCplus: V/f linear | - |
| Pump and fan drives with quadratic load characteristic | VFCplusEco | SLVC or SLPSM |
| Simple hoists | VFCplus: V/f linear | - |
| **Group drives** *(several motors connected to inverter)* | | |
| Identical motors and loads | VFCplus: V/f linear | - |
| Different motors and/or alternating loads | VFCplus: V/f linear | - |

[5-1] Standard applications without speed feedback

| Application | recommended | Alternatively |
|---|---|---|
| **Single drives** | | |
| With constant load | SC | VFCplus / SLVC |
| With extremely alternating loads | SC | VFCplus / SLVC |
| With high starting duty | SC | VFCplus / SLVC |
| With speed control (speed feedback) | SC | VFCplus |
| With high dynamic performance e.g. for positioning and infeed drives | SC | - |
| Torque limitation | SC | SLVC |
| With torque limitation (power control) | - | - |
| Winder with dancer position control | SC | VFCplus |
| Unwinder with dancer position control | SC | VFCplus |
| Three-phase reluctance motor | - | - |
| Three-phase sliding rotor motor | - | - |
| Three-phase AC motors with permanently assigned frequency/voltage characteristic | - | - |
| Pump and fan drives with quadratic load characteristic | - | - |
| Simple hoists | VFCplus: V/f linear | - |
| **Group drives** *(several motors connected to inverter)* | | |
| Identical motors and loads | VFCplus | - |
| Different motors and/or alternating loads | - | VFCplus |

[5-2] Standard applications with speed feedback

## 5.3 Defining current and speed limits

### Limitation of the speed setpoint

Parameterising the reference speed in C00011 means that the drive must rotate at the set speed if a speed setpoint of 100% is specified.

All speed setpoint selections are provided in % and always refer to the reference speed set in C00011.

**Tip!**

For reasons of achievable resolution and the accuracy involved, the reference speed should be geared to the speed range required for the respective application.

Lenze recommendation: Reference speed (C00011) = 1500 ... 3000 rpm

Irrespective of the selected motor control, there are more limitation options:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00909/1 | Max. positive speed | 120 | % |
| C00909/2 | Max. negative speed | 120 | % |
| C00910/1 | Max. positive output frequency | 599 | Hz |
| C00910/2 | Max. negative output frequency | 599 | Hz |

**Note!**

In the torque-controlled operation (*bTorquemodeOn* = TRUE), the limitation of the speed setpoint set in C00909/x has no impact!

For the torque-controlled operation, a permissible speed range can be defined via the speed limitation (inputs *nSpeedHighLimit_a* and *nSpeedLowLimit_a* at the SB LS_MotorInterface).

For a correct function of the speed limitation in both direction, the following applies:

The speed limitation operates with a internal hysteresis of 50 $min^{-1}$. The upper or lower speed limit resulting from the settings has to be higher than this hysteresis!

### Current limitation in motor and generator mode

In the various motor control modes, the inverter is provided with functions which determine the dynamic behaviour under load and counteract exceedance of the maximum current in motor or generator mode.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00022 | Imax in motor mode | 47.00 | A |
| C00023 | Imax in generator mode<br>• 100 % ≡ Imax in motor mode (C00022) | 100 | % |

The current limits must be selected depending on

- the permissible maximum current of the motor → recommendation: $I(Mot)_N$ < 1.5 ... 2.0
- the permissible maximum current of the inverter
- the torque in motor/generator mode required for the application

## Note!

**Highly dynamic applications**
(that have e.g. too short acceleration/deceleration times or excessively changing loads)

The overcurrent disconnection may respond (fault message oC1 or oC11) if the setting of the maximum current in motor mode in C00022 approximately corresponds to the maximum permissible value of the respective inverter.

Remedies:

- Increase of the acceleration and deceleration ramp times
- Reduction of the maximum current in motor mode (C00022)
- Reduction of the maximum current in generator mode (C00023)
- Adaptation of the indirect peak current limitation (procedure depends on the selected motor control mode, see below)
- Reduction of the reset time of the current limiting controller (C00074/1)

**Influencing the torque in motor/generator mode**

The torque in motor and generator mode can be limited via the *nTorqueMotLim* and *nTorqueGenLim* process signal inputs.

- If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.
- If sensorless vector control (SLVC), sensorless control for synchronous motors (SLPSM) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.

If keypad control is selected, the *nTorqueMotLim* and *nTorqueGenLim* process signals can be parameterised via C00728/1...2.

**How to adapt the peak current limitation:**

V/f characteristic control (VFCplus):

- Reduce the slip compensation with C00021.

V/f control (VFCplus + encoder):

- Reduce the slip limitation to twice the rated motor slip with C00971.
- Reduce the $V_{min}$ boost in C00016.

Sensorless vector control (SLVC):

- Reduce the slip compensation with C00021.
- Reduce the limitation of the torque in motor mode via *nTorqueMotLimit_a* (C00728/1) and the limitation of the torque in generator mode via *nTorqueGenLimit_a* (C00728/2).

Servo control (SC):

- Reduce the jerk limitation with C00274.
- Reduce the limitation of the torque in motor mode via *nTorqueMotLimit_a* (C00728/1) and the limitation of the torque in generator mode via *nTorqueGenLimit_a* (C00728/2).

## 5.4 V/f characteristic control (VFCplus)

In case of the V/f characteristic control (VFCplus), the motor voltage of the inverter is determined by means of a linear or quadratic characteristic depending on the field frequency or motor speed to be generated. The voltage follows a preselected characteristic.

### Stop!

- The V/f characteristic control is only suitable for asynchronous motors.
- The following must be observed when operating drives with quadratic V/f characteristic:
  - Please always check whether the corresponding drive is suitable for operation with a quadratic V/f characteristic!
  - If you pump or fan drive is not suitable for operation with a square-law V/f characteristic, we recommend using the energy-saving V/f characteristic control (VFCplusEco). Alternatively, you can use the V/f characteristic control with linear V/f characteristic or the sensorless vector control (SLVC) or servo control (SC).
- For adjustment, observe the thermal performance of the connected asynchronous motor at low output frequencies.
  - Usually, standard asynchronous motors with insulation class B can be operated for a short time with their rated current in the frequency range 0 Hz ... 25 Hz.
  - Contact the motor manufacturer to get the exact setting values for the max. permissible motor current of self-ventilated motors in the lower speed range.
  - If you select square-law V/f characteristics, we recommend setting a lower $V_{min}$ or using the energy-saving V/f characteristic control (VFCplusEco).
- The nameplate data of the motor (at least rated speed and rated frequency) must be entered if, instead of a standard motor, an asynchronous motor is used with the following values:
  - rated frequency ≠ 50 Hz (star) or
  - rated frequency ≠ 87 Hz (delta) or
  - number of pole pairs ≠ 2

### Note!

When the auto DCB threshold (C00019) is set > 0 rpm, there is no torque at the motor shaft in the lower speed range!

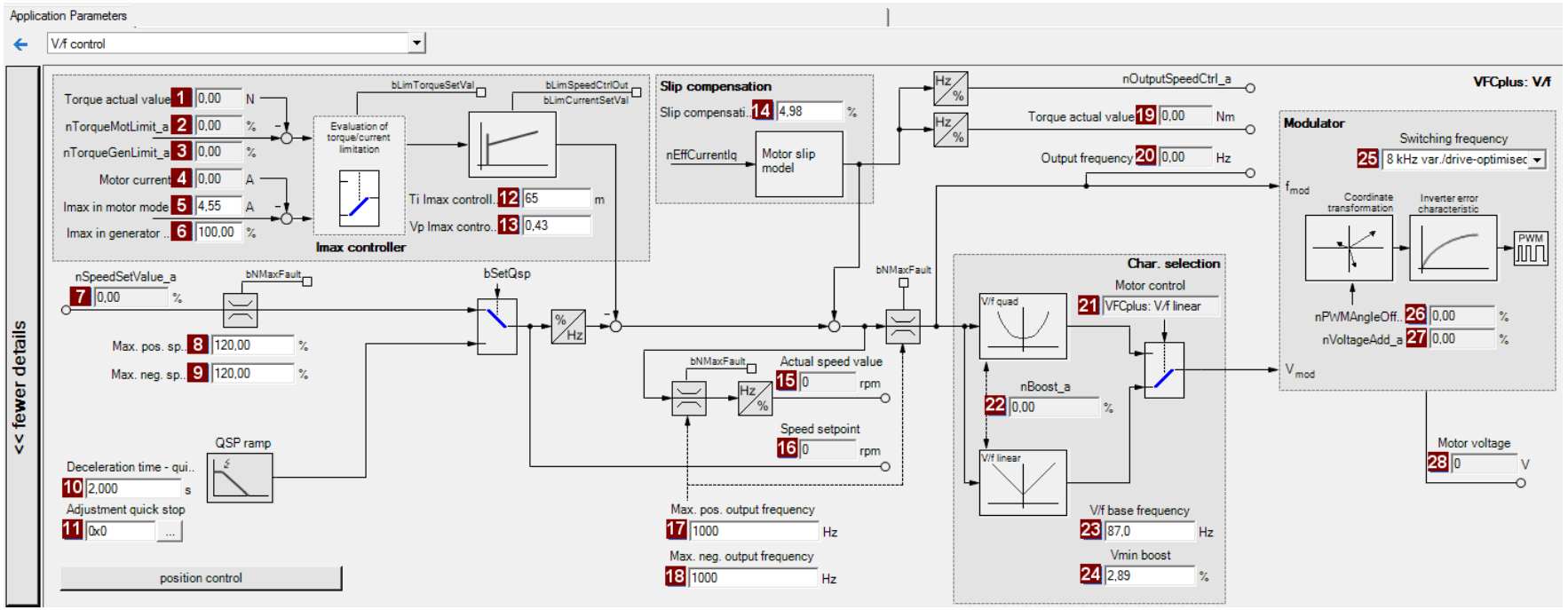
▸ Automatic DC-injection braking (Auto-DCB) (📖 275)

### 5.4.1 Parameterisation dialog/signal flow

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Select the motor control from the *Overview* dialog level in the **Motor control** list field:
   - "6: VFCplus: V/f linear" for linear characteristic or
   - "8: VFCplus: V/f quadr" for square-law characteristic

   More available V/f characteristic control modes:
   - "10: VFCplus: V/f definable".
     With this motor control, the V/f characteristic can be freely defined. ▸ Defining a user-defined V/f characteristic (📖 176)
   - "11: VFCplusEco: V/f energy-saving".
     With this motor control, the motor is always operated in an optimal efficiency range via a cosφ control and the resulting voltage reduction (reduced copper losses in the asynchronous motor). ▸ V/f characteristic control - energy-saving (VFCplusEco) (📖 180)
4. Click the **Motor control V/f** button to change to the *Overview* → *Motor control V/f* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed.




| Parameters | | Info |
|---|---|---|
| 1 | C00056/2 | Actual torque |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00054 | Motor current |
| 5 | C00022 | Imax in motor mode |
| 6 | C00023 | Imax in generator mode |
| 7 | C00830/22 | Speed setpoint |
| 8 | C00909/1 | Max. pos. speed |
| 9 | C00909/2 | Max. neg. speed |
| 10 | C00105 | Decel. time - quick stop |
| 11 | C00104/1 | Quick stop setting |

| Parameters | | Info |
|---|---|---|
| 12 | C00074/1 | Ti Imax controller |
| 13 | C00073/1 | Vp Imax controller |
| 14 | C00021 | Slip compensation |
| 15 | C00051 | Actual speed value |
| 16 | C00050 | Speed setpoint |
| 17 | C00910/1 | Max. pos. output frequency |
| 18 | C00910/2 | Max. neg. output frequency |

| Parameters | | Info |
|---|---|---|
| 19 | C00056/2 | Actual torque |
| 20 | C00058 | Output frequency |
| 21 | C00006 | Motor control |
| 22 | C00830/26 | MCTRL: nBoost_a |
| 23 | C00015 | V/f base frequency |
| 24 | C00016 | Vmin boost |
| 25 | C00018 | Switching frequency |
| 26 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 27 | C00830/31 | MCTRL: nVoltageAdd_a |
| 28 | C00052 | Motor voltage |

### 5.4.2 Basic settings

The "Initial commissioning steps" listed in the table below are sufficient for a simple characteristic control.

- Detailed information on the individual steps can be found in the following subchapters.

| Initial commissioning steps | |
|---|---|
| 1. | Defining the V/f characteristic shape. (📖 165) |
| 2nd | Defining current limits (Imax controller). (📖 166) |

**Tip!**

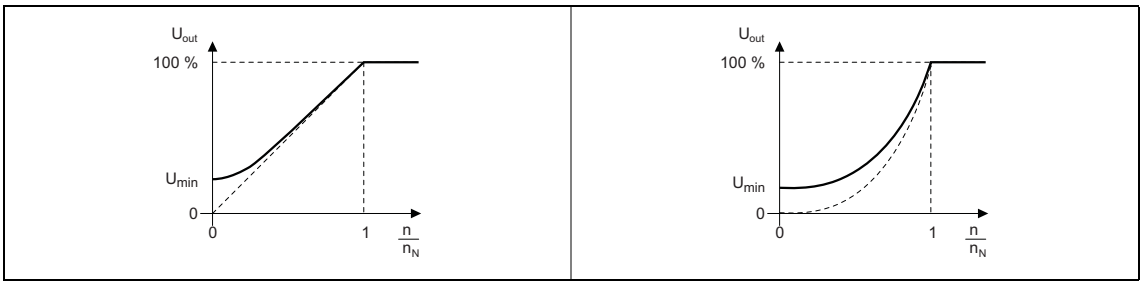
Information on the optimisation of the control mode and the adaptation to the real application is provided in chapter "Optimising the control mode". (📖 167)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 263)

### 5.4.2.1 Defining the V/f characteristic shape

In principle, four different characteristic shapes can be stipulated:

1. **Linear V/f characteristic**:
   For drives for a constant, speed-independent load torque.
2. **Quadratic V/f characteristic**:
   For drives with a load torque curve which is quadratic or in relation to speed. Quadratic V/f characteristics are preferred in the case of centrifugal pumps and fan drives.
3. **Freely definable V/f characteristic**:
   For drives that require adaptation of the magnetising current by means of the output speed. The freely definable V/f characteristic can be used e.g. for operation in conjunction with special machines such as reluctance motors in order to suppress oscillations at the machine or to optimise energy consumption.




[5-3] Principle of a linear V/f characteristic (on the left) and a quadratic V/f characteristic (on the right)

4. **Linear V/f characteristic with voltage reduction**:
   For drives which often work in partial load operation, the energy-saving V/f characteristic control (VFCplusEco) offers the opportunity to reduce the voltage at low load in order to save energy. At higher loads, the voltage reduction is cancelled and a linear characteristic is caused.

The V/f characteristic shape is defined by selecting the corresponding motor control mode in C00006:

| V/f characteristic shape | Motor control to be selected (C00006) |
|---|---|
| Linear V/f characteristic | 6: VFCplus: V/f linear |
| Square-law V/f characteristic | 8: VFCplus: V/f quadr |
| User-definable V/f characteristic | 10: VFCplus: V/f definable |
| Linear V/f characteristic with voltage reduction | 11: VFCplusEco: V/f energy-saving |

**Tip!**

- You can find detailed information on freely definable V/f characteristics in the subchapter entitled "Defining a user-defined V/f characteristic". (📖 176)
- You can find detailed information on the linear V/f characteristic with voltage reduction in the chapter entitled "V/f characteristic control - energy-saving (VFCplusEco)". (📖 180)

### 5.4.2.2 Defining current limits (Imax controller)

The V/f characteristic control (VFCplus) and the V/f control (VFCplus + encoder) operating modes are provided with a current limitation control which is decisive for the dynamic behaviour under load and counteracts exceedance of the maximum current in motor or generator mode. This current limitation control is called $I_{max}$ control.

- The efficiency (motor current) measured by the $I_{max}$ control is compared with the current limit value for motor load set in C00022 and the current limit value for generator load set in C00023.
- If the current limit values are exceeded, the inverter changes its dynamic behaviour.

**Motor overload during acceleration**

The inverter prolongs the acceleration ramp to keep the current on or below the current limit.

**Generator overload during deceleration**

The inverter prolongs the acceleration ramp to keep the current on or below the current limit.

**Increasing load with constant speed**

- If the motor current limit value is reached:
  - The inverter reduces the effective speed setpoint until a stable working point is set or an effective speed setpoint of 0 rpm is reached.
  - If the load is reduced, the inverter increases the effective speed setpoint until the setpoint speed is reached or the load reaches the current limit value again.
- When the generator current limit value is reached:
  - The inverter increases the effective speed setpoint until a stable working point is set or the maximally permissible speed (C00909) or output frequency is reached (C000910).
  - If the load is reduced, the inverter reduces the effective speed setpoint until the setpoint speed is reached or the load reaches the current limit value again.
- If a sudden load is built up at the motor shaft (e.g. drive is blocked), the overcurrent disconnection may respond (fault message oC1 or oC11).

### 5.4.3 Optimising the control mode

The V/f characteristic control (VFCplus) is generally ready for operation. It can be adapted subsequently by adapting the characteristic and/or the drive behaviour.

**Note!**

Following successful motor parameter identification, the V/f base frequency (C00015) and the $V_{min}$ boost (C00016) as well as the slip constant for slip compensation (C00021) are calculated automatically.

From version 12.00.00:

- Following successful motor parameter identification, the gain of the Imax controller (C00073/1) is calculated automatically.
  - If these parameters are not to be calculated, bit 4 of C02865/1 must be set to "1".
- Following successful motor parameter identification, other controller parameters (C00011, C00022, C00966) can be calculated automatically.
  - If these parameters are to be calculated, bit 6 of C02865/1 must be set to "6".

#### Adapting characteristic

For the linear and quadratic characteristic, it is also possible to match its curve to different load profiles or motors by adapting the V/f base frequency (C00015) and the $V_{min}$ boost (C00016).

▸ Adapting the V/f base frequency (📖 168)
▸ Adapting the Vmin boost (📖 170)

#### Freely defining the characteristic

The V/f characteristic can also be defined freely if the linear and quadratic characteristics are not suitable.

▸ Defining a user-defined V/f characteristic (📖 176)

#### Adapting drive behaviour

- Limitation of the maximum current by a current limitation controller (e.g. to prevent the motor from stalling or to limit to the maximally permissible motor current). ▸ Optimising the Imax controller (📖 172)
- Adaptation of the field frequency by a load-dependent slip compensation (improved speed accuracy for systems without feedback)
- Adaptation of the controller parameters of the slip regulator if V/f control (VFCplus + encoder) is selected. ▸ Parameterising the slip regulator (📖 194)

### 5.4.3.1 Adapting the V/f base frequency

The V/f base frequency (C00015) determines the slope of the V/f characteristic and has considerable influence on the current, torque, and power performance of the motor.

- The setting in C00015 applies to all permitted mains voltages.
- Mains fluctuations or fluctuations of the DC-bus voltage (operation in generator mode) do not need to be considered when the V/f base frequency is set. They are automatically compensated for by the internal mains voltage compensation of the device.
- Depending on the setting in C00015, it may be required to adapt the reference speed (C00011) to traverse the entire speed range of the motor.
- The V/f base frequency is automatically calculated from the stored motor nameplate data by the motor parameter identification:

$$C00015\ [Hz] = \frac{U_{FI}\ [V]}{U_{Ratedmot}\ [V]} \cdot f_{Rated}\ Hz$$

$V_{FI}$: Mains voltage 400 V or 230 V
$V_{ratedmot}$: Rated motor voltage depending on the connection method
$f_{rated}$: Rated motor frequency

[5-4] Calculation of the V/f base frequency

#### Typical values of the V/f base frequency

| Inverter with 400-V mains connection | | | |
|---|---|---|---|
| **Motor voltage [V]** | **Motor frequency [Hz]** | **Motor connection** | **V/f base frequency (C00015)** |
| 230 / 400 | 50 | ⅄ | 50 Hz |
| 220 / 380 | 50 | ⅄ | 52.6 Hz |
| 280 / 480 | 60 | ⅄ | 50 Hz |
| 400 / 690<br>400 | 50<br>50 | △ | 50 Hz |
| 230 / 400<br>280 / 480<br>400 | 50<br>60<br>87 | △ | 87 Hz |
| 220 / 380 | 50 | △ | 90.9 Hz |

| Inverter with 230-V mains connection | | | |
|---|---|---|---|
| **Motor voltage [V]** | **Motor frequency [Hz]** | **Motor connection** | **V/f base frequency (C00015)** |
| 230 | 50 | △ | 50 Hz |
| 220 / 380 | 50 | △ | 52.3 Hz |

## Note!

**87-Hz operation**

4-pole asynchronous motors which are designed for a rated frequency of f = 50 Hz in star connection can be operated in delta connection when being constantly excited up to f = 87 Hz.

- Advantages:
  - Higher speed-setting range
  - 73% higher power output in case of standard motors
- Motor current and motor power increase by the factor $\sqrt{3}$.
- The field weakening range starts above 87 Hz.
- Generally, this process can also be used with motors which have different numbers of pole pairs. In case of 2-pole asynchronous motors, the mechanical limit speed must be maintained.

### 5.4.3.2 Adapting the Vmin boost

The $V_{min}$ boost (C00016) of the motor voltage serves to select a load independent magnetising current which is required for asynchronous motors. The torque behaviour of the motor can be optimised by adapting the setting in C00016.

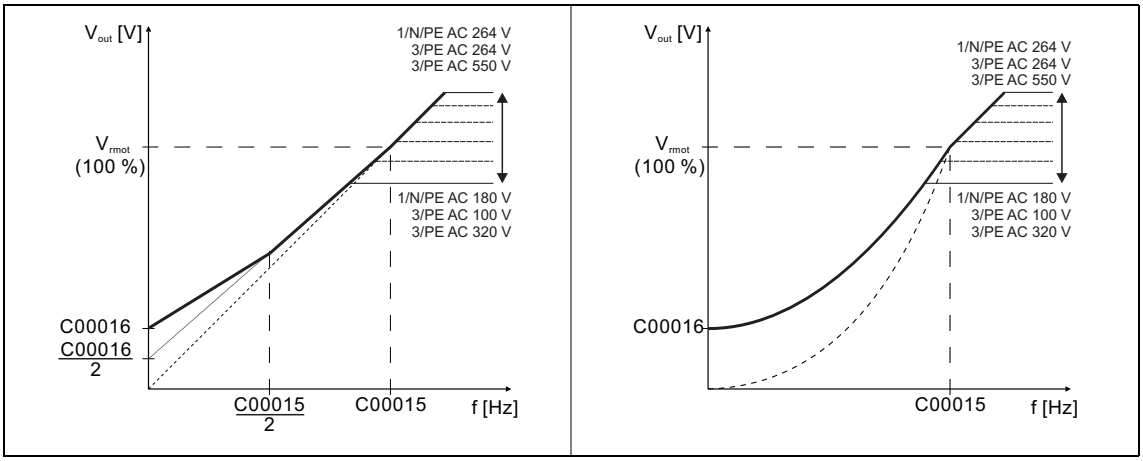
The *nBoost_a* process signal at the SB LS_MotorInterface serves to carry out a $V_{min}$ boost as well:

| Designator<br>DIS code \| data type | Information/possible settings |
|---|---|
| nBoost_a<br>C00830/26 \| INT | Process signal for the $V_{min}$ boost<br>• This signal is added to C00016 and has thus an increasing or decreasing effect.<br>• This signal serves to implement a load-dependent $V_{min}$ boost to improve the torque behaviour in different load states, as for instance in case of operation in generator or motor mode.<br>• Scaling: 16384 ≡ 100 % rated device voltage $V_{FU}$ (400 V or 230 V) |
| | **Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! |

**Note!**

The $V_{min}$ boost has an effect on output frequencies below the V/f base frequency (C00015).

The general linear and quadratic V/f characteristics are shown in the illustrations below. The illustrations show the impacts of the parameters used to adapt the characteristic shape.




[5-5] Representation of the linear V/f characteristic (on the left) and quadratic V/f characteristic (on the right)

**How to set the $V_{min}$boost:**

1. Operate motor in idle state at approx. 6 % of the rated motor speed.
2. Increase $V_{min}$ boost (C00016) until the following motor current is reached:

   Motor in short-time operation up to 0.5 $n_{rated}$
   - for self-ventilated motors: $I_{motor} \approx I_{rated\ motor}$
   - for forced ventilated motors: $I_{motor} \approx I_{rated\ motor}$

   Motor in continuous operation up to 0.5 $n_{rated}$
   - for self-ventilated motors: $I_{motor} \approx 0.8\ I_{rated\ motor}$
   - for forced ventilated motors: $I_{motor} \approx I_{rated\ motor}$

## Note!

$V_{min}$ boost is automatically calculated by the motor parameter identification using the data specified on the motor nameplate so that a no-load current of approx. 0.8 $I_{rated\ motor}$ results at the slip frequency of the machine.

**V/f control (VFCplus + encoder)**

In case of operation in motor mode, we recommend a considerably lower Vmin boost with V/f control (VFCplus + encoder) In this cas, the Vmin boost should be dimensioned in such a way that in case of slip frequency in idle state, approx. 50 % of the rated motor current flows.

In case of operation in generator mode, a light increase of the speed may be caused in certain load ranges with V/f control (VFCplus + encoder) in the lower speed range. This speed increase can be reduced by a higher $V_{min}$ boost.

The *nBoost_a* process signal is added to C00016 and can be used to implement different $V_{min}$ boosts for operation in motor and generator mode. As an alternative, the *nVoltageAdd_a* process signal can be used as well.

### 5.4.3.3 Optimising the Imax controller

Using the Lenze setting of the current limitation controller, the drive is stable:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00073/1 | VFC: Vp Imax controller | 0.25 | |
| C00074/1 | VFC: Ti Imax controller | 65 | ms |

Most applications do not require optimisation.

The setting of the current limitation controller must be adapted if

- power control including great moments of inertia is performed.
  - Recommendation: Increase of the reset time Ti (C00074/1) of the $I_{max}$ controller.
- vibrations occur in the V/f control (VFCplus + encoder) mode during the intervention of the current limitation controller.
  - Recommendation: Increase of the reset time Ti (C00074/1) of the $I_{max}$ controller.
- overcurrent errors (e.g. OC3) occur due to load impulses or too high acceleration ramps.
  - Recommendation: Reduction of the gain Vp (C00073/1) and reset time Ti (C00074/1) of the $I_{max}$ controller.

### 5.4.3.4 Optimising the stalling behaviour

Motor stalling due to a torque overload in the field weakening range is prevented in all characteristic-based motor control types (VFCplus) by means of an inverter-internal stalling current monitoring. In the field weakening range, hence at frequencies above the base frequency, it reduces the maximum current to prevent the motor from stalling. The reduction depends on the current field frequency, the base frequency, the DC-bus voltage and the maximum current (C00022). Generally it applies that a higher field frequency causes a stronger limitation of the maximum current.

The behaviour in the field weakening range can be adapted via the override point of field weakening (C00080). This parameter serves to shift the frequency-dependent maximum current characteristic:

- C00080 > 0 Hz:
  - The maximum current characteristic is shifted by the entered frequency to higher field frequencies.
  - The maximally permissible current and the maximum torque increase in the field weakening range.
  - The risk of motor stalling increases.
- C00080 < 0 Hz:
  - The maximum current characteristic is shifted by the entered frequency to lower field frequencies.
  - The maximally permissible current and the maximum torque are reduced in the field weakening range.
  - The risk of motor stalling is reduced.

**Note!**

We recommend to keep the Lenze setting (0 Hz).

### 5.4.3.5 Torque limitation

The previous chapter, "Optimising the Imax controller", describes how the drive can be protected from overload. During commissioning, these settings are carried out once and remain unchanged afterwards. However, it is often necessary to limit the torque to a lower value for plant or process reasons.

- To avoid overload in the drive train, the torque in motor mode can be limited via the *nTorqueMotLimit_a* process input signal, and the torque in generator mode can be limited via the *nTorqueGenLimit_a* process input signal:

| Designator<br>DIS code \| data type | Information/possible settings |
|---|---|
| nTorqueMotLimit_a<br>C00830/29 \| INT | Torque limitation in motor mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: 0 ... +199.99 %<br>• If keypad control is performed: Parameterisable via C00728/1. |
| nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in generator mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: -199.99 ... 0 %<br>• If keypad control is performed: Parameterisable via C00728/2. |

**Note!**

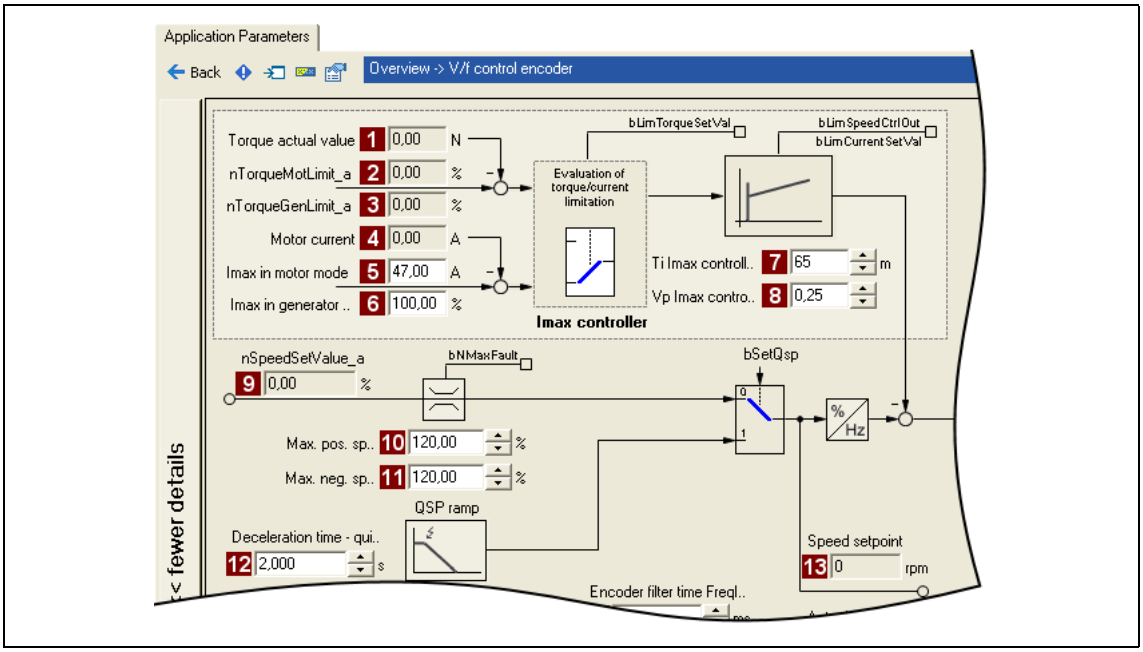
- The actual torque (C00056/2) is directly calculated from the current slip speed of the machine. This requires correct entry of the motor data. (📖 139) ▸ Motor selection/Motor data
- To avoid instabilities during operation with active slip compensation, the torque limit values are internally processed as absolute values.
- If slip compensation is deactivated (C00021 = 0), indirect torque limitation (differential signal between apparent motor current and *nTorqueMotLimit_a* or *nTorqueGenLimit_a*) occurs. Above the no-load current of the motor, the accuracy of the indirect torque limitation is limited.

### V/f characteristic control (VFC)

The accuracy of the torque limitation is limited because the actual torque (C00056/2) is only calculated from the slip speed measured indirectly via the motor current.

### V/f control (VFC + encoder)

The slip speed of the motor is available at the slip controller output. This leads to a high accuracy for the actual torque (C00056/2) and the torque limitation.




[5-6] Extract from the signal flow of the V/f control (VFC + encoder)

| Parameters | | Info |
|---|---|---|
| 1 | C00056/2 | Actual torque |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00054 | Motor current |
| 5 | C00022 | Imax in motor mode |
| 6 | C00023 | Imax in generator mode |
| 7 | C00074/1 | Ti Imax controller |
| 8 | C00073/1 | Vp Imax controller |

| Parameters | | Info |
|---|---|---|
| 9 | C00830/22 | MCTRL: nSpeedSetValue_a |
| 10 | C00909/1 | Max. pos. speed |
| 11 | C00909/2 | Max. neg. speed |
| 12 | C00105 | Decel. time - quick stop |
| 13 | C00050 | Speed setpoint |

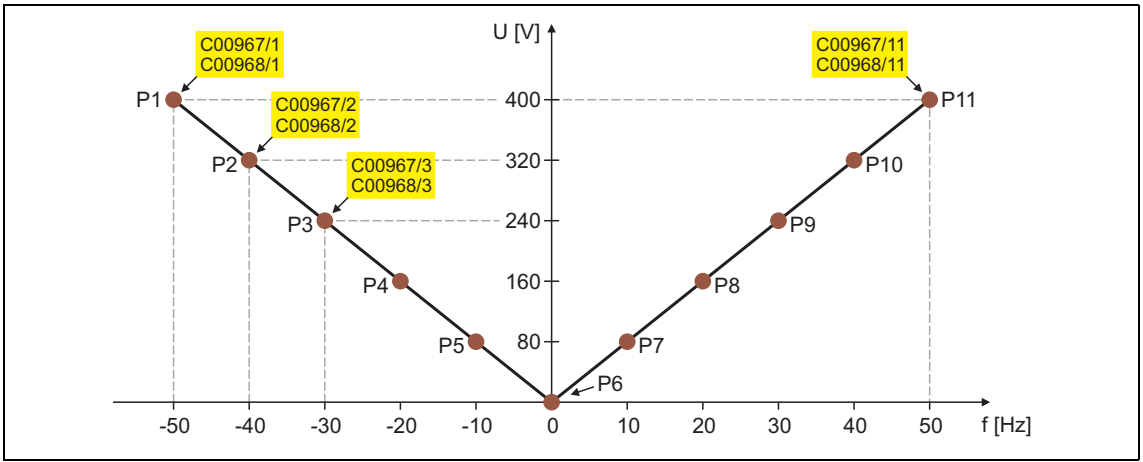
### 5.4.3.6 Defining a user-defined V/f characteristic

For individual adaptation of the motor magnetisation to the actual application, the motor control "10: VFCplus: V/f definable" with a freely definable characteristic can be selected in C00006 as an alternative if the linear and quadratic characteristics are not suitable.

**Note!**

The V/f base frequency (C00015) and the $V_{min}$ boost (C00016) no longer exert an influence if this motor control is chosen.

- The 11 grid points (voltage/frequency values) of the characteristic are selected via the 11 subcodes of C00967 and C00968.
  - It is necessary to set all 11 grid points by means of corresponding subcodes.
  - If fewer grid points (voltage/frequency values) are needed, this can be achieved indirectly by ascribing the same voltage and frequency values to consecutive grid points.
    Example: C00967/**3** = C00967/**4** and C00968/**3** = C00968/**4**
  - The grid points can be specified in any sequence. Internally, they are automatically ordered from the minimum to the maximum frequency value.
  - Above the maximum and below the minimum frequency, the previous rise is continued until the maximum output voltage.
- In the Lenze setting, the 11 grid points represent a linear characteristic.
  - 3-phase devices: Output voltage 400 V at f = 50 Hz
  - 1-phase devices: Output voltage 230 V at f = 50 Hz




| | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **V** | 400 V | 320 V | 240 V | 160 V | 80 V | 0 V | 80 V | 160 V | 240 V | 320 V | 400 V |
| **f** | -50 Hz | -40 Hz | -30 Hz | -20 Hz | -10 Hz | 0 Hz | 10 Hz | 20 Hz | 30 Hz | 40 Hz | 50 Hz |

[5-7] Freely definable characteristic (Lenze setting for 3-phase devices)

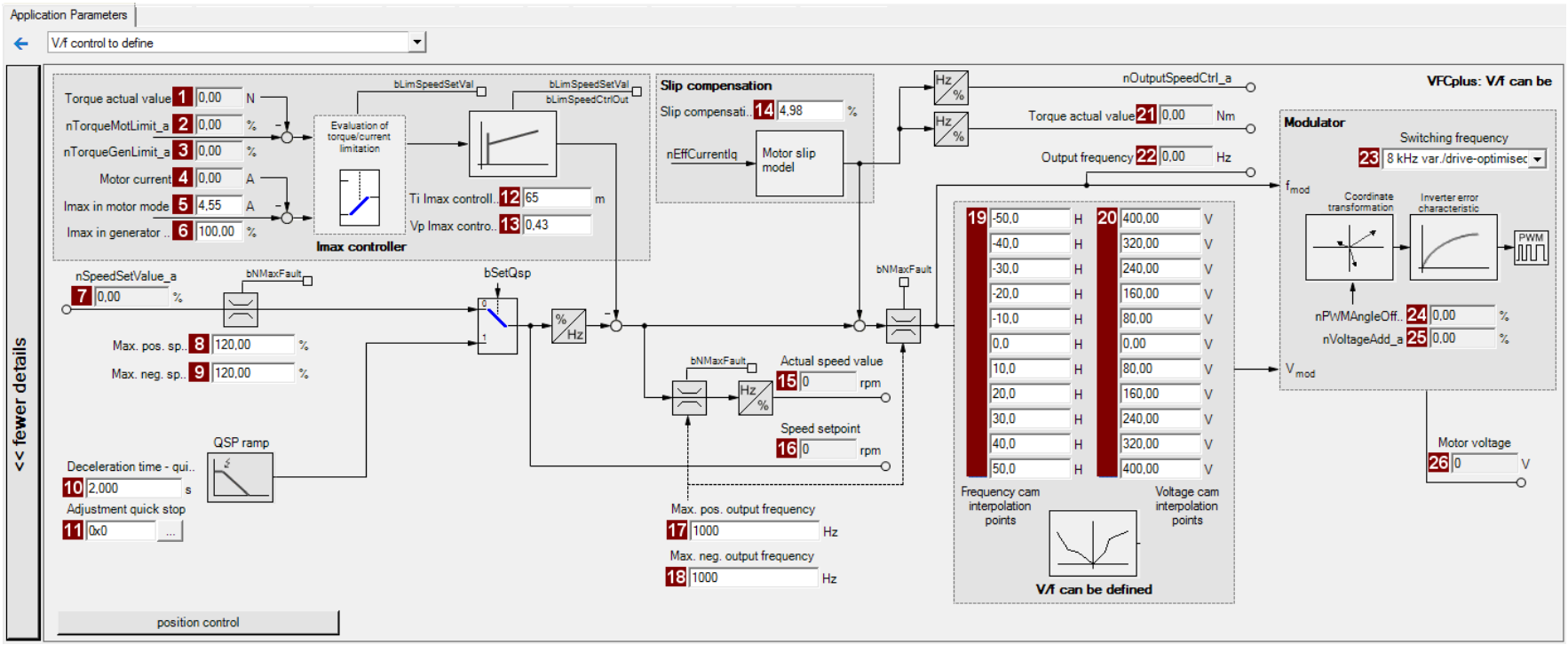
**Tip!**

Cases of application for this function:

- Operation of reluctance motors or synchronous motors during controlled acceleration (reduction of natural frequencies caused by wrong excitation).
- Adaptation of the voltage requirement for the motor, depending on specific load conditions.

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Select the motor control "10: VFCplus: V/f definable" from the *Overview* dialog box in the **Motor control** list field:
4. Click the **Motor control V/f definable** button to change to the *Overview* → *Motor control V/f* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed.




| Parameters | | Info |
|---|---|---|
| 1 | C00056/2 | Actual torque |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00054 | Motor current |
| 5 | C00022 | Imax in motor mode |
| 6 | C00023 | Imax in generator mode |
| 7 | C00830/22 | Speed setpoint |
| 8 | C00909/1 | Max. pos. speed |
| 9 | C00909/2 | Max. neg. speed |
| 10 | C00105 | Decel. time - quick stop |
| 11 | C00104/1 | Quick stop setting |

| Parameters | | Info |
|---|---|---|
| 12 | C00074/1 | Ti Imax controller |
| 13 | C00073/1 | Vp Imax controller |
| 14 | C00021 | Slip compensation |
| 15 | C00051 | Actual speed value |
| 16 | C00050 | Speed setpoint |
| 17 | C00910/1 | Max. pos. output frequency |
| 18 | C00910/2 | Max. neg. output frequency |

| Parameters | | Info |
|---|---|---|
| 19 | C00967/x | Frequency interpol. points |
| 20 | C00968/x | Voltage interpol. points |
| 21 | C00056/2 | Actual torque |
| 22 | C00058 | Output frequency |
| 23 | C00018 | Switching frequency |
| 24 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 25 | C00830/31 | MCTRL: nVoltageAdd_a |
| 26 | C00052 | Motor voltage |

### 5.4.4 Remedies for undesired drive behaviour

| Drive behaviour | Remedy |
|---|---|
| Inadequate smooth running at low speeds, especially in the case of operation with a long motor cable | ▸ Automatic motor data identification (📖 145) |
| Problems in case of high starting duty (great mass inertia) | ▸ Adapting the Vmin boost (📖 170) |
| Drive does not follow the speed setpoint. | The current controller intervenes in the set field frequency to limit the controller output current to the maximum current (C0022, C0023). Therefore:<br>• Prolong acceleration/deceleration times:<br>C00012: Accel. time - main setpoint<br>C00013: Decel. time - main setpoint<br>• Consider a sufficient magnetising time of the motor. Depending on the motor power, the magnetising time amounts to 0.1 ... 0.2 s.<br>• Increase the maximally permissible current:<br>C00022: Imax in motor mode<br>C00023: Imax in generator mode) |
| For operation without speed feedback (C00006 = 6):<br>Insufficient speed constancy at high load (setpoint and motor speed are not proportional anymore) | • Increase slip compensation (C00021).<br>Important: Unstable drive due to overcompensation!<br>• With cyclic load impulses (e. g. centrifugal pump), a smooth motor characteristic is achieved by smaller values in C00021 (possibly negative values).<br>Note: The slip compensation is only active for operation without speed feedback. |
| "Clamp operation active" error message (oC11):<br>Inverter cannot follow dynamic processes, i.e. too short acceleration/deceleration times in terms of load ratios. | • Increase the gain of the $I_{max}$ controller (C00073/1)<br>• Reduce the reset time of the $I_{max}$ controller (C00074/1)<br>• Prolong the acceleration time (C00012)<br>• Prolong the deceleration time (C00013) |
| Motor stalling in the field weakening range<br>(adaptation especially required for small machines) | • Reduce the override point of field weakening (C00080)<br>• If motor power < inverter power:<br>Set C00022 to $I_{max}$ = 2 $I_{rated\ motor}$<br>• Reduce dynamic performance of setpoint generation |

## 5.5 V/f characteristic control - energy-saving (VFCplusEco)

With the energy-saving V/f characteristic control mode (VFCplusEco), the motor voltage of the inverter is detected by means of a linear characteristic depending on the field frequency to be created or the motor speed. Moreover, a cosφ control and the resulting voltage reduction causes the motor to be always operated in the optimum efficiency range (reduction of copper losses in the asynchronous motor).

- Hence, these are the advantages of this motor control mode:
  - Good robustness
  - Easy parameter setting
  - High energy efficiency (lower heating of the motor in partial load operational range)
  - Same speed accuracy and maximum torques as with VFCplus
  - Less noise generation of the motor with active voltage reduction
- Predestinated application areas of this motor control mode are materials handling technology and pump and fan systems.
- This motor control mode serves to improve efficiency of standard asynchronous motors with efficiency class IE1 (standard IEC 60034-30 2008) in the range 0 ... $M_{efficiency_max}$ between 0 ... 20 % (Ø 5 ... 10 %).
  - For asynchronous motors with energy efficiency class IE2 the potential for efficiency improvement is reduced to approx. 0 ... 15 %.
  - Description of $M_{efficiency_max}$: Indicates the torque [%] of $M_{rated_motor}$, where the motor has the max. efficiency.)
- In case of asynchronous motors with a higher energy efficiency class (IE2 and IE3), the absolute energy saving of the motor control mode is lower due to improved efficiency of the machine. However, energy saving is still achieved in a higher load range.
- $M_{efficiency_max}$ is performance-related and listed in the following table for some power values of the energy efficiency class IE1 and IE2:

| Power | $M_{Efficiency_max}$ (related to $M_{rated_motor}$) | |
|---|---|---|
| | IE1 | IE2 |
| 0.25 kW | 75 % | |
| 0.75 kW | 65 % | 75 % |
| 2.2 kW | 55 % | 85 % |
| 7.5 kW | 30 % | 45 % |
| 22 kW | 23 % | |
| 45 kW | 21 % | |

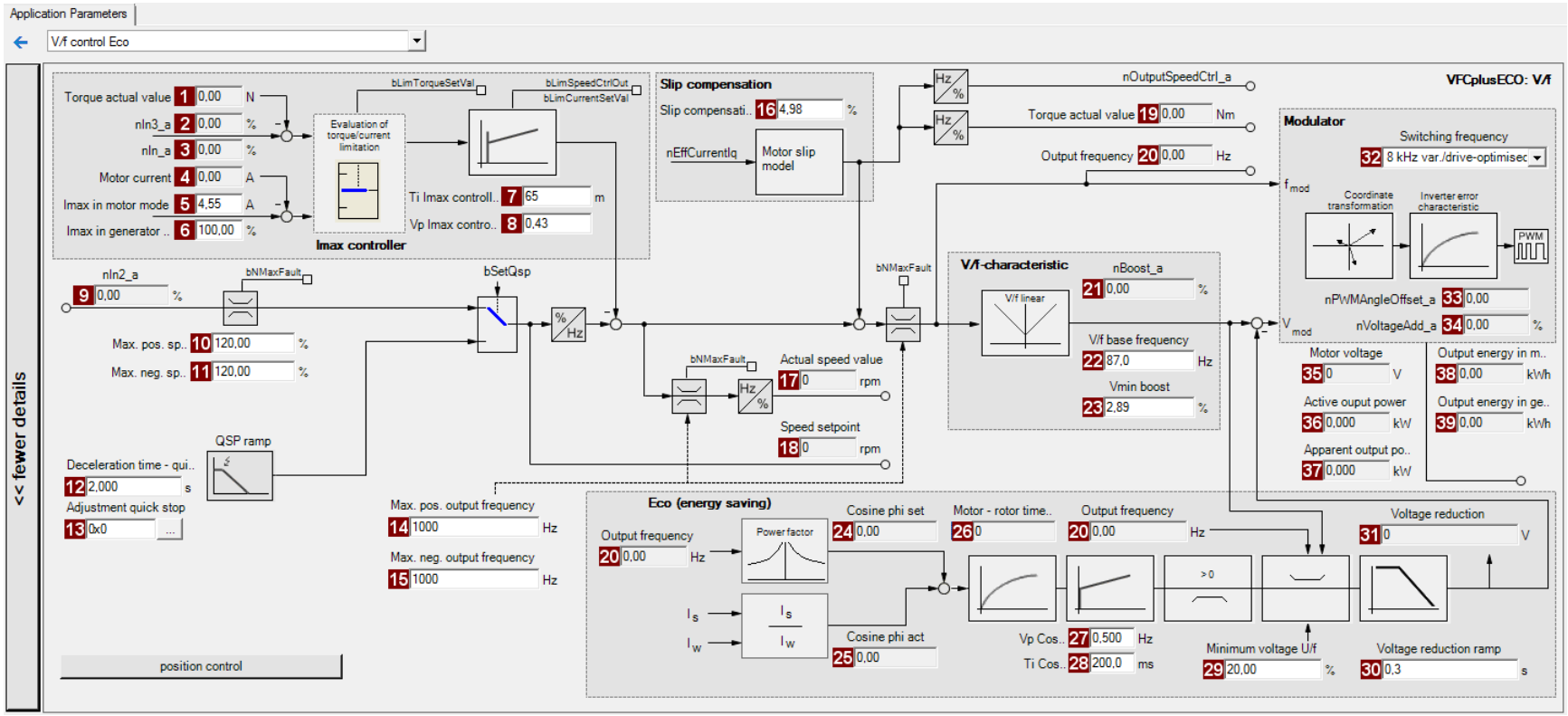
**Stop!**

- For adjustment, observe the thermal performance of the connected asynchronous motor at low output frequencies.
  - Usually, standard asynchronous motors with insulation class B can be operated for a short time with their rated current in the frequency range 0 Hz ... 25 Hz.
  - Contact the motor manufacturer to get the exact setting values for the max. permissible motor current of self-ventilated motors in the lower speed range.
- The nameplate data of the motor (at least rated speed and rated frequency) must be entered if, instead of a standard motor, an asynchronous motor is used with the following values:
  - rated frequency $\neq$ 50 Hz (star) or
  - rated frequency $\neq$ 87 Hz (delta) or
  - number of pole pairs $\neq$ 2

### 5.5.1 Parameterisation dialog/signal flow

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Select the motor control "11: VFCplusEco: V/f energy-saving" from the *Overview* dialog box in the **Motor control** list field:
4. Click the **Motor control V/f Eco** button to change to the *Overview* → *Motor control V/f* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed.




| Parameters | | Info |
|---|---|---|
| 1 | C00056/2 | Actual torque |
| 2 | C00830/4 | Limitation of torque in motor mode |
| 3 | C00830/5 | Limitation of torque in generator mode |
| 4 | C00054 | Motor current |
| 5 | C00022 | Imax in motor mode |
| 6 | C00023 | Imax in generator mode |
| 7 | C00074/1 | Ti Imax controller |
| 8 | C00073/1 | Vp Imax controller |
| 9 | C00830/3 | Speed setpoint |
| 10 | C00909/1 | Max. pos. speed |
| 11 | C00909/2 | Max. neg. speed |
| 12 | C00105 | Decel. time - quick stop |
| 13 | C00104/1 | Quick stop setting |

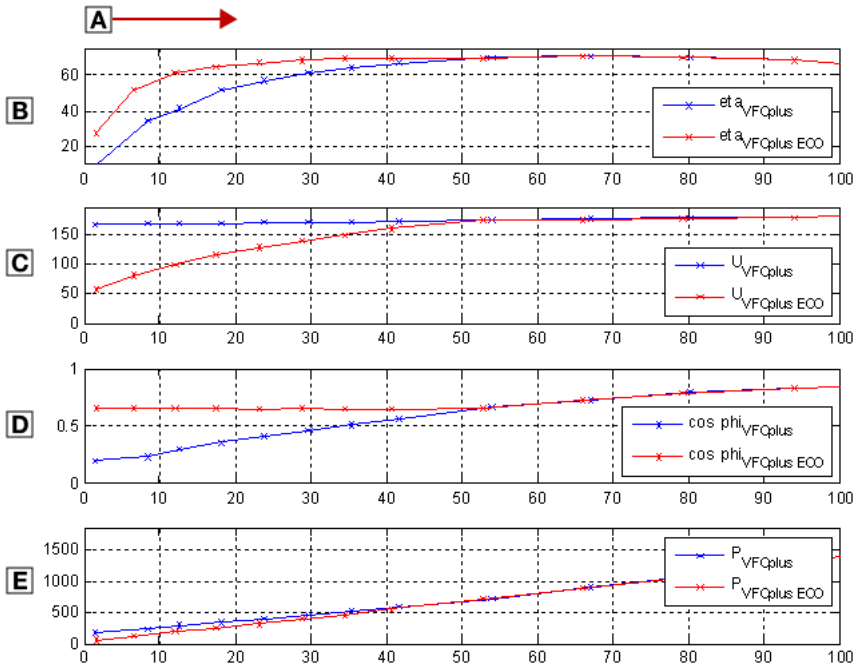
| Parameters | | Info |
|---|---|---|
| 14 | C00910/1 | Max. pos. output frequency |
| 15 | C00910/2 | Max. neg. output frequency |
| 16 | C00021 | Slip compensation |
| 17 | C00051 | Actual speed value |
| 18 | C00050 | Speed setpoint |
| 19 | C00056/2 | Actual torque |
| 20 | C00058 | Output frequency |
| 21 | C00830/26 | MCTRL: nBoost_a |
| 22 | C00015 | V/f base frequency |
| 23 | C00016 | Vmin boost |
| 24 | C00979/2 | Cosine phi set |
| 25 | C00979/1 | Cosine phi act |
| 26 | C00083 | Motor rotor time constant |

| Parameters | | Info |
|---|---|---|
| 27 | C00975 | VFC-ECO: Vp |
| 28 | C00976 | VFC-ECO: Ti |
| 29 | C00977 | VFC-ECO: Minimum voltage V/f |
| 30 | C00982 | VFC-ECO: Voltage reduction ramp |
| 31 | C00978 | VFC-ECO: Voltage reduction |
| 32 | C00018 | Switching frequency |
| 33 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 34 | C00830/31 | MCTRL: nVoltageAdd_a |
| 35 | C00052 | Motor voltage |
| 36 | C00980/1 | Active output power |
| 37 | C00980/2 | Apparent output power |
| 38 | C00981/1 | Output energy in motor mode |
| 39 | C00981/2 | Output energy in generator mode |

### 5.5.2 Comparison of VFCplusEco - VFCplus

The following characteristics show the impact of the energy-saving V/f characteristic control (VFCplusEco) compared to the standard V/f characteristic control (VFCplus).

- The characteristics were recorded with a standard asynchronous motor 2.2 kW with energy efficiency class IE1 at speed = 600 rpm.




A scaled torque in [%]
B efficiency in [%]
C motor voltage in [V]
D power factor
E electrical power in [W]

[5-8] Comparison of VFCplusEco - VFCplus

### 5.5.3 Basic settings

The "Initial commissioning steps" listed in the table below are sufficient for the V/f characteristic control - energy-saving (VFCplusECo).

- Detailed information on the individual steps can be found in the following subchapters.

| Initial commissioning steps | | |
|---|---|---|
| 1. | Determine the motor control:<br>C00006 = "11: VFCplusEco: V/f energy-saving" | |
| 2nd | The required motor data are pre-initialised depending on the device and thus, they do not need to be entered directly.In order to achieve a high energy optimisation, these motor data can be entered (see the following section). | |
| | Set the motor selection/motor data<br>• When selecting and parameterising the motor, the motor nameplate data and the equivalent circuit diagram data are relevant. Detailed information can be found in the "Motor selection/Motor data" chapter. (📖 139)<br>Depending on the motor manufacturer, proceed as follows: | |
| | **Lenze motor:**<br>Selecting a motor from the motor catalogue in the »Engineer«<br>- or -<br>1. Set the motor nameplate data<br>2. Automatic motor data identification | **Third party manufacturer's motor:**<br>1. Set the motor nameplate data<br>2. Automatic motor data identification or set known equivalent circuit diagram manually:<br>C00084: Motor stator resistance<br>C00085: Motor stator leakage inductance<br>C00092: Motor magnetising inductance |
| 3. | Defining current limits (Imax controller). (📖 166) | |

**Tip!**

Information on the optimisation of the control mode and the adaptation to the real application is provided in chapter "Optimising the control mode". (📖 185)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 263)

### 5.5.4 Optimising the control mode

The V/f characteristic control - energy-saving (VFCplus) is generally ready for operation. It can be adapted subsequently by adapting the characteristic and/or the drive behaviour.

**Note!**

Following successful motor parameter identification, the V/f base frequency (C00015) and the $V_{min}$ boost (C00016) as well as the slip constant for slip compensation (C00021) are calculated automatically.

From version 12.00.00 onwards:

- Following successful motor parameter identification, the gain of the Imax controller (C00073/1) is calculated automatically.
  - If these parameters are not to be calculated, bit 4 of C02865/1 must be set to "1".
- Following successful motor parameter identification, other controller parameters (C00011, C00022, C00966, C00982) can be calculated automatically.
  - If these parameters are to be calculated, bit 6 of C02865/1 must be set to "6".

#### Adapting characteristic

For the linear characteristic as part of the V/f characteristic control - energy-saving (VFCplusEco), it is also possible (like in case of the standard V/f characteristic control) to match its curve to different load profiles or motors by adapting the V/f base frequency (C00015) and the $V_{min}$ boost (C00016).

▸ Adapting the V/f base frequency (📖 168)
▸ Adapting the Vmin boost (📖 170)

#### Adapting drive behaviour

- Limitation of the maximum current by a current limitation controller (e.g. to prevent the motor from stalling or to limit to the maximally permissible motor current). ▸ Optimising the Imax controller (📖 172)
- Adaptation of the field frequency by a load-dependent slip compensation (improved speed accuracy for systems without feedback).
- Improving the behaviour at high dynamic load changes. (📖 186)
- Adapting the slope limitation for lowering the Eco function. (📖 187)
- Optimising the cos/phi controller. (📖 187)

#### Torque limitation

Limit the torque to a lower value. ▸ Torque limitation (📖 174)

### 5.5.4.1 Improving the behaviour at high dynamic load changes

Due to the voltage reduction executed via the cosφ control, the motor may stall in the Lenze setting at high dynamic load torque changes. This is caused by the flux reduction and the connected reduction of the stalling torque of the motor current:

$$M_{Max(t)} = M_{Stalling} \cdot \frac{U^2_{Motor(t)}}{(U_{Motor(t)} - U_{Reduction})^2} \qquad \text{with } M_{Stalling} = 1.6 \ldots 2.5 \cdot M_{Rated_motor}$$

$V_{Motor}$ = display in C00052
$V_{Reduction}$ = display in C00978

It generally applies that when the output voltage is divided in halves, the maximum torque is approx. reduced by the factor 4. A reduction by the factor 3 reduces the torque to approx. 15 %.

The minimum voltage and thus the maximum influence access of the Eco function on the output voltage can be defined in C00977. With full influence of the Eco function, the following stalling torque can be ensured depending on the setting in C00977:

| Minimum voltage V/f (C00977) | Maximum torque |
|---|---|
| 100 % | 160 % ... 250 % $M_{rated}$ |
| 70 % | 80 % ... 130 % $M_{rated}$ |
| 50 % | 40 % ... 70 % $M_{rated}$ |
| 20 % | 15 % ... 50 % $M_{rated}$ |

An adaptation of the minimum voltage V/f (C00977) improves the stability in case of load impulses.

- In the Lenze setting, the minimum voltage V/f is set to 20 % for the highest energy optimisation. This setting serves to respond to load torques if these amount to approx. 25 % of the rated torque or occur with low dynamics.
- An increase of the minimum voltage V/f to 70 % permits to apply a dynamic load impulse from 0 to 100 % rated motor torque without the motor stalling. This reduces the energy optimisation to be achieved by approx. 75 %.
- A further increase of the stability at still higher dynamic load impulses can be achieved by a further increase of the minimum voltage V/f, but means a further loss in energy optimisation.

**Note!**

In case of applications with very high dynamic sudden load variations from the unloaded operation, this motor control mode should not be used or the energy optimisation should be switched off, since a motor stalling cannot be excluded.

- The energy optimisation can be switched off by setting the minimum voltage V/f (C00977) to 100 %. Then, the behaviour corresponds to the V/f characteristic control (VFCplus) with linear characteristic.
- From version 13.00.00, the energy optimisation can be switched off via the *bVfcEcoDisable* process signal if a dynamic load change is known to take place.

### 5.5.4.2 Adapting the slope limitation for lowering the Eco function

The ramp set in C00982 for voltage reduction serves as slope limitation in order to prevent that voltage is suddenly applied to the motor when the Eco function is deactivated. Otherwise, the overvoltage limitation (Imax, Clamp) would be activated.

- This ramp is, depending on the device, pre-initialised to approx. the triple rotor time constant. An adaptation of this parameter is not required.

When the Eco function is switched off, a quick reaction (high dynamic performance) is required, but with a low current overshoot and a small torque jump. Thus, the Lenze setting of C00982 is a compromise regarding the switch-off of the Eco function (voltage reduction = 0).

- To increase the dynamics when switching off the Eco function:
  Reduce → setting in C00982.
  (Current compensation actions increase when the Eco function is switched off.)
- In order to reduce current compensation actions when switching off the Eco function:
  Increase → setting in C00982.
  (The dynamics when switching off the eco function is reduced)

### 5.5.4.3 Optimising the cos/phi controller

With the Lenze setting, the cosφ controller is set such that usually no adaptation is required for all power ratings and application cases.

| Behaviour | Remedy/recommendation |
|---|---|
| The cosφ actual value (C00979/1) varies greatly. | Reduce gain Vp (C00975) and reset time Ti (C00976). |
| The cosφ actual value (C00979/1) is permanently lower than the cosφ setpoint (C00979/2). | Increase gain Vp (C00975) and reset time Ti (C00976). |

### 5.5.5 Remedies for undesired drive behaviour

| Drive behaviour | Remedy |
|---|---|
| Inadequate smooth running at low speeds, especially in the case of operation with a long motor cable | ▸ Automatic motor data identification (📖 145)<br>Reduce the influence of the Eco function by increasing the minimum voltage V/f (C00977) if necessary. |
| Problems in case of high starting duty (great mass inertia) | 1. Set motor control VFCplus with linear characteristic (C00006 = 6).<br>2. Adapting the Vmin boost. (📖 170)<br>3. Again set motor control VFCplusEco (C00006 = 11). |
| Drive does not follow the speed setpoint | The current controller intervenes in the set field frequency to limit the controller output current to the maximum current (C0022, C0023). Therefore:<br>• Prolong acceleration/deceleration times:<br>C00012: Accel. time - main setpoint<br>C00013: Decel. time - main setpoint<br>• Consider a sufficient magnetising time of the motor. Depending on the motor power, the magnetising time amounts to 0.1 ... 0.2 s.<br>• Increase the maximally permissible current:<br>C00022: Imax in motor mode<br>C00023: Imax in generator mode<br>• Make adaptations for the Eco function:<br>  • Improving the behaviour at high dynamic load changes. (📖 186)<br>  • Adapting the slope limitation for lowering the Eco function. (📖 187)<br>  • Optimising the cos/phi controller. (📖 187) |
| Insufficient speed constancy at high load (setpoint and motor speed are not proportional anymore) | • Increase slip compensation (C00021).<br>Important: Unstable drive due to overcompensation!<br>• With cyclic load impulses (e. g. centrifugal pump), a smooth motor characteristic is achieved by smaller values in C00021 (possibly negative values).<br>Note: The slip compensation is only active for operation without speed feedback. |
| "Clamp operation active" error message (oC11):<br>Inverter cannot follow dynamic processes, i.e. too short acceleration/deceleration times in terms of load ratios. | • Increase the gain of the $I_{max}$ controller (C00073)<br>• Reduce the reset time of the $I_{max}$ controller (C00074)<br>• Prolong the acceleration time (C00012)<br>• Prolong the deceleration time (C00013)<br>• Make adaptations for the Eco function:<br>  • Improving the behaviour at high dynamic load changes. (📖 186)<br>  • Adapting the slope limitation for lowering the Eco function. (📖 187) |
| Motor stalling in the field weakening range<br>(adaptation especially required for small machines) | • If motor power < inverter power:<br>Set C00022 to $I_{max}$ = 2 $I_{rated\ motor}$<br>• Reduce dynamic performance of setpoint generation<br>• Make adaptations for the Eco function:<br>  • Improving the behaviour at high dynamic load changes. (📖 186)<br>  • Adapting the slope limitation for lowering the Eco function. (📖 187) |
| Speed variations in no-load operation for speeds > 1/3 rated speed. | Minimise speed oscillations with oscillation damping (C00234). |

| Drive behaviour | Remedy |
|---|---|
| Speed variations in no-load operation and with load for speeds > rated speed. | Minimise speed oscillations with increasing the oscillation damping field weakening (C00236).<br>Caution: If C00236 is increased, the maximum output voltage of the device is reduced! |
| Output voltage is too low. There is a too low maximum torque in the high field weakening range. | Reduction of the oscillation damping field weakening (C00236).<br>Caution: When C00236 = 0, oscillation damping field weakening is inactive. Thus, a maximum output voltage is available but the tendency to speed oscillations in the field weakening range at no-load operation and with load increases. |

## 5.6 V/f control (VFCplus + encoder)

The V/f characteristic control (VFCplus) described above can be operated with a speed feedback. This has the following advantages:

- Steady-state accuracy of the speed
- Less parameterisation effort compared to the sensorless vector control (SLVC)
- Improved dynamics compared to V/f characteristic control without feedback or to sensorless vector control (SLVC).
- Suitability for group drives

### Stop!

The V/f control requires a speed feedback!

- The speed sensor used has to be set in C00495. This setting is not made automatically with the selection of the motor from the »Engineer« motor catalogue!
- If no speed sensor is set in C00495 and the controller is enabled, an impermissibly high motor current occurs which may destroy the motor thermally!
- From version 14.00.00, the error response set in C00571/2 (Lenze setting: "Fault") takes place if in case of controller enable it is detected that a motor control type with feedback is set in C00006 but no speed encoder is set in C00495.

### Stop!

**V/f emergency operation**

From version 15.00.00 , it is internally switched to the encoderless V/f characteristic control in case of an encoder open circuit in order to avoid impermissible motor movements.

- In order that this "V/f emergency operation" works properly, the parameters relevant for the V/f characteristic control (base frequency, Vmin boost, slip compensation, etc.) have to be set correctly. As an alternative, a motor parameter identification can be executed as well.
- The change-over to "V/f emergency operation" is reported via bit 4 in C01000 and via the *bWirebreakUfLinearActive* status signal at the SB LS_DeviceMonitor.
- The change-over to "V/f emergency operation" can be suppressed by setting bit 8 to "1" in C02864/1.

### Note!

As the slip is calculated in the feedback V/f operation and injected through the slip regulator, the slip compensation (C00021) is deactivated with V/f control.

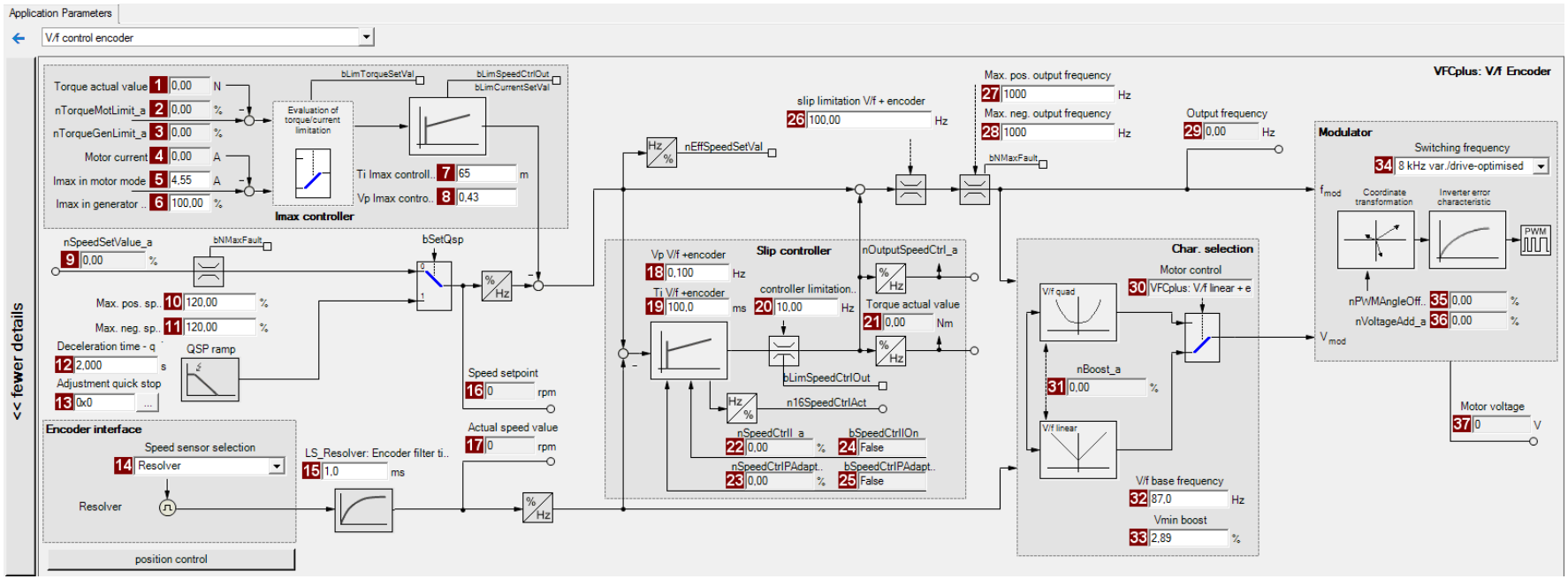
The descriptions in chapter "V/f characteristic control (VFCplus)" also apply to the V/f control. (📖 161)

Detailed information on the speed feedback can be found in the chapter "Encoder/feedback system". (📖 321)

### 5.6.1 Parameterisation dialog/signal flow

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Select the motor control from the *Overview* dialog level in the **Motor control** (C00006) list field:
   - "7: VFCplus: V/f linear +encoder" for linear characteristic or
   - "9: VFCplus: V/f quadr +encoder" for quadratic characteristic
4. Click the **Motor control V/f encoder** button to change to the *Overview* → *Motor control V/f* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed, as shown in the following subchapter.




| | Parameters | Info |
|---|---|---|
| 1 | C00056/2 | Actual torque |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00054 | Motor current |
| 5 | C00022 | Imax in motor mode |
| 6 | C00023 | Imax in generator mode |
| 7 | C00074/1 | Ti Imax controller |
| 8 | C00073/1 | Vp Imax controller |
| 9 | C00830/22 | Speed setpoint |
| 10 | C00909/1 | Max. pos. speed |
| 11 | C00909/2 | Max. neg. speed |
| 12 | C00105 | Decel. time - quick stop |
| 13 | C00104/1 | Quick stop setting |
| 14 | C00495 | Speed sensor selection |
| 15 | C00497/4 | Encoder filter time FreqIn12 |

| | Parameters | Info |
|---|---|---|
| 16 | C00050 | Speed setpoint |
| 17 | C00051 | Actual speed value |
| 18 | C00972 | Vp Vf+encoder |
| 19 | C00973 | Ti Vf+encoder |
| 20 | C00971/1 | Controller limitation Vf+encoder |
| 21 | C00056/2 | Actual torque |
| 22 | C00830/24 | MCTRL: nSpeedCtrlI_a |
| 23 | C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| 24 | C00833/31 | MCTRL: bSpeedCtrlIOn |
| 25 | C00833/69 | MCTRL: bSpeedCtrlPAdaptOn |
| 26 | C00971/2 | Slip limitation Vf+encoder |
| 27 | C00910/1 | Max. pos. output frequency |
| 28 | C00910/2 | Max. neg. output frequency |

| | Parameters | Info |
|---|---|---|
| 29 | C00058 | Output frequency |
| 30 | C00006 | Motor control |
| 31 | C00830/26 | MCTRL: nBoost_a |
| 32 | C00015 | V/f base frequency |
| 33 | C00016 | Vmin boost |
| 34 | C00018 | Switching frequency |
| 35 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 36 | C00830/31 | MCTRL: nVoltageAdd_a |
| 37 | C00052 | Motor voltage |

### 5.6.2 Basic settings

In order to protect the drive system, carry out the commissioning of the V/f control and the slip regulator in several steps.

- Detailed information on the single steps can be found in the following subchapters or in the corresponding subchapters for V/f characteristic control.

| Initial commissioning steps | |
|---|---|
| 1. | Define the V/f characteristic:<br>• C00006 = 7: Linear characteristic<br>• C00006 = 9: Quadratic characteristic |
| 2nd | Defining current limits (Imax controller). (📖 166) |
| 3. | Parameterise encoder/feedback system.<br>▸ Encoder/feedback system (📖 321) |
| 4. | If special motors with a rated frequency other than 50 Hz or with a number of pole pairs ≠ 2 are used, set the motor parameters according to the motor nameplate.<br>▸ Motor selection/Motor data (📖 139) |
| 5th | Define speed setpoint (e.g. 20 % of the rated speed) and enable inverter. |
| 6. | Check whether the actual speed value (C00051) ≈ speed setpoint (C00050) and then inhibit the inverter again.<br>• In case of a sign reversal between actual value and setpoint, check the connection or setting of the encoder.<br>• In case the actual value differs considerably from the setpoint (factor 2), set the motor parameters according to motor nameplate. Then repeat step 5. |
| 6. | To protect the drive, reduce the slip regulator limitation in C00971/1.<br>• e.g. reduction to half the slip frequency (≈ 2 Hz) |
| 7. | Define speed setpoint (e.g. 20 % of the rated speed) and enable inverter. |
| 8. | In case of a semi-stable operational performance, reduce the reset time (C00972) or the proportional gain (C00973) of the slip regulator until a stable operation has been achieved.<br>▸ Parameterising the slip regulator (📖 194) |
| 10. | In a final step, increase the slip regulator limitation again in C00971/1.<br>• e.g. increase to twice the slip frequency |

**Tip!**

Information on the further optimisation of the control mode and the adaptation to the real application is provided in the "Optimising the control mode" chapter for the V/f characteristic control (VFCplus). (📖 167)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 263)

### 5.6.2.1 Parameterising the slip regulator

The slip regulator is designed as a PI controller. In order to improve the response to setpoint changes, the setpoint speed or setpoint frequency is added to the output (correcting variable) of the slip regulator as feedforward control value.

- Unlike traditional speed controllers, the slip regulator only controls the slip.
- In the Lenze setting, the configuration of the slip regulator provides robustness and moderate dynamics.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00971/1 | VFC: Controller limitation V/f +encoder | 10.00 | Hz |
| C00971/2 | VFC: Slip limitation V/f +encoder | 100.00 | Hz |
| C00972 | VFC: Vp V/f +encoder | 0.100 | Hz/Hz |
| C00973 | VFC: Ti V/f +encoder | 100.0 | ms |

#### Slip regulator gain Vp

The setting range of the slip regulator gain Vp (C00972) which leads to a stable operational performance, mainly depends on the resolution of the speed sensor. There is a direct relationship between encoder resolution and gain:

- The higher the encoder resolution, the higher the gain can be set.

The following table provides maximum and recommended slip regulator gains for encoder with standard encoder increments:

| Encoder increment [Increments/revolution] | Slip regulator gain Vp | |
|---|---|---|
| | maximum | recommended |
| 8 | 0.09 | 0.06 |
| 64 | 0.52 | 0,31 |
| 100 | 0.79 | 0.47 |
| 120 | 0.94 | 0.57 |
| 128 | 1.00 | 0.60 |
| 256 | 1.29 | 0.77 |
| 386 | 1.63 | 0.98 |
| 512 | 1.97 | 1.18 |
| 640 | 2.31 | 1.38 |
| 768 | 2.65 | 1.59 |
| 896 | 2.99 | 1.79 |
| 1014 | 3.33 | 2.00 |
| 1536 | 4.69 | 2.81 |
| 2048 | 6.05 | 3.63 |
| 3072 | 8.77 | 5.26 |
| 4096 | 11.49 | 6.90 |

[5-1] Slip regulator gain Vp based on the encoder increment

**How to adapt the slip regulator gain to the operating conditions:**

1. Adapt the slip regulator gain (C00972) to the encoder increment according to table [5-1].
2. Set controller limitation (C00971/1) to half the slip frequency (≈ 2 Hz).
3. Select speed setpoint (e.g. 20 % of the rated speed).
4. Enable inverter.
5. Increase the slip regulator gain (C00972) until the drive is semi-stable.
   - This can be recognised by motor noises or "humming" of the motor or by a noise on the actual speed signal.
6. Reduce slip regulator gain (C00972) until the drive runs stable again (no motor "humming").
7. Reduce slip regulator gain (C00972) to approx. half the value.
   - With low encoder resolutions, another reduction of the slip regulator gain for low speeds may be necessary (speed setpoint ≈ 0).
   - We recommend to finally check the behaviour at setpoint speed = 0 and to further reduce the slip regulator gain if irregular running occurs.
8. Increase controller limitation (C00971/1) again (e.g. to twice the slip frequency).

### Slip regulator time constant Ti

**How to set the slip regulator time constant:**

1. Set controller limitation (C00971/1) to half the slip frequency (≈ 2 Hz).
2. Select speed setpoint (e.g. 20 % of the rated speed).
3. Enable inverter.
4. Reduce the slip regulator time constant (C00973) until the drive is semi-stable.
   - This can be recognised by motor noise, "motor vibrations" or resonance on the actual speed value signal.
5. Increase slip regulator time constant (C00973) until the drive runs stable again (no motor "oscillation").
6. Increase the slip regulator time constant (C00973) to approx. twice the value.
7. Increase controller limitation (C00971/1) again (e.g. to twice the slip frequency).

### Controller limitation

Max. intervention of the controller is limited by the controller limitation (C00971/1).

- The controller can be limited depending on the application.
- We recommend to limit the max. intervention to twice the rated slip of the motor.
- The rated slip is calculated as follows:

$$f_{Slip_{Rated}}[Hz] = f_{Rated}[Hz] - \left(\frac{n_{Motor_{Rated}}[rpm]}{60} \cdot p_{Number\ of\ pole\ pairs}\right)$$

[5-9] Calculation of the rated slip

**Note!**

A setting of C00971/1 = 0 Hz deactivates the slip regulator. In this case, the structure of the V/f control corresponds to the structure of a V/f characteristic control without feedback.

### Slip limitation

In addition to limiting the slip regulator, the field frequency to be injected can also be limited by another limiting element, the slip limitation (C00971/2).

- If the slip is e.g. limited to twice the rated slip of the motor, a stalling of the motor during very dynamic processes can be avoided.
- Motor stalling is caused by:
  - a high overcurrent at very steep speed ramps
  - very fast speed changes due to load, e.g. abrupt stopping of the drive due to an encounter with a stop or a load that is not moving.

## 5.7 Sensorless vector control (SLVC)

Sensorless vector control (SLVC) is based on a better motor current control according to a field-oriented control mode by Lenze.

### Stop!

- The sensorless vector control (SLVC) is only suitable for asynchronous motors.
- The connected motor may be maximally two power classes lower than the motor assigned to the inverter.
- Operation of the sensorless vector control (SLVC) is only permissible for one single drive!
- Operation of the sensorless vector control (SLVC) is not permissible for hoists!
- The Lenze setting permits the operation of a power-adapted motor. Optimal operation is only possible if either:
  - the motor is selected via the »Engineer« motor catalogue,
  - the motor nameplate data are entered and motor parameter identification is carried out afterwards
    *- or -*
  - the nameplate data and equivalent circuit data of the motor (motor leakage inductance and mutual motor inductance, slip compensation and motor stator resistance) are entered manually.
- When you enter the motor nameplate data, take into account the phase connection implemented for the motor (star or delta connection). Only enter the data applying to the selected connection type.
  - In this context, also observe the instructions in chapter "Adapting the V/f base frequency" relating to V/f characteristic control. (📖 168)

### Note!

Optimal operation of the sensorless vector control (SLVC) can be achieved from a minimum speed of approx. 0.5-fold slip speed. At lower speed values below the 0.5-fold slip speed, the maximum torque is reduced.

The maximum field frequency with this motor control mode is 650 Hz.

In comparison to the V/f characteristic control without feedback, the following can be achieved by means of sensorless vector control SLVC:

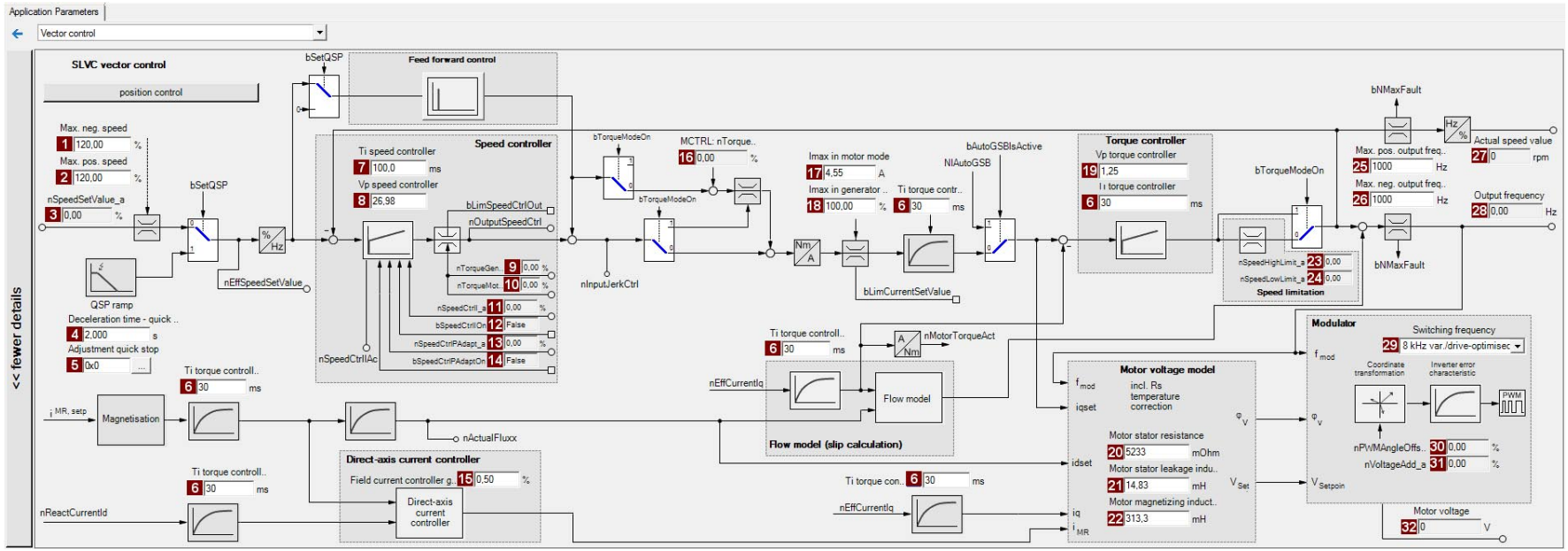
- A higher maximum torque throughout the entire speed range
- A higher speed accuracy
- A higher concentricity factor
- A higher level of efficiency
- The implementation of torque-controlled operation with speed limitation
- The limitation of the maximum torque in motor and generator mode for speed-controlled operation

### 5.7.1 Parameterisation dialog/signal flow



**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Select the motor control "4: SLVC: Vector control" from the *Overview* dialog level in the **Motor control** list field (C00006):
4. Click the **Motor control vector** button to change to the *Overview* → *Motor control vector* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed, as shown in the following subchapter.




| Parameters | | Info |
|---|---|---|
| 1 | C00909/2 | Max. neg. speed |
| 2 | C00909/1 | Max. pos. speed |
| 3 | C00830/22 | Speed setpoint |
| 4 | C00105 | Decel. time - quick stop |
| 5 | C00104/1 | Quick stop setting |
| 6 | C00074/2 | SLVC: Ti torque controller |
| 7 | C00071/1 | SLVC: Ti speed controller |
| 8 | C00070/1 | SLVC: Vp speed controller |
| 9 | C00830/28 | Limitation of torque in generator mode |
| 10 | C00830/29 | Limitation of torque in motor mode |
| 11 | C00830/24 | MCTRL: nSpeedCtrlI_a |
| 12 | C00833/31 | MCTRL: bSpeedCtrlIOn |
| 13 | C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| 14 | C00833/69 | MCTRL: bSpeedCtrlPAdaptOn |
| 15 | C00985 | SLVC: Gain of field current controller |

| Parameters | | Info |
|---|---|---|
| 16 | C00830/27 | MCTRL: nTorqueSetValue_a |
| 17 | C00022 | Imax in motor mode |
| 18 | C00023 | Imax in generator mode |

| Parameters | | Info |
|---|---|---|
| 19 | C00073/2 | SLVC: Vp torque controller |
| 20 | C00084 | Motor stator resistance |
| 21 | C00085 | Motor stator leakage inductance |
| 22 | C00092 | Motor magnetising inductance |
| 23 | C00830/88 | MCTRL: nSpeedHighLimit_a |
| 24 | C00830/23 | MCTRL: nSpeedLowLimit_a |
| 25 | C00910/1 | Max. pos. output frequency |
| 26 | C00910/2 | Max. neg. output frequency |
| 27 | C00051 | Actual speed value |
| 28 | C00058 | Output frequency |
| 29 | C00018 | Switching frequency |
| 30 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 31 | C00830/31 | MCTRL: nVoltageAdd_a |
| 32 | C00052 | Motor voltage |

### 5.7.2 Types of control

The sensorless vector control can be operated in two different modes:

- Speed control with torque limitation (*bTorquemodeOn* = FALSE)
- Torque control with speed limitation (*bTorquemodeOn* = TRUE)

#### 5.7.2.1 Speed control with torque limitation

When *bTorquemodeOn* = FALSE, the drive system is operated with a selected speed setpoint in a speed-controlled manner.

**Note!**

From version 13.00.00, the torque setpoint *nSpeedSetValue_a* is set to 0 by quick stop (QSP) device-internally and the torque limit values *nTorqueMotLimit_a* and *nTorqueGenLimit_a* are set to 100 % to stop the drive quickly and safely anytime. The previous behaviour can be set in C2865/1 via bit 12 and bit 13.

The operational performance can be adapted in the following ways:

A. Overload limitation in the drive train
B. Motor current limitation
C. Slip compensation

**Overload limitation in the drive train**

The torque is limited via the torque setpoint.

- The torque setpoint is identical to the value at the output of the speed controller, *nOutputSpeedCtrl*.
- To avoid overload in the drive train, the torque in motor mode can be limited via the *nTorqueMotLimit_a* process input signal, and the torque in generator mode can be limited via the *nTorqueGenLimit_a* process input signal:

| Designator<br>DIS code \| data type | Information/possible settings |
|---|---|
| nTorqueMotLimit_a<br>C00830/29 \| INT | Torque limitation in motor mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: 0 ... +199.99 %<br>• If keypad control is performed: Parameterisable via C00728/1. |
| nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in generator mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: -199.99 ... 0 %<br>• If keypad control is performed: Parameterisable via C00728/2. |

**Note!**

To avoid instabilities during operation, the torque limit values are internally processed as absolute values.

**Motor current limitation**

A cross current setpoint is calculated from the torque setpoint which is limited depending on the magnetising current, the max. current in motor mode (C00022) and the max. current in generator mode (C00023). The total current injected into the motor does not exceed the max. currents in motor mode and in generator mode.

**Slip compensation**

The slip of the machine is reconstructed using the slip model. An influencing parameter is the slip constant (C00021). ▸ Slip compensation (📖 278)

### 5.7.2.2 Torque control with speed limitation

When *bTorquemodeOn* = TRUE, a torque-controlled operation is activated. The setpoint torque directly follows the default value *nTorqueSetValue_a*.

Due to its speed limitation, the torque-controlled drive can only rotate within a speed range whose positive speed is limited by *nSpeedHighLimit_a* and whose negative speed is limited by *nSpeedLowLimit_a*.

**Note!**

- Absolute speed limitation to speed 0 rpm (*nSpeedLowLimit_a* or *nSpeedHighLimit_a* = 0) is only possible from version 12.00.00.
- Quick stop (QSP) is used to switch over to Speed control with torque limitation.
  - From version 13.00.00, the torque setpoint *nSpeedSetValue_a* is set to 0 by quick stop (QSP) device-internally and the two torque limit values *nTorqueMotLimit_a* and *nTorqueGenLimit_a* are set to 100 % to stop the drive quickly and safely anytime. The previous behaviour can be set in C2865/1 via bit 12 and bit 13.

- From version 13.00.00, the *bLimSpeedTorquemodeOn* status signal is used to show that the speed limitation is active.
- The speed is defined by the process.
- The torque setpoint is calculated directly from *nTorqueSetValue_a*.
  - From version 12.00.00 onwards, the torque limitation is active via *nTorqueMotLimit_a* and *nTorqueGenLimit_a* in this control mode, too, for the torque setpoint to be limited. The torque limitation can be deactivated in C2865/1 via bit 0 to obtain the previous function.

| Designator<br>DIS code \| data type | Information/possible settings |
|---|---|
| nTorqueSetValue_a<br>C00830/27 \| INT | Torque setpoint / additive torque<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| nSpeedHighLimit_a<br>C00830/88 \| INT | Upper speed limit for the speed limitation<br>• During torque-controlled operation only (*bTorquemodeOn* = TRUE)<br>• Scaling: 16384 ≡ 100 % rated speed (C00011) |
| nSpeedLowLimit_a<br>C00830/23 \| INT | Lower speed limit for speed limitation<br>• During torque-controlled operation only (*bTorquemodeOn* = TRUE)<br>• Scaling: 16384 ≡ 100 % rated speed (C00011) |

| Designator<br>DIS code \| data type | Information/possible settings |
|---|---|
| nTorqueMotLimit_a<br>C00830/29 \| INT | Torque limitation in motor mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: 0 ... +199.99 %<br>• If keypad control is performed: Parameterisable via C00728/1. |
| nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in generator mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: -199.99 ... 0 %<br>• If keypad control is performed: Parameterisable via C00728/2. |

### 5.7.3 Basic settings

The following "Initial commissioning steps" must be performed to commission the sensorless vector control:

| Initial commissioning steps | | |
|---|---|---|
| 1. | Determine the motor control:<br>C00006 = "4: SLVC: Vector control" | |
| 2nd | Set the motor selection/motor data<br>• When selecting and parameterising the motor, the motor nameplate data and the equivalent circuit diagram data are relevant. Detailed information can be found in the "Motor selection/Motor data" chapter. (📖 139)<br>Depending on the motor manufacturer, proceed as follows: | |
| | **Lenze motor:**<br>Selecting a motor from the motor catalogue in the »Engineer«<br>- or -<br>1. Set the motor nameplate data<br>2. Automatic motor data identification | **Third party manufacturer's motor:**<br>1. Set the motor nameplate data<br>2. Automatic motor data identification or set known equivalent circuit diagram data manually:<br>C00082: Motor rotor resistance<br>C00084: Motor stator resistance<br>C00085: Motor stator leakage inductance<br>C00092: Motor magnetising inductance<br>C00095: Motor magnetising current |
| 3. | Define the type of control:<br>*bTorquemodeOn* = FALSE: Speed control with torque limitation<br>*bTorquemodeOn* = TRUE: Torque control with speed limitation | |
| 4. | Set the slip compensation (C00021). ▸ Slip compensation (📖 278) | |

**Tip!**

Information on the optimisation of the control mode and the adaptation to the real application is provided in chapter "Optimising the control mode". (📖 203)

We recommend to use the flying restart function for connecting/synchronising the inverter to an already rotating drive system. ▸ Flying restart function (📖 271)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 263)

### 5.7.4 Optimising the control mode

**Note!**

From version 12.00.00:

- Following successful motor parameter identification, the current controller parameters (C00075, C00076) are calculated automatically.
  - If these parameters are not to be calculated, bit 4 of C02865/1 must be set to "1".
- Following successful motor parameter identification, the speed controller parameters (C00070/1, C00071/1) can be calculated automatically.
  - If these parameters are to be calculated, bit 6 of C02865/1 must be set to "5".
- Following successful motor parameter identification, other controller parameters (C00011, C00022) can be calculated automatically.
  - If these parameters are to be calculated, bit 6 of C02865/1 must be set to "6".

#### 5.7.4.1 Optimising the starting performance after a controller enable

After the inverter has been enabled, the starting action of the motor is delayed due to the magnetisation of the motor. Under consideration of the motor rotor time constant (C00083), the time delay is calculated as follows:

Magnetisation = 1.5 * motor rotor time constant

If this delay cannot be tolerated for specific applications, the motor must always be operated in an energised condition. For this, select one of the following options:

**Procedure without setting a controller inhibit**

1. Deactivate the auto DCB function with C00019 = 0.
2. Do not activate the controller inhibit. Instead, stop the drive by selecting a setpoint of 0 or by activating the quick stop function.

**Procedure with setting a controller inhibit due to application requirements**

1. Deactivate the auto DCB function with C00019 = 0.
2. Enter a greater value for the motor rotor resistance (max. factor 2!) to reduce the magnetisation time in C00082.

**Note!**

During the starting action, a jerk may occur in the machine due to the temporarily increased motor current!

### 5.7.4.2 Optimise speed controller

The speed controller is designed as a PI controller.

- In the Lenze setting, the configuration of the speed controller provides robustness and moderate dynamics.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00070/1 | SLVC: Vp speed controller | 15.00 | |
| C00071/1 | SLVC: Ti speed controller | 100.0 | ms |

#### Speed controller gain Vp

The gain Vp (C00070/1) of the speed controller is defined in a scaled representation which enables a comparable parameterisation almost independent of the power of the motor or inverter. Here, the speed input difference of the controller is scaled to the rated motor speed whereas the output torque refers to the rated motor torque. A gain of 10 means that a speed difference of 1 % is gained through the P component with 10 % torque.

If the rated data of the motor and the mass inertia of the drive system are known, we recommend the following setting:

$$V_P \approx 1.5 \ldots 3 \cdot \frac{T_M[s]}{0.01[s]}$$

$$T_M[s] = \frac{2 \cdot \pi \cdot n_N[rpm]}{M_N[Nm] \cdot 60} \cdot J_{Drive,\,total}[kgm^2]$$

$$M_N[Nm] = \frac{P_N[W] \cdot 60}{2 \cdot \pi \cdot n_N[rpm]}$$

$V_P$ = Gain of the speed controller (C00070/1)
$T_M$ = Time constant for the acceleration of the motor
$M_N$ = Rated motor torque
$n_N$ = Rated motor speed
$J_{drive,\,total}$ = Total moment of inertia of the drive

[5-10] Recommendation for the setting of the gain of the speed controller

**Tip!**

Values recommended by Lenze for the setting of the (proportional) gain:

- For drive systems without feedback: Vp = 6 ... 25
- For drive systems with a good disturbance behaviour: Vp > 15
  In this case, we recommend the optimisation of the dynamic performance of the torque controller.

**Speed controller reset time Ti**

Apart from setting the P component, C00071/1 provides the possibility to take influence on the I component of the PI controller.

**Tip!**

Value range recommended by Lenze for the setting of the reset time:

Ti = 20 ms ... 150 ms

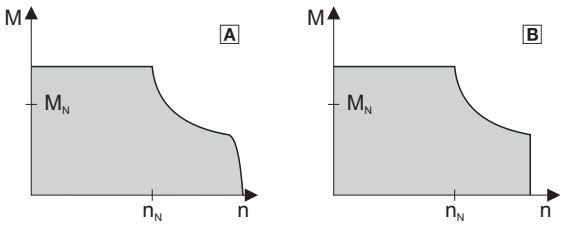
### 5.7.4.3 Optimising dynamic performance and field weakening behaviour

In the Lenze setting, the torque controller has been preset in such a way that robust and stable operation with a moderate dynamic response is enabled over the entire speed range. Retrospective optimisation of the controller parameters is not necessary.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00073/2 | SLVC: Vp torque controller | 1.25 | |
| C00074/2 | SLVC: Ti torque controller | 30 | ms |

A greater dynamic performance of the sensorless vector control can be achieved by reducing time constant Ti of the speed controller (C00074/2).

A greater dynamic performance of the field weakening function can be achieved by setting a time constant ≤ 15 ms. This means for actual speeds above rated speed a better torque-speed-characteristic in the field weakening range:




A Torque controller time constant Ti > 15 ms
B Torque controller time constant Ti ≤ 15 ms

[5-11] Speed / torque characteristic diagram in the field weakening range

- For Ti > 15 ms (see A), the actual speed value slightly drops in the field weakening range if the load torque increases in the motor mode.
- For Ti ≤ 15 ms (see B), the speed remains stable in the field weakening range if the torque is within the M/n characteristic field highlighted in grey.

**Tip!**

For applications with high dynamic performance and speed/torque accuracy requirements in the field weakening range, we recommend a time constant Ti ≤ 15 ms.

In this case, the maximum torque should be limited via the *nTorqueMotLimit_a* and *nTorqueGenLimit_a* process input signals to 1.5 x $M_N$ to ensure stable operation in the field weakening range.

### 5.7.4.4 Optimising the stalling behaviour

Motor stalling due to a torque overload in the field weakening range is prevented in sensorless vector control by means of an inverter-internal stalling current monitoring. In the field weakening range, hence at frequencies above the base frequency, it reduces the maximum current to prevent the motor from stalling. The reduction depends on the current field frequency, the base frequency, the DC-bus voltage and the maximum current (C00022). Generally it applies that a higher field frequency causes a stronger limitation of the maximum current.

The field weakening behaviour of the sensorless vector control depends on the setting of the reset time Ti of the torque controller (C00074/2).

**The following applies to the reset time Ti (C00074/2) > 15 ms:**

The behaviour in the field weakening range can be adapted via the override point of field weakening (C00080). This parameter serves to shift the frequency-dependent maximum current characteristic:

- C00080 > 0 Hz:
  - The maximum current characteristic is shifted by the entered frequency to higher field frequencies.
  - The maximally permissible current and the maximum torque increase in the field weakening range.
  - The risk of motor stalling increases.
- C00080 < 0 Hz:
  - The maximum current characteristic is shifted by the entered frequency to lower field frequencies.
  - The maximally permissible current and the maximum torque are reduced in the field weakening range.
  - The risk of motor stalling is reduced.

**Note!**

We recommend to keep the Lenze setting (0 Hz).

**The following applies to the reset time Ti (C00074/2) <= 15 ms:**

The reduction of the magnetising current in the field weakening range can be adapted via the override point of field weakening (C00080):

- C00080 > 0 Hz:
  The reduction of the magnetising current is shifted to higher field frequencies. Here, there is a risk of the motor being magnetised too much and having too little voltage reserve for the torque-creating current.
- C00080 < 0 Hz:
  The reduction of the magnetising current is shifted to lower field frequencies.

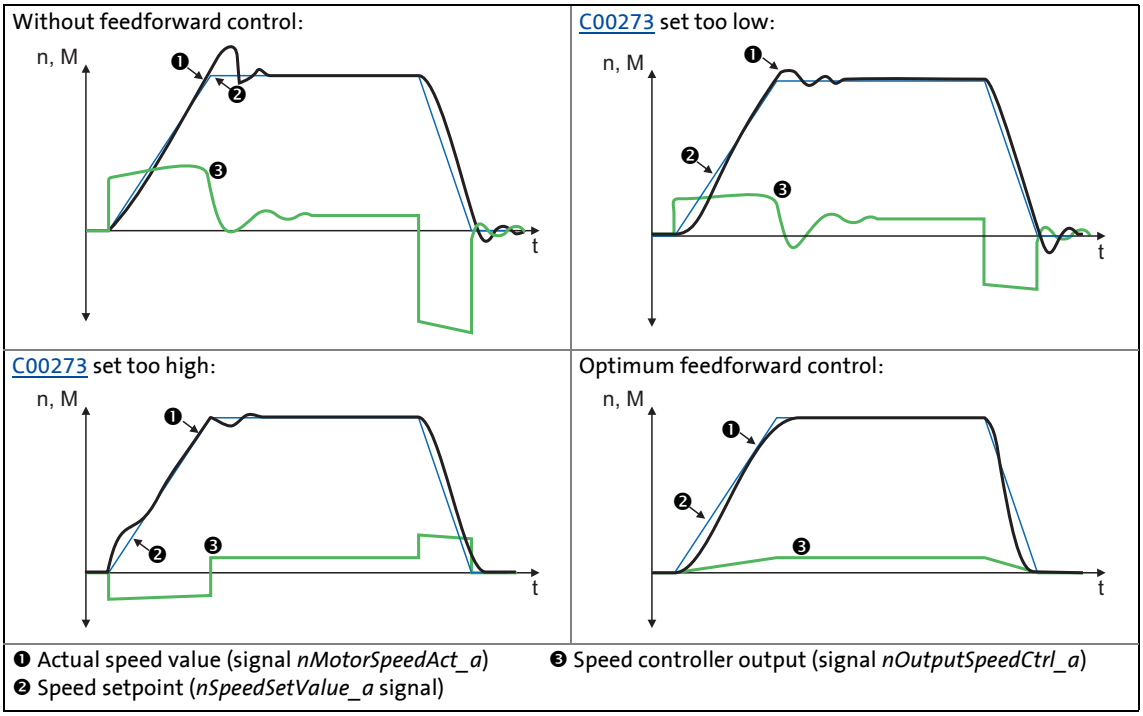
**Note!**

A function for enabling a stable operation can only be implemented to a limited extent with a reset time Ti <= 15 ms For applications with speeds above the 2-fold rated speed, we recommend a reset time Ti (C00074/2) > 15 ms.

### 5.7.4.5 Optimise response to setpoint changes and determine mass inertia

#### Optimisation at constant mass inertia

Setting the total moment of inertia under C00273 provides the optimum torque feedforward control. Depending on the application, an adjustment of the setting under C00273 may be necessary to optimise the response to position/speed setpoint changes by means of the torque feedforward control.




[5-12] Typical signal characteristics for different settings of the load moment of inertia

**How to optimise the torque feedforward control:**

1. Run a typical speed profile and record the inputs and outputs of the speed controller with the data logger.
   - Motor control variables to be recorded:
     *nSpeedSetValue_a* (speed setpoint)
     *nMotorSpeedAct_a* (actual speed value)
     *nOutputSpeedCtrl_a* (speed controller output)
2. Estimate the moment of inertia and set it in C00273 in relation to the motor end (i.e. with account being taken of the gearbox factors).
3. Repeat the data logger recording (see step 1).

   Now the data logger should show that part of the required torque is generated by the feedforward control and the speed controller output signal (*nOutputSpeedCtrl_a*) is correspondingly smaller. The resulting following error decreases.
4. Change the setting in C00273 and repeat the data logger recording until the intended response to setpoint changes is reached.
   - The optimisation could aim at the speed controller being completely relieved (see signal characteristics in Fig. [5-12]).
5. Save the parameter set (device command: C00002/11).

**Optimisation at variable mass inertia**

From version V12.00.00, mass inertia that changes during the process (e.g. a reel) can be taken into account when optimising the response to setpoint changes.

How to proceed:

1. In C00273 the known constant total moment of inertia (motor, gearbox, shaft, etc.) must be set or determined according to previous instructions ("How to optimise ...").
   - The determination requires travelling the typical speed profile without variable mass inertia (e.g. reels).
2. At the LS_MotorInterface SB, the *nInertiaAdapt_a* process signal must be interconnected in a way which ensures that a value of "100 %" is applied at this input.
3. In C00919/1 set the known maximum value of the variable moment of inertia or determine the value according to previous instructions ("How to optimise ...").
   - The determination requires travelling the typical speed profile including variable mass inertia (e.g. maximum reels).
4. The *nInertiaAdapt_a* process signal can be used during the process to dynamically control the percentage of the variable moment of inertia set in C00919/1 which is to be considered for setpoint feedforward control.

   Example :
   - If there is no variable moment of inertia (e.g. no reel), the *nInertiaAdapt_a* process signal must be set to "0 %".
   - If the maximally variable moment of inertia is available (e.g. maximum reels), the *nInertiaAdapt_a* process signal must be set to "100 %".



**Tip!**

Via the *nTorqueSetValue_a* process signal at the LS_MotorInterface SB, any differential signal can be defined for torque feedforward control. First the speed controller, then this additive torque is connected which is hence not derived from the differential change in speed setpoint.

**Other functions for differential setpoint feedforward control**

From version V12.00.00 onwards, the following additional functions are available for differential setpoint feedforward control (torque feedforward control):

- In C00653/1, the sensitivity of setpoint feedforward control can be adapted.
- In C00654/1, alternatively to the *nSpeedSetValue_a* process signal, the new *nSpeedSetValueInertia_a* process signal for the setpoint feedforward control can be selected at the LS_MotorInterface SB. Via the *nSpeedSetValueInertia_a* process signal, an optional input value (e.g. setpoint of the position or process controller) for the torque feedforward control can be specified.
- For *bTorqueModeOn* = TRUE, the setpoint feedforward control is added to the torque setpoint *nTorqueSetValue_a*. In this way, feedforward control of torque is also possible for torque-controlled operation (e.g. for winder applications).




**Stop!**

In the Lenze setting of C00654/1, the speed setpoint of the speed controller (*nSpeedSetValue_a*) is used for the torque feedforward control which is why it is also called "differential setpoint feedforward control". A very sharp change of the speed setpoint at the speed controller thus causes a strong torque impulse at the machine!

A very sharp change of the speed setpoint of the speed controller occurs if a control creates the setpoint ramp itself and the speed setpoint is only written every 20 ms to the drive. (In this case, the speed setpoint is changed every 20 ms.)

Deactivate the speed feedforward control for the speed setpoint in C00654/1 if the speed setpoint changes very sharply or the mass inertia is unknown (e.g. in case of hoists)!

### 5.7.4.6 Slip calculation from motor equivalent circuit diagram data

This function extension is available from version 02.00.00!

In order to achieve a better speed stability and torque accuracy, the slip calculation can be either derived from the motor nameplate data (e.g. rated motor speed) or the motor equivalent circuit diagram data (stator resistance, rotor resistance etc.).

The data to be used for sensorless vector control is selected via bit 0 in C02879/1:

| Setting | | Info |
|---|---|---|
| Bit 0 | SLVC | In case of sensorless vector control:<br>• "0" ≡ Slip calculation from motor nameplate data (Lenze setting)<br>• "1" ≡ Slip calculation from motor equivalent circuit diagram data |
| Bit 1 | SC_ASM | In case of servo control for asynchronous motors:<br>• "0" ≡ Slip calculation from motor nameplate data<br>• "1" ≡ Slip calculation from motor equivalent circuit diagram data (Lenze setting) |
| Bit 2 ... 7 | Reserved | |

**Note!**

In order that the slip can be calculated from the motor equivalent circuit diagram data, the equivalent circuit data (stator resistance, rotor resistance etc.) must be known as exactly as possible.

- Selecting a motor in the »Engineer« motor catalogue loads the exact motor equivalent circuit diagram data.
- When the motor nameplate data is entered manually and the motor equivalent circuit diagram data is then detected via the motor parameter identification, the "extended identification" (C02867/1 = 2) must be used. ▸ Automatic motor data identification (📖 145)

In the slip calculation from the motor equivalent circuit diagram data, the slip compensation (C00021) has no influence anymore.

### 5.7.4.7 Optimising field feedforward control and torque feedforward control

At the start of an acceleration process and at the end of a deceleration process, the field current (*nReaktCurrentId_a*) may oscillate.

- Especially when accelerating via small acceleration ramps, these oscillations become obvious by the speed being unable to follow the setpoint or speed drops during acceleration.
- Oscillating can be reduced by increasing the gain of the field current controller in C00985. A small increase of the cross current controller gain in C00986 can further improve the starting performance for small acceleration ramps.

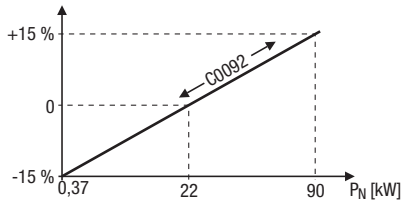
In the field weakening range, the cross current (*nEffCurrentIq_a*) may oscillate when an acceleration process starts or a deceleration process ends.

- In case of some motors, this cross current oscillation (*nEffCurrentIq_a*) can also occur in the motor setting range/field weakening range transitions which may also cause an overcurrent interruption.
- These oscillations can be reduced by again increasing the cross current controller gain slightly in C00986.

**Note!**

The setting of C00985 and C00986 reduces the torque setting range.

### 5.7.5 Remedies for undesired drive behaviour

| Drive behaviour | Remedy |
|---|---|
| Deviation between no-load current and magnetising current or bad speed or torque accuracy. | Adapt the motor magnetising inductance (C00092) for no-load operation.<br>• If the no-load current is greater than the magnetising current (C00095) at 0.5-fold rated motor speed, the magnetising inductance must be reduced until the no-load current and the magnetising current have the same values.<br>• Otherwise, the magnetising inductance must be increased.<br>Tendency of the correction of C00092:<br>  <br>PN: Rated motor power |
| Insufficient speed constancy at high load: Setpoint and motor speed are not proportional anymore.<br>**Caution:** Overcompensation of the settings mentioned under " Remedy" may result in unstable behaviour! | Via the slip compensation (C00021), the speed stability under high loads can be affected:<br>• If $n_{act} > n_{slip}$, reduce the value in C00021<br>• If $n_{act} < n_{slip}$, increase the value in C00021 |
| Unstable control with higher speeds. | • Check the setting of the magnetising inductance (C00092) by comparing the current consumption in no-load operation with the rated magnetising current (C00095).<br>• Optimise oscillation damping (C00234). |
| "Short circuit" (oC1) or "Clamp operation active" (oC11) error messages at short acceleration time (C00012) in proportion to the load (inverter cannot follow the dynamic processes). | • Increase the gain of the torque controller (C00073/2).<br>• Reduce the reset time of the torque controller (C00074/2).<br>• Increase the acceleration (C00012)/deceleration ( C00013) time. |
| Mechanical resonance at certain speeds. | The L_NSet_1 function block masks out those speed ranges that include resonance. |
| Speed variations in no-load operation for speeds > 1/3 rated speed. | Minimise speed oscillations with oscillation damping (C00234). |
| Drive runs unstable. | Check set motor data (nameplate data and equivalent circuit diagram data).<br>▸ Motor selection/Motor data (📖 139) |
| Setpoint speed and actual speed differ strongly. | |
| The torque required is not generated at standstill. | Increase motor magnetising current (C00095). |
| Current overshoots occur when heavy loads are accelerated from standstill (OC1 or OC11 error). | |
| The machine runs uneven. | |

## 5.8 Sensorless control for synchronous motors (SLPSM)

The sensorless control for synchronous motors is based on a decoupled and separated control of the torque-creating and field-creating current share of synchronous motors. In contrast to the servo control, the actual speed value and the rotor position are reconstructed via a motor model.

### Stop!

- When the maximum output frequency is limited to ± 599 Hz, the devices are not subject to the export restrictions of the "EC-Dual-Use Regulation" - EC 428/2009. This applies to the devices supplied from the middle of the year 2015. The maximum possible output frequency is given on the nameplate.
  - For certain applications, the devices can be supplied with the previous maximum output frequency of ± 1000 Hz. If required, get in touch with your Lenze contact person.
- The sensorless control for synchronous motors is only possible up to a maximum output frequency of 650 Hz (from version 14.00.00: 1000 Hz)!
  - Depending on the number of motor pole pairs, the reference speed (C00011) may only be selected that high that the output frequency displayed in C00059 is less than or equal to 650 Hz (from version 14.00.00: 1000 Hz).
- We recommend to select a power-adapted combination of inverter and motor.
- The Lenze setting permits the operation of a power-adapted motor. Optimal operation is only possible if either:
  - the motor is selected via the »Engineer« motor catalogue,
  - the motor nameplate data are entered and motor parameter identification is carried out afterwards
    *- or -*
  - the nameplate data and equivalent circuit data of the motor (motor leakage inductance and motor stator resistance) are entered manually.
- When you enter the motor nameplate data, take into account the phase connection implemented for the motor (star or delta connection). Only enter the data applying to the selected connection type.
- In order to protect the motor (e.g. from demagnetisation) we recommend setting the ultimate motor current in C00939. This ensures motor protection even with an unstable operation. ▸ Maximum current monitoring (📖 311)
- Controller enable is only possible if the motor is at standstill.
  - When the controller is enabled, a jerk may occur due to an angle jump since the rotor displacement angle is not known after controller enable. For some applications, this jerk in the machine is not acceptable.
  - From version 02.00.00, the rotor displacement angle is identified with every controller enable in the Lenze setting, and thus a jerk in the machine after controller enable can be avoided. ▸ Pole position identification without motion (📖 372)
  - A flying restart circuit for synchronising to rotating motors is in preparation.
- The injection of a constant current may cause an unwanted heating of the motor at controlled operation.
  - We recommend the use of a temperature feedback via KTY, PTC or thermal contact.
    ▸ Motor temperature monitoring (KTY) (📖 381)
    ▸ Motor temperature monitoring (PTC) (📖 304)

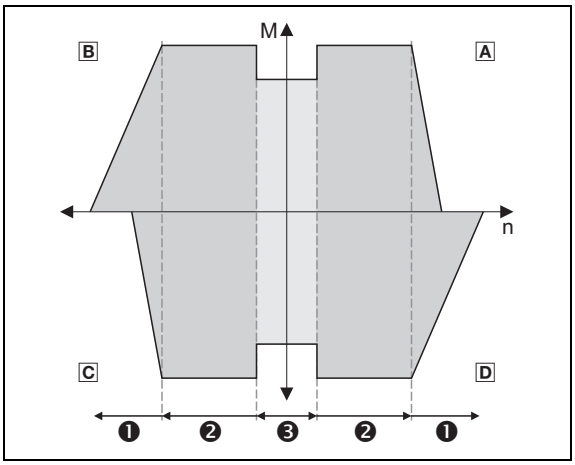
**Note!**

Currently, the sensorless control does not contain a flying restart function that enables a synchronisation of the inverter to a rotating machine.

- Thus, we recommend taking measures for preventing overvoltages at operation in generator mode (e.g. brake resistor).
- By any means, the delay time for the "DC-bus overvoltage" error trigger in C00601/1 must be set to 0 s.

The motor model-based speed monitoring requires a rotating machine. Thus, the operational performance of the sensorless control for synchronous motors is divided into two categories:

1. Open-loop controlled operation ($|n_{setpoint}| < n_{C00996}$)
   - In the range of low speeds, the speed of a synchronous motor is not possible. Thus, only an adjustable and constant current is injected that enables an acceleration.
2. Closed-loop controlled operation ($|n_{setpoint}| > n_{C00996}$)
   - In this range, the rotor flux position and the speed are reconstructed via an observer. The control is carried out field-oriented. Only the current is injected that is needed for the required torque.




Ⓐ Operation in motor mode (CW rotation)
Ⓑ Operation in generator mode (CCW rotation)
Ⓒ Operation in motor mode (CCW rotation)
Ⓓ Operation in generator mode (CW rotation)

❶ Field weakening (in preparation)
❷ Closed-loop controlled operation
❸ Open-loop controlled operation

[5-13] Operating ranges of the sensorless control for synchronous motors

The sensorless control for synchronous motors has similar advantages for the closed-loop controlled operating range and the servo control (SC) for synchronous motors. Compared to asynchronous motors, there are the following advantages:

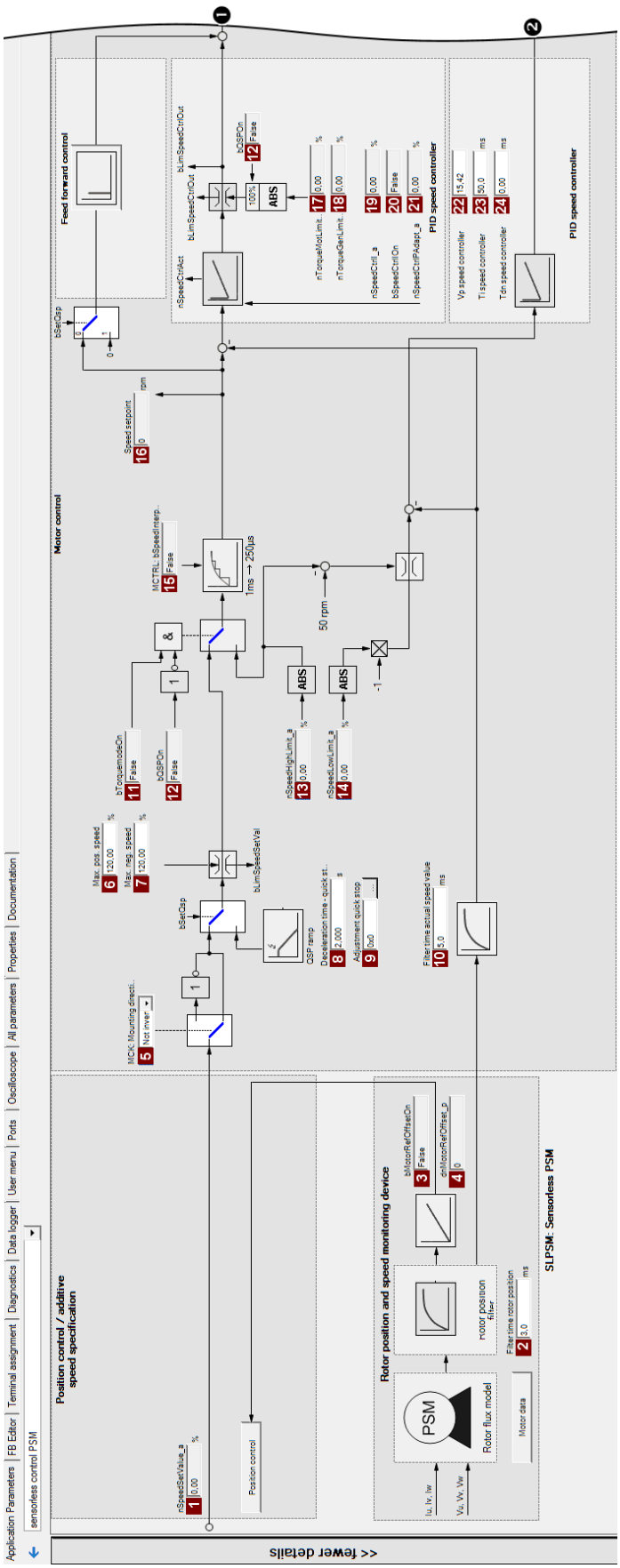
- Higher power density of the motor
- Higher efficiency
- Limitation of the maximum torque in motor mode and generator mode in closed-loop operating range
- Implementation of simple positioning

### 5.8.1 Parameterisation dialog/signal flow

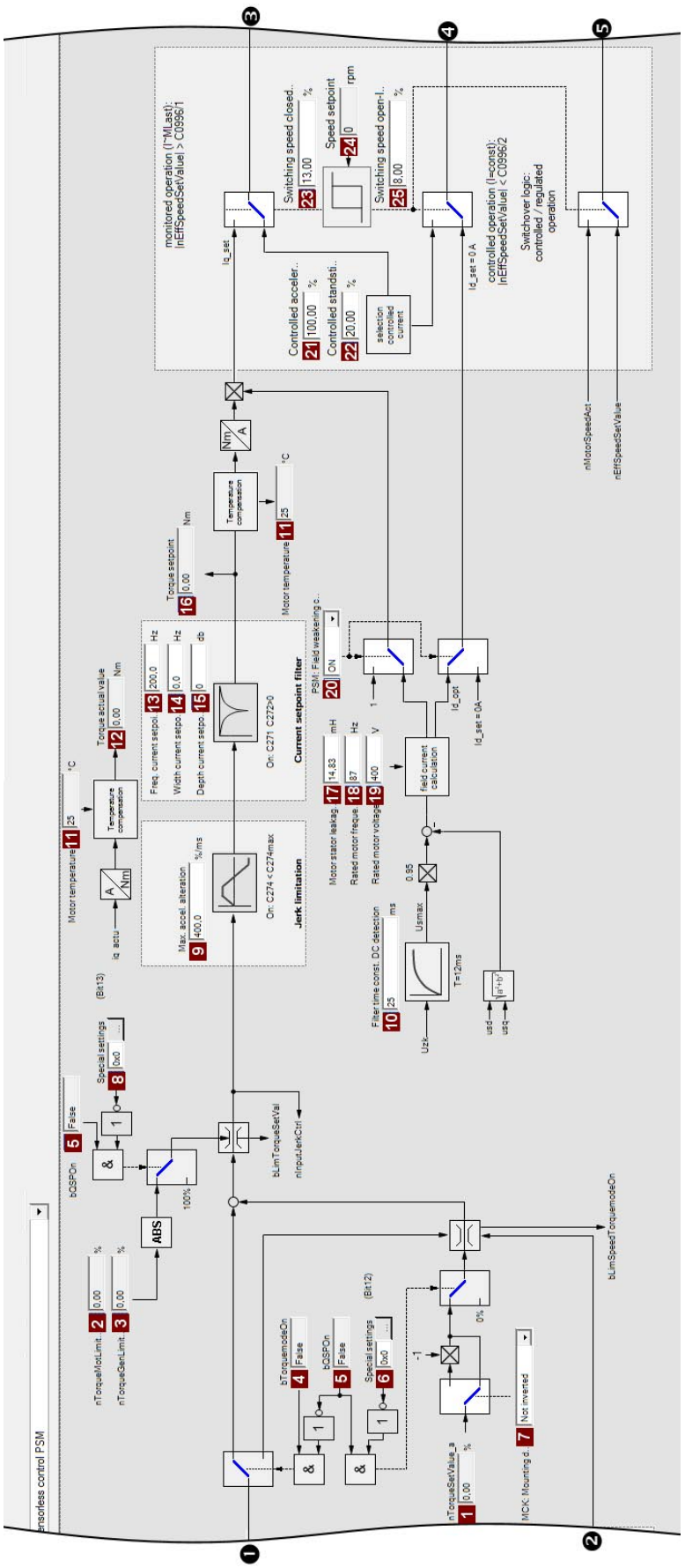


**Proceed as follows to open the dialog for parameterising the motor control:**

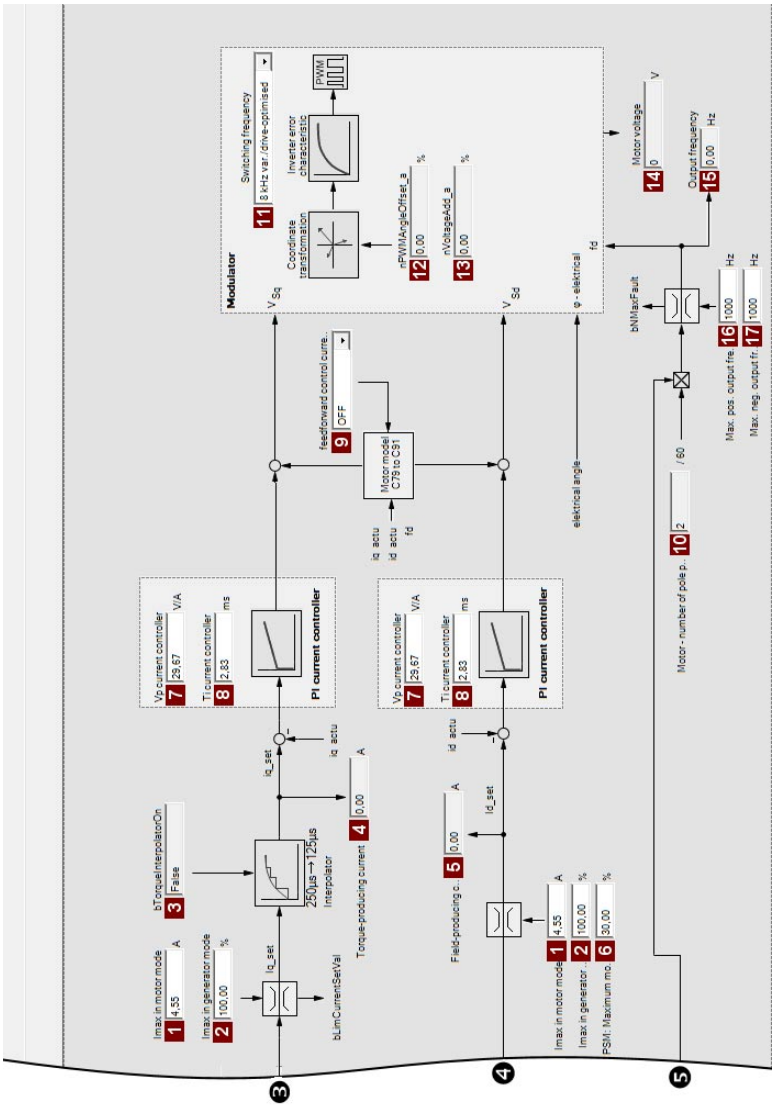
1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Select the motor control "3: SLPSM: Sensorless PSM" from the *Overview* dialog level in the **Motor control** list field:
4. Click the **Motor control servo SLPSM** button to change to the *Overview → Motor control vector* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed.




| | Parameters | Info |
|---|---|---|
| 1 | C00830/22 | Speed setpoint |
| 2 | C00998/1 | SLPSM: Filter time rotor position |
| 3 | C00833/68 | MCTRL: bMotorRefOffsetOn |
| 4 | C00834/6 | MCTRL: dnMotorRefOffset_p |
| 5 | C01206/1 | MCK: Mounting direction: Motor |
| 6 | C00909/1 | Max. pos. speed |
| 7 | C00909/2 | Max. neg. speed |
| 8 | C00105 | Decel. time - quick stop |
| 9 | C00104/1 | Quick stop setting |
| 10 | C00998/2 | SLPSM: Filter time actual speed value |
| 11 | C00833/30 | Actual speed value |
| 12 | C00833/33 | bQSPOn |
| 13 | C00830/88 | MCTRL: nSpeedHighLimit_a |
| 14 | C00830/23 | MCTRL: nSpeedLowLimit_a |
| 15 | C00833/28 | MCTRL: bSpeedInterp |
| 16 | C00050 | Speed setpoint |
| 17 | C00830/29 | Limitation of torque in motor mode |
| 18 | C00830/28 | Limitation of torque in generator mode |
| 19 | C00830/24 | MCTRL: nSpeedCtrlI_a |
| 20 | C00833/31 | MCTRL: bSpeedCtrlIOn |
| 21 | C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| 22 | C00070/2 | SLPSM: Vp speed controller |
| 23 | C00071/2 | SLPSM: Ti speed controller |
| 24 | C00072 | SC: Tdn speed controller |




| | Parameters | Info |
|---|---|---|
| 1 | C00830/27 | MCTRL: nTorqueSetValue_a |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00833/30 | bTorquemodeOn |
| 5 | C00833/33 | bQSPOn |
| 6 | C02865/1 | Special settings (bit 12) |
| 7 | C01206/1 | MCK: Mounting direction |
| 8 | C02865/1 | Sondereinstellungen (Bit 13) |
| 9 | C00274 | SC: Max. change in acceleration |
| 10 | C00280 | SC: Filter time const. DC detection |
| 11 | C00063/1 | Motor temperature |
| 12 | C00056/2 | Actual torque |
| 13 | C00270 | SC: Freq. current setpoint filter |
| 14 | C00271 | SC: Current setpoint filter width |
| 15 | C00272 | SC: Current setpoint filter depth |
| 16 | C00056/1 | Torque setpoint |
| 17 | C00085 | Motor stator leakage inductance |
| 18 | C00089 | Rated motor frequency |
| 19 | C00090 | Rated motor voltage |
| 20 | C00079/4 | Field weakening |
| 21 | C00995/1 | SLPSM: Open-loop controlled acceleration current |
| 22 | C00995/2 | SLPSM: Open-loop controlled standstill current |
| 23 | C00996/1 | SLPSM: Closed-loop controlled switching speed |
| 24 | C00050 | Speed setpoint |
| 25 | C00996/2 | SLPSM: Open-loop controlled switching speed |




| | Parameters | Info |
|---|---|---|
| 1 | C00022 | Imax in motor mode |
| 2 | C00023 | Imax in generator mode |
| 3 | C00833/29 | MCTRL: bTorqueInterpolatorOn |
| 4 | C00937/2 | Torque-producing current |
| 5 | C00937/1 | Maximally effective field-producing motor current |
| 6 | C00938 | Limitation of the field-producing motor current |
| 7 | C00075 | Vp current controller |
| 8 | C00076 | Ti current controller |
| 9 | C00079/1 | SC: Current controller - feedforward control |
| 10 | C00969/1 | Motor - number of pole pairs |

| | Parameters | Info |
|---|---|---|
| 11 | C00018 | Switching frequency |
| 12 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 13 | C00830/31 | MCTRL: nVoltageAdd_a |
| 14 | C00052 | Motor voltage |
| 15 | C00058 | Output frequency |
| 16 | C00910/1 | Max. pos. output frequency |
| 17 | C00910/2 | Max. neg. output frequency |

### 5.8.2 Types of control

Sensorless control for synchronous motors can only be executed in the "Speed control with torque limitation" (*bTorquemodeOn* = FALSE) mode.

#### Speed control with torque limitation

A speed setpoint is selected and the drive system is operated in a speed-controlled manner. For adapting the operational performance, the overload in the drive train can be limited:

- The torque is limited via the torque setpoint.
- The torque setpoint is identical to the value at the output of the speed controller, *nOutputSpeedCtrl*.
- To avoid overload in the drive train, the torque in motor mode can be limited via the *nTorqueMotLimit_a* process input signal, and the torque in generator mode can be limited via the *nTorqueGenLimit_a* process input signal:

| Designator<br>DIS code \| data type | Information/possible settings |
|---|---|
| nTorqueMotLimit_a<br>C00830/29 \| INT | Torque limitation in motor mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: 0 ... +199.99 %<br>• If keypad control is performed: Parameterisable via C00728/1. |
| nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in generator mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: -199.99 ... 0 %<br>• If keypad control is performed: Parameterisable via C00728/2. |

**Stop!**

Torque limitation is only active in the closed-loop controlled operation ($|n_{Setpoint}| > n_{C00996}$)!

- It must be prevented that the actual speed value is braked into the non-observable area due to the torque limitation!

**Note!**

To avoid instabilities during operation, the torque limit values are internally processed as absolute values.

### 5.8.3 Basic settings

The following "Initial commissioning steps" must be performed to commission the sensorless control for synchronous motors:

<table>
<tr><th colspan="3">Initial commissioning steps</th></tr>
<tr><td>1.</td><td colspan="2">Select motor control:<br>C00006 = "3: SLPSM: Sensorless PSM"</td></tr>
<tr><td>2nd</td><td colspan="2">Set the motor selection/motor data<br>• When selecting and parameterising the motor, the motor nameplate data and the equivalent circuit diagram data are relevant. Detailed information can be found in the "Motor selection/Motor data" chapter. (📖 139)<br>Depending on the motor manufacturer, proceed as follows:</td></tr>
<tr><td></td><td>Lenze motor:<br>Selecting a motor from the motor catalogue in the »Engineer«<br>- or -<br>1. Set the motor nameplate data<br>2. Automatic motor data identification</td><td>Third party manufacturer's motor:<br>1. Set the motor nameplate data<br>2. Automatic motor data identification or set known equivalent circuit diagram manually:<br>C00084: Motor stator resistance<br>C00085: Motor stator leakage inductance</td></tr>
<tr><td>3.</td><td colspan="2">Set speed switching thresholds between open-loop and closed-loop controlled operation:<br>• Set transition speed from closed-loop to open-loop operation in C00996/1 in [%] with regard to the rated motor speed (C00087).<br>• Set transition speed from closed-loop to open-loop operation in C00996/2 in [%] with regard to the rated motor speed (C00087).<br>Tip!<br>• With voltage-adjusted motors, a speed switching threshold of 10 % is recommended.<br>• As a rule of thumb, the speed switching threshold should be selected as follows:<br>$C00996/1...2\ [\%] = \frac{U_{Rated,\ motor}[V]}{U_{Rated,\ FI}[V]} \cdot 10$</td></tr>
<tr><td>4.</td><td colspan="2">Set open-loop accelerating current in C00995/1 in [%] with regard to the rated motor current (C00088).<br>• This value defines the height of the current that is injected during the acceleration process.<br>• The accelerating current must be dimensioned so that the required torque in the lower speed range can always be reached (acceleration torque + load torque):<br>$C00995/1\ [\%] = \frac{M_{Meax}[Nm]}{M_{Rated}[Nm]} \cdot I_{Rated,\ motor}[A] \cdot 1.3$</td></tr>
<tr><td>5th</td><td colspan="2">Set open-loop steady-state current in C00995/2 in [%] with regard to the rated motor current (C00088).<br>• This value defines the height of the current for processes without acceleration (e.g. standstill or constant setpoint speed ).</td></tr>
<tr><td>6.</td><td colspan="2">For improving the operating characteristics:<br>If required, adapt the filter time for reconstructing the rotor position and the actual speed value through the motor model in C00998/1 and C00998/2.<br>• We recommend using the Lenze setting:<br>Filter time rotor position (C00998/1) = 3 ms<br>Filter time actual speed value (C00998/2) = 5 ms<br>• Deviant from this, the following value range can be used:<br>Filter time rotor position (C00998/1) = 2 ... 5 ms<br>Filter time actual speed value (C00998/2) = 3 ... 8 ms</td></tr>
<tr><td>6.</td><td colspan="2">For protecting the motor from demagnetisation:<br>Set the ultimate current in C00939.</td></tr>
</table>

**Note!**

The Lenze settings of the current controller are predefined for a power-adapted motor. For an optimal drive behaviour of a synchronous motor, we recommend to adapt the controller settings.



**Tip!**

Information on the optimisation of the control mode and the adaptation to the real application is provided in the "Optimising the control mode" chapter.

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 263)

### 5.8.4 Optimising the control mode

**Note!**

From version 12.00.00:

- Following successful motor parameter identification, the current controller parameters (C00075, C00076) are calculated automatically.
  - If these parameters are not to be calculated, bit 4 of C02865/1 must be set to "1".
- Following successful motor parameter identification, the speed controller parameters (C00070/3, C00071/3) can be calculated automatically.
  - If these parameters are to be calculated, bit 6 of C02865/1 must be set to "5".
- Following successful motor parameter identification, other controller parameters (C00011, C00022) can be calculated automatically.
  - If these parameters are to be calculated, bit 6 of C02865/1 must be set to "6".

The "optimisation steps" given in the table below serve to further optimise the control behaviour of the sensorless control for synchronous motors and adjust it to the concrete application.

- Detailed information on the individual steps can be found in the following subchapters.

Generally, the following optimisation steps are recommended:

| Optimisation steps | |
|---|---|
| 1. | Optimise current controller. (📖 223)<br>• The current controller should always be optimised if a motor of a third-party manufacturer with unknown motor data is used! |
| 2nd | Optimise speed controller. (📖 224)<br>• The setting of the speed controller must be adapted depending on the mechanical path. |
| 3. | Optimise response to setpoint changes and determine mass inertia. (📖 227)<br>• For an optimal reference behaviour, the total moment of inertia can be used to make a feedforward control of the speed setpoint. |

**Note!**

**Current setpoint filter (band-stop filter) / jerk limitation**

The use of the functions is only recommended in exceptional cases.

The functions are described in chapter "Servo control (SC)":

- Setting the current setpoint filter (band-stop filter) (📖 251)
- Adapting the max. acceleration change (jerk limitation) (📖 252)

### 5.8.4.1 Optimise current controller

**Note!**

An optimisation of the current controller should generally be carried out unless a power-adapted standard motor is used or the motor has been selected from the motor catalogue of the »Engineer«!

An optimisation of the current controller is sensible since the two control parameters gain (C00075) and reset time (C00076) depend on the required maximum current and the set switching frequency.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00075 | Vp current controller | 7.00 | V/A |
| C00076 | Ti current controller | 10.61 | ms |

- Gain and reset time can be calculated as per the following formulae:

$$V_p = \frac{L_{ss}[H]}{T_E[s]}$$

$$T_i = \frac{L_{ss}[H]}{R_s[\Omega]}$$

$V_p$ = Current controller gain (C00075)
$T_i$ = Current controller reset time (C00076)
$L_{ss}$ = Motor stator leakage inductance (C00085)
$R_s$ = Motor stator resistance (C00084)
$T_E$ = Equivalent time constant (= 500 µs)

### 5.8.4.2 Optimise speed controller

The speed controller is in the form of a PID controller with an additional differential speed-setpoint gain. For optimum behaviour, the PID speed controller has to be optimised and the overall mass inertia of the drive train has to be determined.

- In the Lenze setting, the configuration of the speed controller provides robustness and moderate dynamics.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00070/3 | SLPSM: Vp speed controller | 3.00 | |
| C00071/3 | SLPSM: Ti speed controller | 100.0 | ms |
| C00072 | SC: Tdn speed controller | 0.00 | ms |

#### Speed controller gain Vp

The gain Vp (C00070/3) of the speed controller is defined in a scaled representation which enables a comparable parameterisation almost independent of the power of the motor or inverter. Here, the speed input difference of the controller is scaled to the rated motor speed whereas the output torque refers to the rated motor torque. A gain of 10 means that a speed difference of 1 % is gained through the P component with 10 % torque.

If the rated data of the motor and the mass inertia of the drive system are known, we recommend the following setting:

$$V_P \approx 0.2 \ldots 0.5 \cdot \frac{T_M[s]}{0.01[s]}$$

$$T_M[s] = \frac{2 \cdot \pi \cdot n_N[rpm]}{M_N[Nm] \cdot 60} \cdot J_{Drive,\,total}[kgm^2]$$

$$M_N[Nm] = \frac{P_N[W] \cdot 60}{2 \cdot \pi \cdot n_N[rpm]}$$

$V_P$ = Gain of the speed controller (C00070/3)
$T_M$ = Time constant for the acceleration of the motor
$M_N$ = Rated motor torque
$n_N$ = Rated motor speed
$J_{drive,\,total}$ = Total moment of inertia of the drive

[5-14] Recommendation for the setting of the gain of the speed controller

If the mass inertia of the drive is unknown, the optimisation can be achieved as follows:

1. Specify speed setpoint.
   - A small speed just above the switching threshold is recommended in the closed-loop controlled operation.
2. Increase Vp (C00070/3) until the drive starts to oscillate (observe engine noise).
3. Reduce Vp (C00070/3) until the drive runs stable again.
4. Reduce Vp (C00070/3) to approx. half the value.
5. Afterwards check results of the optimisation in the entire speed range (one-time passing through of the speed range).

**Tip!**

Values recommended by Lenze for the setting of the (proportional) gain:
- For drive systems without feedback: Vp = 2 ... 8
- For drive systems with a good disturbance behaviour: Vp > 6

#### Speed controller reset time Ti

Apart from setting the P component, C00071/3 provides the possibility to take influence on the I component of the PI controller.

If the mass inertia of the drive is unknown, the optimisation can be achieved as follows:

1. Specify speed setpoint.
2. Reduce Ti (C00071/3) until the drive starts to oscillate (observe engine noise).
3. Increase Ti (C00071/3) until the drive runs stable again.
4. Increase Ti (C00071/3) to approx. twice the value.

**Tip!**

Value range recommended by Lenze for the setting of the reset time:

Ti = 20 ms ... 150 ms

#### Using the ramp response for setting the speed controller

If the mechanical components cannot be operated at the stability limit, the ramp response can also be used for setting the speed controller.

**Stop!**

If the controller parameters are preset unfavourably, the control can tend to heavy overshoots up to instability!
- Following and speed errors can adopt very high values.
- If the mechanics are sensitive, the corresponding monitoring functions are to be activated.

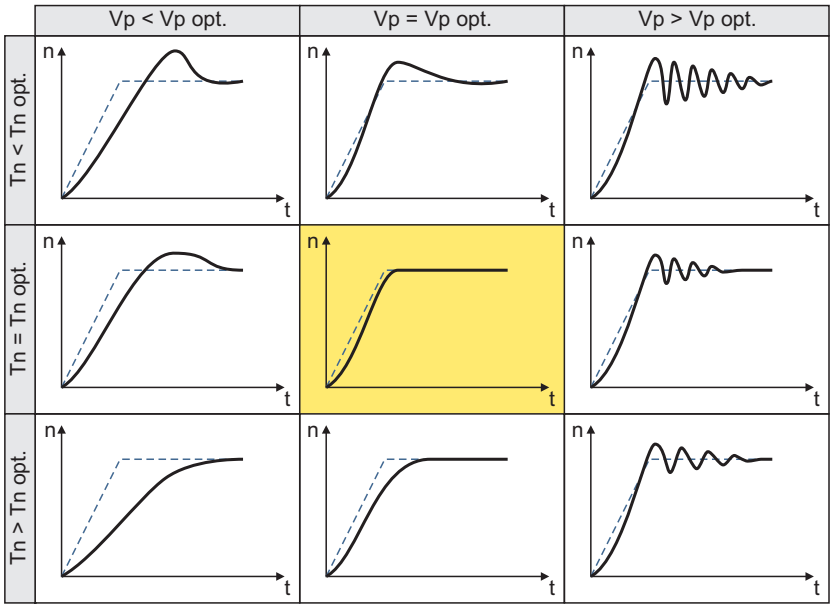
**Note!**

For an optimal setting, we recommend to determine the mass inertia (optimal response to setpoint changes) first.

▸ Optimise response to setpoint changes and determine mass inertia (📖 227)

**How to optimise the speed controller setting by means of the ramp response:**

1. Run a typical speed profile and record the ramp response of the speed using the data logger.
   - Motor control variables to be recorded:
     *nSpeedSetValue_a* (speed setpoint)
     *nMotorSpeedAct_a* (actual speed value)
2. Evaluate the ramp response:

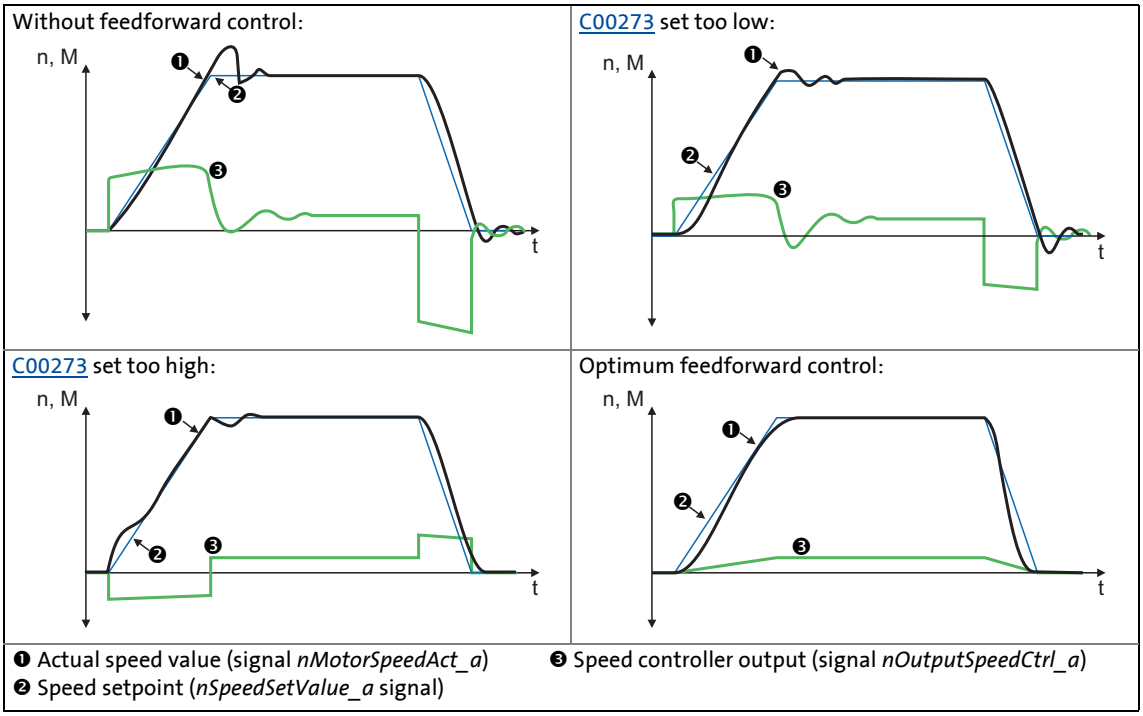



   - Solid line = ramp response (actual speed value)
   - Dash line = speed setpoint
3. Change gain Vp in C00070/3 and reset time Tn in C00071/3.
4. Repeat steps 1 ... 3 until the optimum ramp response is reached.

### 5.8.4.3 Optimise response to setpoint changes and determine mass inertia

#### Optimisation at constant mass inertia

Setting the total moment of inertia under C00273 provides the optimum torque feedforward control. Depending on the application, an adjustment of the setting under C00273 may be necessary to optimise the response to position/speed setpoint changes by means of the torque feedforward control.




[5-15] Typical signal characteristics for different settings of the load moment of inertia

**How to optimise the torque feedforward control:**

1. Run a typical speed profile and record the inputs and outputs of the speed controller with the data logger.
   - Motor control variables to be recorded:
     *nSpeedSetValue_a* (speed setpoint)
     *nMotorSpeedAct_a* (actual speed value)
     *nOutputSpeedCtrl_a* (speed controller output)
2. Estimate the moment of inertia and set it in C00273 in relation to the motor end (i.e. with account being taken of the gearbox factors).
3. Repeat the data logger recording (see step 1).

   Now the data logger should show that part of the required torque is generated by the feedforward control and the speed controller output signal (*nOutputSpeedCtrl_a*) is correspondingly smaller. The resulting following error decreases.
4. Change the setting in C00273 and repeat the data logger recording until the intended response to setpoint changes is reached.
   - The optimisation could aim at the speed controller being completely relieved (see signal characteristics in Fig. [5-15]).
5. Save the parameter set (device command: C00002/11).

**Optimisation at variable mass inertia**

From version V12.00.00, mass inertia that changes during the process (e.g. a reel) can be taken into account when optimising the response to setpoint changes.

How to proceed:

1. In C00273 the known constant total moment of inertia (motor, gearbox, shaft, etc.) must be set or determined according to previous instructions ("How to optimise ...").
   - The determination requires travelling the typical speed profile without variable mass inertia (e.g. reels).
2. At the LS_MotorInterface SB, the *nInertiaAdapt_a* process signal must be interconnected in a way which ensures that a value of "100 %" is applied at this input.
3. In C00919/1 set the known maximum value of the variable moment of inertia or determine the value according to previous instructions ("How to optimise ...").
   - The determination requires travelling the typical speed profile including variable mass inertia (e.g. maximum reels).
4. The *nInertiaAdapt_a* process signal can be used during the process to dynamically control the percentage of the variable moment of inertia set in C00919/1 which is to be considered for setpoint feedforward control.

   Example :
   - If there is no variable moment of inertia (e.g. no reel), the *nInertiaAdapt_a* process signal must be set to "0 %".
   - If the maximally variable moment of inertia is available (e.g. maximum reels), the *nInertiaAdapt_a* process signal must be set to "100 %".

**Tip!**

Via the *nTorqueSetValue_a* process signal at the LS_MotorInterface SB, any differential signal can be defined for torque feedforward control. First the speed controller, then this additive torque is connected which is hence not derived from the differential change in speed setpoint.

**Other functions for differential setpoint feedforward control**

From version V12.00.00 onwards, the following additional functions are available for differential setpoint feedforward control (torque feedforward control):

- In C00653/1, the sensitivity of setpoint feedforward control can be adapted.
- In C00654/1, alternatively to the *nSpeedSetValue_a* process signal, the new *nSpeedSetValueInertia_a* process signal for the setpoint feedforward control can be selected at the LS_MotorInterface SB. Via the *nSpeedSetValueInertia_a* process signal, an optional input value (e.g. setpoint of the position or process controller) for the torque feedforward control can be specified.
- For *bTorqueModeOn* = TRUE, the setpoint feedforward control is added to the torque setpoint *nTorqueSetValue_a*. In this way, feedforward control of torque is also possible for torque-controlled operation (e.g. for winder applications).




**Stop!**

In the Lenze setting of C00654/1, the speed setpoint of the speed controller (*nSpeedSetValue_a*) is used for the torque feedforward control which is why it is also called "differential setpoint feedforward control". A very sharp change of the speed setpoint at the speed controller thus causes a strong torque impulse at the machine!

A very sharp change of the speed setpoint of the speed controller occurs if a control creates the setpoint ramp itself and the speed setpoint is only written every 20 ms to the drive. (In this case, the speed setpoint is changed every 20 ms.)

Deactivate the speed feedforward control for the speed setpoint in C00654/1 if the speed setpoint changes very sharply or the mass inertia is unknown (e.g. in case of hoists)!

*This page has been left blank intentionally,*
*to present the following information more clearly.*

## 5.9 Servo control (SC)

Field-oriented servo control (SC) is based on a decoupled, separate control of the torque-producing and the field-producing current component. The motor control is based on a field-oriented, cascaded controller structure with feedback function and enables dynamic and stable operation in all of the four quadrants. It can be used for synchronous motors (PSM) and asynchronous motors (ASM).

### Stop!

- Servo control requires a speed feedback!
  - The speed sensor used has to be set in C00495. This setting is not made automatically with the selection of the motor from the »Engineer« motor catalogue!
  - If no speed sensor is set in C00495 and the controller is enabled, an impermissibly high motor current occurs which may destroy the motor thermally!
  - From version 14.00.00, the error response set in C00571/2 (Lenze setting: "Fault") takes place if in case of controller enable it is detected that a motor control type with feedback is set in C00006 but no speed encoder is set in C00495.
- We recommend to select a power-adapted combination of inverter and motor.
- The Lenze setting permits the operation of a power-adapted motor. Optimal operation is only possible if either:
  - the motor is selected via the »Engineer« motor catalogue,
  - the motor nameplate data are entered and motor parameter identification is carried out afterwards
    *- or -*
  - the nameplate data and equivalent circuit data of the motor are entered manually.
- When you enter the motor nameplate data, take into account the phase connection implemented for the motor (star or delta connection). Only enter the data applying to the selected connection type.

### Stop!

**V/f emergency operation**

From version 15.00.00 , it is internally switched to the encoderless V/f characteristic control in case of an encoder open circuit in order to avoid impermissible motor movements.

- In order that this "V/f emergency operation" works properly, the parameters relevant for the V/f characteristic control (base frequency, Vmin boost, slip compensation, etc.) have to be set correctly. As an alternative, a motor parameter identification can be executed as well.
- The change-over to "V/f emergency operation" is reported via bit 4 in C01000 and via the *bWirebreakUfLinearActive* status signal at the SB LS_DeviceMonitor.
- The change-over to "V/f emergency operation" can be suppressed by setting bit 8 to "1" in C02864/1.
- In case of servo control for synchronous motors (PSM), no change-over to "F/f emergency operation" takes place.

**Note!**

- For closed-loop control of a synchronous motor (PSM), the pole position of the motor must be known! ▸ Pole position identification (PPI) (📖 366)
- For closed-loop control of an asynchronous motor (ASM), the maximum current (C00022) should be higher than the magnetising current as otherwise the motor does not generate a torque.

Detailed information on the speed feedback can be found in the chapter "Encoder/feedback system". (📖 321)

Generally, the servo control offers the same advantages as the sensorless vector control (SLVC), i.e. compared to the V/f characteristic control, the servo control (SC) can be used to achieve

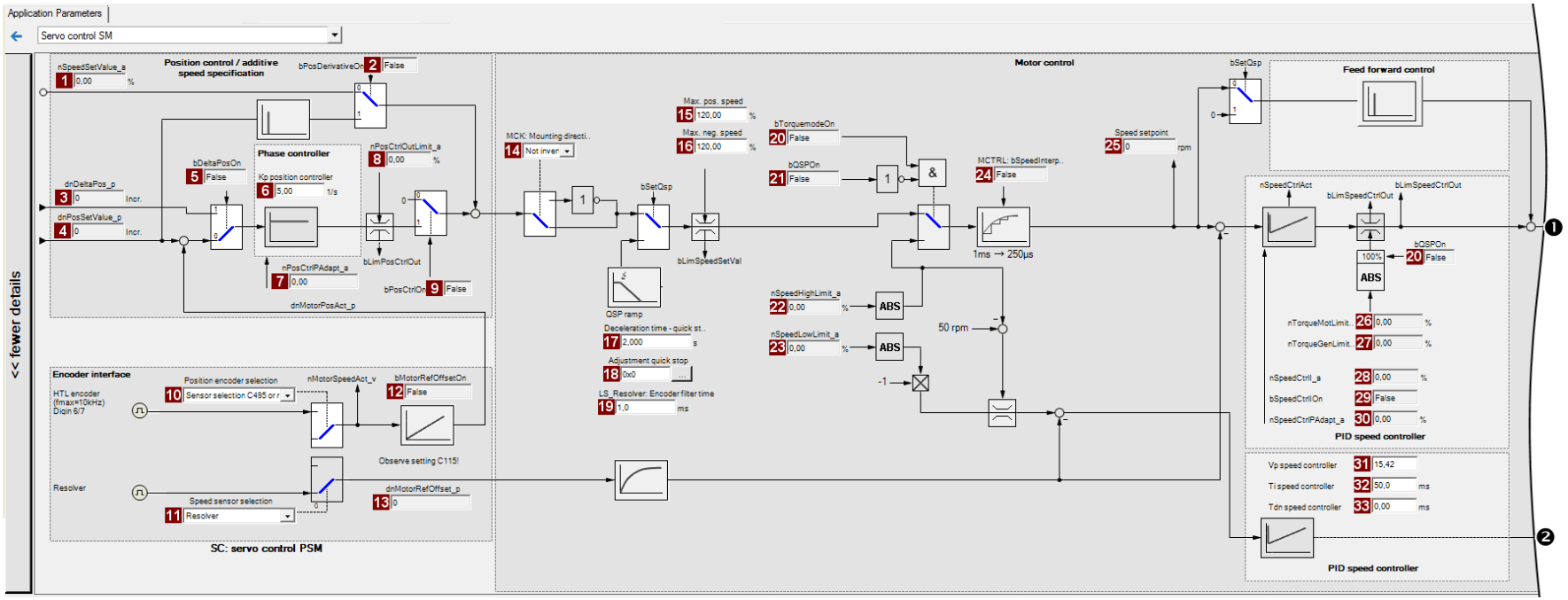
- A higher maximum torque throughout the entire speed range
- A higher speed accuracy
- A higher concentricity factor
- A higher level of efficiency
- The implementation of torque-controlled operation with speed limitation
- The limitation of the maximum torque in motor and generator mode for speed-controlled operation

### 5.9.1 Parameterisation dialog/signal flow

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Select the motor control from the *Overview* dialog level in the **Motor control** (C00006) list field:
   - "1: SC: Servo control PSM" for synchronous motor
     - or -
   - "2: SC: Servo control ASM" for asynchronous motor
4. Click the **Motor control servo** button to change to the *Overview* → *Motor control vector* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed, as shown in the following subchapter.

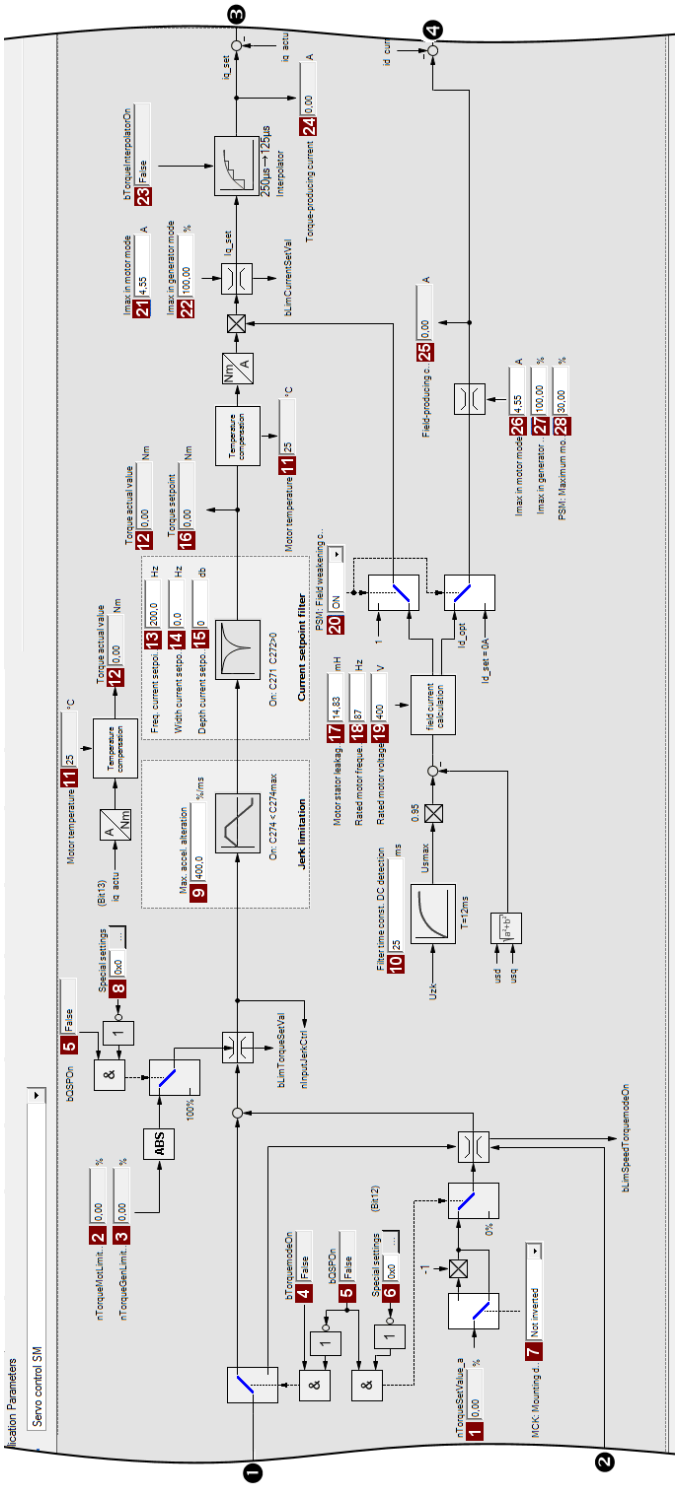
## Signal flow for servo control for synchronous motor (PSM):



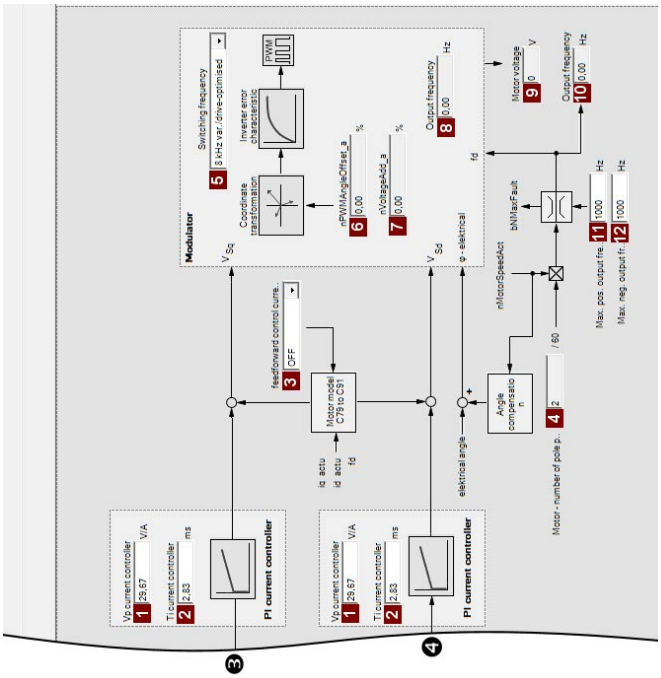

| Parameters | | Info |
|---|---|---|
| 1 | C00830/22 | Speed setpoint |
| 2 | C00833/67 | MCTRL: bPosDerivativeOn |
| 3 | C00834/4 | MCTRL: dnDeltaPos_p |
| 4 | C00834/5 | MCTRL: dnPosSetValue_p |
| 5 | C00833/35 | MCTRL: bDeltaPosOn |
| 6 | C00254 | Kp position controller |
| 7 | C00830/20 | MCTRL: nPosCtrlPAdapt_a |
| 8 | C00830/21 | MCTRL: nPosCtrlOutLimit_a |
| 9 | C00833/27 | MCTRL: bPosCtrlOn |
| 10 | C00490 | Position encoder selection |
| 11 | C00495 | Speed sensor selection |
| 12 | C00833/68 | MCTRL: bMotorRefOffsetOn |
| 13 | C00834/6 | MCTRL: dnMotorRefOffset_p |

| Parameters | | Info |
|---|---|---|
| 14 | C01206/1 | MCK: Mounting direction: Motor |
| 15 | C00909/1 | Max. pos. speed |
| 16 | C00909/2 | Max. neg. speed |
| 17 | C00105 | Decel. time - quick stop |
| 18 | C00104/1 | Quick stop setting |
| 19 | C00497/4 | LS_Resolver: Encoder filter time FreqIn67 |
| 20 | C00833/30 | bTorquemodeOn |
| 21 | C00833/33 | bQSPOn |
| 22 | C00830/88 | MCTRL: nSpeedHighLimit_a |
| 23 | C00830/23 | MCTRL: nSpeedLowLimit_a |
| 24 | C00833/28 | MCTRL: bSpeedInterp. |
| 25 | C00050 | Speed setpoint |

| Parameters | | Info |
|---|---|---|
| 26 | C00830/29 | Limitation of torque in motor mode |
| 27 | C00830/28 | Limitation of torque in generator mode |
| 28 | C00830/24 | MCTRL: nSpeedCtrlI_a |
| 29 | C00833/31 | MCTRL: bSpeedCtrlIOn |
| 30 | C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| 31 | C00070/2 | SC: Vp speed controller |
| 32 | C00071/2 | SC: Ti speed controller |
| 33 | C00072 | SC: Tdn speed controller |

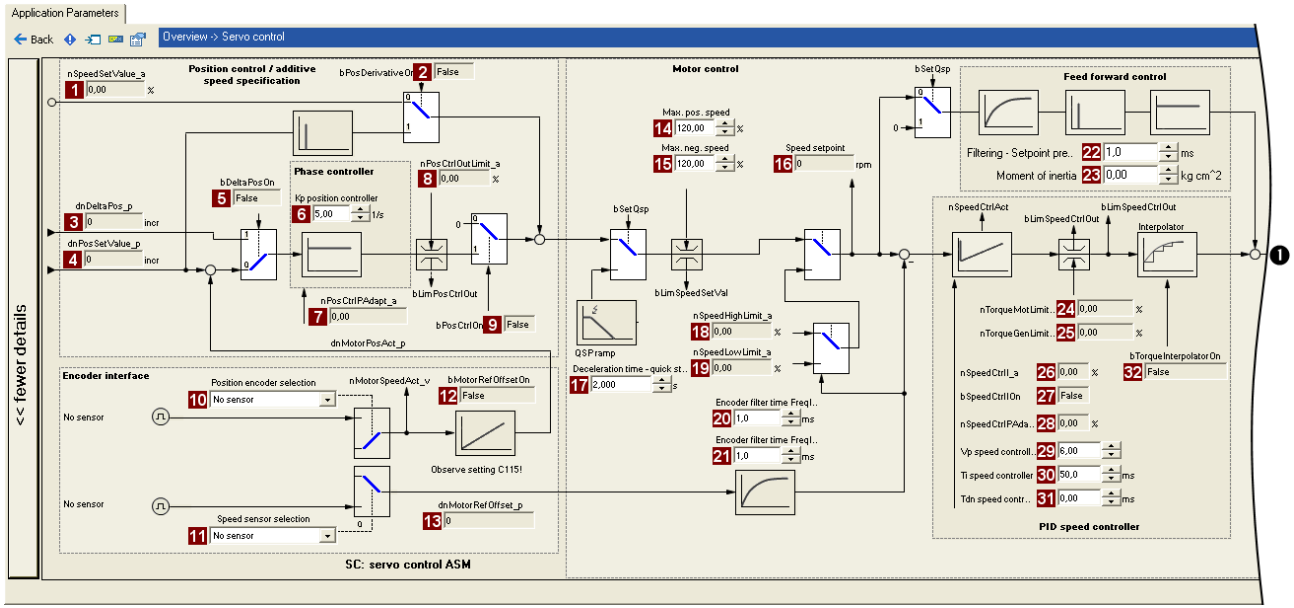


| | Parameters | Info |
|---|---|---|
| 1 | C00830/27 | MCTRL: nTorqueSetValue_a |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00833/30 | bTorquemodeOn |
| 5 | C00833/33 | bQSPOn |
| 6 | C02865/1 | Special settings (bit 12) |
| 7 | C01206/1 | MCK: Mounting direction: Motor |
| 8 | C02865/1 | Sondereinstellungen (Bit 13) |
| 9 | C00274 | SC: Max. change in acceleration |
| 10 | C00280 | SC: Filter time const. DC detection |
| 11 | C00063/1 | Motor temperature |
| 12 | C00056/2 | Actual torque |
| 13 | C00270 | SC: Freq. current setpoint filter |
| 14 | C00271 | SC: Current setpoint filter width |
| 15 | C00272 | SC: Current setpoint filter depth |
| 16 | C00056/1 | Torque setpoint |
| 17 | C00085 | Motor stator leakage inductance |
| 18 | C00089 | Rated motor frequency |
| 19 | C00090 | Rated motor voltage |
| 20 | C00079/4 | Field weakening |
| 21 | C00022 | Imax in motor mode |
| 22 | C00023 | Imax in generator mode |
| 23 | C00833/29 | MCTRL: bTorqueInterpolatorOn |
| 24 | C00937/2 | Torque-producing current |
| 25 | C00937/1 | Maximally effective field-producing motor current |
| 26 | C00022 | Imax in motor mode |
| 27 | C00023 | Imax in generator mode |
| 28 | C00938 | Limitation of the field-producing motor current |




| | Parameters | Info |
|---|---|---|
| 1 | C00075 | Vp current controller |
| 2 | C00076 | Ti current controller |
| 3 | C00079/1 | SC: Current controller - feedforward control |
| 4 | C00969/1 | Motor - number of pole pairs |
| 5 | C00018 | Switching frequency |
| 6 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 7 | C00830/31 | MCTRL: nVoltageAdd_a |
| 8 | C00058 | Output frequency |
| 9 | C00052 | Motor voltage |
| 10 | C00058 | Output frequency |
| 11 | C00910/1 | Max. pos. output frequency |
| 12 | C00910/2 | Max. neg. output frequency |

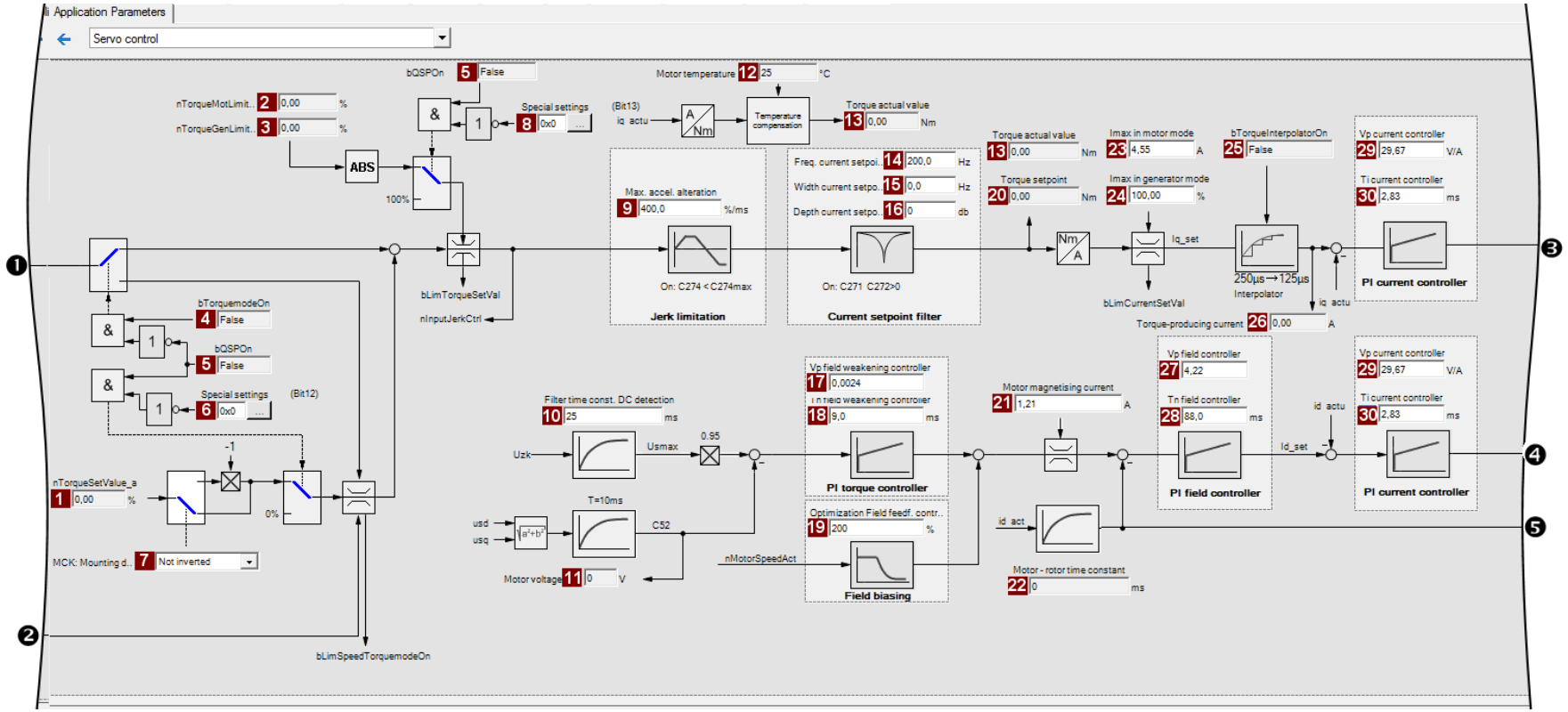
## Signal flow for servo control for asynchronous motor (ASM):




| Parameters | | Info |
|---|---|---|
| 1 | C00830/22 | Speed setpoint |
| 2 | C00833/67 | MCTRL: bPosDerivativeOn |
| 3 | C00834/4 | MCTRL: dnDeltaPos_p |
| 4 | C00834/5 | MCTRL: dnPosSetValue_p |
| 5 | C00833/35 | MCTRL: bDeltaPosOn |
| 6 | C00254 | Kp position controller |
| 7 | C00830/20 | MCTRL: nPosCtrlPAdapt_a |
| 8 | C00830/21 | MCTRL: nPosCtrlOutLimit_a |
| 9 | C00833/27 | MCTRL: bPosCtrlOn |
| 10 | C00490 | Position encoder selection |
| 11 | C00495 | Speed sensor selection |
| 12 | C00833/68 | MCTRL: bMotorRefOffsetOn |
| 13 | C00834/6 | MCTRL: dnMotorRefOffset_p |

| Parameters | | Info |
|---|---|---|
| 14 | C01206/1 | MCK: Mounting direction: Motor |
| 15 | C00909/1 | Max. pos. speed |
| 16 | C00909/2 | Max. neg. speed |
| 17 | C00105 | Decel. time - quick stop |
| 18 | C00104/1 | Quick stop setting |
| 19 | C00497/4 | LS_Resolver: Encoder filter time FreqIn67 |
| 20 | C00833/30 | bTorquemodeOn |
| 21 | C00833/33 | bQSPOn |
| 22 | C00830/88 | MCTRL: nSpeedHighLimit_a |
| 23 | C00830/23 | MCTRL: nSpeedLowLimit_a |
| 24 | C00833/28 | MCTRL: bSpeedInterp. |
| 25 | C00050 | Speed setpoint |

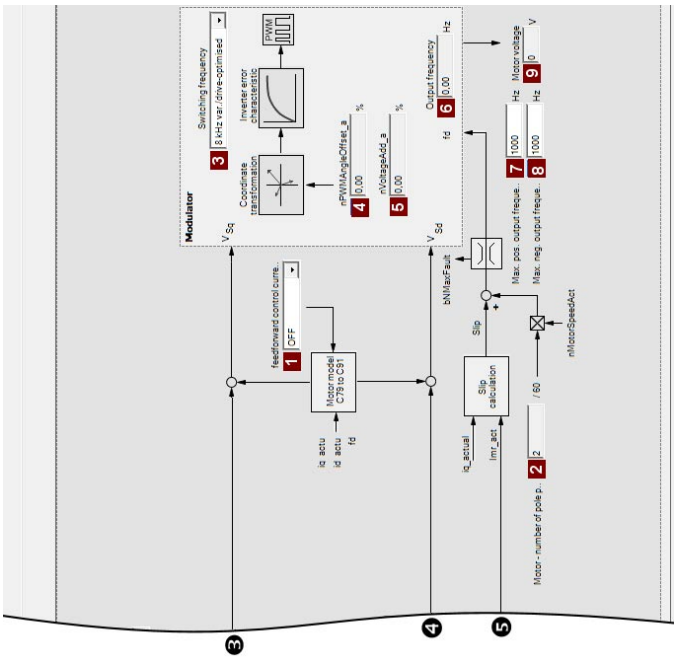
| Parameters | | Info |
|---|---|---|
| 26 | C00830/29 | Limitation of torque in motor mode |
| 27 | C00830/28 | Limitation of torque in generator mode |
| 28 | C00830/24 | MCTRL: nSpeedCtrlI_a |
| 29 | C00833/31 | MCTRL: bSpeedCtrlIOn |
| 30 | C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| 31 | C00070/2 | SC: Vp speed controller |
| 32 | C00071/2 | SC: Ti speed controller |
| 33 | C00072 | SC: Tdn speed controller |




| Parameters | | Info |
|---|---|---|
| 1 | C00830/27 | MCTRL: nTorqueSetValue_a |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00833/30 | MCTRL: bTorqueModeOn |
| 5 | C00833/33 | bQSPOn |
| 6 | C02865/1 | Special settings (bit 12) |
| 7 | C01206/1 | MCK: Mounting direction: Motor |
| 8 | C02865/1 | Sondereinstellungen (Bit 13) |
| 9 | C00274 | SC: Max. change in acceleration |
| 10 | C00280 | SC: Filter time const. DC detection |

| Parameters | | Info |
|---|---|---|
| 11 | C00052 | Motor voltage |
| 12 | C00063/1 | Motor temperature |
| 13 | C00056/2 | Actual torque |
| 14 | C00270 | SC: Freq. current setpoint filter |
| 15 | C00271 | SC: Current setpoint filter width |
| 16 | C00272 | SC: Current setpoint filter depth |
| 17 | C00577 | SC: Vp field weakening controller |
| 18 | C00578 | SC: Tn field weakening controller |
| 19 | C00576 | SC: Field feedforward control |
| 20 | C00056/1 | Torque setpoint |
| 21 | C00095 | Motor magnetising current |
| 22 | C00083 | Motor rotor time constant |

| Parameters | | Info |
|---|---|---|
| 23 | C00022 | Imax in motor mode |
| 24 | C00023 | Imax in generator mode |
| 25 | C00833/29 | MCTRL: bTorqueInterpolatorOn |
| 26 | C00937/2 | Torque-producing current |
| 27 | C00077 | SC: Vp field controller |
| 28 | C00078 | SC: Tn field controller |
| 29 | C00075 | Vp current controller |
| 30 | C00076 | Ti current controller |




| | Parameters | Info |
|---|---|---|
| 1 | C00079/1 | SC: Current controller - feedforward control |
| 2 | C00969/1 | Motor - number of pole pairs |
| 3 | C00018 | Switching frequency |
| 4 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 5 | C00830/31 | MCTRL: nVoltageAdd_a |
| 6 | C00058 | Output frequency |
| 7 | C00910/1 | Max. pos. output frequency |
| 8 | C00910/2 | Max. neg. output frequency |
| 9 | C00052 | Motor voltage |

### 5.9.2 Types of control

The servo control can be operated in two different modes:

- Speed control with torque limitation (*bTorquemodeOn* = FALSE)
- Torque control with speed limitation (*bTorquemodeOn* = TRUE)

**Tip!**

A position-controlled application requires a speed control with torque limitation (*bTorquemodeOn* = FALSE).

#### 5.9.2.1 Speed control with torque limitation

When *bTorquemodeOn* = FALSE, the drive system is operated with a selected speed setpoint in a speed-controlled manner.

**Note!**

From version 13.00.00, the torque setpoint *nSpeedSetValue_a* is set to 0 by quick stop (QSP) device-internally and the torque limit values *nTorqueMotLimit_a* and *nTorqueGenLimit_a* are set to 100 % to stop the drive quickly and safely anytime. The previous behaviour can be set in C2865/1 via bit 12 and bit 13.

The operational performance can be adapted in the following ways:

A. Overload limitation in the drive train

B. Motor current limitation

**Overload limitation in the drive train**

The torque is limited via the torque setpoint.

- The torque setpoint is identical to the value at the output of the speed controller, *nOutputSpeedCtrl*.
- To avoid overload in the drive train, the torque in motor mode can be limited via the *nTorqueMotLimit_a* process input signal, and the torque in generator mode can be limited via the *nTorqueGenLimit_a* process input signal:

| Designator<br>DIS code \| data type | Information/possible settings |
|---|---|
| nTorqueMotLimit_a<br>C00830/29 \| INT | Torque limitation in motor mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: 0 ... +199.99 %<br>• If keypad control is performed: Parameterisable via C00728/1. |
| nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in generator mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: -199.99 ... 0 %<br>• If keypad control is performed: Parameterisable via C00728/2. |

**Note!**

To avoid instabilities during operation, the torque limit values are internally processed as absolute values.

**Motor current limitation**

A cross current setpoint is calculated from the torque setpoint which is limited depending on the magnetising current, the max. current in motor mode (C00022) and the max. current in generator mode (C00023). The total current injected into the motor does not exceed the max. currents in motor mode and in generator mode.

**Note!**

For synchronous motors, the Lenze setting of the torque limits *nTorqueMotLimit_a* and *nTorqueGenLimit_a* to 100 % can result in the activation of the torque limitation for motor temperature < max. motor temperature before the set current limit values (C00022, C00023) are reached.

- With a motor temperature of approx. 20°C and maximum load, the maximum current will then be set to approx. 15 % below the set current limit values.
- Remedy: If the torque limits *nTorqueMotLimit_a* and *nTorqueGenLimit_a* are increased to 115 %, the set current limit values can also be reached for a motor temperature of 20°C and maximum load. Alternatively, the motor temperature compensation can be switched off via C02878. In this case, however, the torque accuracy decreases.

### 5.9.2.2 Torque control with speed limitation

When *bTorquemodeOn* = TRUE, a torque-controlled operation is activated. The setpoint torque directly follows the default value *nTorqueSetValue_a*.

Due to its speed limitation, the torque-controlled drive can only rotate within a speed range whose positive speed is limited by *nSpeedHighLimit_a* and whose negative speed is limited by *nSpeedLowLimit_a*.

**Note!**

- Absolute speed limitation to speed 0 rpm (*nSpeedLowLimit_a* or *nSpeedHighLimit_a* = 0) is only possible from version 12.00.00.
- Quick stop (QSP) is used to switch over to Speed control with torque limitation.
  - From version 13.00.00, the torque setpoint *nSpeedSetValue_a* is set to 0 by quick stop (QSP) device-internally and the two torque limit values *nTorqueMotLimit_a* and *nTorqueGenLimit_a* are set to 100 % to stop the drive quickly and safely anytime. The previous behaviour can be set in C2865/1 via bit 12 and bit 13.

- From version 13.00.00, the *bLimSpeedTorquemodeOn* status signal is used to show that the speed limitation is active.
- The speed is defined by the process.
- The torque setpoint is calculated directly from *nTorqueSetValue_a*.
  - From version 12.00.00 onwards, the torque limitation is active via *nTorqueMotLimit_a* and *nTorqueGenLimit_a* in this control mode, too, for the torque setpoint to be limited. The torque limitation can be deactivated in C2865/1 via bit 0 to obtain the previous function.

| Designator<br>DIS code \| data type | Information/possible settings |
|---|---|
| nTorqueSetValue_a<br>C00830/27 \| INT | Torque setpoint / additive torque<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| nSpeedHighLimit_a<br>C00830/88 \| INT | Upper speed limit for the speed limitation<br>• During torque-controlled operation only (*bTorquemodeOn* = TRUE)<br>• Scaling: 16384 ≡ 100 % rated speed (C00011) |
| nSpeedLowLimit_a<br>C00830/23 \| INT | Lower speed limit for speed limitation<br>• During torque-controlled operation only (*bTorquemodeOn* = TRUE)<br>• Scaling: 16384 ≡ 100 % rated speed (C00011) |
| nTorqueMotLimit_a<br>C00830/29 \| INT | Torque limitation in motor mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: 0 ... +199.99 %<br>• If keypad control is performed: Parameterisable via C00728/1. |
| nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in generator mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: -199.99 ... 0 %<br>• If keypad control is performed: Parameterisable via C00728/2. |

### 5.9.3 Basic settings

The following "Initial commissioning steps" must be performed to commission the servo control:

<table>
<tr><th colspan="3">Initial commissioning steps</th></tr>
<tr><td>1.</td><td colspan="2">Determine the motor control:<br>• For the closed-loop control of a synchronous motor (PSM):<br>C00006 = "1: SC: Servo control PSM"<br>• For the closed-loop control of an asynchronous motor (ASM):<br>C00006 = "2: SC: Servo control ASM"</td></tr>
<tr><td>2nd</td><td colspan="2">Set the motor selection/motor data<br>• When selecting and parameterising the motor, the motor nameplate data and the equivalent circuit diagram data are relevant. Detailed information can be found in the "Motor selection/Motor data" chapter. (📖 139)<br>Depending on the motor manufacturer, proceed as follows:</td></tr>
<tr><td></td><td>Lenze motor:<br>Selecting a motor from the motor catalogue in the »Engineer«<br>- or -<br>1. Set the motor nameplate data<br>2. Automatic motor data identification</td><td>Third party manufacturer's motor:<br>1. Set the motor nameplate data<br>2. Automatic motor data identification or set known equivalent circuit diagram data manually:<br>C00082: Motor rotor resistance*<br>C00084: Motor stator resistance<br>C00085: Motor stator leakage inductance<br>C00092: Motor magnetising inductance*<br>C00095: Motor magnetising current*<br>* Setting only required for asynchronous motors.</td></tr>
<tr><td>3.</td><td colspan="2">Define the type of control:<br>bTorquemodeOn = FALSE: Speed control with torque limitation<br>bTorquemodeOn = TRUE: Torque control with speed limitation</td></tr>
<tr><td>4.</td><td colspan="2">Parameterise the encoder/feedback system. ▸ Encoder/feedback system (📖 321)</td></tr>
<tr><td>5th</td><td colspan="2">Only with servo control for synchronous motors (PSM):<br>Detect pole position of the motor. ▸ Pole position identification (PPI) (📖 366)</td></tr>
</table>

**Note!**

- The Lenze settings of the inverter are predefined for a power-adapted standard asynchronous motor. For an optimal drive behaviour, we recommend to adapt the controller settings.
- When controlling an asynchronous motor (ASM), magnetisation is carried out prior to enabling the setpoint after a controller enable in the Lenze setting. The (delayed) setpoint enabling ensures direct availability of the maximum torque at the motor.
  - The delay of the setpoint enabling associated with the magnetisation can be reduced by reducing the threshold set in C00918. However, this also reduces the maximum torque which is directly available after the setpoint enabling.
  - If C00918 = "0 %", magnetisation is not carried out.

**Tip!**

Information on the optimisation of the control mode and the adaptation to the real application is provided in chapter "Optimising the control mode". (📖 243)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 263)

### 5.9.4 Optimising the control mode

**Note!**

When the inverter has been enabled, starting up is delayed by the magnetization process of the motor. The speed setpoint is only enabled for motor control if 87 % of the motor magnetising current are available. If this delay is not tolerable for certain applications, the preset percentage threshold can be reduced in C00918.

From version 12.00.00:

- Following successful motor parameter identification, the current controller parameters (C00075, C00076) and field controller parameters (C00077, C00078) are calculated automatically.
  - If these parameters are not to be calculated, bit 4 of C02865/1 must be set to "1".
- Following successful motor parameter identification, the speed controller parameters (C00070/2, C00071/2, C00072) can be calculated automatically.
  - If these parameters are to be calculated, bit 6 of C02865/1 must be set to "5".
- Following successful motor parameter identification, other controller parameters (C00011, C00022, C00497) can be calculated automatically.
  - If these parameters are to be calculated, bit 6 of C02865/1 must be set to "6".

The "optimisation steps" given in the table below serve to further optimise the control behaviour of the servo control and adjust it to the concrete application.

- Detailed information on the individual steps can be found in the following subchapters.

Generally, the following optimisation steps are recommended:

| Optimisation steps | |
|---|---|
| 1. | Optimise current controller. (📖 244)<br>• The current controller should always be optimised if a motor of a third-party manufacturer with unknown motor data is used! |
| 2nd | Optimise speed controller. (📖 245)<br>• The setting of the speed controller must be adapted depending on the mechanical path. |
| 3. | Optimise response to setpoint changes and determine mass inertia. (📖 248)<br>• For an optimal reference behaviour, the total moment of inertia can be used to make a feedforward control of the speed setpoint. |

Special cases may require further optimisation steps:

| Optimisation steps | |
|---|---|
| 1. | Setting the current setpoint filter (band-stop filter). (📖 251)<br>• In order to suppress or damp (mechanical) resonant frequencies, a current setpoint filter is integrated in the speed control loop which is switched off in the default setting but can be parameterised accordingly, if required.<br>Then readjust the speed controller: Optimise speed controller. (📖 245) |
| 2nd | Adapting the max. acceleration change (jerk limitation). (📖 252) |
| 3. | Only with servo control for asynchronous motors (ASM):<br>Optimising the behaviour of the asynchronous motor in the field weakening range. (📖 257)<br>(For synchronous motors, this function is in preparation) |

**Tip!**

In order to traverse a typical speed profile for optimisation of motor control, you can also use the basic function "Manual jog" with appropriately adapted manual jog parameters if this basic function is supported by the selected technology application. (📖 638)

### 5.9.4.1 Optimise current controller

**Note!**

An optimisation of the current controller should generally be carried out unless a power-adapted standard asynchronous motor is used or the motor has been selected from the motor catalogue of the »Engineer«!

An optimisation of the current controller is sensible since the two control parameters gain (C00075) and reset time (C00076) depend on the required maximum current and the set switching frequency.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00075 | Vp current controller | 7.00 | V/A |
| C00076 | Ti current controller | 10.61 | ms |

- Gain and reset time can be calculated as per the following formulae:

$$V_p = \frac{L_{ss}[H]}{T_E[s]}$$

$$T_i = \frac{L_{ss}[H]}{R_s[\Omega]}$$

$V_p$ = Current controller gain (C00075)
$T_i$ = Current controller reset time (C00076)
$L_{ss}$ = Motor stator leakage inductance (C00085)
$R_s$ = Motor stator resistance (C00084)
$T_E$ = Equivalent time constant (= 500 µs)

### 5.9.4.2 Optimise speed controller

**Note!**

With servo control, an optimisation of the speed controller is recommended in order that the connection to a motor rotating with very high speed after controller enable always functions properly.

The speed controller is in the form of a PID controller with an additional differential speed-setpoint gain. For optimum behaviour, the PID speed controller has to be optimised and the overall mass inertia of the drive train has to be determined.

- In the Lenze setting, the configuration of the speed controller provides robustness and moderate dynamics.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00070/2 | SC: Vp speed controller | 6.00 | |
| C00071/2 | SC: Ti speed controller | 50.0 | ms |
| C00072 | SC: Tdn speed controller | 0.00 | ms |

**Speed controller gain Vp**

The gain Vp (C00070/2) of the speed controller is defined in a scaled representation which enables a comparable parameterisation almost independent of the power of the motor or inverter. Here, the speed input difference of the controller is scaled to the rated motor speed whereas the output torque refers to the rated motor torque. A gain of 10 means that a speed difference of 1 % is gained through the P component with 10 % torque.

If the rated data of the motor and the mass inertia of the drive system are known, we recommend the following setting:

$$V_P \approx 1.5 \ldots 3 \cdot \frac{T_M[s]}{0.01[s]}$$

$$T_M[s] = \frac{2 \cdot \pi \cdot n_N[rpm]}{M_N[Nm] \cdot 60} \cdot J_{Drive,\,total}[kgm^2]$$

$$M_N[Nm] = \frac{P_N[W] \cdot 60}{2 \cdot \pi \cdot n_N[rpm]}$$

$V_P$ = Gain of the speed controller (C00070/1)
$T_M$ = Time constant for the acceleration of the motor
$M_N$ = Rated motor torque
$n_N$ = Rated motor speed
$J_{drive,\,total}$ = Total moment of inertia of the drive

[5-16] Recommendation for the setting of the gain of the speed controller

If the mass inertia of the drive is unknown, the optimisation can be achieved as follows:

1. Specify speed setpoint.
2. Increase Vp (C00070/2) until the drive is unstable (observe motor noise).
3. Reduce Vp (C00070/2) until the drive runs stable again.
4. Reduce Vp (C00070/2) to approx. half the value.

**Tip!**

Values recommended by Lenze for the setting of the (proportional) gain:

- For drive systems without feedback: Vp = 6 ... 20
- For drive systems with a good disturbance behaviour: Vp > 12

#### Speed controller reset time Ti

Apart from setting the P component, C00071/2 provides the possibility to take influence on the I component of the PI controller.

If the mass inertia of the drive is unknown, the optimisation can be achieved as follows:

1. Specify speed setpoint.
2. Reduce Ti (C00071/2) until the drive is unstable (observe motor noise).
3. Increase Ti (C00071/2) until the drive runs stable again.
4. Increase Ti (C00071/2) to approx. twice the value.

**Tip!**

Value range recommended by Lenze for the setting of the reset time:

Ti = 20 ms ... 150 ms

#### Differential time constant Tdn (rate time)

The differential time constant Tdn of the speed controller can be set in C00072.

If the mass inertia of the drive is unknown, the optimisation can be achieved as follows:

- Increase Tdn (C00072) during operation until optimal control mode is reached.

#### Using the ramp response for setting the speed controller

If the mechanical components cannot be operated at the stability limit, the ramp response can also be used for setting the speed controller.

**Stop!**

If the controller parameters are preset unfavourably, the control can tend to heavy overshoots up to instability!

- Following and speed errors can adopt very high values.
- If the mechanics are sensitive, the corresponding monitoring functions are to be activated.

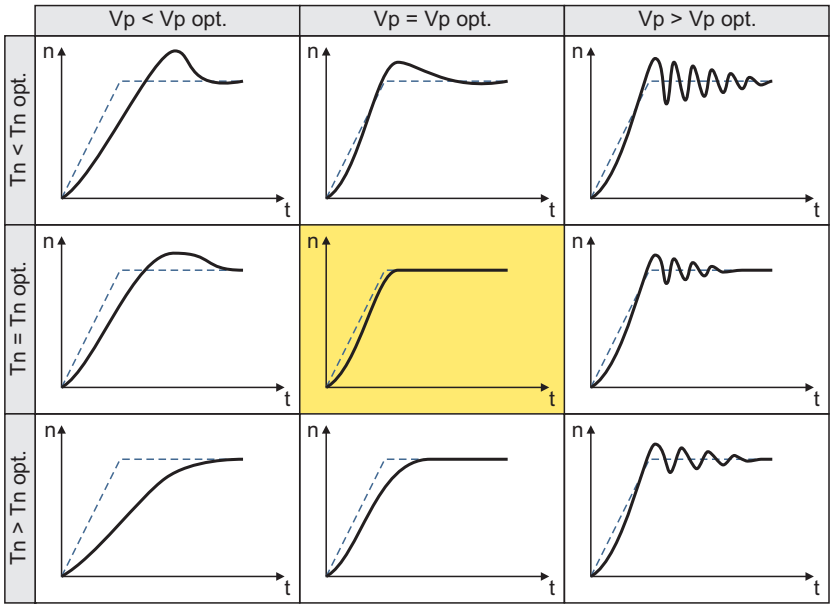
**Note!**

For an optimal setting, we recommend to determine the mass inertia (optimal response to setpoint changes) first.

▸ Optimise response to setpoint changes and determine mass inertia (📖 248)

**How to optimise the speed controller setting by means of the ramp response:**

1. Run a typical speed profile and record the ramp response of the speed using the data logger.
   - Motor control variables to be recorded:
     *nSpeedSetValue_a* (speed setpoint)
     *nMotorSpeedAct_a* (actual speed value)
2. Evaluate the ramp response:

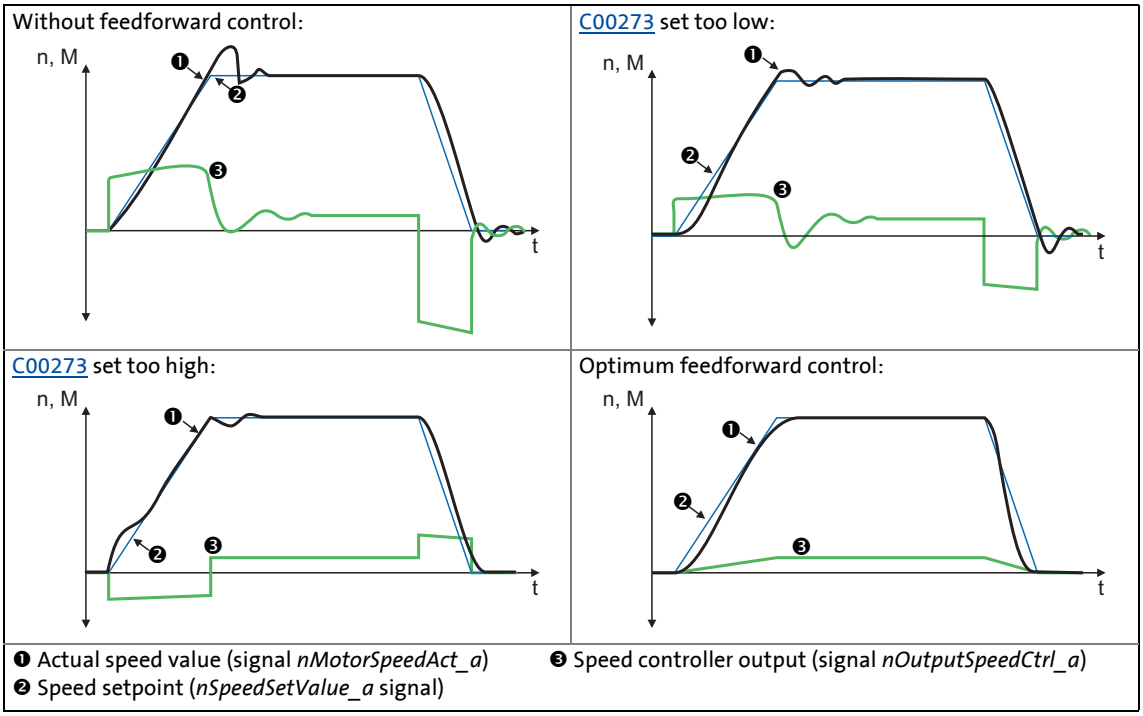



   - Solid line = ramp response (actual speed value)
   - Dash line = speed setpoint
3. Change gain Vp in C00070/2 and reset time Tn in C00071/2.
4. Repeat steps 1 ... 3 until the optimum ramp response is reached.

### 5.9.4.3 Optimise response to setpoint changes and determine mass inertia

#### Optimisation at constant mass inertia

Setting the total moment of inertia under C00273 provides the optimum torque feedforward control. Depending on the application, an adjustment of the setting under C00273 may be necessary to optimise the response to position/speed setpoint changes by means of the torque feedforward control.




[5-17] Typical signal characteristics for different settings of the load moment of inertia

**How to optimise the torque feedforward control:**

1. Run a typical speed profile and record the inputs and outputs of the speed controller with the data logger.
   - Motor control variables to be recorded:
     *nSpeedSetValue_a* (speed setpoint)
     *nMotorSpeedAct_a* (actual speed value)
     *nOutputSpeedCtrl_a* (speed controller output)
2. Estimate the moment of inertia and set it in C00273 in relation to the motor end (i.e. with account being taken of the gearbox factors).
3. Repeat the data logger recording (see step 1).

   Now the data logger should show that part of the required torque is generated by the feedforward control and the speed controller output signal (*nOutputSpeedCtrl_a*) is correspondingly smaller. The resulting following error decreases.
4. Change the setting in C00273 and repeat the data logger recording until the intended response to setpoint changes is reached.
   - The optimisation could aim at the speed controller being completely relieved (see signal characteristics in Fig. [5-17]).
5. Save the parameter set (device command: C00002/11).

**Optimisation at variable mass inertia**

From version V12.00.00, mass inertia that changes during the process (e.g. a reel) can be taken into account when optimising the response to setpoint changes.

How to proceed:

1. In C00273 the known constant total moment of inertia (motor, gearbox, shaft, etc.) must be set or determined according to previous instructions ("How to optimise ...").
   - The determination requires travelling the typical speed profile without variable mass inertia (e.g. reels).
2. At the LS_MotorInterface SB, the *nInertiaAdapt_a* process signal must be interconnected in a way which ensures that a value of "100 %" is applied at this input.
3. In C00919/1 set the known maximum value of the variable moment of inertia or determine the value according to previous instructions ("How to optimise ...").
   - The determination requires travelling the typical speed profile including variable mass inertia (e.g. maximum reels).
4. The *nInertiaAdapt_a* process signal can be used during the process to dynamically control the percentage of the variable moment of inertia set in C00919/1 which is to be considered for setpoint feedforward control.

   Example :
   - If there is no variable moment of inertia (e.g. no reel), the *nInertiaAdapt_a* process signal must be set to "0 %".
   - If the maximally variable moment of inertia is available (e.g. maximum reels), the *nInertiaAdapt_a* process signal must be set to "100 %".



**Tip!**

Via the *nTorqueSetValue_a* process signal at the LS_MotorInterface SB, any differential signal can be defined for torque feedforward control. First the speed controller, then this additive torque is connected which is hence not derived from the differential change in speed setpoint.

**Other functions for differential setpoint feedforward control**

From version V12.00.00 onwards, the following additional functions are available for differential setpoint feedforward control (torque feedforward control):

- In C00653/1, the sensitivity of setpoint feedforward control can be adapted.
- In C00654/1, alternatively to the *nSpeedSetValue_a* process signal, the new *nSpeedSetValueInertia_a* process signal for the setpoint feedforward control can be selected at the LS_MotorInterface SB. Via the *nSpeedSetValueInertia_a* process signal, an optional input value (e.g. setpoint of the position or process controller) for the torque feedforward control can be specified.
- For *bTorqueModeOn* = TRUE, the setpoint feedforward control is added to the torque setpoint *nTorqueSetValue_a*. In this way, feedforward control of torque is also possible for torque-controlled operation (e.g. for winder applications).




**Stop!**

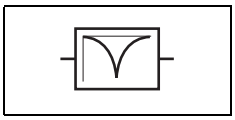
In the Lenze setting of C00654/1, the speed setpoint of the speed controller (*nSpeedSetValue_a*) is used for the torque feedforward control which is why it is also called "differential setpoint feedforward control". A very sharp change of the speed setpoint at the speed controller thus causes a strong torque impulse at the machine!

A very sharp change of the speed setpoint of the speed controller occurs if a control creates the setpoint ramp itself and the speed setpoint is only written every 20 ms to the drive. (In this case, the speed setpoint is changed every 20 ms.)

Deactivate the speed feedforward control for the speed setpoint in C00654/1 if the speed setpoint changes very sharply or the mass inertia is unknown (e.g. in case of hoists)!

### 5.9.4.4 Setting the current setpoint filter (band-stop filter)

Due to the high dynamic performance/limit frequency of the closed current control loop, mechanical natural frequencies can be activated which may lead to an unstable speed control loop.



To mask out or at least damp these resonant frequencies, a so-called current setpoint filter is integrated into the speed control loop of the inverter.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00270 | SC: Freq. current setpoint filter | 200.0 | Hz |
| C00271 | SC: Current setpoint filter width | 0.0 | Hz |
| C00272 | SC: Current setpoint filter depth | 0 | dB |

- In the default setting of 0 db of the filter depth (C00272), the current setpoint filter is switched off.

#### Setting of the current setpoint filter

Since the frequency response of the speed controlled system is only rarely known to such an extent that the current setpoint filter can be adjusted to the controlled system in the run-up, the following example describes how to set the current setpoint filter.

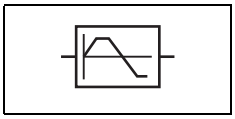
**How to set the current setpoint filter:**

1. Optimise current controller (📖 244).
2. Optimise speed controller (📖 245)
3. Measure the oscillation frequency (observe current or speed).
4. Set the measured oscillation frequency in C00270 as filter frequency.
5. Set "25%" of the filter frequency in C00271 as filter width.
   - Example: Filter frequency = 200 Hz → filter width = 50 Hz.
6. Set "40 dB" in C00272 as filter depth.
   - If the filter depth is set to "0 dB" (default setting), the filter is not active.

**Note!**

Readjust the speed controller after setting the current setpoint filter. ▸ Optimise speed controller. (📖 245)

### 5.9.4.5 Adapting the max. acceleration change (jerk limitation)



Via the max. acceleration change that can be set in C00274, the change of the setpoint torque can be limited for jerk limitation. Hence, sudden torque step changes can be avoided. The entire speed characteristic is smoothed.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00274 | SC: Max. change in acceleration | 400.0 | %/ms |

In the default setting of 400 %/ms of the max. acceleration change (C00274), jerk limitation is switched off.

The setting defines the permissible maximum torque change per ms (based on the rated motor torque).

**Note!**

Only activate this jerk limitation for speed-controlled applications!

If table positioning or a free function block interconnection with a positioning operating mode is selected, jerk limitation must be switched off.

- Here, jerk limitation is provided for in the travel profile generator. Setting this jerk limitation in the motor control would lead to following errors!

### 5.9.4.6 Slip calculation from motor equivalent circuit diagram data

This function extension is available from version 02.00.00!

In order to achieve a better speed stability and torque accuracy, the slip calculation can be either derived from the motor nameplate data (e.g. rated motor speed) or the motor equivalent circuit diagram data (stator resistance, rotor resistance etc.).

In case of servo control for asynchronous motors (C00006 = "2"), the slip calculation is always made

- up to and including version 01.xx.xx from the motor nameplate data.
- from version 02.00.00 onwards in the Lenze setting from the motor equivalent circuit diagram data in order to obtain a higher torque accuracy.

The data to be used for servo control (for asynchronous motors) is selected via bit 1 in C02879/1:

| Setting | | Info |
|---|---|---|
| Bit 0 | SLVC | In case of sensorless vector control:<br>• "0" ≡ Slip calculation from motor nameplate data (Lenze setting)<br>• "1" ≡ Slip calculation from motor equivalent circuit diagram data |
| Bit 1 | SC_ASM | In case of servo control for asynchronous motors:<br>• "0" ≡ Slip calculation from motor nameplate data<br>• "1" ≡ Slip calculation from motor equivalent circuit diagram data (Lenze setting) |
| Bit 2 ... 7 | Reserved | |

**Note!**

In order that the slip can be calculated from the motor equivalent circuit diagram data, the equivalent circuit data (stator resistance, rotor resistance etc.) must be known as exactly as possible.

- Selecting a motor in the »Engineer« motor catalogue loads the exact motor equivalent circuit diagram data.
- When the motor nameplate data is entered manually and the motor equivalent circuit diagram data is then detected via the motor parameter identification, the "extended identification" (C02867/1 = 2) must be used. ▸ Automatic motor data identification (📖 145)

In case of servo control for asynchronous motors, the slip calculation can also be made from the motor nameplate data (C02879/Bit 1 = "0"). In this case, the slip is too high by a factor of 1.35 as the rated motor speed (C00087) is assumed for a cold machine. This has the following consequences:

- The higher slip provides a better behaviour for the field weakening range.
- In the motor setting range, a considerably too high torque is detected (torque accuracy is worse).
- The achieved maximum torque in the motor setting range is smaller.

From version 15.00.00, the excessive slip with the activated "slip calculation from motor nameplate data" option (C02879/bit 1 = "0") can be corrected by setting bit 7 to "1" in C02864/1. With this setting, the rated motor speed (C00087) for a warm machine is assumed.

- By selecting the motor in the »Engineer« motor catalogue or executing the motor parameter identification, bit 7 is set to "1" in C02864/1.

## Note!

**Exchange of devices**

When devices with a parameterised servo control for asynchronous motors are exchanged, problems regarding the control quality may occur if motor parameters set for a device with version 01.xx.xx are tranferred to a replacement device of a higher version.

- In this case, we recommend to load the very precise motor data available from »Engineer« V2.17 or to execute the "advanced motor parameter identification" (C02867/1 = 2). ▸ Automatic motor data identification (📖 145)
- As an alternative, the slip calculation from the motor nameplate data can be set again in C02879/1 (bit 1 = "0") to obtain the previous behaviour.

### 5.9.4.7 Temperature compensation within the motor control

This function extension is available from version 02.00.00!

A temperature compensation over the detected motor temperature (display in C00063/1) serves to compensate inaccuracies in the output torque within the motor control in case of temperature changes of the asynchronous or synchronous motor.

**Note!**

In the Lenze setting, the temperature compensation within the motor control is activated. The temperature compensation, however, is only active with speed encoder selection "3: Multi encoder" or "4: Resolver" in C00495 as well as error-free KTY temperature detection (display in C00063/1 ≠ 255 °C).

**Behaviour in case the temperature compensation is not active**

If the temperature compensation is not active,

- we assume a motor temperature of 20 °C when using an asynchronous motor. If the asynchronous motor is at operating temperature, however, a too low actual torque value is detected, especially in the field weakening range. This leads to a strong decrease in the torque setting range.
- we assume a motor temperature of 150 °C when using a synchronous motor. If the synchronous motor is cold, however, a too high actual torque value is detected. This leads to a decrease in the torque setting range.

From version 14.00.00 onwards, this negative behaviour in case of warm asynchronous motors or cold synchronous motors can be improved by setting the option "warm machine" (bit= "1") in C02864. We then assume a motor temperature of 90 °C.

The temperature compensation is switched on/off bit-coded in C02878/1:

| Setting | | Info |
|---|---|---|
| Bit 0 | SC_PSM | KTY motor temperature compensation with servo control for synchronous motors:<br>• "0" ≡ Off<br>• "1" ≡ On (Lenze setting) |
| Bit 1 | SC_ASM | KTY motor temperature compensation with servo control for asynchronous motors:<br>• "0" ≡ Off<br>• "1" ≡ On (Lenze setting) |
| Bit 2 ... 7 | Reserved | |

**General information on temperature behaviour of the asynchronous and synchronous motor**

In case of the **asynchronous motor**, the nameplate data always refer to rated data, i.e. on the rated load and thus on the rated temperature of the motor. For the 8400 device series, the equivalent circuit data for stator resistance (C00084) and rotor resistance (C00082) are always given for a cold asynchronous motor (20 °C) and converted to star connection. In case of a cold motor (20 °C), the real slip frequency thus corresponds to the calculated setpoint slip. In case of a warm motor, the real slip frequency is higher. This becomes especially apparent at high torques as an inaccuracy of the torque over the motor temperature.

In case of the **synchronous motor**. the field strength generated by the permanent magnet in the rotor is temperature-dependent. With a rated motor temperature, a rated field strength is generated. A negative temperature coefficient of -0.11 %/°C means that with a cold motor at the same setpoint torque a higher output torque is output than with a warm motor.

These errors must be compensated by means of the temperature compensation for asynchronous and synchronous motors in order to have a constant output torque over the motor temperature independent of the motor temperature.

### 5.9.4.8 Optimising the behaviour of the asynchronous motor in the field weakening range

The behaviour of the asynchronous motor in the field weakening range is influenced by:

- the field controller
- the field weakening controller
- the field feedforward control unit
- adaptive adaptation of the P component of the field and speed controllers
- An additional limitation of the I component of the controller when the voltage limit is reached

**Note!**

By default, these control parts are called every 1 ms. However, the 1-ms cycle is not always sufficient to achieve an optimum jerk-free transition to the field weakening range.

From version 14.00.00, the control parts can be called for an optimum jerk-free transition to the field weakening range in a 500-µs cycle. For this purpose, bit 5 has to be set to "1" int in C02864/1.

- With this setting, C00015 and C00576 do not have any influence anymore on the field feedforward control.
- We recommend this setting for new applications.
- In existing applications, this setting may require an increase of the reset times of the field weakening controller (C00578) and the field controller (C00078) by the factor 2 in order to achieve a stable operation.

In general, these settings are pre-optimised so that further optimisation is not required.

Optimisation for special motors (e.g. mid-frequency motors) or for standard asynchronous motors whose power is not adapted can be carried out according to the algorithms shown in the following sections.

From version 16.00.00 onwards, the dynamics of the field weakening control can be improved with C02864/1 (bit 9 = 1) by admitting also negative setpoints of the field current controller at the field controller output. Please note that this setting may cause higher torque peaks which can be reduced again by decreasing the field weakening controller gain ( C00577) or increasing the reset time (C00578).

### Field controller

The field controller is designed as a PI controller.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00077 | SC: Vp field controller | 12.80 | |
| C00078 | SC: Tn field controller | 256.0 | ms |

The gain Vp (C00077) of the field controller can be calculated using the motor rotor time constant and the equivalent time constant of the current-controlled motor:

$$V_{PFeld} = \frac{T_R[s]}{a^2 \cdot T_{Replacement}[ms]} \quad [\%]$$

$V_{PFeld}$: Gain of the field controller (C00077)
$T_R$: Motor rotor time constant (C00083)
$T_{Equivalent}$: Motor equivalent time constant (approx. 2.5 ms)
a: Measure for damping (z. B. a = 2)

If the rated data of the motor and the mass inertia of the drive system are known, we recommend the following setting:

$$V_{PFeld} \approx \frac{T_R[s]}{4 \cdot T_{Replacement}[ms]} = \frac{T_R[s]}{0.01[s]}$$

$$T_{NFeld}[ms] = T_R[ms]$$

$V_{PFeld}$: Gain of the field controller (C00077)
$T_R$: Motor rotor time constant (C00083)
$T_{Equi}$: Motor equivalent time constant
$T_{NFeld}$: Time constant of the field controller (C00078)

[5-18] Recommendation for the setting of the gain and the time constant of the field controller

**Tip!**

The motor rotor time constant depends on the motor rotor resistance, the magnetising inductance, and the leakage inductance.

- For an optimal calculation, we recommend to select the motor from the motor catalogue of the »Engineer« first if a Lenze motor is used. ▸ Selecting a motor from the motor catalogue in the »Engineer« (📖 143)
- If a third party manufacturer's motor is used, motor parameter identification must be carried out previously. ▸ Automatic motor data identification (📖 145)

### Field weakening controller

The field weakening controller serves to adapt the magnetising current when the maximum control voltage has been reached so that in steady operation approximately 95 % of the maximally possible control voltage is output. Thus, there is a voltage reserve for dynamic load or speed variations.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00577 | SC: Vp field weakening controller | 0.0010 | |
| C00578 | SC: Tn field weakening controller | 20.0 | ms |

$$V_{P,\,FS} = 4774.65 \cdot \frac{L_h[mH] + L_{ss}[mH]}{L_h[mH]^2 \cdot n_N[rpm] \cdot I_N[A]}$$

$$C00577 = 4774.65 \cdot \frac{C00092\ [mH] + C00085\ [mH]}{C00092\ [mH]^2 \cdot C00087\ [rpm] \cdot C00088\ [A]}$$

$$T_{n,\,FS} = 20\ ms \cdot \frac{0.0010}{V_{P,\,FS}}$$

$$C00578 = 20\ ms \cdot \frac{0.0010}{C00577}$$

$L_m$: Mutual motor inductance (C00092)
$L_{sl}$: Motor stator leakage inductance (C00085)
$n_{rated}$ = rated motor speed (C00087)
$I_{rated}$ = rated motor current (C00088)

[5-19] Recommendation for the setting of the gain and the time constant of the field weakening controller

**Tip!**

A field weakening controller set too slow can be recognised by the fact that the actual speed value cannot follow the speed setpoint in the field weakening range. In this case, carry out the setting of the field weakening controller as described in the formula. If the speed setpoint still cannot follow the actual speed value, increase the setting in C00577 to maximally twice the value and reduce the setting in C00578 according to the formula mentioned above.

If the field weakening controller is set too fast, vibrations occur in the magnetising current, in the direct-axis current and in the cross current, in the torque and in the speed in the field weakening range. This is also audible by an increased motor noise (humming). The vibrations can be dampened by a decrease of the proportional gain of the field weakening controller.

Procedure:

- Operate the drive with max. required speed in the field weakening operation.
- Reduce Vp (C00577) of the field weakening controller until the vibration response cannot be detected anymore. Increase Tn (C00578) according to the formula.

**Note!**

Vp (C00577) can also be set to 0. In this case, Tn (C00578) should first be set to 2000 ms.

### Field feedforward control

To have enough voltage reserve available for dynamic acceleration processes, a timely weakening of the field is required. The field weakening is controlled by the field feedforward control.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00576 | SC: Field feedforward control | 200.0 | % |

**Note!**

The function of the field feedforward control and thus the field weakening control requires a correct setting of the V/f base frequency (C00015).

- The selection of the motor from the »Engineer« motor catalogue and the motor parameter identification serve to set the V/f base frequency correctly.
- When third-party motors are entered, either the V/f base frequency (C00015) has to be calculated manually or a motor parameter identification has to be carried out.
  - Adapting the V/f base frequency (📖 168)
  - Automatic motor data identification (📖 145)

Due to an error in the device software, the field feedforward control of the servo control only operates properly if the V/f base frequency (C00015) is reduced by the following factor after being set:

- 400-V devices: Factor 0.173
- 230-V devices: Factor 0.34

The same factor has to be considered for the field feedforward control (C00576).

From version 14.00.00, the control parts can be called for an optimum jerk-free transition to the field weakening range in a 500-µs cycle. For this purpose, bit 5 has to be set to "1" int in C02864/1.

- With this setting, C00015 and C00576 do not have any influence anymore on the field feedforward control.
- We recommend this setting for new applications.

From version 16.00.00 onwards, the dynamic performance of the field weakening control is improved with C02864/1, bit 9 = "1": like this, negative (field) current setpoints are also permitted at the output of the field controller. If this causes higher torque peaks, they can be reduced by either reducing the field weakening controller gain C00577 or by increasing the reset time C00578.

The field feedforward control reciprocally reduces the magnetising current from the V/f base frequency (C00015) on. The starting point of the reduction can be shifted to low frequencies via the field feedforward control (C00576). Thus, more voltage reserve is available for acceleration processes.

The field feedforward control (C00576) must be specified in %, based on the rated slip of the machine.

**Tip!**

Generally, the Lenze setting is sufficient for most applications.

- We recommend to increase the field feedforward control for applications with very dynamic acceleration processes in the field weakening range.
- Reduce the field feedforward control for very slow applications if necessary.

### Adaptive adaptation of the P component of the field and speed controllers

In the field weakening range, the properties of the drive change due to the reduction of the magnetisation and the voltage limitation of the inverter. To be able to continue to provide a stable, well dampened drive behaviour, the servo control has an automatic adaptation of the P component of the field weakening controller and speed controller.

In the Lenze setting, this function is activated. Depending on the application, this function can be deactivated via C00079/2.

| Parameters | Info | Lenze setting |
|---|---|---|
| C00079/2 | SC: Adaptive field weakening controller | 1: On |

**Tip!**

Lenze recommends to always activate the adaptation of the field weakening and speed controller.

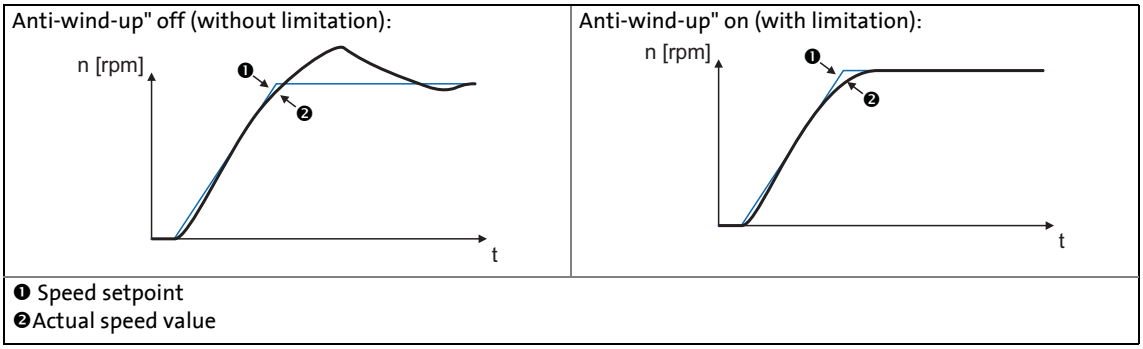
### Additional limitation of the I component of the speed controller when the voltage limit is reached (anti-wind-up)

In dynamic acceleration processes in the field weakening range, the acceleration torque is limited due to the limitation of the maximum output voltage of the inverter. Acceleration ramps that are set too high (C00012) may lead to undesired charging of the integrator of the speed controller, which result in very strong overshoots of the actual speed value when the setpoint is reached.

The so-called anti-wind-up function serves to intelligently limit the integrator part in dynamic acceleration processes in the field weakening range so that an overshoot of the actual speed value can be prevented when the setpoint speed is reached.

This function is deactivated in the Lenze setting because the robustness of the drive may decrease in the field weakening range. Possible consequences are slight speed variations. If required by the application, the function can be activated via C00079/3.

| Parameters | Info | Lenze setting |
|---|---|---|
| C00079/3 | SC: n-Ctrl Anti-Wind-Up | 0: Off |




[5-20] Typical signal characteristics for switched-on/off anti-wind-up function

**Tip!**

Lenze recommend to activate the additional limitation of the speed controller's I component on reaching the voltage limit (C00079 = 1) if dynamic acceleration processes in the field weakening range are wanted.

- The function should be deactivated again if slight speed variations occur in the field weakening range.
- If this function is activated, the actual speed value may not follow the speed setpoint correctly in the field weakening range.

### 5.9.4.9 Feedforward control of the current controller

**Note!**

Function only possible with servo control (SC) for asynchronous motors!

In C00079/1, the feedforward control of the current controller can be switched on. The feedforward control serves to unload the current controller so that the current control loop is able to operate more stable and dynamically. The feedforward control can contribute to dampen vibrations at very high speeds.

Before the feedforward control is activated in C00079/1, we recommend the execution of the motor parameter identification in order to get an optimum result with the feedforward control.
▸ Automatic motor data identification (📖 145)

**Note!**

If the data of the motor equivalent circuit (motor magnetising inductance, motor stator leakage inductance, motor stator resistance, motor rotor resistance) are not known, the feedforward control of the current controller must not be switched on as otherwise it might cause instabilities in the control loop!

Up to and including version 13.xx.xx the following applies:

The feedforward control of the current controller may only be switched on in C00079/1 if the motor magnetising inductance set in C00092 is not more than 4 % higher than the real magnetising inductance of the connected motor detected by an identification process. Otherwise, instabilities may occur at high speeds in the control loop and the motor can accelerate in an uncontrolled manner up to maximum speed.

- The magnetising inductance of a motor has a tolerance of more than 4 % so that this problem might occur when the motor parameters are loaded from the »Engineer« motor catalogue and the feedforward control of the current controller is activated.
- If the feedforward control is to be used and instabilities occur: Reduce the motor magnetising inductance (C00092) until a stable behaviour occurs or carry out a motor parameter identification.

From version 14.00.00 the following applies:

The feedforward control of the current controller operates properly within the tolerances of the motor. It is sufficient to select the motor from the »Engineer« motor catalog or enter the data of the motor equivalent circuit manually.

## 5.10 Parameterisable additional functions

### 5.10.1 Selection of switching frequency

The switching frequency of the inverter that can be selected in C00018 influences the smooth running performance and the noise generation in the connected motor as well as the power losses in the inverter.

The lower the switching frequency the higher the concentricity factor, the smaller the losses, and the higher the noise generation.

**Stop!**

If operated at a switching frequency of 16 kHz, the output current of the inverter must not exceed the current limit values specified in the technical data! (See chapter "Rated data" of the hardware manual.)

**Note!**

- Operate mid-frequency motors only at a switching frequency of 8 kHz or 16 kHz (var./drive-opt.).
- If operated at a switching frequency of 16 kHz, the Ixt evaluation (C00064) is considered including the required derating to 67 % of the rated device current at switching frequencies of 2.4 and 8 kHz.
- Ensure that the value does not fall below the permissible switching frequency,
  - if the motor is not approved for the setting switching frequency
    - or -
  - if a sinusoidal filter has been connected between the output of the inverter and the motor which is not approved for the setting switching frequency.
    (see section "Preventing a decrease of the switching frequency")

**Short overview of the relevant parameters:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00018 | Switching frequency | 2: "8 kHz var./drive-opt." | |
| C00144 | Switching frequency reduction (temp.) | 1: On | |
| C00725 | Current switching frequency | - | |
| C00910/1 | Max. pos. output frequency | 599 | Hz |
| C00910/2 | Max. neg. output frequency | 599 | Hz |
| C02864/1<br>Bit 4: | MCTRL: Special settings 2<br>Option "Motor ident: Switching frequency 8 kHz"<br>(from version 14.00.00) | 0x0000<br>(Off) | |
| Greyed out = display parameter | | | |

### Settable switching frequencies

| Selection in C00018 | | | |
|---|---|---|---|
| 1 | 4 kHz var./drive-optimised | 21 | 8 kHz var./drive-opt./4 kHz min |
| **2** | **8 kHz var./drive-optimised** | 22 | 16 kHz var./drive-opt./4 kHz min |
| 3 | 16 kHz var./drive-optimised | 23 | 16 kHz var./drive-opt./8 kHz min |
| 5 | 2 kHz constant/drive-optimised | 31 | 8 kHz var./min. Pv/4 kHz min |
| 6 | 4 kHz constant/drive-optimised | 32 | 16 kHz var./min. Pv/4 kHz min |
| 7 | 8 kHz constant/drive-optimised | 33 | 16 kHz var./min. Pv/8 kHz min |
| 8 | 16 kHz constant/drive-optimised | Abbreviations used:<br>• "var.": Adaptation of the switching frequency depending on the current<br>• "drive-opt.": drive-optimised modulation ("sine/delta modulation")<br>• "fixed": fixed switching frequencies<br>• "min. Pv": additional reduction of power loss | |
| 11 | 4 kHz var./min. Pv | | |
| 12 | 8 kHz var./min. Pv | | |
| 13 | 16 kHz var./min. Pv | | |
| 15 | 2 kHz constant/min. Pv | | |
| 16 | 4 kHz constant/min. Pv | | |
| 17 | 8 kHz constant/min. Pv | | |
| 18 | 16 kHz constant/min. Pv | | |

**Tip!**

The Lenze setting C00018 = "2: 8 kHz var.drive-opt." is the optimal value for standard applications.

### Preventing a decrease of the switching frequency

The value must not fall below the permissible switching frequency if the motor is not approved for the setting switching frequency or if a sinusoidal filter is connected between the output of the inverter and the motor that is not approved for the setting switching frequency.

In order to prevent a decrease of the switching frequency, the following settings have to be made:

1. The temporary switching frequency reduction has to be activated by the setting C00144 = "0: Off".
2. The switching frequency has to be set in C00018 so that the value is not fallen below the permissible switching frequency.
3. If a sinusoidal filter is available, only a drive-optimised switching frequency ("drive-opt.") is permissible.

Example: A sinusoidal filter is connected between the output of the inverter and the motor which only is to be operated with a minimum switching frequency of 8 kHz.

In this case, only the following settings in C00018 are permissible:

| Selection in C00018 | |
|---|---|
| 7 | 8 kHz constant/drive-optimised |
| 8 | 16 kHz constant/drive-optimised |
| 23 | 16 kHz var./drive-opt./8 kHz min |

## Note!

**Switching frequency for the motor parameter identification**

Up to and including version 13.xx.xx, the motor parameter identification is executed with a switching frequency of 4 kHz.

- Thus, a motor parameter identification is not permissible for the previously mentioned example as the sinusoidal filter may only be operated with a minimum switching frequency of 8 kHz!

From version 14.00.00, the motor parameter identification can also be executed with a switching frequency of 8 kHz instead of 4 kHz. For this purpose, the option "Motor ident.: Switching frequency 8 kHz" (Bit 4 = "1") has to be set in C02864/1.

- With this setting, a motor parameter identification would be permissible for the previously mentioned example.

Generally, a switching frequency of 4 kHz is recommended for the motor parameter identification as it serves to obtain the most accurate results.

▸ Automatic motor data identification (📖 145)

### Lowering the switching frequency due to high heatsink temperatures

Exceeding the maximally permissible heatsink temperature would lead to an inhibited drive due to the "Overtemperature" error and a torquelessly coasting motor. Therefore, if the Lenze setting is selected, the switching frequency is reduced to the next frequency below when the heatsink temperature has risen to 5 °C below the maximally permissible temperature. After the heatsink has cooled down, the inverter automatically switches to the next frequency above until the set switching frequency is reached.

Switching frequency reduction due to high heatsink temperature can be deactivated via C00144. If the switching frequency reduction is deactivated, the "oH1: Heatsink overtemperature" error message will be issued when the maximally permissible heatsink temperature is reached. An "Error" response is the result and the motor is coasting.

| Parameters | Info | Lenze setting |
|---|---|---|
| C00144 | Switching frequency reduction (temp.) | 1: On |

### Lowering of the switching frequency depending on the output current

"Variable" switching frequencies can be selected for the inverter in C00018, where the inverter automatically lowers the switching frequency depending on the inverter output current. The modulation mode will not be changed.

The changeover thresholds are listed in the **8400 hardware manual** in the rated data. The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 inverter.

When a "fixed" switching frequency is selected, no switching frequency changeover takes place. In case of fixed frequencies, the inverter output current is limited to the permissible value of the corresponding switching frequency. In case of larger load impulses, the overcurrent interruption may be activated, to which the inverter responds with "Error".

### Limiting the maximum output frequency

**Note!**

If the servo control mode (SC) is selected, the drive assumes the 'maximum current limitation' state if the maximum output frequency (C00910) has been reached.

▸ Defining current and speed limits (📖 158)

The maximum output frequency (C00910) of the inverter is not limited depending on the switching frequency. Therefore, adapt the maximum output frequency according to our recommendation:

$$\text{Maximum output frequency} \leq \frac{1}{8}\text{Switching frequency}$$

- At a switching frequency of 4 kHz, for instance, 500 Hz for the maximum output frequency should not be exceeded.

Carry out further measures:

- If required, deactivate the switching frequency changeover by the heatsink temperature via C00144.
- If required, ensure that the changeover threshold of the inverter output current to the next switching frequency below will not be exceeded. If required, select a constant switching frequency in C00018.

### Operation at an ambient temperature of 45°C

The inverter is designed so that operation at an ambient temperature of 45° C without derating is permissible at a switching frequency of 4 kHz.

### 5.10.2 Operation with increased rated power

Under the operating conditions described here and under continuous operation, the inverter can be operated with a higher power motor (increased rated power). The remaining overload capacity of the drive system (for 60 s/3 s) is reduced accordingly to approx. 120 %/160 %.

Typical applications stand out due to low dynamic requirements, e.g. pumps and fans, general horizontal materials handling technology and line drives.

The inverters which can be operated at an increased rated power are listed in the **8400 hardware manual** in the rated data. The hardware manual is stored in electronic form on the data carrier which is supplied with the 8400 inverter.

**Stop!**

Operation at increased rated power is only permitted ...

- with the inverters listed in the **8400 hardware manual** for this type of operation in the stated mains voltage range.
- at switching frequencies of 2 kHz and 4 kHz.
- at a max. ambient temperature of 40 °C.
- with the types of installation stated in the **8400 hardware manual**.
- with the fuses, cable cross-sections, mains chokes, and filters as required in the **8400 hardware manual** for this operation.
- after parameterisation according to the specifications below.

#### Required parameterisation

Operation at increased rated power requires the following settings to be made particularly for the V/f characteristic control (VFCplus), but also for all other types of control:

| Parameters | Info | Required setting |
|---|---|---|
| C00016 | VFC: Vmin boost | adapt to motor (reduce) |
| C00018 | Switching frequency | 1: 4 kHz var./drive-opt. |
| C00021 | Slip compensation | adapt to motor |
| C00120 | Setting of motor overload (I²xt) | adapt to motor |
| C00123 | Device utilisat. threshold (Ixt) | 120 % |
| C00173 | Mains voltage | see hardware manual → Rated data |

All other types of control require the following settings in addition:

| Parameters | Info | Required setting |
|---|---|---|
| C00022 | Imax in motor mode | higher than rated motor current (max. 160 % rated motor current) |
| C00081 | Rated motor power | adapt motor data (see motor nameplate), then carry out identification run ▸ Automatic motor data identification (📖 145) |
| C00087 | Rated motor speed | |
| C00088 | Rated motor current | |
| C00089 | Rated motor frequency | |
| C00090 | Rated motor voltage | |
| C00091 | Motor cos φ | |

### 5.10.3 Current-dependent stator leakage inductance Ppp(I)

**Note!**

Function only possible with:
- Servo control (SC) for synchronous motors
- Sensorless control for synchronous motors (SLPSM)

The current controller must be adjusted to the electrical characteristics of the motor stator resistance (C00084) and stator leakage inductance (C00085). In case of modern motors, the stator leakage inductance changes with the height of the current so that a new current controller setting is required for each current height.

When the motor is operated with very low and very high currents (e.g. in *Pick and place* applications), it is not always possible to achieve a satisfactory current controller setting for all operating points. For this purpose, the correction of the stator leakage inductance and current controller parameters is now possible via an adjustable saturation characteristic (17 interpolation points).

**Short overview of the relevant parameters:**

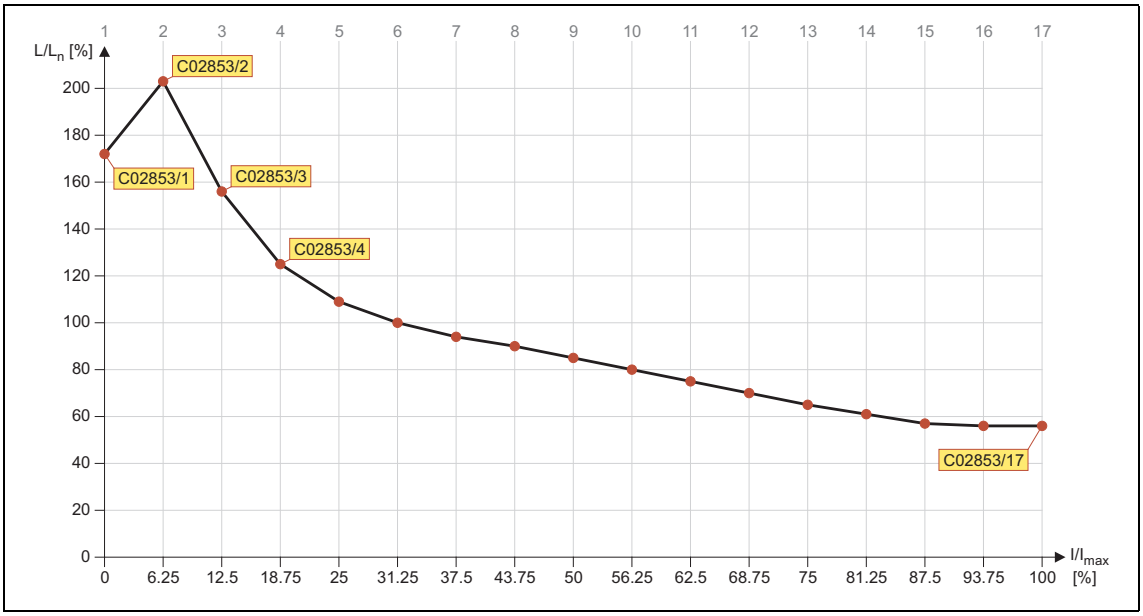
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02853/1...17 | PSM: Ppp saturation characteristic (17 interpolation points) | 100 | % |
| C02855 | PSM: Imax Lss saturation characteristic | 3000.0 | A |
| C02859 | PSM: Activate Ppp saturation char. | 0: Off | |

**Note!**

- The saturation characteristic is not only used for the correction of the current controller but also influences the current controller feedforward control (C00079/1).
- When a Lenze motor is selected from the »Engineer« motor catalogue, the corresponding saturation characteristic is set in C02853/1...17 and – if required – the correction via this saturation characteristic is switched on in C02859.
- For third-party motors: If the current controller becomes unstable with high currents, contact the motor manufacturer to find out whether the stator leakage inductance changes with the current level. If required, the saturation characteristic of this motor must be set in C02853/1...17 and then activated in C02859.

**Distribution of the grid points**

- The saturation characteristic is represented by 17 interpolation points linearly distributed on the X axis (C02853/1...17).
- Interpolation point 17 represents 100 % of the maximum motor current in the process (C02855).
- The following diagram shows the saturation characteristic stored in the »Engineer« motor catalogue for the Lenze motor "MCS12H15" as an example:




[5-21] Saturation characteristic: Inductance referring to the inductance for rated current

### 5.10.4 Flying restart function

The flying restart circuit for asynchronous motors uses a simple motor model which requires knowledge of the motor stator resistance RS and the rated motor current.

**Note!**

- Currently, the flying restart circuit is only available for asynchronous motors. (A flying restart circuit for synchronous motors is in preparation.)
- For a correct functioning of the flying restart circuit, we recommend to perform a parameter identification first. ▸ Automatic motor data identification (📖 145)
- The flying restart function works safely and reliably for drives with great centrifugal masses.
- Do not use the flying restart function if several motors with different centrifugal masses are connected to an inverter.
- After the controller is enabled, the motor can start for a short time or reverse when machines with low friction and low mass inertia are used.
- The flying restart function serves to identify max. field frequencies up to ±200 Hz.
- When power-adapted standard asynchronous motors are used (rated motor power approximately corresponds to the rated inverter power), a motor parameter identification is not required.
- On drive systems with feedback, you do not need to use the flying restart function because the synchronisation to the speed detected by the feedback is always carried out in a jerk-free manner.
- Especially in case of very high power, very high mass inertia and a mains voltage higher than 440 V, an overvoltage in the DC bus may be caused temporarily. Using a brake resistor may avoid this behaviour.
  - From version 15.00.00, frequency can be set as a remedy in C00989 which is added to the found flying restart frequency.
    Here, the setting in C00989 has to be increased in small steps up to double the slip frequency until the behaviour does not occur anymore.

**Tip!**

In association with the flying restart function, we recommend information provided in this documentation on the following topic:
▸ Automatic DC-injection braking (Auto-DCB) (📖 275)

### General information

This function serves to activate a mode which is used to "catch" a coasting motor during operation without speed feedback. This means that the synchronicity between inverter and motor is to be adjusted in such a way that a jerk-free transition to the rotating machines is achieved in the instant of connection.

The inverter determines the synchronicity by identifying the synchronous field frequency.

### Duration

The "catching" process is completed after approx. 0.5 ... 1.5 seconds. The duration is influenced by the starting value. If the field frequency is not known, we recommend a fixed starting value of 10 Hz (or -10 Hz with systems rotating in negative direction).

### Short overview of the relevant parameters:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00989 | Restart on the fly: Flying restart frequency fd_add (from version 15.00.00) | 0.00 | Hz |
| C00990 | Flying restart fct.: Activate | Off | |
| C00991 | Flying restart fct.: Process | -n...+n \| Start: +10 Hz | |
| C00992 | Flying restart: Start frequency | 5 | Hz |
| C00993 | Flying restart: Integration time | 300 | ms |
| C00994 | Flying restart: Current | 25.00 | % |



### How to parameterise the flying restart function:

1. Activate the flying restart circuit by selecting "1: On" in C00990.
   - Every time the controller is enabled, a synchronisation to the rotating or standing drive is carried out.

   When the Lenze setting is used, most applications do not require additional inverter settings.

   If additional settings are necessary, proceed as follows:
2. Define the process and hence the speed range/rotational frequency range in C00991 which is to be examined by the flying restart circuit:
   - positive speed range (n ≥ 0 rpm)
   - negative speed range (n ≤ 0 rpm)
   - total speed range
3. Define the starting frequency.

   The starting frequency which defines the starting point of the flying restart function is 10 or -10 Hz for processes 0 ... 3 and has been pre-optimised for standard motors.

   If process 4 is selected in C00991, an arbitrary starting frequency can be defined via C00992. This is especially recommended for motors with higher rated frequencies.
   - We recommend to define a starting frequency of approximately 20 % of the rated motor frequency to enable a safe and fast connection to standing drive systems.
   - For systems with a known search speed (e.g. torque-controlled drive systems which are to synchronise to a defined speed) the starting value can be adapted to reduce the flying restart time.
4. Set the flying restart current in C00994.

We recommend setting a flying restart current of 10 % ... 25 % of the rated motor current.

- During a flying restart process, a current is injected into the motor to identify the speed.
- Reducing the current causes a reduction of the motor torque during the flying restart process. A short-time starting action or reversing of the motor is prevented with low flying restart currents.
- An increase of the current improves the robustness of the flying restart function.
- At too high mass inertias and high speeds at the same time, the flying restart circuit may cause an overvoltage in the DC bus ("oU") if no brake resistor is connected. In this case, reduce the flying restart current.

**Tip!**

**Use of motors with higher rated frequencies**

For trouble-free operation, we recommend to manually enter a starting frequency of 20 % of the rated motor frequency in C00992 as well as to accelerate the flying restart process (see above) and to use a lower flying restart current (10 % of the rated motor current) if motors with higher rated frequencies are used.

**Optimisation of the flying restart time**

The duration of the flying restart process can be influenced via the setting of the integration time (C00993). A reduction of the integration time causes the flying restart function to accelerate and thus a reduced flying restart time.

- We recommend not to change the Lenze setting of the integration time.
- When special motor are used (e.g. multi-pole motors or ASM servo motors), a reduced integration time may improve the flying restart behaviour.

**Optimising the current controller if the behaviour is unstable**

During the execution of flying restart function, peak currents/torques are avoided by controlling the current amplitude.

Gain (C00075) and reset time (C00076) of the current controller can be adapted to improve the jerk-free/torque-free connection of the inverter to the supply of the rotating motor.

- We recommend not to change the Lenze setting of the current controller.
- If the behaviour of the current controller is unstable, gain and reset time can be calculated as per the following formulae:

$$V_p = \frac{L_{ss}[H]}{T_E[s]}$$

$$T_i = \frac{L_{ss}[H]}{R_s[\Omega]}$$

$V_p$ = Current controller gain (C00075)
$T_i$ = Current controller reset time (C00076)
$L_{ss}$ = Motor stator leakage inductance (C00085)
$R_s$ = Motor stator resistance (C00084)
$T_E$ = Equivalent time constant (= 500 µs)

[5-22] Formulae for the calculation of the gain and reset time of the current controller

### 5.10.5 DC-injection braking

**Note!**

The DC-injection braking function (manually or automatically) is only possible for the following motor control modes:

- V/f characteristic control (VFCplus) (📖 161)
- V/f characteristic control - energy-saving (VFCplusEco) (📖 180)
- V/f control (VFCplus + encoder) (📖 190)
- Sensorless vector control (SLVC) (📖 197)

**Danger!**

Holding braking is not possible when this braking mode is used!

For low-wear control of a holding brake, use the basic function "Holding brake control". (📖 673)

DC-injection braking allows the drive to be quickly braked to a standstill without the need to use an external brake resistor.

- The braking current is set in C00036.
- The maximum braking torque to be generated by the DC braking current is approx. 20 ... 30 % of the rated motor torque. It is lower than that for braking in generator mode with an external brake resistor.
- The DC-injection braking can be activated manually via a control input of the application or automatically ("Auto-DCB"). The two different modes are described in the following subchapters.

**Tip!**

DC-injection braking has the advantage that it is possible to influence the braking time by changing the motor current or the braking torque..

The automatic DC-injection braking (Auto-DCB) improves the starting performance of the motor in case of V/f operation without speed feedback.

**Short overview of the relevant parameters:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00019 | Auto-DCB: Threshold<br>• Operating threshold for activating DC-injection braking | 3 | rpm |
| C00036 | DC braking: Current<br>• Braking current in [%] based on rated device current (C00098) | 50 | % |
| C00106 | Auto-DCB: Hold time | 0.500 | s |
| C00107 | DC braking: Hold time<br>• When "999.000 s" is set, the hold time is infinite. | 999.000 | s |

### 5.10.5.1 Manual DC-injection braking (DCB)

DC-injection braking can be activated manually for the two technology applications "speed actuating drive" and "switch-off positioning" by connecting the *bSetDCBrake* input of the LA_NCtrl or LA_SwitchPos application block to a digital signal source (e.g. via a digital signal source *bCtrl1_B3* of the LP_CanIn1 port block).

- For HIGH-active inputs, DC-injection braking is active as long as the signal is at HIGH level.
- After the hold time (C00107) has expired, the inverter set the pulse inhibit.

**Short overview of the relevant configuration parameters:**

| Parameters | Info | Lenze setting |
|---|---|---|
| C00701/4 | LA_NCtrl: bSetDCBrake<br>• "Speed actuating drive" application: Selection of the signal source for manually activating DC-injection braking. | Dependent on the selected control mode |
| C00761/4 | LA_SwitchPos: bSetDCBrake<br>• "Switch-off positioning" application: Selection of the signal source for manually activating DC-injection braking. | Dependent on the selected control mode |

### 5.10.5.2 Automatic DC-injection braking (Auto-DCB)

"Automatic DC-injection braking" (referred to in the following as "auto DCB") can be used if there is a requirement that the drive be isolated from the supply at n ≈ 0.

**Note!**

**Deactivate automatic DC-injection braking when a holding brake is used!**

- For this purpose, go to C00019 and set the auto DCB threshold to "0".
- Background: Controller inhibit is already activated by the Holding brake control. (📖 673)

**Function**

For understanding the auto DCB function, it is necessary to distinguish between three different types of operation:

A. The drive has been enabled and, in the course of operation, the speed setpoint falls below the auto DCB threshold.
   - In case of V/f operation without speed feedback, a braking current (C00036) is injected. After the auto DCB hold time (C00106) has expired, the motor is deenergised via the auto DCB function, i.e. pulse inhibit is set.
   - In case of V/f operation with speed feedback, the motor is deenergised via the auto DCB function after the auto DCB hold time (C00106) has expired, i.e. pulse inhibit is set. The braking current which can be parameterised in C00036 does not have any effect during operation with speed feedback.

B. When the controller is enabled, the drive is at standstill (n = 0).
   If the enabled drive is to start, the speed setpoint passed via the acceleration ramp must exceed the auto DCB threshold (C00019). Below this threshold, the motor will not be energised.

C. When the controller is enabled, the motor (still) rotates at a speed which is above the auto DCB threshold. If the speed setpoint reached via the acceleration ramp exceeds the auto DCB threshold (C00019), the motor will be energised and the following action will take place:
   - During V/f operation without speed feedback, the drive is "caught". ▸ Flying restart function (📖 271)
   - During V/f operation with speed feedback, the drive synchronises to the current actual speed value.

**Tip!**

We recommend to deactivate the auto DCB function during V/f operation with speed feedback via a setting of C00019 = 0.

#### Auto DCB function during V/f operation with speed feedback

**Stop!**

If the DC-injection braking operation is too long and the braking current or braking voltage is too high, the connected motor may overheat!

If you want to use the auto DCB function contrary to our recommendation (see above), the auto DCB threshold must not fall below the following values depending on the number of encoder increments:

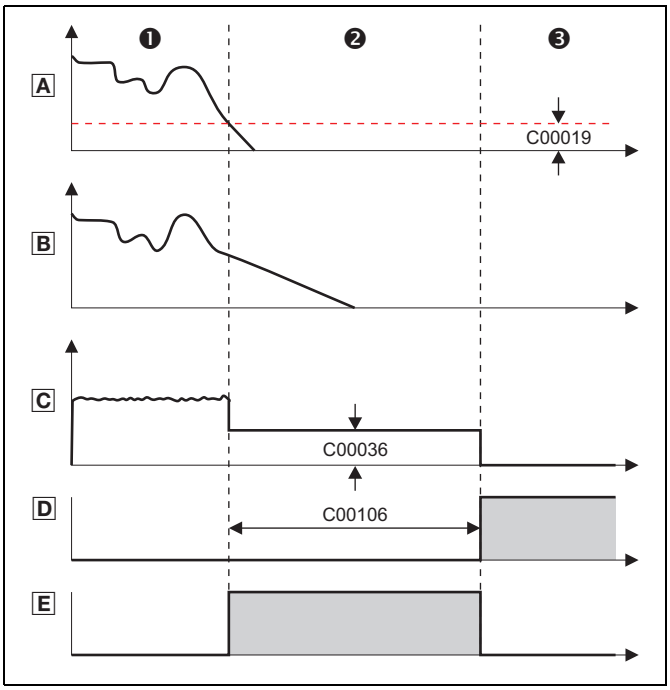
| Number of encoder increments (C00420) | Auto-DCB threshold (C00019) |
|---|---|
| 8 | 16 |
| 16 | 8 |
| 32 | 4 |
| 64 | 2 |
| > 128 | No restrictions |

#### Setting the auto-DCB function

**How to set the automatic DC-injection braking:**

1. Set a hold time in C00106 > 0 s.
   - Automatic DC-injection braking is active for the time set.
   - In case of V/f operation without speed feedback, the braking set in C00036 is injected.
   - After the set hold time has expired, the inverter sets pulse inhibit.
2. Set the operating threshold in C00019.
   - The operating threshold can serve to set a dead band in the setpoint. If DC-injection braking is not to be active then, C00106 must be set to a value of "0".

### Explanation of the automatic DC-injection braking function by means of two examples



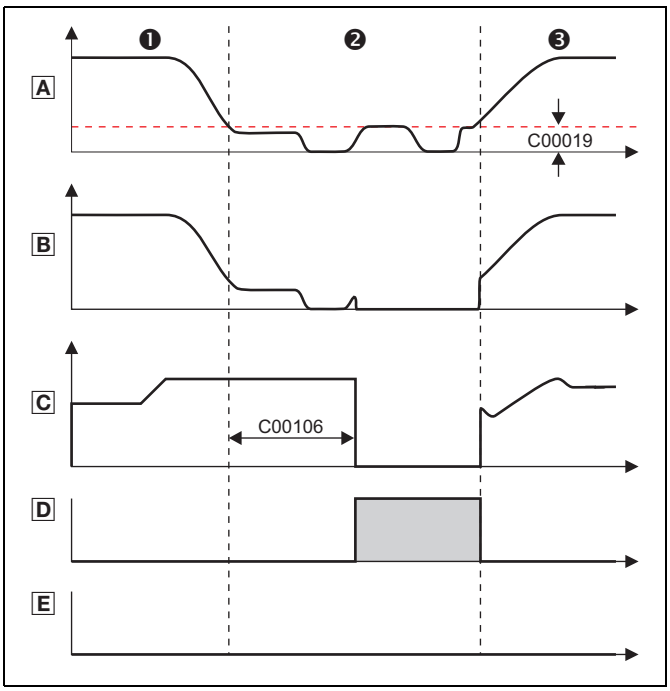

❶ The motor rotates at a specified speed. The current adjusts itself to the load, see C.

❷ The DC braking current set in C00036 is injected.

❸ After the hold time (C00106) has expired, a pulse inhibit is set.

A Speed setpoint
B Actual speed value of the motor
C Controller output current
D Pulse inhibit
E DC-injection braking is active

[5-23] Example 1: Auto-GSB signal characteristic during V/f operation without speed feedback




❶ The motor rotates at a specified speed. The current adjusts itself to the load, see C.

❷ The current sets itself load-dependent until the hold time (C00106) has expired.

❸ The actual speed value of the motor follows the speed setpoint. The resulting current depends on the load.

A Speed setpoint
B Actual speed value of the motor
C Controller output current
D Pulse inhibit
E DC-injection braking is active

[5-24] Example 2: Auto-GSB signal characteristic during V/f operation with speed feedback

### 5.10.6 Slip compensation

**Note!**

Slip compensation is only active with the following motor control modes:
- V/f characteristic control (VFCplus) (📖 161)
- Sensorless vector control (SLVC) (📖 197)

Under load, the speed of an asynchronous motor decreases. This load-dependent speed drop is called slip. The slip can partly be compensated for by the setting in C00021.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00021 | Slip compensation | 2.11 | % |

- The setting of C00021 can be done automatically in the course of motor parameter identification. ▸ Automatic motor data identification (📖 145)
- The setting must be made manually if the motor parameter identification cannot be called up.

**How to set the slip compensation manually:**

1. Set rated motor current (C00088) and rated motor frequency (C00089).
2. Calculate the slip compensation according to motor nameplate data:

$$s = \frac{n_{rsyn} - n_r}{n_{rsyn}} \cdot 100\%$$

$$n_{rsyn} = \frac{f_r \cdot 60}{p}$$

| | |
|---|---|
| $s$ | Slip constant (C00021) [%] |
| $n_{rsyn}$ | Synchronous motor speed [rpm] |
| $n_r$ | Rated motor speed according to the motor nameplate [rpm] |
| $f_r$ | Rated motor frequency according to the motor nameplate [Hz] |
| $p$ | Number of motor pole pairs (1, 2, 3 ...) |

3. Transfer the calculated slip constant s to C00021.
4. Correct the setting in C00021 while the drive is running until the load-dependent speed drop does not occur anymore between idling and maximum load of the motor in the desired speed range.

**Tip!**

The following guide value applies to a correctly set slip compensation:
- Deviation from the rated motor speed ≤ 1% for the speed range of 10 % ... 100 % of the rated motor speed and loads ≤ rated motor torque.
- Greater deviations are possible in the field weakening range.
- If C00021 is set too high, the drive may get unstable.
- Negative slip (C00021 < 0) with V/f characteristic control results in "smoother" drive behaviour at heavy load impulses or applications requiring a significant speed drop under load.

### 5.10.7 Oscillation damping

Mechanical oscillations are undesirable effects in every process and they may have an adverse effect on the single system components and/or the production output.

Mechanical oscillations in the form of speed oscillations are suppressed by the oscillation damping function.

Mechanical oscillations may occur:

- In the voltage range (output voltage is lower than max. voltage)
  - Here, the oscillations occur in no-load operation.
  - Here, speeds of 40 ... 80 % of the rated speed are typical.
  - See subchapter "Oscillation damping voltage range". (📖 280)
- In the field weakening range (output voltage has reached maximum voltage)
  - Here, the oscillations occur in no-load operation and with load.
  - Here, speeds higher than the rated speed are typical, especially when the output frequency is close to the mains frequency.
  - See subchapter "Oscillation damping in the field weakening range". (📖 281)

**Note!**

With servo control (SC), the Oscillation damping voltage range has no influence.

Mechanical natural frequencies can be suppressed or at least dampened in the speed control loop of the servo control by means of a current setpoint filter. ▸ Setting the current setpoint filter (band-stop filter) (📖 251)

Oscillation dampening, especially in the field weakening range, is also possible by activating the current controller feedforward control (C00079/1).

### 5.10.7.1 Oscillation damping voltage range

The oscillation damping voltage range is successfully used with

- unloaded motors (no-load oscillations)
- motors whose rated power deviates from the rated power of the inverter.
  - e.g. during operation at high switching frequency including the power derating involved.
- operation with higher-pole motors
- operation with special motors
- compensation of resonance in the drive
  - At an output frequency of approx. 20 ... 40 Hz, some asynchronous motors can show resonance which causes current and speed variations and thus destabilise the running operation.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00234 | Oscillation damping influence | 5.00 | % |
| C00235 | Oscillation damping filter time | 32 | ms |

**Note!**

Compensate the resonance during operation with feedback (closed loop, feedback of $n_{act}$) via the parameters of the slip regulator.

▸ Parameterising the slip regulator (📖 194)

**How to eliminate speed oscillations in no-load operation at speeds with 40 ... 80 % of the rated speed:**

1. Approach the area where the speed oscillations occur.
2. Reduce the speed oscillations by changing C00234 step by step (increment 1 %).
   - The filter time oscillation damping (C00235) should not be changed.
3. These can be indicators for smooth running:
   - Constant motor current characteristic
   - Reduction of the mechanical oscillations in the bearing seat

### 5.10.7.2 Oscillation damping in the field weakening range

When the max. possible output voltage (full modulation) has been reached, a voltage dip in the DC bus causes a voltage fluctuation in the motor. With load and during no-load operation this voltage fluctuation can cause mechanical oscillations.

The "oscillation damping field weakening" adjustable in C00236 serves to limit the maximum output voltage. This can be used to always compensate voltage dips in the DC bus to the output voltage (constant output voltage). This serves to prevent mechanical oscillations due to these voltage dips.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00236 | Oscillation damping field weakening<br>• Setting "0" ≡ 100 % output voltage can be reached | 14 | |

- With the Lenze setting of C00236 the limitation of the output voltage is set so that voltage dips in the DC bus in the output voltage for the single-phase and three-phase devices can largely be compensated so that no speed oscillations may be expected. Thus, an adaptation of C00236 is not required in the majority of cases.
- Maximum output voltage to be reached with Lenze setting of C00236:
  - Single-phase devices: 98.2 %
  - Three-phase devices: 99.7 %

**Note!**

The limitation of the output voltage via C00236 in the extreme field weakening range (high speeds) causes a reduction of the max. possible output torque (stalling torque).

- If the output torque to be reached in the extreme field weakening range is not sufficient (motor is stalling too early), reduce the setting in C00236.

With servo control (SC), the Lenze setting of C00236 should not be reduced. Otherwise the field weakening control could not work optimally anymore (the behaviour at high speeds may get worse). When servo control (SC) has been selected, the oscillation damping in C00236 should never be set to "0".

**How to eliminate speed oscillations in the field weakening range:**

1. Approach the area where the speed oscillations occur.
2. Reduce the speed oscillations by changing C00236 step by step (increment 1).
3. These can be indicators for smooth running:
   - Constant motor current characteristic
   - Reduction of the mechanical oscillations in the bearing seat

### 5.10.8 Phase sequence reversal for correcting misconnected UVW motor phases

**Note!**

Up to and including version 02.00.00, this function may only be activated for the following motor control types:

- V/f characteristic control (VFCplus) (📖 161)
- V/f characteristic control - energy-saving (VFCplusEco) (📖 180)

For all other motor control types, this function must not be activated because the set control mode would not work in that case!

From version 12.00.00, this function may be activated for all motor control types with the exception of the ones for synchronous motor.

The activation of this function does not affect the control types for synchronous motor since these control types require an in-phase connection of the synchronous motor.

If the motor phases are misconnected at the inverter output (e.g. phase u takes the place of phase v), the motor will rotate in the wrong direction.

To correct such misconnected motor phases, the rotating field of the inverter's output can be reversed by selecting "1: Inverted" in C00905. In this case, a phase will be reversed at the output of the inverter.

This function does not have any effect on setpoints and actual values, i.e. the polarity of the speed setpoint/actual speed value, actual torque, output frequency, and AngleOffset do not change.

**Tip!**

Cases of application for this function:

- Phase sequence reversal in case of misconnected motor phases.
- Setting of the correctly signed direction of rotation for inversely mounted motors.

### 5.10.9 Field weakening for synchronous motors

This function extension is available from version 02.00.00!

**Note!**

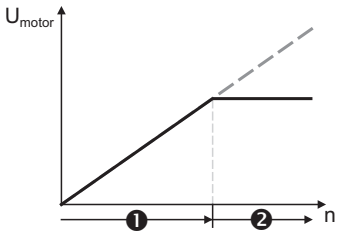
Function only possible with:
- Servo control (SC)
- Sensorless control for synchronous motors (SLPSM) (from version 10.00.00)

In the Lenze setting, the field weakening for synchronous motors is activated in C00079/4.
- If a high energy efficiency is required, keep the field weakening switched off or restrict the field weakening operation via C00938.

**Stop!**

In the field weakening operation, a current is injected into the synchronous motor even in idle state which can rise to maximum current (C00022).

Ensure that this no-load current does not cause the motor to be heated impermissibly!
- The use of a temperature feedback via KTY is recommended. ▸ Motor temperature monitoring (KTY) (📖 381)



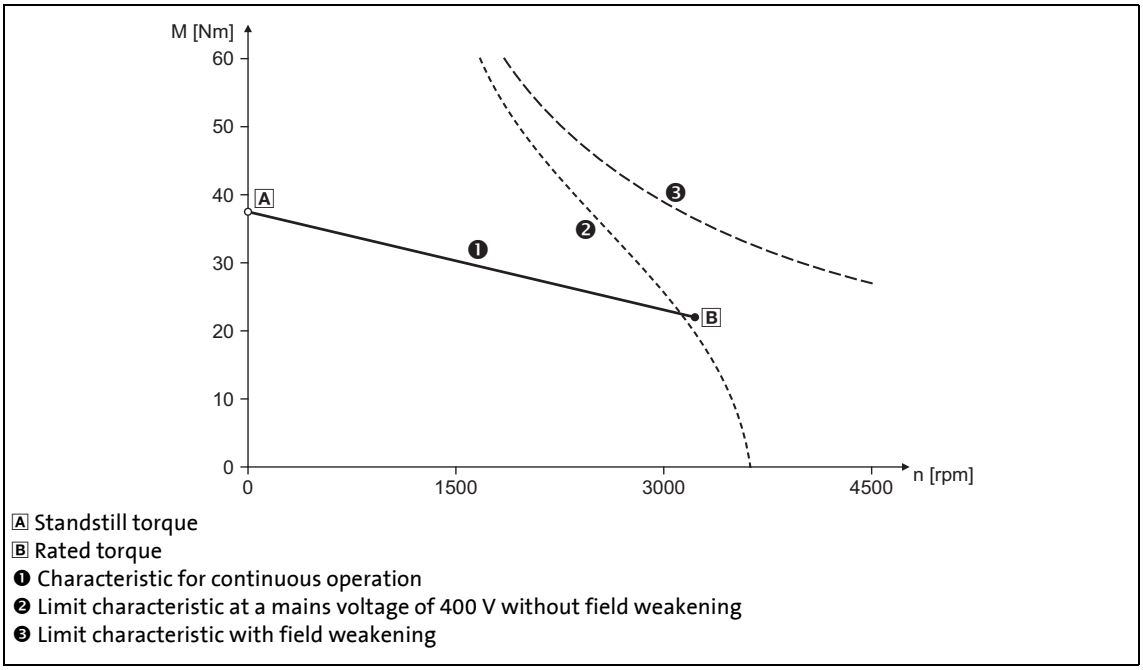

❶ Armature control range
❷ Field weakening range

- When field weakening is switched on, the motor magnetising current is increased from 0 A to the maximally effective magnetising current via an internal control loop when the voltage limit is reached.
- As a result, a higher speed can be reached at the same motor voltage or DC-bus voltage.

[5-25] Voltage/speed characteristic with switched-on field weakening

$$n_{max} = n_{nenn_mot} \cdot \frac{800V}{\sqrt{2} \cdot U_{nenn_mot}}$$

[5-26] Calculation of the maximally reachable speed with switched-on field weakening




A Standstill torque
B Rated torque
❶ Characteristic for continuous operation
❷ Limit characteristic at a mains voltage of 400 V without field weakening
❸ Limit characteristic with field weakening

[5-27] Speed/torque characteristics of a synchronous servo motor with field weakening

**Short overview of the relevant parameters:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00079/4 | Field weakening | 1: On | |
| C00938 | Limitation of maximally effective field-producing motor current<br>• With regard to rated motor current (C00088) | 30 | % |
| C00937/1 | Maximally effective field-producing motor current | - | A |
| Greyed out = display parameter | | | |

- The maximally effective field-producing motor current is calculated based on the motor data set in C00085, C00089 and C00098. Then, the value is internally limited to 98 % of the set maximum current (C00022 or maximally permissible current for the permanent switching frequency set in C00018).
- C00938 serves to limit the maximally effective field-producing motor current as well.
  - In the Lenze setting, the field weakening for synchronous motors is active (C00079/4). However, the field-producing motor current is limited via C00938 to 30 % of the rated motor current (C00088). Hence, the maximum speed is limited during field weakening operation and, at the same time, the temperature rise of the motor during field weakening operation and no-load operation is also limited.
  - If a higher speed for the field weakening operation is required or the current in the field weakening operation is to be limited (e.g. since no motor temperature detection is available and/or heating in the field weakening operation is to be limited), the value must be increased or reduced accordingly in C00938.

- In C000937/1, the actually used maximally effective field-producing motor current is displayed.
  - With switched-on and active field weakening: 0.00 A ... -x.xx A
  - With sensorless control for synchronous motors (SLPSM), the injected current is displayed in open-loop controlled operation: 0.00 A ... +x.xx A
  - If neither field weakening nor open-loop controlled operation are active,e "0.00 A" is displayed.

**Note!**

**If a Lenze motor is used:**

The inverter is automatically parameterised so that field weakening operates optimally and the maximally permissible speed is monitored.

**Stop!**

**If an OEM motor is used:**

If pulse inhibit is set in the inverter, the DC bus is loaded with the voltage that corresponds to the current speed of the machine.

Since with switched-on field weakening higher speeds can be achieved at a correspondingly higher rotor voltage of the motor, the DC bus can be loaded to a voltage higher than the set DC-bus voltage in case of pulse inhibit and a currently high motor speed and even exceed the maximally permissible voltage of 800 V!

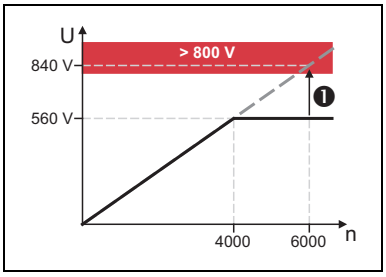
For device protection, either use a brake chopper or parameterise the motor speed monitoring via C00965 in such a way that only a maximum speed is possible which would be also reachable without field weakening with a DC-bus voltage of = 800 V.

▸ Motor speed monitoring (📖 311)

**Example: Voltage increase in the DC bus when field weakening is switched off**

(For instance by an active setting of the controller inhibit or by tripping a fault or error at high motor speed.)

| Field weakening | Speed n | Motor voltage peak value |
|---|---|---|
| Switched off | 4000 rpm | 560 V |
| | 5700 rpm | 800 V |
| | 6000 rpm | 840 V |
| Switched on | 6000 rpm | 560 V |




[5-28] Example: Possible DC-bus voltage > 800 V if field weakening gets lost

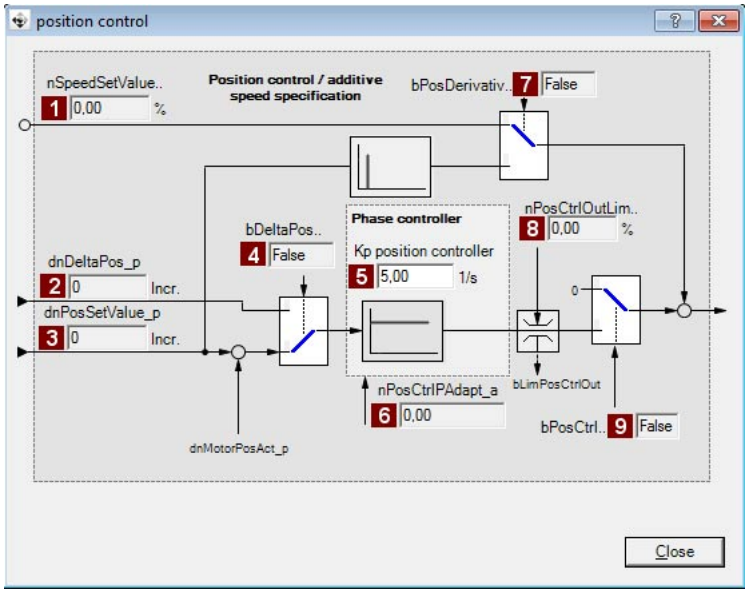
- If pulse inhibit occurs at 6000 rpm and switched-on field weakening, the DC bus is loaded to more than 800 V (❶).
- A speed limitation to 5700 rpm is required since this speed causes a DC-bus voltage of 800 V if field weakening is switched off.

## 5.11 Position control/additive speed specification

In the Lenze setting, the position control is only active with TA "Table positioning".

**Proceed as follows to open the dialog for parameterising the position control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Go to the *Overview* dialog level and click the **Motor control...** button to change to the *Overview* ➞ *motor control...* dialog level.
4. Click the **Position control** button in the displayed signal flow.




| Parameters | | Info | |
|---|---|---|---|
| 1 | C00830/22 | MCTRL: nSpeedSetValue_a | Speed setpoint |
| 2 | C00834/4 | MCTRL: dnDeltaPos_p | Position difference (following error input) |
| 3 | C00834/5 | MCTRL: dnPosSetValue_p | Absolute position setpoint |
| 4 | C00833/35 | MCTRL: bDeltaPosOn | TRUE = Position difference is active as setpoint selection |
| 5 | C00254 | Kp position controller | Gain for following error compensation |
| 6 | C00830/20 | MCTRL: nPosCtrlPAdapt_a | Adaptation of the position controller gain |
| 7 | C00833/67 | MCTRL: bPosDerivativeOn | TRUE = Setpoint for the speed controller is created from the position setpoint |
| 8 | C00830/21 | MCTRL: nPosCtrlOutLimit_a | Limitation of the position controller output |
| 9 | C00833/27 | MCTRL: bPosCtrlOn | TRUE = Position/angle control active |

## 5.12 Braking operation/brake energy management

When braking electrical motors, the kinetic energy of the drive train is fed back regeneratively to the DC bus. This energy results in a boost of the DC-bus voltage.

- Several different strategies can serve to avoid DC-bus overvoltages:
  - Use of a brake resistor
  - Stopping of the ramp function generator if brake chopper threshold exceeded (RFG_Stop)
  - Use of the "Inverter motor brake" function
  - Combination of the above named options
- In the case of inverters with a 3-phase supply, the following is also possible:
  - Coupling of the inverters in a DC-bus connection
  - Recovery of regenerative energy with a regenerative module




**Stop!**

If the connected brake resistor is smaller than required, the brake chopper can be destroyed!

- Appropriate protective measures are described in subchapter "Avoiding thermal overload of the brake resistor". (📖 296)

If no brake resistor or regenerative module is used, the overvoltage switch-off ("oU") may respond, e.g. in case of short deceleration times during feedback operation. ▸ Error messages of the operating system (📖 715)

### Using the integrated brake chopper

We recommend to use the brake chopper (brake transistor) which is integrated into the inverter for the braking operation, regardless of the selected motor mode.

- Connect the required brake resistor to the $R_{B1}$ and $R_{B2}$ terminals of the inverter.
- In C00175, a ramp function generator stop (FB L_NSet_1) can be set for instances when the brake resistor is controlled. This prevents overvoltage deactivation in the case of short deceleration times. ▸ Selecting the response to an increase of the DC-bus voltage (📖 291)

**Note!**

The brake transistor will be switched off if it remains switched on for a period of 4 seconds.

- If the DC-bus voltage falls under the brake chopper threshold for a short time again, the brake transistor can switch on again for a maximum of 4 seconds, without interruption.
- This protective function is used to prevent the brake chopper from being switched on permanently due to e.g. too high voltages at all times or incorrect interconnection of the *bBrakeChopperOn* signal (from version V12.00.00).
- From version 14.00.00, the limitation of the operating time can be deactivated to max. 4 seconds. For this purpose, set bit 1 to "1" in C02864/1.

The integrated brake chopper is always triggered via the DC-bus voltage with exceedance of the "voltage threshold for braking operation" independent of the device status.

From version 14.00.00, the brake chopper can also be switched off at pulse inhibit or when the "Fault" device status is activated.

- For a switch-off at pulse inhibit, set bit 7 to "1" in C02865/1.
- For a switch-off in the "Fault" device status, set bit 2 to "1" in C02864/1.

### DC-bus system

For a DC-bus connection with other devices, we recommend to connect the regenerative power supply module to terminals +UG and –UG.

**Note!**

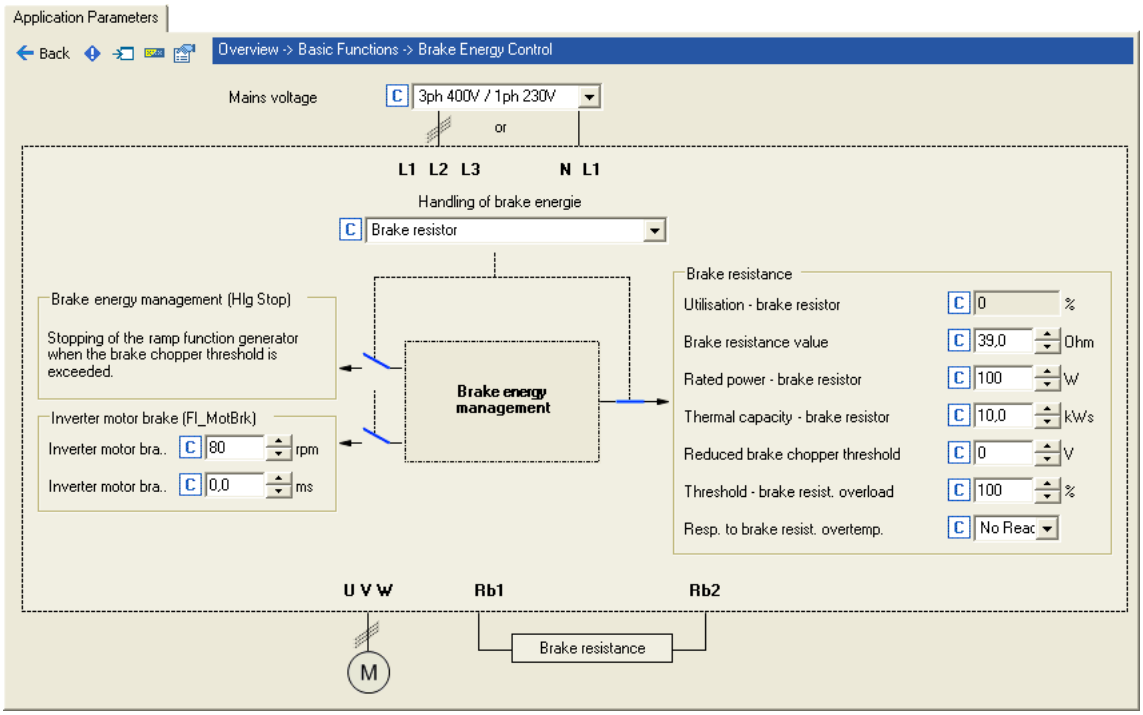
DC-bus system without the use of a regenerative module:

- Up to and including version V02.xx.xx, only one internal brake chopper can be used in the DC-bus system to dissipate regenerative energy.
- From version V12.00.00, all internal brake choppers can be used in the DC-bus system to dissipate regenerative energy ("Master-slave operation"). ▸ Control of multiple internal brake choppers in the DC-bus system (📖 296)

In order to install the regenerative module, follow the notes in the **8400 hardware manual**. The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 inverter.

**Proceed as follows to open the dialog for parameterising the brake energy management:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Go to the *Overview* dialog level and click the "basic functions" button.
4. Go to the *Overview* → *basic functions* dialog box and click the **Brake energy management** button.




**Short overview of the relevant parameters:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00173 | Mains voltage | 3ph 400 V / 1ph 230 V | |
| C00175 | Brake energy management | R_Brake (brake resistance) | |
| **Brake resistor** | | | |
| C00133 | Brake resistor utilisation | - | % |
| C00129 | Brake resistance value | 39.0 | Ohm |
| C00130 | Rated brake resistor power | 100 | W |
| C00131 | Thermal capacity - brake resistor | 10.0 | kWs |
| C00174 | Reduced brake chopper threshold | 0 | V |
| C00572 | Threshold - brake resist. overload | 100 | % |
| C00574 | Resp. to brake resist. overtemp. | No response | |
| **Inverter motor brake** | | | |
| C00987 | Inverter motor brake: nAdd | 80 | rpm |
| C00988 | Inverter motor brake: PT1 filter time | 0.0 | ms |
| Greyed out = display parameter | | | |

### 5.12.1 Setting the voltage source for braking operation

The voltage threshold for braking operation is set via the mains voltage (C00173) and the reduced brake chopper threshold (C00174). When this "brake chopper threshold" is exceeded, the response selected in C00175 takes place in the DC bus. The selected function (e.g. use of a brake resistor) serves to dissipate energy in the DC bus and reduce the DC-bus voltage.

- The "brake chopper threshold" is preset as follows so that it is higher than the specified mains voltage (C00173):

| Mains voltage | | | Brake chopper threshold | |
|---|---|---|---|---|
| C00173 | 1-phase | 3-phase | 1-phase | 3-phase |
| 0 | 1ph 230V | 3ph 400V | DC380V | DC725V |
| 1 | 1ph 230V | 3ph 440V | DC380V | DC735V |
| 2 | 1ph 230V | 3ph 480V | DC380V | DC775V |
| 3 | 1ph 230V | 3ph 500V | DC380V | DC790V |

- This brake chopper threshold can be reduced by 0 ... 150 V by means of C00174.

**Stop!**

The brake chopper threshold resulting from C00173 and C00174 must not exceed the stabilised DC-bus voltage!

Example :

- A 400 V device has a maximum mains voltage of 420 V AC.
  - Maximum stationary DC-bus voltage: 420 V AC * 1.414 = 594 V DC
  - C00173 has been set with the selection "0" for 400 V AC mains.
- This means that C00174 can be set to a maximum of 131 V DC (725 V DC - 594 V DC).

### 5.12.2 Selecting the response to an increase of the DC-bus voltage

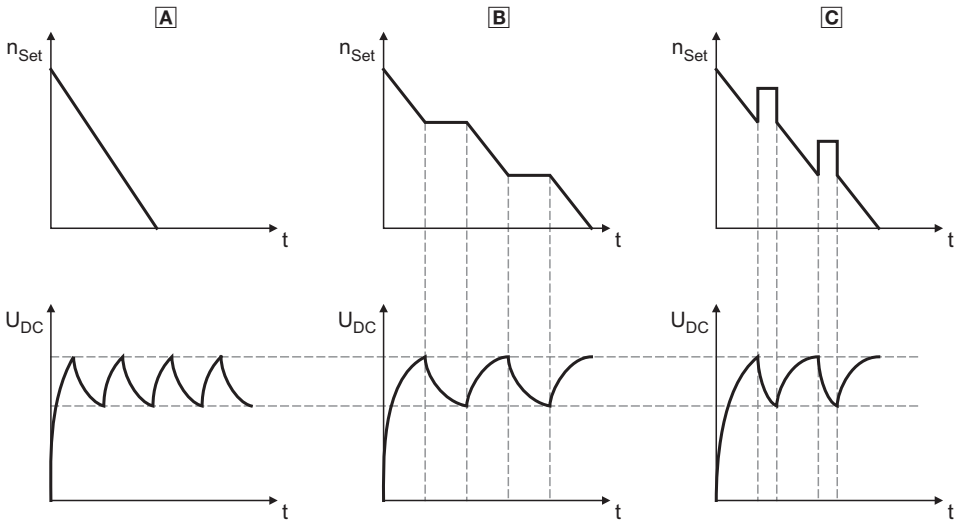
If the brake chopper threshold resulting from C00173 and C00174 is exceeded in the DC bus, the reaction selected in C00175 takes place (use of the brake resistor and/or stop of the ramp function generator and/or inverter-motor brake).

- Optimum following of the actual speed value until the speed setpoint is reached (e.g. the motor is stopped rapidly) is always achieved with the help of a brake resistor.
- Stopping the ramp function generator enables smoother deceleration with lower torque oscillation..
- The inverter-motor brake is available for selection in C00175. This function enables rapid braking without a brake resistor. Torque oscillations can occur due to the traversing dynamics. ▸ Inverter motor brake (📖 293)

## Stop!

- The two braking procedures "Stopping of the ramp function generator" and "Inverter motor brake" can only be used for speed-controlled applications without the influence of a position controller!
- When the "inverter motor brake" function is used, the Motor load monitoring (I2xt) is not adapted. If it is braked too frequently, there is a risk of the motor being thermally overloaded or the motor overload monitoring does not work properly!
- The "inverter motor brake" function must not be used with vertical conveyors (hoists) or with active loads!

The way in which the different braking procedures work is demonstrated schematically in the following illustration:




A Use of the brake resistor (C00175 = "0: Brake resistor")
B Stopping of the ramp function generator (C00175 = "1: HlgStop")
C Inverter motor brake (C00175 = "3: FI_MotBrk + RfgStop)

[5-29] Graph of the effective speed setpoint and the DC bus voltage during braking

## Tip!

Independent of the selected motor control, all procedures given in C00175 can be used. The actual speed value always follows the speed setpoint in an optimal way when a brake resistor is used.

If it is possible to dispense with exact adherence to the deceleration ramp in simple applications, selection of a braking method without an external brake resistor enables costs to be reduced due to the avoidance of having to use a brake resistor .

With the "inverter motor brake" function, an effective braking torque of 10 ... 20 % of the rated motor torque can be achieved.

A combination of all three braking procedures is also possible, e.g. for emergency braking if the brake resistor fails
(C00175 = "4: Brake resistor + FI_MotBrk + RfgStop").

#### 5.12.2.1 Inverter motor brake

With this braking method, which can be selected as an alternative in C00175, the regenerative energy in the motor is converted as a result of dynamic acceleration/deceleration with down-ramping of the ramp function generator..

**Stop!**

- This braking method only works without intervention of a position controller in the case of speed-controlled applications!
- When the "inverter motor brake" function is used, the Motor load monitoring (I2xt) is not adapted. If it is braked too frequently, there is a risk of the motor being thermally overloaded or the motor overload monitoring does not work properly!
- The "inverter motor brake" function must not be used with vertical conveyors (hoists) or with active loads!

**Tip!**

If no brake resistor is used, the DC injection brake can also be used for a braking process in addition to the "inverter motor brake" and "Stopping of the ramp function generator".
▸ DC-injection braking (📖 274)

In applications with high mass inertia and long braking times (> 2 s), we recommend the use of the DC injection brake.

- The DC injection brake provides for an oscillation-minimised braking. The braking process generally takes more time than the "inverter motor brake" function with an optimised setting. Moreover, the function is only recommended for braking to a standstill.

In the following cases we recommend the "inverter motor brake" function:

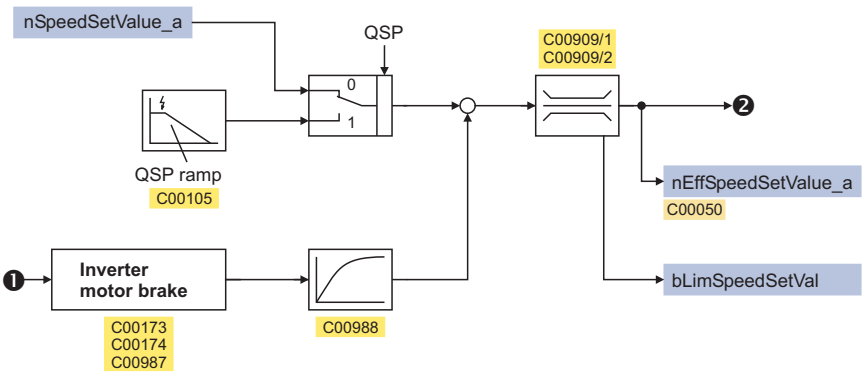
- With servo control (SC).
- For all applications that do not require braking to a standstill (e.g.braking to a lower speed setpoint) or the braking process can be interrupted by selecting a new speed setpoint.
- For applications with low mass inertias and a short braking time (< 1 s).
- For all applications where braking should be as quick as possible.

### Operating mode of the inverter motor brake

The ramp function generator is stopped during acceleration. The speed set in C00987 is added to the speed setpoint by means of a hysteresis-type 2-point DC bus voltage controller, whereby the sign of the current actual speed is taken into account. In addition, the ramp function generator is stopped during overvoltage.

If the DC bus voltage falls below a defined DC bus voltage potential of the hysteresis controller, the added speed is subtracted again and the ramp function generator is activated again.

The energy is converted into heat in the motor due to alternating instances of acceleration and deceleration as a result of this switching operation.




❶DC-bus voltage
❷ Speed setpoint for motor control

[5-30] Signal flow of the "Inverter motor brake" function

- In case of an asynchronous motor, the additive speed setpoint (C00987) should be 1 ... 4 times the slip of the machine:

$$C00987\,[rpm] = 1 \ldots 4 \cdot (n_{Sync}[rpm] - n_{Rated}[rpm])$$

$$n_{Sync}[rpm] = \frac{f_{Rated}Hz \cdot 60}{p}$$

p = number of pole pairs
$n_{Rat}$ = Rated speed of the motor
$f_{Rat}$ = Rated frequency of the motor
$n_{Sync}$ = Synchronous speed of the motor

[5-31] Formula for calculating the additive speed setpoint for an asynchronous motor

- In case of a synchronous motor, the additive speed setpoint (C00987) should be 5 ... 20 % of the rated machine speed.

**Short overview of the relevant parameters:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00173 | Mains voltage | 3ph 400V / 1ph 230V | |
| C00174 | Reduced brake chopper threshold | 0 | V |
| C00175 | Resp. to brake resistor control | Brake resistor | |
| C00987 | Inverter motor brake: nAdd<br>• Speed lift which is connected in pulses to the brake ramp when the motor is braked. | 80 | rpm |
| C00988 | Inverter motor brake: PT1 filter time<br>• PT1 filter time for smoothing the speed lift which is added in pulses. | 0.0 | ms |

## Note!

When the "inverter motor brake" function is used, torque oscillations occur which have may have a negative effect on the service life of the components of the mechanical drive train (e.g. gearbox).

- The extent of the occurring oscillations depends on the drive train (mass inertia, natural frequencies, etc.) and the function setting.
- We recommend optimising the "inverter motor brake" function for an oscillation-free operation as described in the following. Usually, this setting does not cause any torque oscillations which affect the service life of the gearbox.
- The settings of implementing a maximum acceleration ramp are only recommended if the inverter motor brake is used infrequently (e.g. in case of quick stop).

**How to set the "inverter motor brake" function for an oscillation-reduced operation:**

For V/f characteristic open-loop control/closed-loop control (VFCplus):

- Set reduced brake chopper threshold (C00174) to approx. 70 V.
- Set additive speed (C00987) to rated slip speed.
- Adapt the deceleration ramp so that the deceleration time is slightly below (10 ... 30 %) the deceleration time that can be realised with the inverter motor brake.

For sensorless vector control (SLVC) and servo control (SC):

- Set reduced brake chopper threshold (C00174) to approx. 50 V.
- Set additive speed (C00987) to 1 ... 2-fold rated slip speed.
- Adapt the deceleration ramp so that the deceleration time is slightly below (10 ... 30 %) the deceleration time that can be realised with the inverter motor brake.

**How to set the "inverter motor brake" function for a maximum acceleration ramp:**

For V/f characteristic open-loop control/closed-loop control (VFCplus):

- Set reduced brake chopper threshold (C00174) to approx. 70 V.
- Set additive speed (C00987) to 1,5 ... 2,5-fold rated slip speed.
- Adapt the deceleration ramp so that the deceleration time is slightly below (10 ... 30 %) the deceleration time that can be realised with the inverter motor brake.

For sensorless vector control (SLVC) and servo control (SC):

- Set reduced brake chopper threshold (C00174) to approx. 70 V.
- Set additive speed (C00987) to 2 ... 4-fold rated slip speed.
- Adapt the deceleration ramp so that the deceleration time is slightly below (10 ... 30 %) the deceleration time that can be realised with the inverter motor brake.

### 5.12.3 Avoiding thermal overload of the brake resistor

- Parameterisation of an error response in C00574 and evaluation of the parameterised error message within the application or within the machine control system.
  - See chapter entitled "Brake resistor monitoring (I2xt)". (📖 305)
- External interconnection using the thermal contact on the brake resistor (e.g. supply interruption via the mains contactor and activation of the mechanical brakes).

### 5.12.4 Control of multiple internal brake choppers in the DC-bus system

This function extension is available from version 12.00.00!

If an additional control signal is used, all internal brake choppers can be used in the DC-bus system to dissipate regenerative energy ("Master-slave operation").

**Stop!**

The integration of external brake choppers (e.g. brake chopper 9352) in the above-described "Master-slave operation" is not permissible since the voltage levels for the input and output of the external brake chopper are not suitable for the brake transistor control or, more precisely, for the output of the state of the brake transistor.

If the internal brake choppers of the DC-bus system are not sufficient, they should be replaced by an external brake chopper. It can be synchronised with other external brake choppers, if need be, so that simultaneous switch-on of all external brake choppers is ensured.

**Note!**

For trouble-free operation, the setting of the mains voltage in C00173 must be identical for all inverters of the DC-bus system since this setting also influences the brake chopper threshold for switching on the brake chopper.

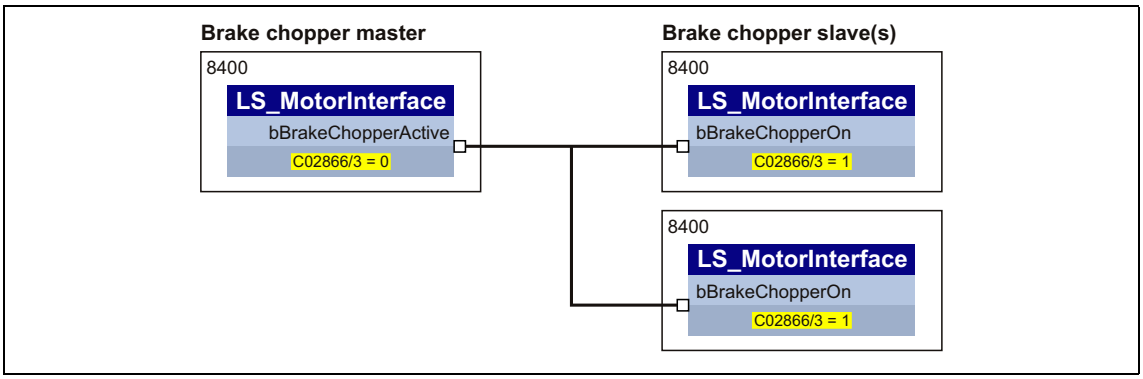
### Functional principle

One of the inverters of the DC-bus system is assigned the role of the "brake chopper master".

- For logical reasons, the "brake chopper master" should be the most powerful inverter.
- The "brake chopper master" controls its internal brake chopper via the DC-bus voltage as before. In addition, the "brake chopper master" transmits the *bBrakeChopperActive* status signal of its internal brake chopper control to the other inverters of the DC-bus system via fieldbus or digital output.

All the other inverters of the DC-bus system are "brake chopper slaves".

- The "brake chopper slaves" have the *bBrakeChopperActive* status signal received from the "brake chopper master" connected to the *bBrakeChopperOn* control input.
- If the internal brake transistor of the "brake chopper master" is switched on, the internal brake transistors of the "brake chopper slaves" are switched on at the same time.




[5-32] Functional principle of the "Brake chopper master-slave operation" (simplified representation)

### Procedure

1. Assign the role of the "brake chopper master" to one of the inverters of the DC-bus system.
2. Adapt the function block interconnection for the "brake chopper master" so that the *bBrakeChopperActive* status signal of the LS_MotorInterface SB is provided to the other inverters for controlling the internal brake chopper.
   - The *bBrakeChopperActive* signal can e.g. be output via port block to the fieldbus or via digital output.
   - A free output of the application block can be used to transfer the signal from application level to I/O level.
3. Configure all the other inverters of the DC-bus system as "brake chopper slaves". Select "1: Yes" in C2866/3 for these inverters.
   - With this setting, the brake chopper is not controlled via the DC-bus voltage anymore. Its control now depends on the *bBrakeChopperOn* control signal.
4. Adapt the function block interconnection for the "brake chopper slaves" so that the *bBrakeChopperActive* signal received from the "brake chopper master" is connected to the *bBrakeChopperOn* input of the LS_MotorInterface SB.
   - Depending on the output at the "brake chopper master", the signal must be read in e.g. via port block or digital input.
   - A free input of the application block can be used to transfer the signal from I/O level to application level.
   - If the digital inputs/outputs are used for transmitting the signal, they must be connected electrically accordingly.

**Note!**

If the brake chopper master-slave operation is activated, plausibility monitoring is carried out in the "brake chopper slaves":

- The brake transistor can only be switched on via the *bBrakeChopperOn* control signal if the DC-bus voltage is greater than the brake chopper threshold minus 40 $V_{DC}$.
- Exception: From a mains voltage of 513 $V_{AC}$ (or 725 $V_{DC}$) and a mains voltage of 480 V or 500 V set in C00173, the plausibility monitoring is not effective anymore.

**Fast discharge of the DC bus**

The *bBrakeChopperOn* control input of the "brake chopper master" can be optionally used for a fast discharge of the DC bus after the supply voltage has been switched off. The FB interconnection f the "brake chopper master" has to be adapted so that the *bBrakeChopperOn* control input of the SB LS_MotorInterface is connected to a digital bus or hardware signal (e.g. digital input on HIGH level).

**Note!**

Only the brake transistor of the "brake chopper master" is switched on with this function (max. 4 seconds, without interruption).

A complete discharge of the DC-bus cannot be achieved using this function!

From version 14.00.00, the limitation of the operating time can be deactivated to max. 4 seconds. For this purpose, set bit 1 to "1" in C02864/1.

- We recommend this setting when this function is used in order to achieve a higher discharge of the DC bus - especially in case of high device power.

## 5.13 Monitoring

Many monitoring functions that are integrated in the inverter can detect errors and thus protect the device/motor from damage or overload.

- Detailed information on the individual monitoring functions can be found in the following subchapters.

| Monitoring | Response | | Error message (with activated monitoring) |
|---|---|---|---|
| | Lenze setting | Configuration | |
| Device overload monitoring (Ixt) | Warning | C00604 | oC5 |
| Motor load monitoring (I2xt) | Warning | C00606 | oC6 |
| Motor overcurrent monitoring | Fault | - | oC7 |
| Motor temperature monitoring (PTC) | Fault | C00585 | oH3 |
| Brake resistor monitoring (I2xt) | No Reaction | C00574 | oC12 |
| Motor phase failure monitoring | No Reaction | C00597 | LP1 |
| Motor phase error monitoring before operation | No Reaction | C02866/2 | |
| Mains phase failure monitoring | Warning | C00565 | Su02 |
| Current monitoring for overload | No Reaction | C00584/1 | oC18 |
| Maximum current monitoring | No Reaction | C00609 | oC10 |
| Maximum torque monitoring | No Reaction | C00608 | ot1 |
| Motor speed monitoring | Fault | - | oS2 |

### Parameterisable responses

If a monitoring function trips, the response set via the corresponding parameter is carried out. The following responses can be selected:

- "No response": Response/monitoring is deactivated.
- "Fault": Change of the operating status by a pulse inhibit of the power output stage.
- "Warning": Operating status of the inverter remains unchanged. Only a message is entered into the Logbook of the inverter.

### Related topics:


- Device state machine and device states (📖 116)
- Diagnostics & error management (📖 691)
- Basics on error handling in the inverter (📖 691)
- Error messages of the operating system (📖 715)

### 5.13.1 Device overload monitoring (Ixt)

C00064/1...3 displays the device utilisation (ixt) in [%] in different time intervals:

| Parameters | Info |
|---|---|
| C00064/1 | Device utilisation (Ixt)<br>• Maximum value of pulse utilisation (C00064/2) and permanent utilisation (C00064/3). |
| C00064/2 | Device utilisation (Ixt) 15s<br>• Pulse utilisation over the last 15 seconds (only for loads >160 %). |
| C00064/3 | Device utilisation (Ixt) 3 min<br>• Permanent utilisation over the last 3 minutes. |
| Greyed out = display parameter | |

- If the device utilisation reaches the switch-off threshold set in C00123:
  - The error response set in C00604 will be carried out (Lenze setting: "Warning").
  - The "oC5: Ixt overload" error message will be entered into the Logbook.
  - The *bMctrlIxtOverload* status output of the LS_DeviceMonitor system block will be set to TRUE.
- A setting of C00604 = "0: No Reaction" deactivates the monitoring.

### 5.13.2 Motor load monitoring (I2xt)

The Inverter Drives 8400 are provided with a simple, sensorless, thermal $I^2xt$ motor monitoring of self-ventilated standard motors which is based on a mathematical model.

- C00066 displays the calculated motor load in [%].
- If the calculated motor load reaches the motor load setting (C00120):
  - The error response set in C00606 will be carried out (Lenze setting: "Warning").
  - The "oC6: I2xt motor overload" error message will be entered into the Logbook.
  - The *bMctrlI2xtOverload* status output of the LS_DeviceMonitor system block will be set to TRUE.
- A setting of C00606 = "0: No Reaction" deactivates the monitoring.

**Stop!**

$I^2xt$ motor monitoring does not provide full motor protection!
As the motor utilisation calculated in the thermal motor model is lost after mains switching, for instance the following operating states cannot be detected correctly:

- Restarting (after mains switching) of a motor that is already very hot.
- Change of the cooling conditions (e.g. cooling air flow interrupted or too warm).

Full motor protection requires additional measures such as the evaluation of temperature sensors that are located directly in the winding or the use of thermal contacts.

For the installation according to UL or UR, the safety instructions provided in the hardware manual must be observed! Among other things, the activation of the motor overload monitoring (I2xt) is required here.

**Note!**

From version 12.00.00, the thermal motor load displayed in C00066 can be pre-initialised when the device is connected to the mains, optionally using a fixed value or the value used last at the time when the device was switched off. The desired initialisation is selected in C00122. In the Lenze setting of C00122, the behaviour remains unchanged (no initialisation).

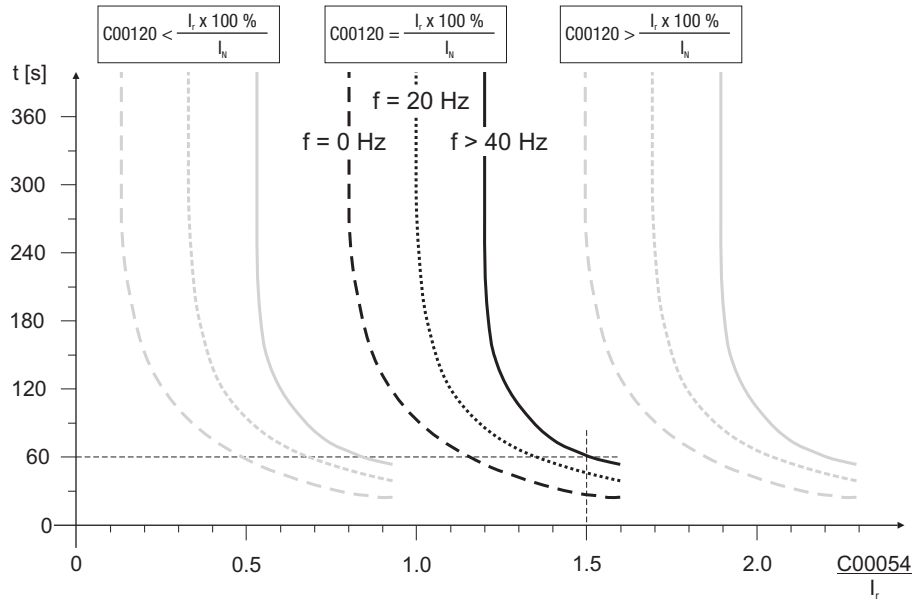
**Adjustment of the motor utilisation meter**

The motor utilisation meter for indicating the motor load in C00066 begins to count when the apparent motor current (C00054) is greater than the motor overload setting (C00120).

The overload threshold (C00120) is to be set as follows:

$$C00120 = \frac{\text{Rated motor current (C00088)}}{\text{Rated device current (C00098)}} \cdot 100\ \%$$

- If you reduce C00120 starting from the calculated value, the motor utilisation meter will already be counted up before the rated overload threshold is reached.
- If you increase C00120 starting from the calculated value, the motor utilisation meter will not be counted up until the rated overload threshold is reached.




f: Output frequency
t: Release time
$I_N$: Rated device current at a switching frequency of f = 8 kHz
$I_r$ : Rated motor current (see nameplate of motor)
C00054: Apparent motor current
C00120: Motor overload setting

[5-33] Tripping characteristic of the $I^2xt$ monitoring

Example in Figure [5-33]:

C00120 = $I_r / I_{rated}$ x 100 %

C00054 = 150 % rated motor current

- After approx. 60 seconds, C00066 has reached the final value (100 %) at output frequencies f > 40 Hz.
- The inverter outputs the "oC6: I2xt overload motor" error message and triggers the response set in C00606 (default setting: "Warning").

**Tip!**

- If forced ventilated motors are used, a premature response of the overload threshold can be avoided by deactivating this function if necessary (C00606 = "0: No Reaction").
- The current limits set in C00022 and C00023 influence the $I^2xt$ calculation only in an indirect way. However, the operation of the motor at maximum possible load can be averted. ▸ Defining current and speed limits (📖 158)

### 5.13.3 Motor overcurrent monitoring

The ultimate motor current to be parameterised in C00939 is a limit value to protect the motor from destruction, influence of the rated data and demagnetisation.

- This limit value must not be travelled cyclically in the drive process.
- If the instantaneous value of the motor current exceeds the limit value set in C00939, the error response "Fault" occurs to protect the motor and the error message "oC7: Motor overcurrent" is entered into the logbook.
- The maximum currents to be parameterised in C00022 and C00023 should have a sufficient distance to this limit value.

**Note!**

If a Lenze motor is selected from the catalogue whose plant parameters are transferred into the inverter, the setting of the maximum current in C00022 and C00023 will automatically be adapted to the selected motor.

**Related topics:**

▸ Maximum current monitoring (📖 311)

### 5.13.4 Motor temperature monitoring (PTC)

For detecting and monitoring of the motor temperature, a PTC thermistor (DIN 44081/DIN 44082) or a thermal contact (NC contact) can be connected to the terminals X106/T1 and X106/T2.

**Stop!**

- The inverter can only evaluate one PTC thermistor!
  Do not connect several PTC thermistors in series or parallel.
- If several motors are operated on one inverter, use thermal contacts (NC contacts) connected in series.
- To achieve full motor protection, an additional temperature monitoring with separate evaluation must be installed.

**Note!**

- In the Lenze setting (C00585 = "1: Fault"), motor temperature monitoring is activated!
- There is a wire jumper between the terminals X106/T1 and X106/T2 by default.
- Lenze three-phase AC motors are provided with a thermal contact on delivery.

- If 1.6 kΩ < R < 4 kΩ at the terminals X106/T1 and X106/T2, the monitoring will respond, see functional test below.
- If the monitoring responds:
  - The error response set in C00585 is activated (Lenze setting: "Fault").
  - The "oH3: Motor temperature (X106) triggered" error message is entered into the Logbook.
  - The *bMctrlMotorPtc* status output of the LS_DeviceMonitor system block is set to TRUE.
- A setting of C00585 = "0: No Reaction" deactivates the monitoring.

**Tip!**

We recommend to always activate the PTC input when using motors which are equipped with PTC thermistors or thermostats. This prevents the motor from being destroyed by overheating.

#### Functional test

Connect a fixed resistor to the PTC input:

- R > 4 kΩ : Fault message must be activated.
- R < 1 kΩ : Fault message must not be activated.

### 5.13.5 Brake resistor monitoring (I2xt)

Due to the converted braking power, the brake resistor is thermally stressed and can even be thermally destroyed by excessive braking power.

The monitoring of the $I^2xt$ utilisation of the inverter serves to protect the brake resistor. It acts in proportion to the converted braking power.

**Danger!**

In the Lenze setting (C00574 = "0: No Reaction"), the response of the monitoring function does not stop the braking process!

In particular for applications such as hoists or applications with a DC-bus connection, it must be checked if a stopping of the braking process due to a setting of C00574 = "1: Fault" is permissible.

**Stop!**

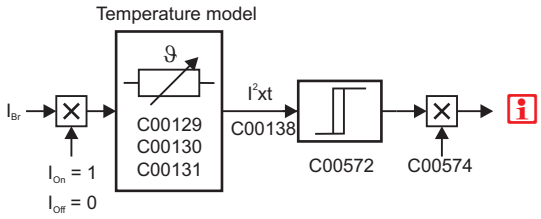
Implement appropriate protective measures against thermal overload of the brake resistor!

Examples:

- Parameterisation of an error response in C00574 and evaluation of the parameterised error message within the application or the machine control system.
- Interruption of the mains supply by means of the temperature contact at the brake resistor and a simultaneous activation of the mechanical brake.

- If the $I^2xt$ utilisation reaches the switch-off threshold set in C00572:
  - The error response set in C00574 will take place.
  - The "oC12: I2xt brake resistor overload" error message is entered into the Logbook.
  - The *bMctrlBrakeChopperFault* status output of the LS_DeviceMonitor system block will be set to TRUE.
- If the system is dimensioned correctly, the monitoring should not be activated. If individual pieces of rated data of the actually connected brake resistor are not known, they have to be identified.
- If the DC-bus voltage exceeds the overvoltage threshold due to a braking energy that is too high, the monitoring for overvoltage in the DC bus is activated ("OU: DC-bus overvoltage" error message).
- Apart from the threshold of the $I^2xt$ utilisation that can be set in C00572, there is the switching threshold of the brake transistor which results from the mains voltage (C00173) and the reduced brake chopper threshold (C00174).

### Temperature model




[5-34] Signal flow for monitoring the brake resistor

The monitoring function calculates the braking current $I_{Br}$ from the current DC-bus voltage $U_{DC_act}$ and the brake resistance parameterised in C00129:

$$I_{Br} = \frac{U_{DC_act}}{C00129}$$

**Note!**

The monitoring function can also be triggered due to a value entered in C00129 although a brake resistor is not even connected.

- The calculation considers the thermal utilisation of the brake resistor based on the following parameters:
  - Resistance value (C00129)
  - Continuous power (C00130)
  - Thermal capacity (C00131)
- In the Lenze setting these parameters are preset with the corresponding power-adapted Lenze brake resistor.
- C00133 indicates the calculated utilisation of the brake resistor in [%].
  - A utilisation of 100 % corresponds to the continuous power of the brake resistor depending on the maximally permissible temperature limit.

**Related topics:**

▸ Braking operation/brake energy management (📖 288)

### 5.13.6 Motor phase failure monitoring

**Note!**

In the Lenze setting (C00597 = "0: No Reaction"), the motor phase failure monitoring is not activated!

In the case of a synchronous motor,

- the motor phase failure monitoring is basically deactivated. (Due to the low no-load current, the monitoring would be permanently activated.)
- only the Motor phase error monitoring before operation is active (for Lenze setting C2866/2 = "1: Yes").

In order to safely detect the failure of a motor phase, a certain motor current must flow for the current sensor system. Thus, the response set in C00597 (Lenze setting: "No Reaction") is caused after a delay time of maximally 2 s after controller enable if a current-carrying motor phase U, V, W fails or if motor connection is missing. If the current threshold value set in C00599 is already exceeded within the delay time, the motor phase failure monitoring starts from this point in time.

The monitoring mode checks the current flow for each motor phase as a function of the commutation angle. Monitoring is activated if a commutation angle of approx. 140° is covered without the current set in C00599 being exceeded. Monitoring is activated at an output frequency of 0 Hz if none of the three motor phases reaches the threshold value set in C00599.

- If the motor phase failure monitoring is tripped:
  - The response set in C00597 will take place.
  - The error message "LP1: Motor phase failure" is entered into the logbook.
  - The *bMctrlMotorPhaseFault* status output of the LS_DeviceMonitor system block is set to TRUE.

**Note!**

If an error response of "1: Fault" is set in C00597, the *bMctrlMotorPhaseFault* status output of the LS_DeviceMonitor SB will be set to TRUE for only 1 second in the event of a motor phase failure because it is no longer possible to detect a motor phase fault via the error response with a pulse inhibit. However, the Logbook and C00561/3...5 still display the cause of the motor phase failure.

- The motor phase failure monitoring is inactive if
  - a controller inhibit is set,
  - connection to a rotating machine is carried out (flying restart circuit or connection to actual speed value),
  - an error is pending due to a DC-bus overvoltage ("OU"),
  - motor parameter identification is carried out,
  - DC-injection braking is active.

### 5.13.7 Motor phase error monitoring before operation

This function extension is available from version 02.00.00!

This extended motor phase failure monitoring can both detect a phase failure on the basis of test signals and check for the existence of the motor.

- The "motor phase error monitoring before operation" is only directly active after controller enable if
  - an error response is set in C00597 **AND**
  - the motor phase error monitoring is switched on (C2866/2 = "1: Yes").
- The following parameters show the cause of the motor phase failure:
  - C00561/3: Motor phase U
  - C00561/4: Motor phase V
  - C00561/5: Motor phase W

**Note!**

The motor phase error monitoring before operation must not be connected to a rotating or coasting machine (high compensation currents and effect of the DC injection braking).

- In case of motor control with feedback, no motor phase error monitoring is executed if the actual speed value is > 10 rpm.
- In case of motor control without feedback, the user must ensure that the motor phase error monitoring will only be executed if the speed is 0.

If the motor is at quick stop and the brake is applied, no motor phase error monitoring is executed when quick stop is deactivated (same with "0" speed and applied brake).

If the rated current of the connected motor is lower than 10 % of the rated device current, the motor phase error monitoring can be activated although no motor phase error has occurred. In this case, the motor phase error monitoring must be switched off before operation (C2866/2 = "0: No").

## Note!

**With automatic brake control:**

In case of automatic brake control, the brake will only be released if no motor phase failure exists and the magnetisation of the field-oriented control types is completed.

**With manual brake control:**

In case of manual brake control and forced release of the brake, the brake will be controlled directly as before.

The user himself must ensure that the brake will only be opened if all of the following conditions are met:

- Motor phase failure monitoring (C00597) and motor phase error monitoring before operation (C2866/2) are active.
- The inverter is enabled (controller enable).
- The *bMctrlMotorPhaseFault* status output of the SB LS_DeviceMonitor is set to FALSE.
- Bit 10 of the *MCTRL_Status3* status word must be set to 0 before the brake opening is triggered.
  - When the controller is enabled, this bit is set to 1 and will not be set to 0 again before the "Motor phase error monitoring before operation" is successfully completed.
  - The *MCTRL_Status3* status word can be integrated in the application via configuration parameters (e.g. C00620) (*MCTRL_Status3* = selection 34906 in Selection list - analog signals).

### 5.13.8 Mains phase failure monitoring

**Stop!**

Under load, the mains input of a three-phase inverter can be destroyed if the device is only supplied by two phases (e.g. if a mains phase fails).

The inverter has a simple mains-phase failure detection function with which a mains phase failure can be detected under load.

- In the case of power-adapted machines, approx. 50 % of the rated motor power must be exceeded so that a main-phase failure can be detected.
- If the mains phase failure monitoring is tripped:
  - The error response set in C00565 will be carried out (Lenze setting: "Warning").
  - The "Su02: One mains phase is missing" error message will be entered into the logbook.
  - The *bMctrlMainsFault* status output of the LS_DeviceMonitor system block will be set to TRUE.

### 5.13.9 Current monitoring for overload

This function extension is available from version 16.00.00!

If the apparent motor current exceeds a defined threshold value C00124/1) for a certain time (C00563/1) an overload has taken place.

Monitoring responds as follows:

- The *bCurrentMonitoringOverload* signal is set to TRUE
  See selection list - digital signals
- The response set in C00584/1 is activated (Lenze setting: "No response")
- The **OC18** error message, current monitoring overload, is entered into the logbook.
- The *bCurrentMonitoringOverload* status output of the **LS_DeviceMonitor** FB is set to TRUE.

If the overload decreases, the apparent motor current has to decrease below the value C00124/1 - 0.05 x $I_{rated}$ in order that the *bCurrentMonitoringOverload* signal can accept the FALSE state.

When *bCurrentMonitoringOverload* = FALSE, the delay time in the resolution is set to the value 0 s again.

### 5.13.10 Maximum current monitoring

**Note!**

In the Lenze setting (C00609 = "0: No Reaction"), the maximum current monitoring is not activated!

If a Lenze motor is selected from the catalogue whose plant parameters are transferred into the inverter, the setting of the maximum current in C00022 and C00023 will automatically be adapted to the selected motor.

If the parameterised maximum current is reached, the response set in C00609 is triggered (Lenze setting: "0: No Reaction").

If the activated monitoring is tripped:

- The "oC10: Maximum current reached" error message is entered in the logbook.

**Related topics:**

▸ Motor overcurrent monitoring (📖 303)

### 5.13.11 Maximum torque monitoring

**Note!**

In the Lenze setting (C00608 = "0: No Reaction"), the maximum torque monitoring is not activated!

If the maximum possible torque C00057 is reached at the motor shaft, the response set in C00608 will be carried out (Lenze setting: "0: No Reaction").

If the activated monitoring is tripped:

- The "ot1: Maximum torque reached" error message is entered into the logbook.
- The *bMctrlTorqueMax* status output of the LS_DeviceMonitor system block will be set to TRUE.

### 5.13.12 Motor speed monitoring

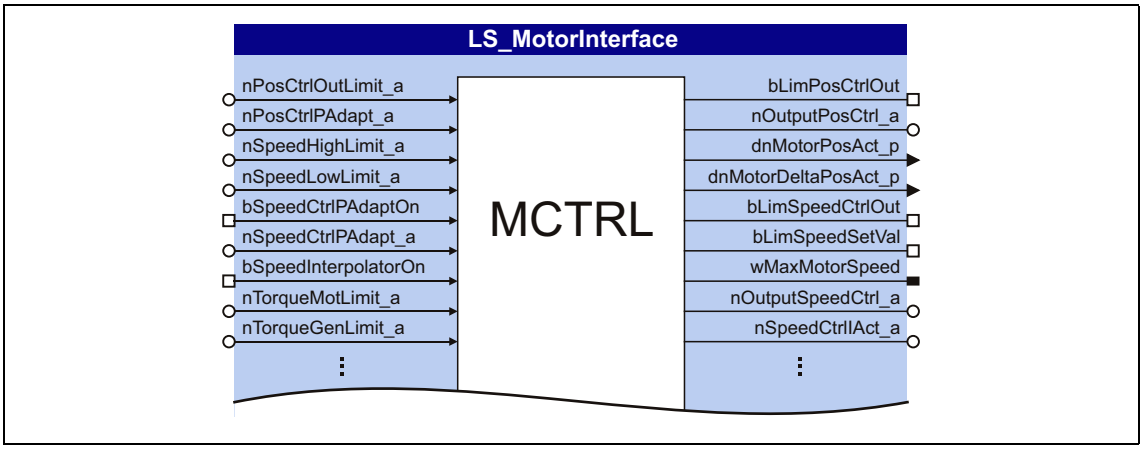
This function extension is available from version 02.00.00!

If the drive reaches the maximally permissible motor speed (C00965):

- The error response "Fault" occurs, i.e. the inverter is inhibited and the motor changes to torque-free operation (coasts down).
- The error message "oS2: Max. motor speed reached" is entered into the logbook.

## 5.14 Internal interfaces | System block "LS_MotorInterface"

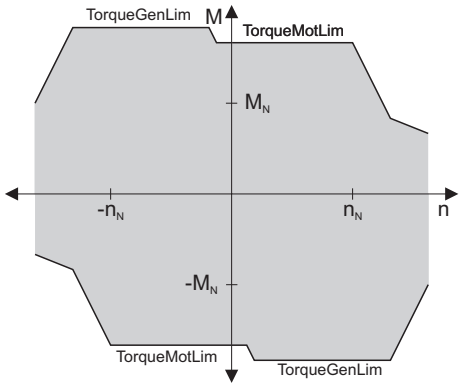
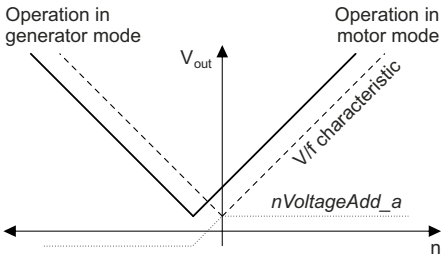
The **LS_MotorInterface** system block provides the internal interfaces to the driving machine in the function block editor.




[5-35] LS_MotorInterface system block (excerpt)

### Inputs

| Designator<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| nPosCtrlOutLimit_a<br>C00830/21 \| INT | Limitation of the position controller output<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nPosCtrlPAdapt_a<br>C00830/20 \| INT | Adaptation of the position controller gain<br>• Scaling: 16384 ≡ 100 % Vp (C00254) | |
| nSpeedHighLimit_a<br>C00830/88 \| INT | Upper speed limit for the speed limitation<br>• During torque-controlled operation only (*bTorquemodeOn* = TRUE)<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>**Note:**<br>For a correct function of the speed limitation in both direction, the following applies: The speed limitation operates with a internal hysteresis of 50 rpm. The upper or lower speed limit resulting from the settings has to be higher than this hysteresis! | |
| nSpeedLowLimit_a<br>C00830/23 \| INT | Lower speed limit for speed limitation<br>• During torque-controlled operation only (*bTorquemodeOn* = TRUE)<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>**Note:**<br>For a correct function of the speed limitation in both direction, the following applies: The speed limitation operates with a internal hysteresis of 50 rpm. The upper or lower speed limit resulting from the settings has to be higher than this hysteresis! | |
| bSpeedCtrlPAdaptOn<br>C00833/69 \| BOOL | Adaptation of the speed controller gain | |
| | FALSE | Deactivate adaptive adaptation. |
| | TRUE | Activate adaptive adaptation. |
| nSpeedCtrlPAdapt_a<br>C00830/25 \| INT | Adaptation of the speed controller gain<br>• Scaling: 16384 ≡ 100 % Vp (C00070) | |
| bSpeedInterpolatorOn<br>C00833/28 \| BOOL | Speed setpoint interpolation | |
| | FALSE | Deactivate interpolation |
| | TRUE | Activate interpolation |

| Designator<br>DIS code \| data type | Information/possible settings |
|---|---|
| nTorqueMotLimit_a<br>C00830/29 \| INT<br>nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in motor mode and in generator mode<br>• The drive cannot output a higher torque in motor/generator mode than set here.<br>• The applied values (any polarity) are internally interpreted as absolute values.<br>• If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.<br>• If sensorless vector control (SLVC) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| | Torque limits in motor and generator mode:<br>  <br> |
| bTorqueInterpolatorOn<br>C00833/29 \| BOOL | Torque setpoint interpolation |
| | FALSE: Deactivate interpolation |
| | TRUE: Activate interpolation |
| nVoltageAdd_a<br>C00830/31 \| INT | Additive voltage impression<br>• This process input serves to specify an additional setpoint for the motor voltage. If there are, for instance, different loads at the motor output end, it is possible to apply a voltage boost at the starting time.<br>• If the value is negative, the voltage is reduced.<br>• Scaling: 16384 ≡ 1000 V |
| | This process signal also serves to implement a load adaptation.<br>Example of V/f characteristic control (VFCplus): Adaptation of the voltage characteristic as a function of the load in motor mode/in generator mode in case of CW/CCW rotation<br>  <br><br>• CW rotation in motor mode: Increase voltage<br>• CCW rotation in generator mode: Reduce voltage |
| | **Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! |
| bAutoBoostOn<br>C00833/32 \| BOOL | Reserved |

| Designator<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| nBoost_a<br>C00830/26 \| INT | Process signal for the $V_{min}$ boost<br>• This signal is added to C00016 and has thus an increasing or decreasing effect.<br>• This signal serves to implement a load-dependent $V_{min}$ boost to improve the torque behaviour in different load states, as for instance in case of operation in generator or motor mode.<br>• Scaling: 16384 ≡ 100 % rated device voltage $V_{FU}$ (400 V or 230 V)<br>▸ Adapting the Vmin boost (📖 170) | |
| | **STOP Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! | |
| bPosCtrlOn<br>C00833/27 \| BOOL | Position/angle control | |
| | FALSE | Deactivate position/angle control. |
| | TRUE | Activate position/angle control. |
| bDeltaPosOn<br>C00833/35 \| BOOL | Activate position difference as setpoint selection<br>• In order to position the motor shaft, the position control function can work within the motor control function with the absolute position setpoint *dnPosSetValue_p* or alternatively with the speed setpoint *nSpeedSetValue_a* and the position difference *dnDeltaPos_p*. | |
| | FALSE | Positioning with position setpoint *dnPosSetValue_p*. |
| | TRUE | Positioning with speed setpoint *nSpeedSetValue_a* and position difference *dnDeltaPos_p*. |
| dnDeltaPos_p<br>C00834/4 \| DINT | Position difference (following error input)<br>• Difference between setpoint position and actual position in [increments]<br>• Is used for position control if *bDeltaPosOn* = TRUE.<br>• Scaling: 65535 ≡ 1 revolution | |
| dnPosSetValue_p<br>C00834/5 \| DINT | Absolute position setpoint in [increments]<br>• Is used for position control if *bDeltaPosOn* = FALSE.<br>• Scaling: 65535 ≡ 1 revolution | |
| bPosDerivativeOn<br>C00833/67 \|BOOL | Create a setpoint for the speed controller from the position setpoint<br>• For highly dynamic control systems, the setpoint for the speed controller can be created from the absolute position setpoint *dnPosSetValue_p* instead of the speed setpoint *nSpeedSetValue_a* ▸ Position control/additive speed specification | |
| | TRUE | Create a speed setpoint from the position setpoint.<br>• The absolute position setpoint *dnPosSetValue_p* is differentiated and a speed value is created which is the setpoint for the speed controller.<br>• Internal limitation of 65536 increments/ms. |
| bMotorRefOffsetOn<br>C00833/68 \| BOOL | Set home position ("referencing on the fly") | |
| | TRUE | Set home position to value *dnMotorRefOffset_p*. |
| dnMotorRefOffset_p<br>C00834/6 \| DINT | Home position in [increments]<br>• Scaling: 65535 ≡ 1 revolution | |
| bQspOn<br>C00833/33 \| BOOL | Quick stop | |
| | FALSE | Deactivate quick stop |
| | TRUE | Activate quick stop |
| nPWMAngleOffset_a<br>C00830/32 \| INT | Angle step change of output voltage phasor<br>• Scaling: 65535 ≡ 1 revolution | |
| bSpeedCtrlIOn<br>C00833/31 \| BOOL | Directly set the I-component of speed controller<br>• In order to statically specify a minimum torque, e.g. when a load is being lifted. | |
| | TRUE | Se the I-component of the speed controller to the value *nSpeedCtrlI_a*. |

| Designator<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| nSpeedCtrlI_a<br>C00830/24 \| INT | Value of the speed controller integrator<br>• Scaling depends on the selected motor control:<br>• V/f control (VFCplus + encoder):<br>16384 ≡ 100 % reference speed (C00011)<br>• Servo control (SC) or vector control (SLVC):<br>16384 ≡ 100 % $M_{max}$ (C00057) | |
| nSpeedSetValue_a<br>C00830/22 \| INT | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| bTorquemodeOn<br>C00833/30 \| BOOL | Selection: Speed/Torque control | |
| | FALSE | Speed control with torque limitation |
| | TRUE | Torque control with speed limitation |
| nTorqueSetValue_a<br>C00830/27 \| INT | Torque setpoint / additive torque<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| bDcBrakeOn<br>C00833/34 \| BOOL | Activate DC injection brake | |
| | FALSE | Deactivate DC-injection braking |
| | TRUE | Activate DC-injection braking |
| bTorqueLimitAdaptOn<br>C00833/98 \| BOOL | Adaptation of torque limitation | |
| | TRUE | Activate adaptation of torque limitation. |
| nTorqueLimitAdapt_a<br>C00830/70 \| INT | Value for adaptation of torque limitation<br>• Scaling: 16384 ≡ 100 % *nTorqueMotLimit_a* and *nTorqueGenLimit_a* | |
| nInertiaAdapt_a<br>C00830/96 \| INT<br>(from version 12.00.00) | Adaptation of the moment of inertia<br>• This process signal can be used during the process to dynamically control the percentage of the variable moment of inertia (e.g. a reel) set in C00919/1 which is to be considered for setpoint feedforward control.<br>• Scaling: 16384 ≡ 100 % moment of inertia - process (C00919/1) | |
| bBrakeChopperOn<br>C00833/130 \| BOOL<br>(from version 12.00.00) | Switch on the internal brake chopper as "brake chopper slave" (C02866/3 = "1: Yes") when the inverter is configured<br>▸ Control of multiple internal brake choppers in the DC-bus system (📖 296) | |
| | TRUE | Switch on internal brake chopper. |
| nSpeedSetValueInertia_a<br>C00830/97 \| INT<br>(from version 12.00.00) | Input for differential setpoint feedforward control (torque feedforward control)<br>• If the selection "1: nSpeedSetValueInertia_a" is set in C00654/1, this process signal can be used to preselect any input value (e.g. setpoint of the position or process controller) for torque feedforward control.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| bVfcEcoDisable<br>C00833/131 \| BOOL<br>(from version 13.00.00) | Deactivate energy optimisation for VFCplusEco<br>▸ Improving the behaviour at high dynamic load changes | |
| | FALSE | Activate energy optimisation. |
| | TRUE | Deactivate energy optimisation. |

### Outputs

| Designator<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| bLimPosCtrlOut<br>BOOL | "Position controller output inside the limitation" status signal | |
| | TRUE | The position controller output is internally limited |
| nOutputPosCtrl_a<br>INT | Position controller output<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |

| Designator<br>DIS code \| data type | Value/meaning |
|---|---|
| dnMotorPosAct_p<br>DINT | Current position in [increments]<br>• If the gearbox factor position encoder (C01203/1..2) is equal to "1", the motor position is output.<br>• Otherwise, if the gearbox factor position encoder (C01203/1..2) is not equal to "1" and a position encoder has been set in C00490, a position referring to the position encoder is output. |
| dnMotorDeltaPosAct_p<br>DINT | Current following error in [increments]<br>• Following error = Difference between set position and actual position<br>• If the gearbox factor position encoder (C01203/1..2) is equal to "1", the motor following error is output.<br>• Otherwise, if the gearbox factor position encoder (C01203/1..2) is not equal to "1" and a position encoder has been set in C00490, a following error referring to the position encoder is output. |
| bLimSpeedCtrlOut<br>BOOL | "Speed controller or manipulating variable of the slip regulator inside the limitation" status signal<br>TRUE: The speed controller output is internally limited |
| bLimSpeedSetVal<br>BOOL | "Reduction or increase of the setpoint speed active" status signal<br>TRUE: Reduction or increase of the setpoint speed by the $I_{max}$ controller is active |
| wMaxMotorSpeed<br>C00011 \| BOOL | Reference speed (C00011) |
| nOutputSpeedCtrl_a<br>INT | Speed controller output<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• In case of the "VFC (+encoder)" motor control, this output is the output of the slip regulator. In this case, the scaling depends on the reference frequency (display in C00059):<br>• C00059 < 650 Hz: 16384 ≡ 327.68 Hz (24000 ≡ 480.00 Hz)<br>• C00059 > 650 Hz: 16384 ≡ 655.36 Hz (12000 ≡ 480.00 Hz) |
| nSpeedCtrlIAct_a<br>INT | Current value of speed controller integrator<br>• Scaling depends on the selected motor control:<br>• V/f control (VFCplus + encoder):<br>16384 ≡ 100 % reference speed (C00011)<br>• Servo control (SC) or vector control (SLVC):<br>16384 ≡ 100 % $M_{max}$ (C00057) |
| nEffSpeedSetValue_a<br>INT | Effective speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| nMotorSpeedAct_a<br>C00051 \| INT | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| nMotorSpeedAct_v<br>INT | Actual speed value<br>• Scaling: 65535 ≡ 1 revolution |
| nMotorFreqAct_a<br>C00058 \| INT | Current field frequency<br>• Scaling depends on the reference frequency (display in C00059):<br>• C00059 < 650 Hz: 16384 ≡ 327.68 Hz (24000 ≡ 480.00 Hz)<br>• C00059 > 650 Hz: 16384 ≡ 655.36 Hz (12000 ≡ 480.00 Hz) |
| bLimTorqueSetVal<br>BOOL | "Setpoint torque inside the limitation" status signal<br>TRUE: The setpoint torque is internally limited |
| wMaxMotorTorque<br>C00057 | Maximum motor torque<br>• *wMaxMotorTorque* = 10 * $M_{max}$ (C00057) |
| nInputTorqueCtrl_a<br>INT | Input value of the torque control (torque setpoint)<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| nMotorTorqueAct_a<br>C00056/2 \| INT | Actual torque<br>• In the "VFC (+encoder)" motor control mode, this value is determined from the current motor current and only approximately corresponds to the actual torque value..<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |

| Designator<br>DIS code \| data type | Value/meaning |
|---|---|
| nInputJerkCtrl_a<br>INT | Input value of the jerk limitation<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| bLimCurrentSetVal<br>BOOL | "Current setpoint inside the limitation" status signal |
| | TRUE: The current setpoint is internally limited |
| nStatorCurrentIS_a<br>INT | Current stator current/effective motor current<br>• Scaling: 16384 ≡ 100 % $I_{max_mot}$ (C00022) |
| nEffCurrentIq_a<br>INT | Current torque-producing cross current<br>• Scaling: 16384 ≡ 100 % $I_{max_mot}$ (C00022) |
| nReaktCurrentId_a<br>INT | Current field-producing direct-axis current<br>• Scaling: 16384 ≡ 100 % $I_{max_mot}$ (C00022) |
| nActualFluxx_a<br>INT | Current magnetising current<br>• Scaling: 16384 ≡ 100 % $I_{max_mot}$ (C00022) |
| nDCVoltage_a<br>INT | Actual DC-bus voltage<br>• Scaling: 16384 ≡ 1000 V |
| nMotorVoltage_a<br>INT | Current motor voltage/inverter output voltage<br>• Scaling: 16384 ≡ 1000 V |
| bQspActive<br>BOOL | "Quick stop active" status signal |
| | TRUE: Quick stop is active |
| bAutoDCBActive<br>BOOL | "Automatic DC-injection braking active" status signal<br>▸ DC-injection braking (📖 274) |
| | TRUE: Automatic DC-injection braking is active |
| bIdentificationActive<br>BOOL | "Motor parameter identification active" status signal<br>▸ Automatic motor data identification (📖 145) |
| | TRUE: Motor parameter identification is active |
| bFlyingSyncActive<br>BOOL | "Flying restart function active" status signal<br>▸ Flying restart function (📖 271) |
| | TRUE: Flying restart function is active |
| bHlgLoad<br>BOOL | Control signal for an additional loading function of the ramp function generator<br>• ➞ L_NSet_1.*bExternalCINH*<br>• To enable the ramp function generator to follow automatically when the controller is inhibited, for jerk-free setpoint connection. |
| | TRUE: Set the ramp function generator to a setpoint of *nHlgSetValue_a* |
| nHlgSetValue_a<br>INT | Setpoint for an additional loading function of the ramp function generator<br>• ➞ L_NSet_1.*nCInhVal_a*<br>• For speed-controlled drive tasks, the current actual speed value (e.g. in case of an active pulse inhibit, flying restart function, controller inhibit) is provided at this output.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| bHlgStop<br>BOOL | Control signal for stopping the ramp function generator (L_NSet_1) |
| | TRUE: Stop the ramp function generator |
| bBrakeChopperActive<br>BOOL<br>(from version 12.00.00) | Status signal of the internal brake chopper control<br>▸ Control of multiple internal brake choppers in the DC-bus system (📖 296) |
| | TRUE: Internal brake chopper is switched-on. |
| nVoltageAngleAct_a<br>INT<br>(from version 13.00.00) | Current electrical voltage output angle of the inverter<br>• Scaling: 16384 ≡ 360° |
| bLimSpeedTorquemodeOn<br>BOOL<br>(from version 13.00.00) | Status signal of torque control with speed limitation |
| | TRUE: Speed limitation for torque control is active. |

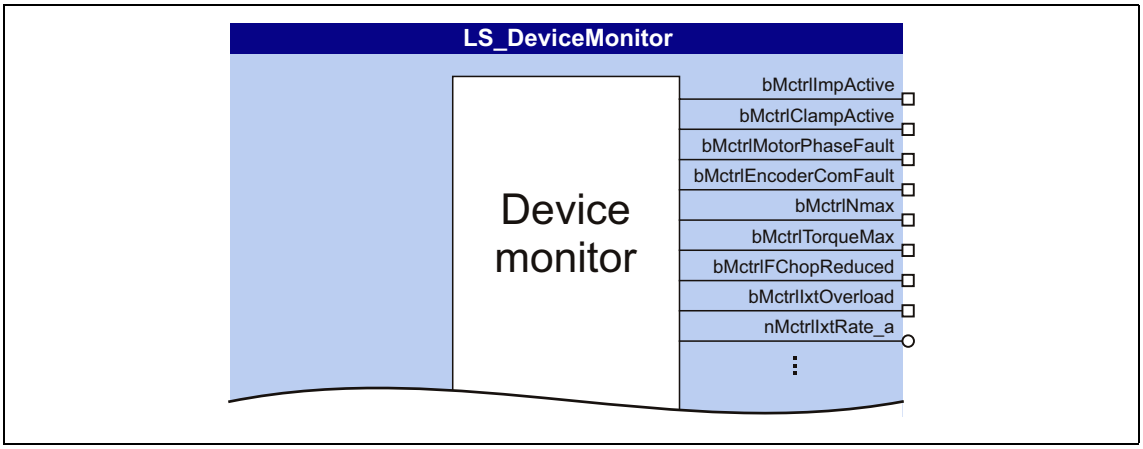
## 5.15 Internal status signals | System block "LS_DeviceMonitor"

The **LS_DeviceMonitor** system block provides the status signals of the motor control in the function block editor.

### Note!

The **LS_DeviceMonitor** system block can only be inserted on the application level.

If status signals of the motor control function are to be output via digital outputs or example, you can use the free *bFreeOut1* ... *bFreeOut8* outputs of the application block to transfer the desired status signals from the application level to the I/O level. On the I/O level, you can then establish the logical link to the digital output terminals.




[5-36] LS_DeviceMonitor system block (excerpt)

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bMctrlImpActive<br>BOOL | TRUE | Pulse inhibit is active |
| bMctrlClampActive<br>BOOL | TRUE | Clamp current limitation is active |
| bMctrlMotorPhaseFault<br>BOOL | Status of the Motor phase failure monitoring | |
| | TRUE | Motor phase fault has been detected |
| bMctrlEncoderComFault<br>BOOL | TRUE | Encoder error has been detected |
| bMctrlNmax<br>BOOL | TRUE | Max. speed limitation is active |
| bMctrlTorqueMax<br>BOOL | Status of the Maximum torque monitoring | |
| | TRUE | Max. torque limitation is active |
| bMctrlFChopReduced<br>BOOL | TRUE | PWM frequency reduction is active |
| bMctrlIxtOverload<br>BOOL | Status of the Device overload monitoring (Ixt) | |
| | TRUE | Device utilisation (Ixt) ≥ device utilisation threshold (C00123)<br>• Lenze setting: C00123 = 100 % |
| nMctrlIxtRate_a<br>INT | Current device utilisation (Ixt)<br>• Scaling: 16384 ≡ 100 % | |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bMctrlI2xtOverload<br>BOOL | Status of the Motor load monitoring (I2xt) | |
| | TRUE | Thermal motor overload ($I^2xt$) ≥ motor overload setting (C00120)<br>• Lenze setting: C00120 = 100 %. |
| nMctrlI2xtRate_a<br>INT | Current thermal motor load (I2xt)<br>• Scaling: 16384 ≡ 100 % | |
| bMctrlMotorPTC<br>BOOL | Status of the Motor temperature monitoring (PTC) | |
| | TRUE | Temperature monitoring: An error has been detected. |
| bMctrlMotorTemp<br>BOOL | Status of the Motor temperature monitoring (KTY) | |
| | TRUE | Thermal motor overload<br>• One of the following error messages is pending:<br>"oH7: Motor temperature resolver > C121"<br>"oH9: Motor overtemperature resolver"<br>"oH6: Motor temperature MultiEncoder > C121"<br>"oH12: Motor overtemperature MultiEncoder" |
| bMctrlHeatSinkTemp<br>BOOL | TRUE | Thermal inverter overload<br>• The heatsink temperature (display in C00061) has reached the maximally permissible temperature. The "oH1: Overtemperature heatsink" error message is pending and the "Fault" response is activated.<br>• Furthermore, this output is set if the heatsink temperature has exceeded the maximally permissible temperature for the switching frequency set. A response with an "oH4: Heatsink temp.. > switch-off temp. -5°C" error message can be set in C00582. In the Lenze setting, no response will be effected. |
| bMctrlMainsFault<br>BOOL | Status of the Mains phase failure monitoring | |
| | TRUE | Mains phase failure/Mains failure |
| bMctrlFanFault<br>BOOL | TRUE | Fan monitoring: An error has been detected. |
| bMctrlNmaxForFChop<br>BOOL | TRUE | The maximum field frequency for the respective switching frequency has been exceeded. |
| bMctrlShortCircuit<br>BOOL | TRUE | Motor short circuit has been detected |
| bMctrlEarthFault<br>BOOL | TRUE | Earth fault has been detected |
| bMctrlUVDetected<br>BOOL | TRUE | An undervoltage has been detected |
| bMctrlOVDetected<br>BOOL | TRUE | An overvoltage has been detected |
| bMctrlBrakeChopperFault<br>BOOL | Status of the Brake resistor monitoring (I2xt)<br>• This output is set independent of the set error response of the monitoring function. | |
| | TRUE | The $I^2xt$ utilisation has reached the switch-off threshold set in C00572.<br>• Lenze setting: C00572 = 100 %. |
| wUB24V<br>WORD | Current 24 V supply voltage<br>• Scaling: 1000 ≡ 1.000 V | |
| nMctrlActiveOutputPower_a<br>INT<br>(from version 14.00.00) | Current active output power<br>• Scaling: 16384 ≡ double rated power of the inverter | |
| nMctrl ApparentOutputPower_a<br>INT<br>(from version 14.00.00) | Current apparent output power<br>• Scaling: 16384 ≡ double rated power of the inverter | |
| nMctrlCosinePhiAct_a<br>INT<br>(from version 14.00.00) | Current cosine phi<br>• Scaling: 16384 ≡ cosine phi of 1.0 | |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bWirebreakUfLinearActive<br>BOOL<br>(from version 15.00.00) | Status "V/f emergency operation due to encoder open circuit active"<br>• The internal change-over to "V/f emergency operation" can be suppressed by setting bit 8 to "1" in C02864/1. | |
| | TRUE | Due to an encoder open circuit, it is internally switched to the encoderless V/f characteristic control in order to avoid impermissible motor movements. This does not influence the setting in C00006. |
| bCurrentMonitoring Overload<br>BOOL<br>(from version 16.00.00) | Status of the Current monitoring for overload | |
| | TRUE | The monitoring function is activated and the apparent motor current has exceeded the switch-off threshold set in C00124/1 for the delay time set in C00563/1.<br>**Note:** This status output is reset with a hysteresis of 5 % of the rated device current (C00098). |

# 6 Encoder/feedback system

The 8400 TopLine uses various encoder/feedback systems for the motor controls with speed feedback and the position control:

- Resolver at X7
- Multi-Encoder at X8
  - Incremental TTL encoders
  - Incremental SinCos encoders
  - SinCos absolute value encoder with HIPERFACE® protocol
  - SSI absolute value encoder with Stegmann-SSI protocol
  - Digital frequency coupling (from version 12.00.00)
- HTL encoder at the digital DI1/DI2 or DI6/DI7 input terminals

**Danger!**

- If the encoder/resolver is used as motor encoder:
  Safe operation of the motor is no longer ensured in the event of an error!
- If servo control (SC) or V/f control (VFCplus + encoder) are used: For safety reasons, always select "Fault" (Lenze setting) in the following parameters as a response for the (open-circuit) monitoring of the encoder!
  - C00586: Resp. open circuit HTL encoder
  - C00603/1: Resp. open circuit MultiEncoder
  - C00603/2: Resp. open circuit resolver
- If an HTL encoder is used at the digital input terminals:
  Observe the maximum input frequencies of the digital inputs!
  - DI1/DI2: max. 200 kHz
  - DI6/DI7: max. 10 kHz
- Generally, we recommend the use of encoder signals with increments ≥ 4
  - With increments < 4, the evaluation may cause implausible values or a number range overflow may occur when further processing the value.
  - Increments > 2 are at least required for encoder signals that are exclusively processed in the FB interconnection.
- To avoid the injection of interference when an encoder is being used, only use shielded motor and encoder cables!

**Note!**

Speed feedback is essential for the following motor control modes with feedback:

- Servo control (SC)
- V/f control (VFCplus + encoder)

**Note!**

The encoder position can be saved safe against mains failure in the inverter.

- For this purpose, bit 0 must be set in C02652. The actual position of the motor control is then saved in the inverter (not in the memory module) and thus remains known to the drive control after mains switching.
  ▸ Option "Actual MCTRL position received at mains switch-off" (📖 635)
- In case of a device replacement, this information gets lost and the home position might be approached or set again once. A device replacement can be recognised by using the SB LS_RetainData and a suitable function block interconnection.

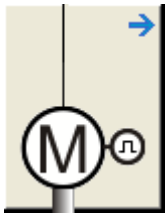
If an absolute value encoder (Multiturn) is used, the actual position will always be reconstructed after mains switch-off/switch-on, no matter if the bit 0 in C02652 has been set to "0" or "1".

Wiring diagram, assignment and electrical data of the terminals for resolvers and multi encoders can be found in the **hardware manual 8400** in the chapter "technical data". The hardware manual is stored in electronic form on the data carrier supplied with the 8400 inverter.

## 6.1 Encoder selection

**How to get to the parameterisation dialog of the encoder/feedback system:**

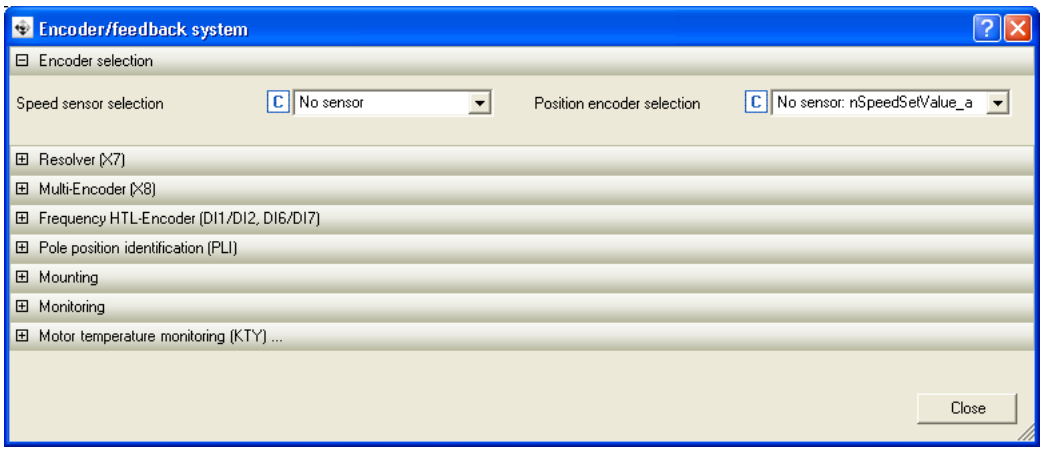
1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Go to the *Overview* dialog level and click the following button:



4. Go to the *Overview* → *Motor data* dialog level and click the **Encoder/Feedback system...** button to open the parameterisation dialog for the encoder/feedback system.

**Parameterisation dialog in the »Engineer«**

From the »Engineer« V2.19, the following parameterisation dialog is available for parameterising the encoder/feedback system. For a better overview, the parameterisation dialog contains different categories which can be expanded/collapsed by a simple click. First, only the "encoder selection" category is expanded:



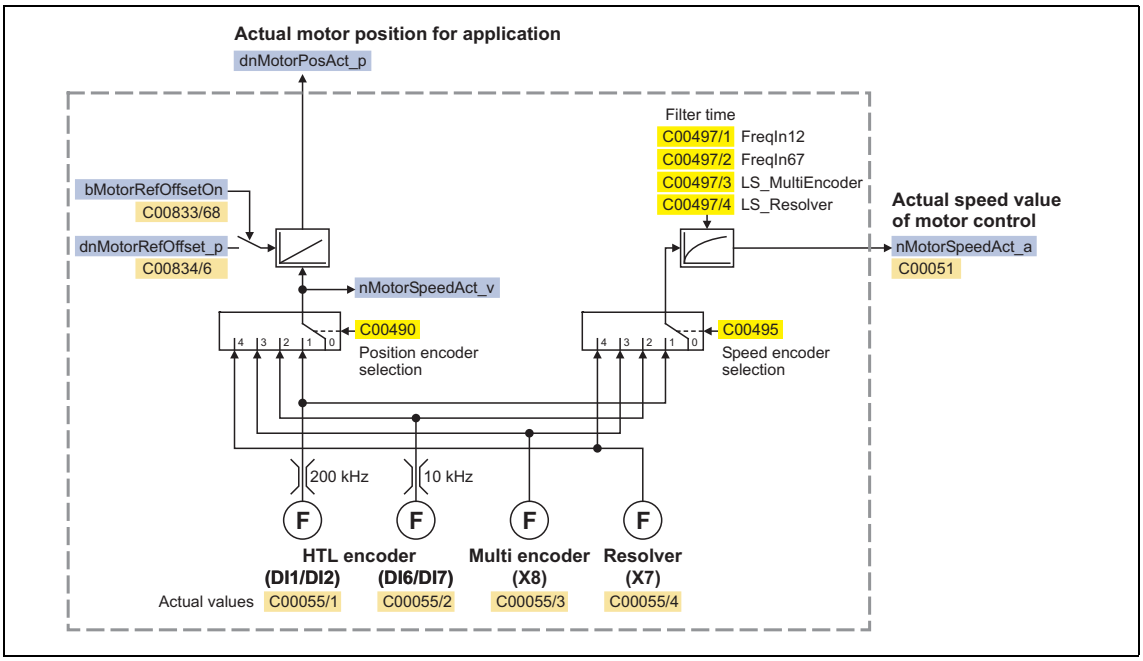

| Parameters | Info | Lenze setting |
|---|---|---|
| C00495 | Speed sensor selection<br>• Source of feedback signal for speed control. | 0: No sensor |
| C00490 | Position encoder selection<br>• Source of feedback signal for position control. | 0: no sensor: nSpeedSetValue_a |

Go to the **Speed sensor selection** list field and select the connected speed sensor.
After the selection, the respective category wit the parameters relevant for the selected encoder is automatically expanded.

Detailed information on the parameters of the other categories can be found in the corresponding chapter:

- Resolver (X7) (📖 326)
- Multi-Encoder (X8) (📖 332)
- Frequency HTL encoder (DI1/DI2, DI6/DI7) (📖 360)
- Pole position identification (PPI) (📖 366)
- Mounting (📖 375)
- Monitoring (📖 376)
- Motor temperature monitoring (KTY) (📖 381)

### 6.1.1 Signal flow - encoder interface




[6-1] Signal flow - encoder interface

#### Generation of the actual speed value

...depending on the encoder selection and mounting direction:

| Speed sensor (C00495) | Position encoder (C00490) | Motor mounting direction (C01206/1) | Direction of rotation of motor shaft (at setpoint = Cw) | Actual speed value (nAct_v) |
|---|---|---|---|---|
| ☐ | ☐ | not inverted | Cw | ΔSetPos |
| | | inverted | Ccw | |
| ☒ | ☐ | not inverted | Cw | C00495 |
| | | inverted | Ccw | |
| ☒ | ☒ | not inverted | Cw | |
| | | inverted | Ccw | |
| ☐ | ☒ | not inverted | Cw | C00490 → ΔActPos |
| | | inverted | Ccw | |
| ☐ no encoder ☒ Encoder set | | | | |

Up to and including version 02.xx.xx the following applies:

- For motor control types without speed feedback (C00495 = "0: No encoder") a speed-proportional unit is taken for calculating the *nMotorSpeedAct_v* speed signal. This derivation, however, is very imprecise so that in case of applications with synchronous motors without speed feedback it is not possible to calculate the current position from the current *nMotorSpeedAct_v* speed signal.
- If no position encoder is available (C00490 = "0: No encoder"), the *dnMotorPosAct_p* position signal is always derived from the *nSpeedSetValue_a* speed setpoint. This derivation, however, is very imprecise since in this case, speed limitations (e.g. by overcurrent limitations) are not considered.

From version 12.00.00 the following applies:

- For applications with synchronous motors without speed feedback, an *nMotorSpeedAct_v* error-free speed signal is available. This is calculated from the electrical voltage output angle of the inverter considering the number of pole pairs of the *nMotorSpeedAct_v* speed signal.
  - From version 13.00.00, the electrical voltage output angle is shown at the *nVoltageAngleAct_a* output of the LS_MotorInterface SB (scaling: 16384 ≡ 360°). This process signal can for instance be used in applications where a change-over from inverter to mains operation is to be carried out (synchronisation of the inverter output voltage to the mains voltage).
- When synchronous or reluctance motors without feedback are used in the motor control types V/f characteristic control (VFCplus) and Sensorless control for synchronous motors (SLPSM), the *nMotorSpeedAct_v* speed signal can be used to create an error-free position signal via a control if the *nMotorSpeedAct_v* signal is read out by the control in a 1 ms cycle.
- If no position encoder is available (C00490 = "0: No encoder"), the *dnMotorPosAct_p* position signal is continued to be derived from the *nSpeedSetValue_a* speed setpoint. The extended selection text "0: No encoder:nSpeedSetValue_a" in C00490 refers to this behaviour.
- The new selection "10: No encoder: C495 or nMotorSpeedSetAct_v" in C00490 is used to calculate the *dnMotorPosAct_p* position signal either from the set speed feedback (when C00495 > 0) or from the *nMotorSpeedAct_v* speed signal (when C00495 = 0).
  - For all motor control types without speed feedback, this selection serves to improve the creation of the *dnMotorPosAct_p* position signal.
  - When synchronous or reluctance motors without feedback are used, the *dnMotorPosAct_p* position signal can be created correctly.
  - In case of the motor control types with speed feedback, the *dnMotorPosAct_p* position signal is directly created from the speed feedback signal.

## 6.2 Resolver (X7)

**Danger!**

- If the resolver is used as motor encoder:
  Safe operation of the motor is no longer ensured in the event of an error!
- If servo control (SC) or V/f control (VFCplus + encoder) are used:
  For safety reasons, always select "Fault" (Lenze setting) in C00603 2 as a response for the (open-circuit) monitoring of the resolver!

### 6.2.1 Parameterising the resolver

**Note!**

If a Lenze motor with resolver is used, the resolver is the only item that can be selected as speed sensor in C00495. Further resolver parameter setting is not required.

Moreover, Lenze synchronous motors do not require a pole position identification (PLI) as the resolvers of the Lenze synchronous motors are pre-adjusted.

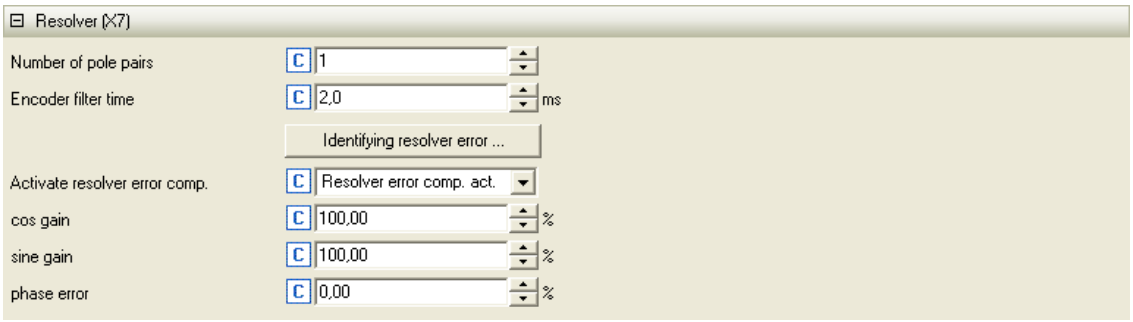
**Note!**

**Acceptance of the resolver pole position from a Servo Drive 9400**

The resolver pole position (C00926/1) cannot be simply accepted from a Servo Drive 9400 if the pole position (C58/1) considerably differs from -90 ° in the Servo Drive 9400.

- We always recommend a pole position identification (PLI). (📖 366)
- As an alternative, the following conversion has to be carried out for a transfer of the resolver pole position from a Servo Drive 9400 to the 8400 TopLine:
  Pole position$_{8400}$ (C00926/1) = - (pole position$_{9400_C58/1}$ + 180 °)

### Parameterisation dialog (cutout)




| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00925 | LS_Resolver: Number of pole pairs | 1 | |
| C00497/4 | LS_Resolver: Encoder filter time | 2.0 | ms |
| C00417 | Activate resolver error comp.<br>▸ Optimising resolver behaviour | 0: Activate resolver error compensation | |
| C02862/1 | Resolver: cos gain | 100.00 | % |
| C02862/2 | Resolver: sine gain | 100.00 | % |
| C02863 | Resolver: Phase error | 0.00 | % |

**How to parameterise the resolver:**

1. If number of resolver pole pairs ≠ 1:
   Set number of pole pairs (C00925).
2. Select resolver as speed sensor (C00495) or/and as position encoder (C00490).
3. Adjust filter time of the actual speed value (C00497/4) if necessary.
   - Lenze setting: 2 ms
   - Lenze recommend a filter time between 1 ... 2 ms.
4. If you use the resolver as motor encoder and the servo control with synchronous motor of a original equipment manufacturer: Carry out Pole position identification (PPI). (📖 366)
5. For detecting and monitoring the motor temperature via the resolver cable:
   Parameterise Motor temperature monitoring (KTY) via resolver. (📖 382)
6. Optional: Parameterise Resolver/multi-encoder open-circuit monitoring. (📖 377)

   In the Lenze setting, the resolver cable and/or encoder cable are monitored with regard to open circuit as a function of the selected speed and position encoder.
7. For a permanent acceptance of the parameter settings:

   Save parameter set.

### 6.2.2 Optimising resolver behaviour

Due to mounting and production tolerances as well as resolver material property leakage, errors may occur which, among other things, result in speed-dependent vibration of the actual speed. These errors are called resolver errors. Resolver errors typically occur in the form of the 1st and 2nd harmonic. They have two different causes:

1. The inductances of the sine and cosine track of the resolver have slightly different values.
2. Sine and cosine track do not magnetise orthogonally to each other.

Resolver errors due to cause 1 can be corrected by adjusting the gains of the digital/analog converters which feed the resolver tracks. In the Lenze setting, the gains of both resolver tracks are preset with identical values.

Resolver errors due to cause 2 can be compensated for by a slight correction of the angle via which both resolver tracks are fed relative to one another.

When the "Resolver error identification" device command (C00002/25) is executed, the gain of the resolver signals and the angular drift of both resolver tracks is corrected to minimise the resolver error.

- Select a speed-controlled operating mode (e.g. servo control) for your machine while you perform a resolver error identification run. During the identification run, speed must be constant and greater than 500 rpm.
- After a successful resolver error identification run, the resolver automatically uses the following resolver error parameters which have been identified during the procedure:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02862/1 | Resolver: cos gain | 100.00 | % |
| C02862/2 | Resolver: sine gain | 100.00 | % |
| C02863 | Resolver: Phase error | 0.00 | % |

- The detected gain can have values from 80 ... 120 %. It makes sense to adjust only one of the two gains during a resolver error compensation. The other one remains at 100 % (Lenze setting).
- Save the parameter set afterwards to accept the identified resolver error parameters permanently (device command C00002/11).
- If the resolver error compensation is deactivated (C00417 = "1: Resolver error comp. deact."), the resolver will resume work with the Lenze setting. However, the identified resolver error parameters remain saved.

**How to perform a resolver error identification run:**

1. Select a motor control mode with speed feedback in C00006 and commission it. ▸ Selecting the control mode
   - Avoid a speed controller setting that is too tight (motor humming).
   - Set the speed ramps as well as the speed setpoint (> 500 rpm).
2. Inhibit the inverter, e.g. via the C00002/16 device command or via a LOW signal at terminal X5/RFR.
   - The inverter must be in the "SwitchedOn" status for the resolver error identification run to be performed.
3. Activate resolver error identification via the C00002/25 device command = "1: On / start".
4. Enable inverter again.
   - The inverter changes to the "Ident" device status.
   - The resolver error identification run starts.
   - The progress of the identification run can be seen in C00002/25.
   - During the resolver error identification run, the direction of rotation of the motor is not important.
   - We recommend to traverse at constant setpoint speed (|nset| > 500 rpm) until resolver error identification has finished.
   - If the application offers only limited space for traversing, reversing is also possible during active resolver error identification. In this case, ensure that the inverter is not inhibited and that the speed profile is traversed for at least 1 second at constant setpoint speed (|nset| > 500 rpm).
   - The identification is completed if the "0: Off / ready" message is displayed in C00002/25.
5. After the resolver error identification has been completed successfully, the inverter can be inhibited again. (When the inverter is inhibited, the "Ident" device state is quit again.)
6. In order to accept the identified resolver error parameters:

   Save parameter set.

**Tip!**

Resolver error identification can be aborted any time by setting a controller inhibit.

**Note!**

Some applications require that the resolver error identification can also be carried out during operation.

From version V12.00.00, the resolver error identification can also be started in the "OperationEnabled" device status (controller enable) using the device command C00002/25 = "1: on / start". Then there is no device state change, i.e. the device status remains at "OperationEnabled".

Furthermore, the following applies to the resolver error identification during operation:

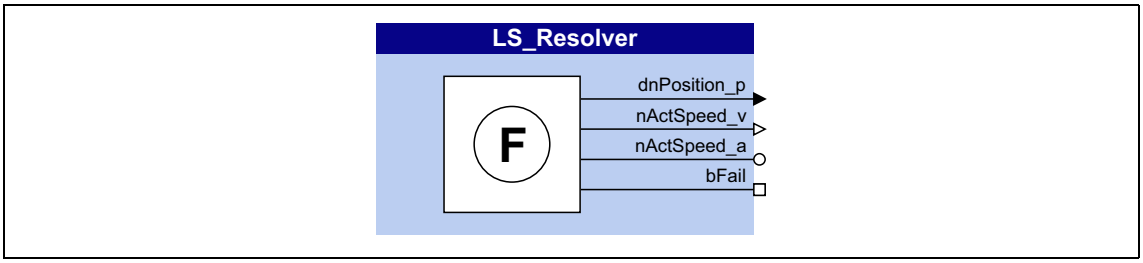
- The progress of the identification run can be seen in C00002/25.
- During the resolver error identification run, the direction of rotation of the motor is not important.
- We recommend to traverse at constant setpoint speed (|nset| > 500 rpm) until resolver error identification has finished.
- If the application offers only limited space for traversing, reversing is also possible during active resolver error identification. In this case, ensure that the inverter is not inhibited and that the speed profile is traversed for at least 1 second at constant setpoint speed (|nset| > 500 rpm).
- The identification is completed if the "0: Off / ready" message is displayed in C00002/25.

**Possible causes for resolver error identification failure:**

- A controller inhibit was set during resolver error identification.
- A time-out occurred while the algorithm was processed.
  - This error may occur if more than 60 seconds pass by between the setting of the "Resolver error identification" device command and the enable of the inverter.
- The setpoint speed was too small (|nset| < 500 rpm).
- The setpoint speed was not traversed for at least 1 second.

### 6.2.3 Internal interfaces | "LS_Resolver" system block

The **LS_Resolver** system block provides the application with speed, position and error information from the resolver in the form of process signals.




#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| dnPosition_p<br>DINT | Actual encoder position in [increments] | |
| nActSpeed_v<br>INT | Actual speed in [increments/ms] | |
| nActSpeed_a<br>INT | Actual speed in [%]<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| bFail<br>BOOL | TRUE | Group error: There is an error (e.g. resolver wire breakage) according to the *bFail* configuration in C00432/1.<br>From version 14.00.00, the error type is displayed bit-coded in C00452/1. ▸ Monitoring |

## 6.3 Multi-Encoder (X8)

**Danger!**

- If the resolver is used as motor encoder:
  Safe operation of the motor is no longer ensured in the event of an error!
- If servo control (SC) or V/f control (VFCplus + encoder) are used: For safety reasons, always select "Fault" (Lenze setting) in C00603 1 as a response for the (open-circuit) monitoring of the encoder!

**Stop!**

Before connecting an encoder, ensure that the encoder supply voltage has the correct setting in C00421:

- Lenze setting: 5 V (setting range: 5 ... 12 V)

If the set supply voltage exceeds the permissible supply voltage of the connected encoder, the encoder may be destroyed!

**Note!**

With regard to their position resolution, higher-level applications are based on the resolution of the encoder which is activated for position control.

The interface of the Multi-Encoder offers various ways in which the encoders can be used / evaluated:

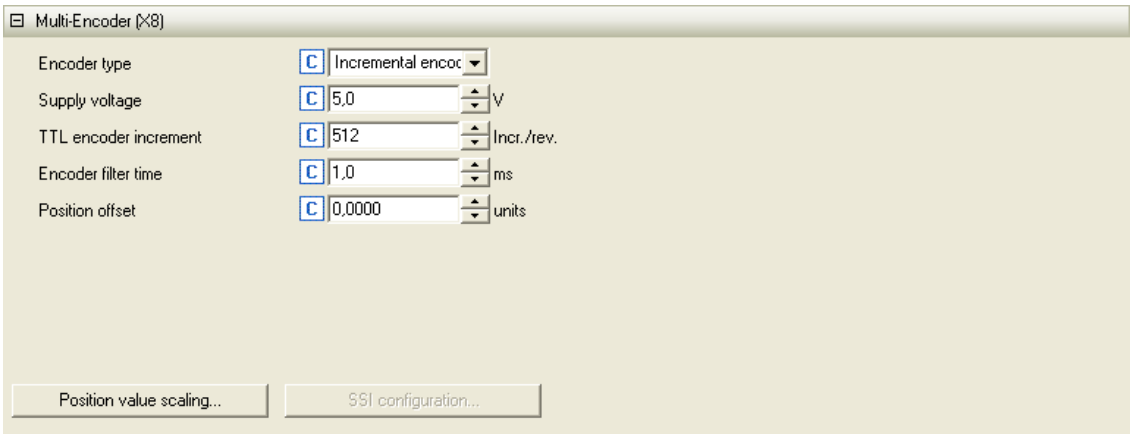
- Incremental TTL and SinCos encoder
- SinCos absolute value encoder with HIPERFACE® protocol
- SSI absolute value encoder with Stegmann-SSI protocol
- Digital frequency coupling (from version 12.00.00 onwards)

### 6.3.1 Incremental TTL and SinCos encoder

**Tip!**

The use of a SinCos encoder instead of a TTL encoder enables a fine interpolation of the position within one encoder increment due to the analog representation of the track signals A and B. This serves to considerably increase the resolution of the position information of a SinCos encoder compared to a conventional TTL incremental encoder.

#### Parameterisation dialog (cutout)




Further parameterisation dialogs: ▸ Scaling of position value

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00422 | LS_MultiEncoder: Encoder type | 0: Incremental encoder (TTL) | |
| C00421 | LS_MultiEncoder: Supply voltage | 5.0 | V |
| C00420/3 | LS_MultiEncoder: Encoder increment | 512 | Incr./rev. |
| C00497/3 | LS_MultiEncoder: Encoder filter time | 1.0 | ms |
| C01112/1 | LS_MultiEncoder: Position offset | 0.0000 | units |

**How to parameterise the incremental TTL or SinCos encoder:**

1. As encoder type (C00422), select "0: Incremental encoder (TTL)" or "1: Sin/Cos encoder" (depending on the type of the connected encoder).
2. Adapt supply voltage (C00421) depending on the cable length if required (see the following section "Voltage setting as a function of the cable length").
3. Set encoder increment (C00420/3).
4. Adjust filter time of the actual speed value (C00497/3) if necessary.
   - Lenze setting: 1 ms
   - Lenze recommend a filter time between 1 ... 2 ms.
5. If you use the encoder as motor encoder and synchronous motor:
   Carry out Pole position identification (PPI). (📖 366)
6. For detecting and monitoring the motor temperature via the Encoder cable:
   Parameterise Motor temperature monitoring (KTY) via encoder. (📖 383)
7. Optional: Parameterise Resolver/multi-encoder open-circuit monitoring. (📖 377)

   In the Lenze setting, the resolver cable and/or encoder cable are monitored with regard to open circuit as a function of the selected speed and position encoder.
8. For a permanent acceptance of the parameter settings:

   Save parameter set.

### Voltage setting as a function of the cable length

The following table shows the settings required for different incremental encoder types as a function of the cable length. The given values apply when Lenze system cables are used at typical ambient temperatures. Other cables, cable cross-sections or extreme ambient temperatures may require metrologically detected adaptations.

| **Encoder type**<br>Product key | Rated voltage $U_N$ | Encoder increment (C00420/3) | Voltage setting in C00421 with a cable length of | | | |
|---|---|---|---|---|---|---|
| | | | 0 - 30 m | 30 - 70 m | 70 - 100 m | 100 - 150 m |
| Incremental TTL encoders | | | | | | |
| IG2048-5V-T | 5 V ±5 % | 2048 | 5.0 V | 5.1 V | 5.2 V | 5.3 V |
| IG4096-5V-T | | 4096 | | | | |
| IK2048-5V-T | | 2048 | | | | |
| IK4096-5V-T | | 4096 | | | | |
| Incremental SinCos encoders | | | | | | |
| IG1024-5V-V | 5 V ±5 % | 1024 | 5.0 V | 5.1 V | 5.2 V | 5.3 V |
| IG2048-5V-S | | 2048 | | | | |

### 6.3.2 SinCos absolute value encoder with HIPERFACE® protocol

SinCos absolute value encoders with HIPERFACE® protocol are feedback systems which transmit an absolute position on request. Moreover, they permanently provide an incremental signal via a sin/cos evaluation. The analog evaluation of the sin/cos tracks provides for a high resolution.

The absolute position is transmitted via an RS485 data interface with 9600 Baud (standard value).This bidirectional parameter channel also serves to request and transmit error codes, encoder data and other information. The analog process data channel is created by a differential transmission of sin/cos signals. The resolutions of these two information channels are generally not identical, thus the channels have to be parameterised independently of each other in the inverter.

**Note!**

If the RS485 cable is interrupted, a communication to the encoder is permanently tried to be built up. After the maximum initialisation time set in C00430/1 has expired, the error "Sd7: Error encoder communication" is triggered. This error is cancelled if a communication could be built up successfully and the encoder initialisation has been executed.

The selection of an error response in C00603/4 serves to optionally monitor the communication between absolute value encoder and inverter.

If the monitoring is tripped:

- The error response set in C00603/4 is triggered.
- The "Sd7: Encoder communication error" error message is entered into the logbook.

#### 6.3.2.1 Initialisation

A connected SinCos absolute value encoder with HIPERFACE® protocol is initialised in the following cases:

A. When the device is switched on after the waiting time of approx. 1 s has expired.

B. After the encoder wire breakage has been removed.
In case of an encoder wire breakage, a speed and position detection cannot be carried out correctly. A renewed initialisation after removing the encoder wire breakage serves to connect to the correct absolute position.

C. After describing one of the following parameters as the setting has an impact on the internal scaling of the encoder position:

| Parameters | Name | Parameters | Name |
|---|---|---|---|
| C00420/3 | LS_MultiEncoder: Encoder increment | C00495 | Speed sensor selection |
| C00422 | LS_MultiEncoder: Encoder type | C01110 | LS_MultiEncoder: Solid measure |
| C00490 | Position encoder selection | C01111/1..3 | LS_MultiEncoder: Encoder constant |
| C00494/1..2 | Hiperface: Resolutions | C01204 | MCK: Feed constant |

### Initialisation process

During initialisation, the inverter requests different information from the encoder. The process of this data transfer is as follows:

1. Reading the encoder status
   The encoder can provide up to 4 error codes which are displayed in C00491/1...4 from version 14.00.00.
2. Reading the encoder type (display of the TypeCode in C00492)
3. Reading the absolute position
   A constant encoder position is assumed to initiate the position value correctly. An initialisation is not permissible when the encoder is rotating.
4. Processing the read position data and linking the SinCos-based position and speed detection to the absolute position of the encoder.

### Encoders detected automatically

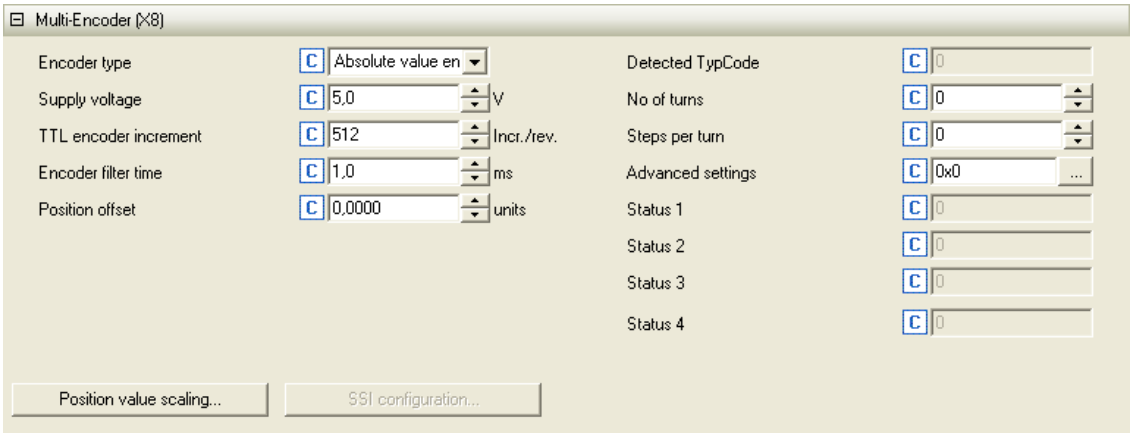
The following encoder types are detected and set automatically by the inverter:

| Type | Increments/ revolution | Absolute revolutions | Bits Singleturn | Bits Multiturn | Bits SinCos track | Type code |
|---|---|---|---|---|---|---|
| AM1024-8V-H (SRM50) | 1024 | 4096 (Multiturn) | 15 | 12 | 10 | 0x27 |
| AM1024-8V-H (SRM60) | | | | | | |
| AM1024-8V-H (SRM64) | | | | | | |
| AM512-8V-H (SCM60) | 512 | | 14 | 12 | 9 | 0x07 |
| AM512-8V-H (SCM70) | | | | | | |
| AM128-8V-H (SKM36) | 128 | | 12 | 12 | 7 | 0x37 |
| AM16-8V-H (SEL37) | 16 | | 9 | 12 | 4 | 0x47 |
| AM16-8V-H (SEL52) | | | | | | |
| AS1024-8V-H (SRS50) | 1024 | 1 (Singleturn) | 15 | 0 | 10 | 0x22 |
| AS1024-8V-H (SRS60) | | | | | | |
| AS1024-8V-H (SRS64) | | | | | | |
| AS512-8V-H (SCS60) | 512 | | 14 | 0 | 9 | 0x02 |
| AS512-8V-H (SCS70) | | | | | | |
| AS128-8V-H (SKS36) | 128 | | 12 | 0 | 7 | 0x32 |
| AS16-8V-H (SEK37) | 16 | | 9 | 0 | 4 | 0x42 |
| AS16-8V-H (SEK52) | | | | | | |

- Regarding the saving of position information, it is distinguished between:
  Singleturn: Saving within one revolution
  Multiturn: Saving within a number of revolutions
- In case of encoders with an extended electronic nameplate (TypCode = "0xFF"), the encoder settings currently cannot be read automatically out of the encoder and thus have to be executed manually. See the instruction "How to parameterise an absolute value encoder with an unknown type code" in the following "Parameter setting" subchapter.
- The encoder settings for all other encoders (TypeCode = "0x00") have to be made manually as well. See the instructions "How to parameterise an absolute value encoder with an unknown type code" in the following "Parameter setting" subchapter.

### 6.3.2.2 Parameterising the encoder

#### Parameterisation dialog (cutout)




Further parameterisation dialogs: ▸ Scaling of position value

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| General settings | | | |
| C00422 | LS_MultiEncoder: Encoder type | 0: Incremental encoder (TTL) | |
| C00421 | LS_MultiEncoder: Supply voltage | 5.0 | V |
| C00420/3 | LS_MultiEncoder: Encoder increment<br>• Setting is relevant for process data channel (incremental SinCos track). | 512 | Incr./rev. |
| C00497/3 | LS_MultiEncoder: Encoder filter time | 1.0 | ms |
| C01112/1 | LS_MultiEncoder: Position offset | 0.0000 | units |
| Settings for SinCos absolute value encoder with HIPERFACE® protocol | | | |
| C00492 | Hiperface: Detected TypCode | - | |
| C00494/1 | Hiperface: Number of revolutions<br>• Number of resolved revolutions of the Hiperface track.<br>• Setting is relevant for parameter channel. | 0 | |
| C00494/2 | Hiperface: Steps per revolution<br>• Resolution of the Hiperface track.<br>• Setting is relevant for parameter channel. | 0 | |
| C00499/1 | Hiperface: Advanced settings | 0x0 | |
| Greyed out = display parameter | | | |

**How to parameterise the SinCos absolute value encoder:**

1. Set the supply voltage (C00421) of the used encoder.
2. As encoder type (C00422), select "2: Absolute value encoder (Hiperface)".
   - From version 14.00.00, the selection "6: Position encoder (Hiperface)" is possible as well.

   When the encoder has already been connected, a change of C00422 causes a renewed Initialisation of the connected encoder.
   - After the initialisation and with a known encoder, the read-out values are displayed in C00494/1..2 and cannot be overwritten.
   - If the value "0xFF" is displayed as TypeCode in C00492, it is an encoder with an extended nameplate. Currently, the settings for this encoder cannot be automatically read out of the encoder yet and thus have to be carried out manually. See the following instructions "How to parameterise an absolute value encoder with an unknown type code".
   - If the value "0x00" is displayed as type code, the encoder has not been detected automatically. In this case, make the settings manually as well. See the following instructions "How to parameterise an absolute value encoder with unknown type code".
3. Only for version 01.xx.xx: Set the number of sine/cosine periods per revolution as "increment" of the encoder in C00420/3.
   - This value can be obtained from the encoder data sheet.
   - This point must be executed last since it initiates a renewed readout of the encoder.
   - From version 02.00.00, this setting will be obtained and set automatically.
4. If you use the encoder as motor encoder and synchronous motor of a third party manufacturer:
   Carry out Pole position identification (PPI). (📖 366)
5. For detecting and monitoring the motor temperature via the Encoder cable:
   Parameterise Motor temperature monitoring (KTY) via encoder. (📖 383)
6. Optional: Parameterise Resolver/multi-encoder open-circuit monitoring. (📖 377)

   In the Lenze setting, the resolver cable and/or encoder cable are monitored with regard to open circuit as a function of the selected speed and position encoder.
7. For a permanent acceptance of the parameter settings:

   Save parameter set.

**How to parameterise a SinCos absolute value encoder with unknown type code:**

If the encoder is not detected automatically, the value "0x00" is displayed in C00492 as TypeCode. In this case, make the settings manually:

1. Set the number of steps per revolution in C00494/2.
2. Set the number of revolutions for a multiturn encoder in C00494/1.
3. Set the number of sine/cosine periods per revolutions as "increment" of the encoder in C00420/3.
   - This value can be obtained from the encoder data sheet.
   - This point must be executed last since it initiates a renewed readout of the encoder.
4. If you use the encoder as motor encoder and synchronous motor of a third party manufacturer:
   Carry out Pole position identification (PPI). (📖 366)
5. For detecting and monitoring the motor temperature via the Encoder cable:
   Parameterise Motor temperature monitoring (KTY) via encoder. (📖 383)
6. Optional: Parameterise Resolver/multi-encoder open-circuit monitoring. (📖 377)

   In the Lenze setting, the resolver cable and/or encoder cable are monitored with regard to open circuit as a function of the selected speed and position encoder.
7. For a permanent acceptance of the parameter settings:

   Save parameter set.

### 6.3.2.3 Display of the current position

In C01119/1, the current actual position of the motor shaft is displayed in [units]. It is calculated from the actual value of the encoder selected in C00422:

$$C01119/1 = \left(Pos \cdot C01204 \cdot \frac{C01202/2}{C01202/1} \cdot \frac{C01203/1}{C01203/2}\right) + C01112/1$$

Pos = encoder position in increments
- For rotative encoders, the following applies: 65536 increments = 1 revolution
- For linear encoders, the following applies: 65536 increment = 1 increment path

[6-2] Actual position calculation

As the term "revolution" does not make any sense in case of linear encoders, the distance covered is defined as the path in which the increments or SinCos periods parameterised in C00420/3 are completely run through once. In of rotative encoders, this corresponds to exactly one encoder revolution.

#### Internal conversion of the encoder information on the motor shaft position

In the inverter, the information of the Hiperface track are factorised so that a revolution of the encoder corresponds to 65536 increments which means a passage of the increments or SinCos periods parameterised in C00420/3. The same applies to linear encoders.

The factor is automatically calculated from the encoder data (C00494/1..2) that have either been saved in the device or manually configured.

Example: 4096 Hiperface increments per revolution ➞ Factor = 65536/4096 = 16

In case of linear encoders, a conversion to one revolution of the motor or gearbox output shaft is made, too. This conversion is made considering the traversing path set in C01111/3 which is covered during a passage of the increments and sin/cos periods parameterised in C00420/3:

$$Pos_{Mot} = Pos_{lin} \cdot \frac{C01111/3}{C01204}$$

[6-3] Conversion to one revolution of the motor or gearbox output shaft in case of linear encoders

**Related topics:**

- Scaling of position value

#### 6.3.2.4 Notes on evaluating absolute value encoder of SEL37 and SELxx types by SICK

For version 02.xx.xx the following applies:

- The 8400 TopLine inverter recognises these encoder types, but sporadically reports the error "Sd4: open circuit MultiEncoder" error if the open-circuit monitoring is activated in C00603/1 (Lenze setting).
  - The reason for this is a faulty open-circuit monitoring during the initialisation of the controller.
  - The error can be acknowledged so that the controller is able to start correctly.
- Moreover, the initialisation time of the encoder is critical. This is the time needed internally by the encoder to respond to a read request of the controller.
  - The maximum initialisation time in the inverter is permanently set to 5.2 s. Depending on the encoder used, it may be possible that this time is not sufficient (see data sheet for the encoder manufacturer)!
- If the monitoring (e.g. die open-circuit monitoring) triggers with the "Fault" error response due to an encoder error during operation, the position is not valid anymore in case of absolute value encoders and must be reread from encoder.
  - After the error message is reset, this does not take place automatically which may cause a wrong position!
  - Remedy: After the error message is reset with the SB LS_ParReadWrite_1, rewrite the code C00422 and thus enforce a re-initialisation of the encoder. Then enable the application.

From version 12.00.00 the following applies:

- The initialisation time of the encoder is not critical anymore since the maximum initialisation time can now be adjusted in C00430/1 (to max. 20 seconds).
  - During the set initialisation time, the inverter remains in the "Init" device state and cannot be enabled.
  - The open-circuit monitoring is also suppressed during the set initialisation time so that the error behaviour of the version 02.xx.xx is remedied as well.
- If the monitoring (e.g. die open-circuit monitoring) triggers with the "Fault" error response due to an encoder error during operation, the position is not valid anymore in case of absolute value encoders and must be reread from encoder.
  - After the error message is reset, this takes place automatically.

### 6.3.2.5 Configuration example 1: Absolute wire draw encoder "XKS09-HTBM0527" by SICK

#### Data

- Wire length: 5 m
- Hiperface resolution: 4096 digits x 4184 "revolutions"
- SinCos resolution: 128 increments (1 period ≡ 1.1953 mm → 153 mm ≡ one "revolution")
- 1 [unit] = 1 mm
- Feed of the mechanics = 4 mm/revolution

#### Scaling of the position value

In order to scale the evaluation of the SinCos track, the traverse path in [units] has to be entered in C01111/3 which is covered in order to pass through once the SinCos increment of the encoder (C00420/3) (here 128 increments x 1.1953 mm period length = 153 mm).

The absolute Hiperface track is scaled by the setting of the "steps per revolution" in C00494/2. Here, a revolution is equated to the path which corresponds to the SinCos number of increments of the encoder (C00420/3).

#### Required settings

| Parameters | Info | Setting for example |
|---|---|---|
| C00494/1 | Hiperface: Number of revolutions | 4184 |
| C00494/2 | Hiperface: Steps per revolution | 4096 |
| C01110 | LS_MultiEncoder: Solid measure | 1: linearly unipolar |
| C01111/3 | LS_MultiEncoder: Traverse path - encoder increment | 153 |

In "Machine parameters", the feed constant and the gearbox factors for motors and position encoders must be set as follows:

- The position encoder gearbox factor must be set to 1:1.
- When the motor gearbox factor is set to 1:1, the path covered at one motor revolution has to be entered as feed constant in C01204 in [units].
- In case of a different motor gearbox factor, the path covered at one defined "drive roll" revolution has to be entered as feed constant in C01204 in [units].

The feed of the mechanics entered in C01204 (here: 4 mm) is displayed with 65536 increments.

### 6.3.2.6 Configuration example 2: Distance sensor "DME5000-117" by Sick

#### Data

from DME5000 operating instructions (8009813/US35/2012-07-02):

| Period Length (mm) | Type code | Resolution RS-485 (mm) | Resolution sin/cos when zero crossings are counted (mm) |
|---|---|---|---|
| 1 | 90h | 1/32 | 0.25 |
| 2 | 91h | 1/16 | 0.5 |
| 4 | 92h | 1/8 | 1 |
| 8 | 93h | 1/4 | 2 |
| 16 | 94h | 1/2 | 4 |

- Maximum distance: 70 m
- Default setting of parameter "Period Length": 1 mm
  (4 zero crossings while passing one period * 0.25 mm = 1 mm)
- Default setting of parameter "Resolution": 125 µm (must not be changed)

These default settings result in the following resolutions:

- 1 "encoder revolution" = 10 mm
- Hiperface resolution: 1/32 mm ➞ 320 digits per "revolution" x 7000 "revolutions"
- Sin/cos resolution: 10 increments per "revolution"
  (1 period ≡ 1 mm ➞ 10 mm ≡ one "revolution" = 10 periods or increments)
- 1 [unit] = 1 mm

#### Scaling of the position value

In order to scale the evaluation of the SinCos track, the traverse path in [units] has to be entered in C01111/3 which is covered in order to pass through once the SinCos increment of the encoder (C00420/3) (here 10 increments x 1 mm period length = 10 mm).

The absolute Hiperface track is scaled by the setting of the "steps per revolution" in C00494/2. Here, a revolution is equated to the path which corresponds to the SinCos number of increments of the encoder (C00420/3).

#### Required settings

| Parameters | Info | Setting for example |
|---|---|---|
| C00494/1 | Hiperface: Number of revolutions | 7000 |
| C00494/2 | Hiperface: Steps per revolution | 320 |
| C01110 | LS_MultiEncoder: Solid measure | 1: linearly unipolar |
| C00420/3 | LS_MultiEncoder: Encoder increment | 10 |
| C01111/3 | LS_MultiEncoder: Traverse path - encoder increment | 10 |

### 6.3.3 SSI absolute value encoder with Stegmann-SSI protocol

All SSI encoders which use the Stegmann SSI protocol are supported.

- Supported bit rates for SSI communication: 100 ... 1000 kbits
- Supported data word widths: 1 ... 31 bits (effective)
- Supported output code of the SSI encoder: Gray or binary
- The SSI encoder can be used as position encoder or master encoder with a minimum cycle time of 1 ms.
- The SSI encoder can be supplied via X8 up to a maximum voltage of 12 V and a maximum current of 0.25 A.
- The received SSI data words are provided to the application via the LS_MultiEncoder system block for further processing in the function block editor.

**Note!**

For some SSi encoder types it is common that despite a high data transfer rate an slow position detection (e.g. every 5, 10 or 20 ms) takes place. In this case, the position control, working in a 1-ms cycle, only functions with low gain and thus low dynamics since the values of the SSI encoder are only updated every x ms.

**When an external SSI laser position encoder is used at X8:**

Also observe the Notes for using an SSI laser position encoder at the end of this chapter. (📖 349)

The technical data of the synchronous serial interfaces (SSI) can be found in the **8400 hardware manual**.

- The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 inverter.

**Tip!**

When C00422 = 7 / 8 (position encoder external/SSI), the drive system is directly connected to the position of the feedback.

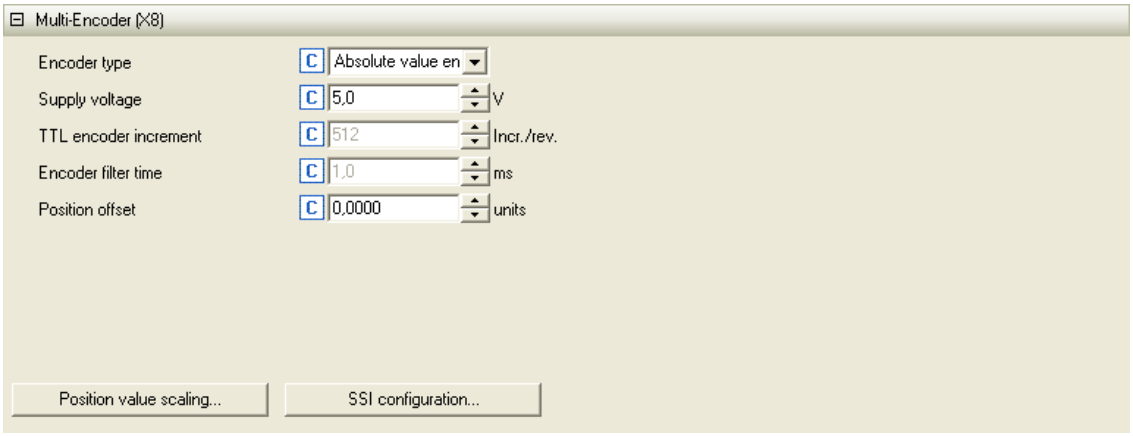
Code C01112/1 ( LS_MultiEncoder : Position offset) does not have any impact on the encoder position.

Finally, only

- a once-only homing has to be executed
  ▸ Homing (📖 618), mode 100: SetRef) or
- "mains switching" can be carried out.

### 6.3.3.1 Parameterising the encoder

#### Parameterisation dialog (cutout)

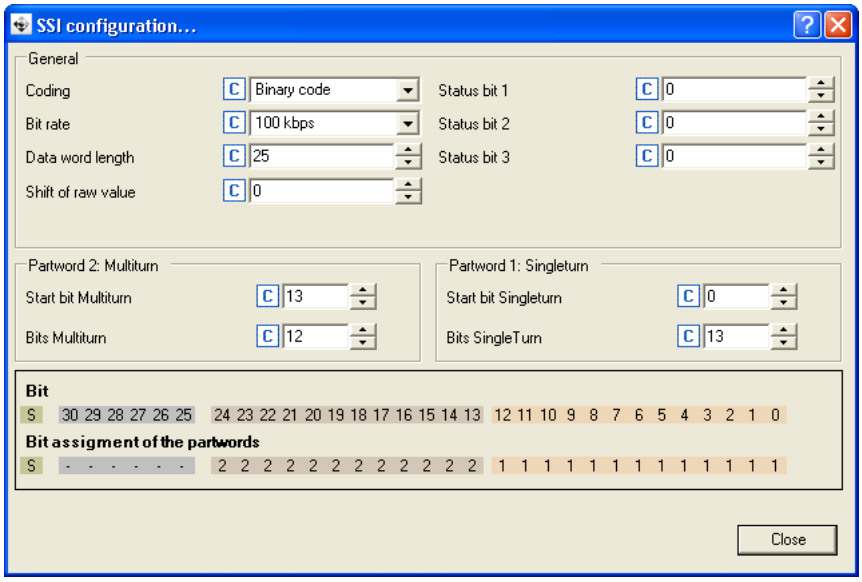



Further parameterisation dialogs: ▸ Scaling of position value ▸ SSI configuration

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00422 | LS_MultiEncoder: Encoder type | 0: Incremental encoder (TTL) | |
| C00421 | LS_MultiEncoder: Supply voltage | 5.0 | V |
| C01112/1 | LS_MultiEncoder: Position offset | 0.0000 | units |

### SSI configuration

Press the **SSI configuration...** button to get to the parameterisation dialog for configuring the SSI encoder:




| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| General | | | |
| C00428 | SSI encoder: Coding | 0: Binary code | |
| C00427 | SSI encoder: Bit rate | 1: 100 kbps | |
| C00426/1 | SSI encoder: Data word length | 25 | |
| C00426/7 | SSI encoder: Shift of raw value | 0 | |
| C00426/6 | SSI encoder: Status bit 1 | 0 | |
| C00426/8 | SSI encoder: Status bit 2 | 0 | |
| C00426/9 | SSI encoder: Status bit 3 | 0 | |
| Partword 2: Multiturn | | | |
| C00426/5 | SSI encoder: Start bit Multiturn | 13 | |
| C00426/4 | SSI encoder: Bits Multiturn | 12 | |
| Partword 1: Singleturn | | | |
| C00426/3 | SSI encoder: Start bit Singleturn | 0 | |
| C00426/2 | SSI encoder: Bits SingleTurn | 13 | |

**How to parameterise the SSI encoder:**

1. Set the supply voltage (C00421) of the used SSI encoder.

   In case of a higher supply power of encoders, e.g. with laser-based length measuring systems, an external supply of the encoders must be provided.

2. As encoder type (C00422), select "4: Absolute value encoder (SSI)".

3. Set the bit rate (C00427) for SSI communication.
   - For SSI protocols, the permissible baud rate is reduced if the cable length is increased. A safe bit rate must be set, depending on the length of the used encoder cable and the electromagnetic interference level.

4. Set the partitioning of the SSI data word in C00426/1...5 according to the used SSI encoder (see Configuration example 1).

5. If an SSI encoder with Gray coding is used:
   Select "1: Gray code" in C00428 to convert from Gray to binary.
   - In the Lenze setting "0: Binary code", no conversion takes place, i.e. an SSI encoder with binary coding is expected.

6. If a linear SSI encoder is used instead of a rotative encoder or the encoder value is to be interpreted with or without sign:
   Set the corresponding standard in C01110. ▸ Scaling of position value

### SSI encoder coding "Position Gray"

From version 14.00.00 onwards, also SSI encoders are supported that transfer the position data in Gray coding and the status bit in binary mode (uncoded). For an encoder with this coding, select "2:Position Gray" in C00428. Then, only the pure position data are converted from Gray to binary code, and status bits provided by the encoder are directly evaluated. The position data converted to binary format are then added to the status bits again and provided as raw value at the outputs *wHighWord* and *wLowWord* of the LS_MultiEncoder. This takes place as a provision of the position raw value in gray code would not make any sense; the value, in case of a linear encoder, would feature jumps instead of a linear characteristic.

### Evaluation of further status bits

From version 14.00.00 onwards, two more status bits of the SSI encoder can be evaluated. The bits to be evaluated in the SSI data current have to be set in C00426/8..9.

If the evaluation of a status bit shall be deactivated completely, the value "32" has to be entered into the corresponding code. Thus, the status bit always has the value "0".

### Left shift of raw value

The data read-in by the encoder is processed via the settings of the start bits for singleturn and multiturn and the number of bits in such a way that a position signal suitable for the inverter is created that can be fed into the function block interconnection.

The raw value of the received data coming from the SSI encoder is provided at the outputs *wHighWord* and *wLowWord* of the LS_MultiEncoder. It is always provided in binary format, i.e. in case of Gray code encoders after the received data is decoded. The MSB of the data is positioned at bit 16 of the *wHighWord* output.

In order to receive a standard data format (LSB at the lowest position) an offset can be set in C00426/7. The setting C00426/7 = 32 - data word length serves to display the raw value of the received data telegram so that the LSB is at bit 0 of the *wLowWord* output:

**Clarification of the data sequence when the data word length is = 26 bits**

| *wHighWord* | | | | | | | | | | | | | | | | *wLowWord* | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data sequence without offset (C00426/7 = 0): | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | |
| Data sequence with an offset by 6 bits (C00426/7 = 6): | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

### 6.3.3.2 Important notes for using an SSI laser position encoder

If you are using an SSI laser position encoder at X8 and, deviating from the Lenze setting, you have activated the position control for quick stop (C00104/1 - bit 0 and/or bit 1 set), please note the following:

If the laser beam is interrupted, due to the "TroubleQuickStop" following error standard response, the drive may move unexpectedly.

- A drive that had already reached the position-controlled standstill can execute a movement.
- A drive that is ramping down can continue running.

Possible countermeasures:

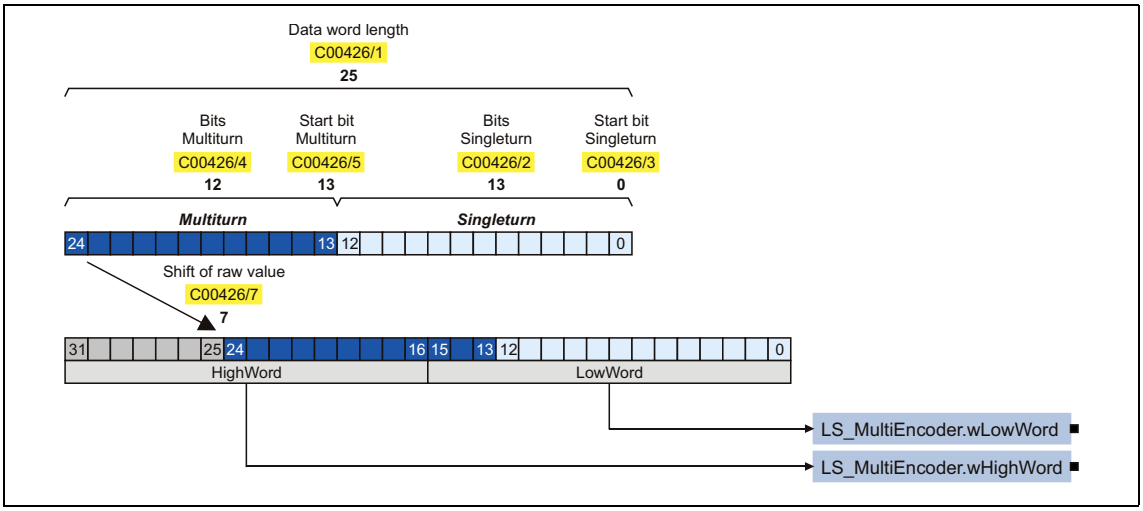
- Do not use the inverter with the above-described parameter setting.
- By implementing suitable other measures, ensure that, in terms of safety aspects, the response described is acceptable for the specific application case.
- Monitor the valid position:
  - Use parameters C01112/2 and C01112/3 to define a value range for valid positions.
  - Configure the LS_MultiEncoder.*bFail* output so that controller inhibit is set when the output at the drive is active. Take other requirements relating to the application into consideration (e.g. application of brake).
  - Resulting drive behaviour: If the value range set is exited, the LS_MultiEncoder.*bFail* output activates controller inhibit. ▸ Encoder value monitoring

**Note!**

The response of the drive to interruptions of the laser beam is always part of the error scenarios to be checked during commissioning.

### 6.3.3.3 Configuration example 1: Optical multi-turn rotary encoder "G0M2H Z04" by Baumer

Coding: Gray code with 25 bit data word length (8192x4096)



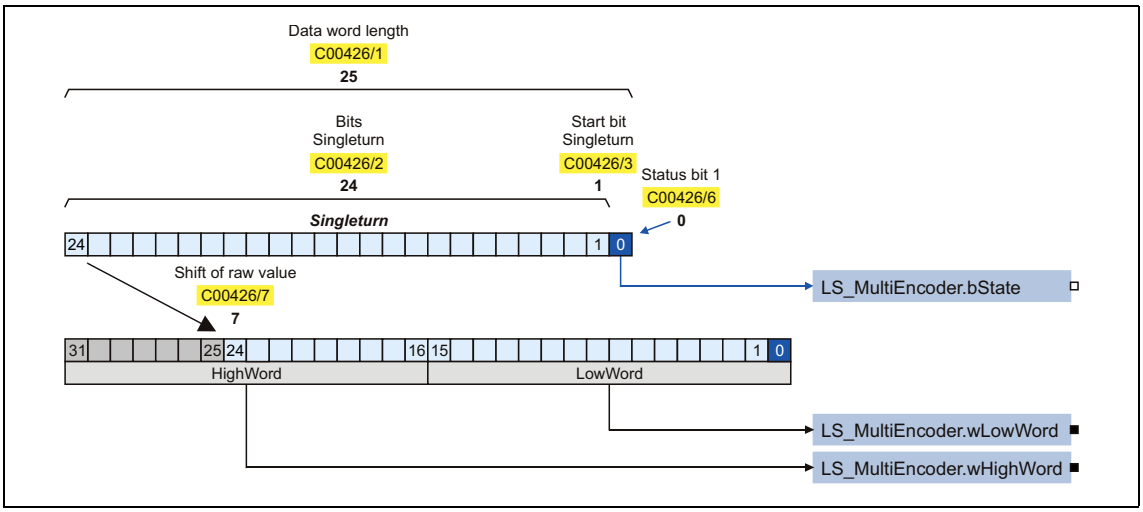

| Parameters | Info | Setting for example 1 |
|---|---|---|
| C00426/1 | SSI encoder: Data word length | 25 |
| C00426/2 | SSI encoder: Bits SingleTurn | 13 |
| C00426/3 | SSI encoder: Start bit Singleturn | 0 |
| C00426/4 | SSI encoder: Bits Multiturn | 12 |
| C00426/5 | SSI encoder: Start bit Multiturn | 13 |
| C00426/6 | SSI encoder: Status bit 1 | 0 (32)* |
| C00426/7 | SSI encoder: Shift of raw value | 7 |
| C00426/8 | SSI encoder: Status bit 2 | 0 (32)* |
| C00426/9 | SSI encoder: Status bit 3 | 0 (32)* |
| C00428 | SSI encoder: Coding | 1: Gray code |
| C01110 | LS_MultiEncoder: Solid measure | 0: rotatively unipolar |

* As the encoder data word in this example does not contain any status bits, you the evaluation of the status bits can also be deactivated completely by entering the value "32" in C00426/6, C00426/8 and C00426/9. Thus, the three status bit outputs of the LS_MultiEncoder always have the value "0".

- In "Machine parameters", the feed constant and the gearbox factors for motor and position encoder must be set correctly. (📖 593)
- One encoder revolution is displayed with 65536 increments.

### 6.3.3.4 Configuration example 2: Distance sensor "DME5000-111" by Sick

Coding: Gray code with 25 bits of data word length
(Bit 0 = status bit in binary format, bit 1 ...24 = position, resolution 1 bit = 0.1 mm)




| Parameters | Info | Setting for example 2 |
|---|---|---|
| C00426/1 | SSI encoder: Data word length | 25 |
| C00426/2 | SSI encoder: Bits SingleTurn | 24 |
| C00426/3 | SSI encoder: Start bit Singleturn | 1 |
| C00426/4 | SSI encoder: Bits Multiturn | 0 |
| C00426/5 | SSI encoder: Start bit Multiturn | 0 |
| C00426/6 | SSI encoder: Status bit 1 | 0 |
| C00426/7 | SSI encoder: Shift of raw value | 7 |
| C00426/8 | SSI encoder: Status bit 2 | 0 (32)* |
| C00426/9 | SSI encoder: Status bit 3 | 0 (32)* |
| C00428 | SSI encoder: Coding | 2: Position Gray |
| C01110 | LS_MultiEncoder: Solid measure | 1: linearly unipolar |

* As the encoder data word in this example contains only one status bit, the evaluation can be deactivated completely by entering the value "32" in C00426/8 and C00426/9. Thus, the status bit outputs *bState2* and *bState3* of the LS_MultiEncoder always have the value "0".

In "Machine parameters", the feed constant and the gearbox factors for motors and position encoders must be set as follows:

- The position encoder gearbox factor must be set to 1:1.
- When the motor gearbox factor is set to 1:1, the path covered at one motor revolution has to be entered as feed constant in C01204 in [units].
- In case of a different motor gearbox factor, the path covered at one defined "drive roll" revolution has to be entered as feed constant in C01204 in [units].
- The feed of the mechanics entered in C01204 is displayed with 65536 increments.

### Scaling of the position value

In order to scale the position change provided by the position encoder at this feed to 65536 as well, enter the traverse path in [units] in C1111/1 and the corresponding difference of the raw value provided by the encoder in C1111/2.

- Only the bits that contain the position data may be taken from the data word for the raw value. If e.g. the bit 0 contains status information, the raw value must be shifted to the right by this bit since only then the SLB of the position value is positioned correctly.

### Example for scaling

1 [unit] = 1 mm

Feed constant (C01204) = 4 mm/revolution

Encoder resolution = 50 µm ➞ raw value difference = 20 [digit]/1 [unit]

➞ Difference - traverse path (C1111/1) = 1 [unit]

➞ Difference - encoder value (C1111/2) = 20 [digit]

Thus, at a feed of the mechanics of 4 mm, the encoder provides a position difference of 65536 increments.

$$\text{Position} = \text{Raw value} \cdot \frac{\text{Encoder feed constant}}{65536}$$

$$\text{with encoder feed constant} = \frac{\text{C01111/1} \cdot 65536^2}{\text{C01111/2} \cdot \text{C01204}}$$

$$\text{Position} = \text{Raw value} \cdot \frac{\text{C01111/1} \cdot 65536}{\text{C01111/2} \cdot \text{C01204}}$$

### 6.3.4 Digital frequency coupling

This function extension is available from version 12.00.00!

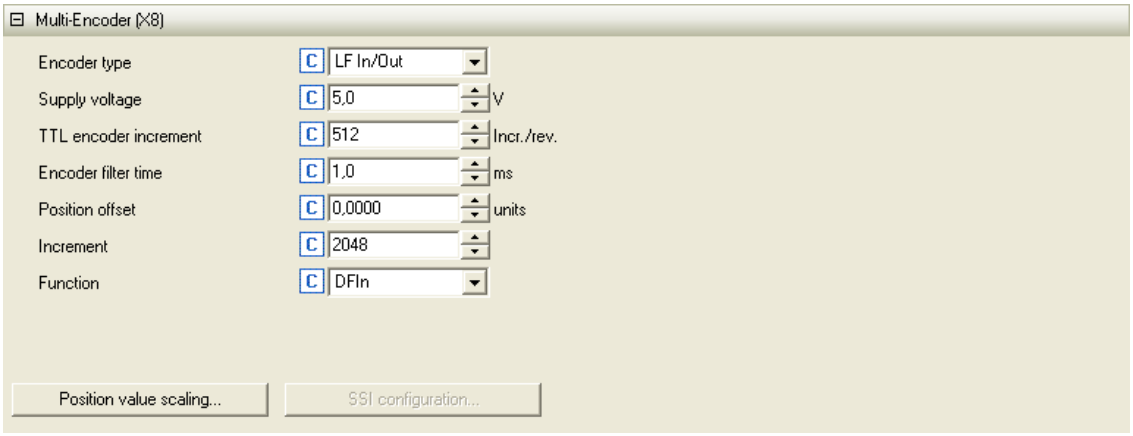
The multi-encoder interface (X8) can also be reconfigured for the output of encoder signals with TTL level (0 ... 500 kHz) in order to realise a "digital frequency coupling".

- The "digital frequency coupling" term describes a digital setpoint transmission and evaluation path between a setpoint source and one or several inverters.

**Note!**

When the digital frequency coupling is used, the multi-encoder interface is not available anymore for further encoder functions! Only encoder signals with TTL level are permissible as input signals at X8!

**Parameterisation dialog (cutout)**




Further parameterisation dialogs: ▸ Scaling of position value

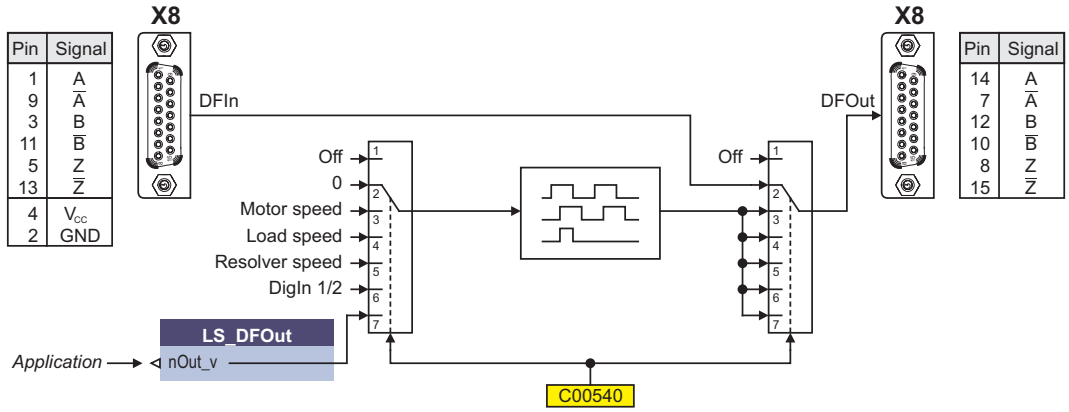
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00422 | LS_MultiEncoder: Encoder type | 0: Incremental encoder (TTL) | |
| C00421 | LS_MultiEncoder: Supply voltage | 5.0 | V |
| C00420/3 | LS_MultiEncoder: Encoder increment | 512 | Incr./rev. |
| C00497/3 | LS_MultiEncoder: Encoder filter time | 1.0 | ms |
| C01112/1 | LS_MultiEncoder: Position offset | 0.0000 | units |
| C00030/1 | LS_DFOut: number of increments | 2048 | |
| C00540 | LS_DFOut: Function | 2: DFIn | |

**How to parameterise the master frequency output:**

1. As encoder type (C00422), select "5: LF In/Out" to activate the master frequency output.
2. Set the number of increments for the digital frequency output in C00030/1.

   The number of increments determines after how many output increments a zero pulse will be generated. Each zero pulse defines a covered "revolution" of the rotary transducer simulated by the digital frequency output.
3. Set the desired function of the digital frequency output in C00540 (see the following section).

**Selecting the function of the digital frequency output**




| Function (C00540) | Info |
|---|---|
| 1: Off | Digital frequency output not active<br>• The frequency "0" is output at the digital frequency output.<br>• All tracks remain on the level output last.<br>• After switching on the inverter, the tracks A, B and Z are set to HIGH level. |
| 2: DFIn | The TTL input signals at X8 are connected through to the digital frequency output. |
| 3: MotorSpeed | Output of the motor encoder<br>• The angle of rotation derived from the motor encoder in [increments] is output as frequency signal after being evaluated with the number of increments set in C00030/1. |
| 4: LoadSpeed | Output of the load encoder<br>• The angle of rotation derived from the load encoder in [increments] is output as frequency signal after being evaluated with the number of increments set in C00030/1. |
| 5: Resolver | Output of the resolver angle<br>• The angle of rotation derived from the resolver input in [increments] is output as frequency signal after being evaluated with the number of increments set in C00030/1.<br>• It is irrelevant for the output whether the resolver input is used as load encoder, motor encoder or not at all within the motor control. |
| 6: DigIn 1/2 | Output of the digital inputs DI1/DI2<br>• The angle of rotation derived from the digital inputs in [increments] is output as frequency signal after being evaluated with the number of increments set in C00030/1. |
| 7: AFB input | Output of a speed signal of the application<br>• The speed signal defined via the *nOut_v* input of the LS_DFOut system block is integrated and output as frequency signal after being evaluated with the number of increments set in C00030/1. |

### 6.3.4.1 Problem description - speed variations

As digital frequency, rectangular encoder signals are processed. The speed can be determined by counting edges and a defined time interval (1 ms). This defined time interval of 1 ms and a finite number of encoder increments per revolution cause the calculated speed signal to only accept discrete values. This gives the impression of a very disturbed and incorrect signal. This effect increases with a lower number of increments.

Example: An encoder with 2048 increments at a speed of 60 rpm is to be simulated via the digital frequency output.

- Setpoint speed *nOut_v* = 60 rpm = 1 $Hz_{mech.}$
- Simulated encoder: 2048 increments
  (by 4-fold evaluation, 4 * 2048 = 8192 edges are counted per revolution)
- Output frequency = *nOut_v* * number of increments (C00030/1) = 1 $Hz_{mech.}$ * 2048 = 2048 Hz

At a measurement time of 1 ms, 8,192 edges per ms are achieved. As only integer counting processes can occur, sometimes 8 and sometimes 9 increments are counted. Hence, the speed is calculated as follows:

$$n_{mess_1} = 60\ rpm \cdot \frac{8}{8.192} = 58.59\ rpm$$

or

$$n_{mess_2} = 60\ rpm \cdot \frac{9}{8.192} = 65.91\ rpm$$

The perceived speed variation amounts to:

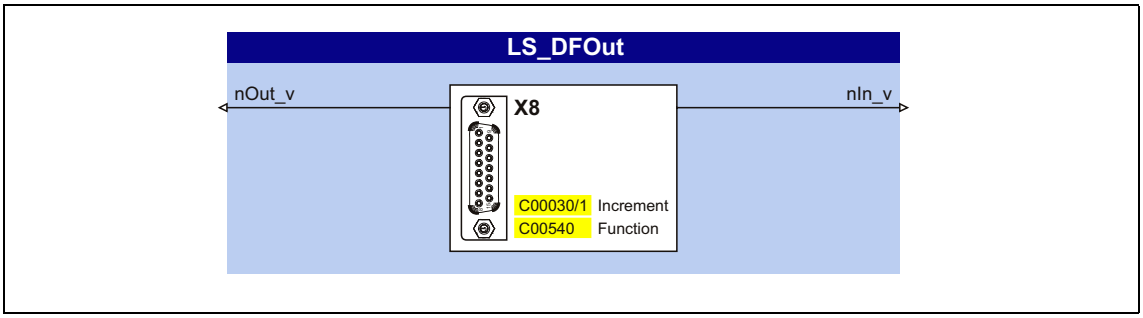
$$\Delta n_{mess} = n_{mess_2} - n_{mess_1} = 7.32\ rpm$$

**General formula for calculating the expected speed variation**

$$\text{ess} = \frac{15000}{\text{Number of increments}}$$

### 6.3.4.2 Internal interfaces | System block "LS_DFOut"

The **LS_DFOut** system block maps the digital frequency output in the FB Editor:




#### Inputs

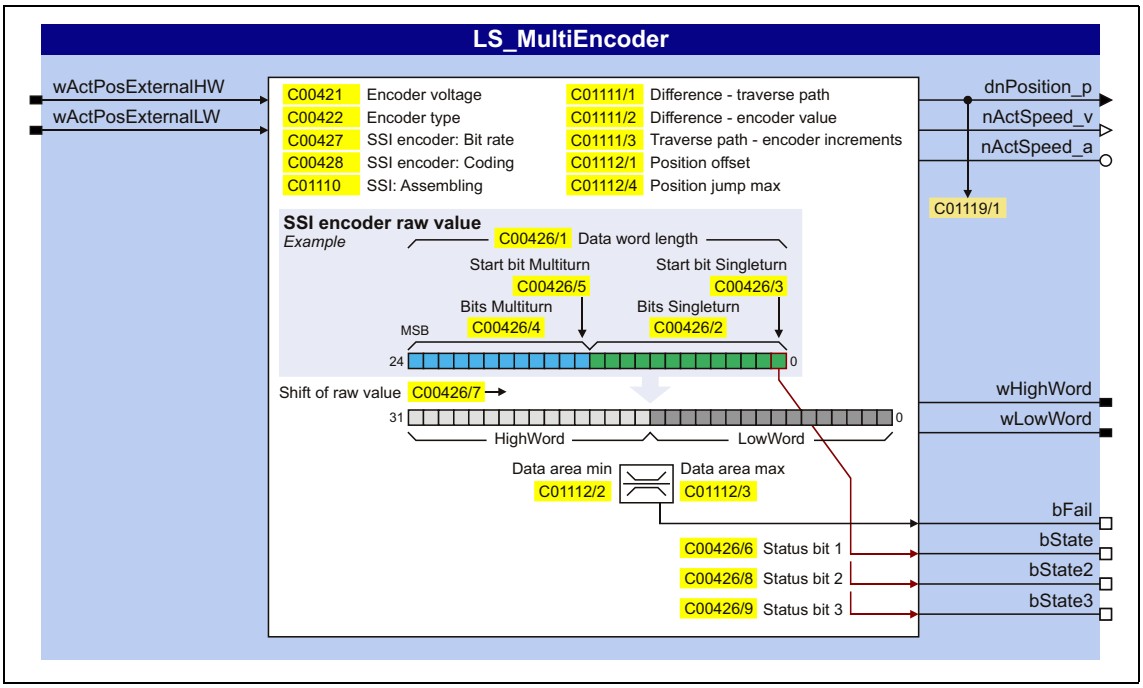
| Designator<br>Data type | Information/possible settings |
|---|---|
| nOut_v<br>INT | Speed in [increments/ms] from the application which is to be output via the digital frequency output X8 in the form of encoder signals with TTL levels.<br>• Scaling: 16384 ≡ 15000 rpm<br>• For selecting this signal source, C00540 must be set to "7: AFB input". |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nIn_v<br>INT | Speed in [increments/ms] that has been detected via the digital frequency input X8 in the form of encoder signals with TTL levels.<br>• Scaling: 16384 ≡ 15000 rpm |

### 6.3.5 Internal interfaces | "LS_MultiEncoder" system block

The **LS_MultiEncoder** system block provides the application with speed, position and error information from the encoder connected to the multi encoder interface X8 in the form of process signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| wActPosExternalHW<br>wActPosExternalLW<br>WORD<br>(from version 14.00.00) | These inputs serves to feed in the position signal of an external encoder (e.g. via CAN or another fieldbus) and use it as position signal.<br>Scaling: LW = 1 revolution, HW = number of revolutions<br>Thus, an encoder revolution is displayed with 65536 increments. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnPosition_p<br>DINT | Actual position value from the encoder in [increments] |
| nActSpeed_v<br>INT | Actual speed from the encoder in [increments/ms] |
| nActSpeed_a<br>INT | Actual speed from the encoder in [%]<br>• 100 % ≡ reference speed (C00011) |
| wHighWord<br>wLowWord<br>WORD | 32-bit data telegram from the encoder (raw data)<br>▸ SSI absolute value encoder with Stegmann-SSI protocol (📖 344)<br>The output is always made in binary format, i.e. in case of Graycode encoders after the received data is decoded. The MSB of the data is at bit 16 of the *wHighWord* output. In order to obtain a standard data format (LSB at the lowest position), an offset can be set in C00426/7.<br>▸ Left shift of raw value (📖 348) |

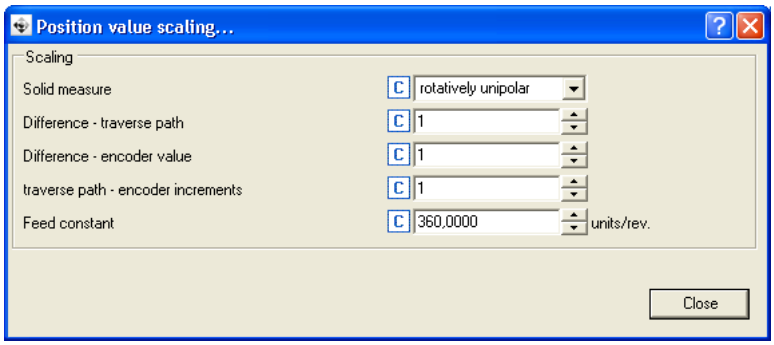
| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bFail<br>BOOL | TRUE | Group error: There is an error (e.g. encoder wire breakage) according to the *bFail* configuration in C00431/1.<br>From version 14.00.00, the error type is displayed bit-coded in C00451/1. ▸ Monitoring |
| bState<br>BOOL<br>(from version 02.00.00) | Status bit 1 of the encoder<br>• The bit position of the status bit within the 32-bit data telegram can be set in C00426/6.<br>• If C00426/6 = "32", the evaluation of the status bit 1 is deactivated and FALSE is always output at *bState*. | |
| bState2<br>BOOL<br>(from version 14.00.00) | Status bit 2 of the encoder<br>• The bit position of the status bit within the 32-bit data telegram can be set in C00426/8.<br>• If C00426/8 = "32", the evaluation of the status bit 2 is deactivated and FALSE is always output at *bState2*. | |
| bState3<br>BOOL<br>(from version 14.00.00) | Status bit 3 of the encoder<br>• The bit position of the status bit within the 32-bit data telegram can be set in C00426/9.<br>• If C00426/9 = "32", the evaluation of the status bit 3 is deactivated and FALSE is always output at *bState3*. | |

### Display parameter

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01119/1 | LS_MultiEncoder: Current position | - | units |
| C01119/2 | LS_MultiEncoder: Maximum travel distance | - | units |
| Greyed out = display parameter | | | |

### 6.3.5.1 Scaling of position value

If you click the **Scaling of position value...** button in the parameterisation dialog for the encoder/feedback system in the category "Multi-Encoder (X8), you get to the respective parameterisation dialog:




**Short overview of the relevant parameters:**

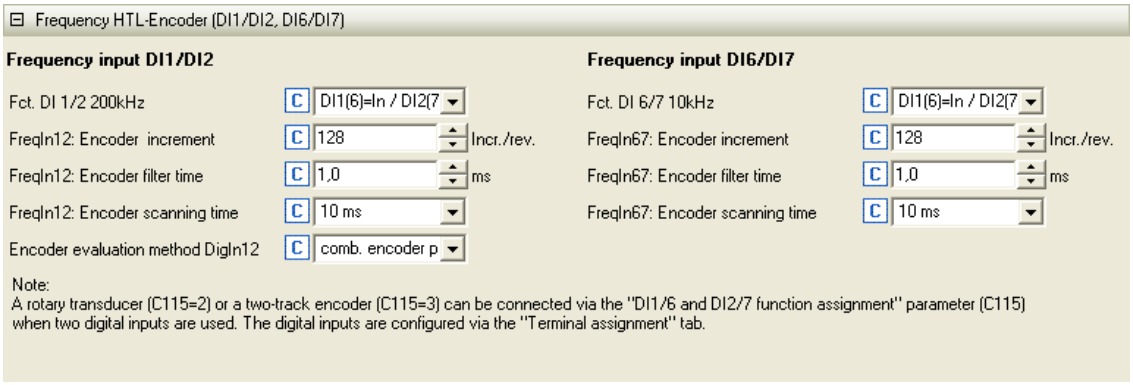
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01110 | LS_MultiEncoder: Solid measure<br>• Setting of rotative or linear encoders.<br>**Note:** The selection "rotative" or "linear" only has an impact if the encoder has not been detected automatically or the encoder data definitely are to be taken from the codes (bit 0 in C00499/1 is set to "1"). The "unipolar" or "bipolar" selection definitely has an impact. | 0: rotatively unipolar | |
| Settings for linear SinCos absolute value encoder with HIPERFACE® protocol | | | |
| C01111/1 | LS_MultiEncoder: Difference - traverse path<br>• Specification of the traverse path while passing through the step number of the Hiperface track given in C01111/2.<br>• Setting is relevant for parameter channel. | 1 | units |
| C01111/2 | LS_MultiEncoder: Difference - encoder path<br>• Step number of the Hiperface track.<br>• Setting is relevant for parameter channel. | 1 | Steps |
| C01111/3 | LS_MultiEncoder: Traverse path - encoder increment<br>• Specification of the traverse path while passing through the number of increments of the SinCos track given in C00420/3.<br>• Setting is relevant for process data channel (incremental SinCos track). | 1 | units |
| Further settings | | | |
| C01204 | MCK: Feed constant | 360.0000 | units/rev. |

### 6.3.5.2 Setting of a static position offset

If, for example, the encoder cannot be mounted mechanically in order that the supplied position matches the mechanics or if for other reasons, a static position offset is required, this value can be set in C01112/1.

## 6.4 Frequency HTL encoder (DI1/DI2, DI6/DI7)

### Parameterisation dialog (cutout)




| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| Settings for HTL encoder at DI1/DI2 | | | |
| C00115/1 | Fct. DI 1/2 200kHz<br>• Function of the digital inputs DI1 and DI2 | 0: DI1(6)=In / DI2(7)=In | |
| C00420/1 | FreqIn12: Encoder increment<br>• If the digital inputs DI1 and DI2 are used as encoder inputs. | 128 | Incr./rev. |
| C00497/1 | FreqIn12: Encoder filter time<br>• If the digital inputs DI1 and DI2 are used as encoder inputs. | 1.0 | ms |
| C00425/1 | FreqIn12: Encoder scanning time<br>• If the digital inputs DI1 and DI2 are used as encoder inputs. | 10 | ms |
| C00496 | ▸ Encoder evaluation method DigIn12 (📖 363) | 2: Comb. encoder method | |
| Settings for HTL encoder at DI6/DI7 | | | |
| C00115/2 | Fct. DI 6/7 10kHz<br>• Function of the digital inputs DI6 and DI7 | 0: DI1(6)=In / DI2(7)=In | |
| C00420/2 | FreqIn67: Encoder increment<br>• If the digital inputs DI6 and DI7 are used as encoder inputs. | 128 | Incr./rev. |
| C00497/2 | FreqIn67: Encoder filter time<br>• If the digital inputs DI6 and DI7 are used as encoder inputs. | 1.0 | ms |
| C00425/2 | FreqIn67: Encoder scanning time<br>• If the digital inputs DI6 and DI7 are used as encoder inputs. | 10 | ms |

### 6.4.1 Parameterising digital inputs as encoder inputs

The function of the digital inputs DI1/DI2 and DI6/DI7 is defined via C00115/1...2.

To be able to use the digital inputs as encoder inputs, select 2, 3, or 4 (Lenze recommendation: 2) in C00115/1 or C00115/2, depending on the input terminals used.

| Selection in C00115/1...2 | Function |
|---|---|
| 2: DI1(6)&DI2(7)=FreqIn (2-track) | DI1/6 and DI2/7 = 2-track frequency input<br>• Permits a two-track evaluation of the encoder including correct detection of the direction of rotation. |
| 3: DI1(6)=FreqIn / DI2(7)=Direction | DI1/6 = 1-track frequency input<br>DI2/7 = specification of direction |
| 4: DI1(6)=CountIn / DI2(7)=In | DI1/6 = counter input<br>DI2/7 = digital input |

**Danger!**

- For single-track evaluation, make sure that the sign is correctly specified. Otherwise, the motor may overspeed.
- If servo control (SC) or V/f control (VFCplus + encoder) are used: For safety reasons, always select "Fault" (Lenze setting) in C00586 as a response for the (open-circuit) monitoring of the encoder!
- If an HTL encoder is used at the digital input terminals:
  Observe the maximum input frequencies of the digital inputs!
  - DI1/DI2: max. 200 kHz
  - DI6/DI7: max. 10 kHz

**Note!**

If the digital inputs are parameterised as encoder inputs, the corresponding output signals (*bIn1*/*bIn2* and *bIn6*/*bIn7*) at the LS_DigitalInput system block are automatically set to FALSE.

Wiring diagram, assignment and electrical data of the digital input terminals can be found in the **hardware manual 8400** in the chapter "technical data". The hardware manual is stored in electronic form on the data carrier supplied with the 8400 inverter.

**General procedure**

(if the encoder is connected to the digital inputs DI1 and DI2)

1. Define the function of the digital inputs DI1 and DI2 in C00115/1.
2. Set the encoder increments in C00420/1.
3. Select "1: Encoder signal FreqIn12" in C00495/1.
4. Adapt the filter time of the speed measurement in C00497/1.
5. In the case of encoders with a very low resolution (number of increments < 120 increments): Change the encoder evaluation procedure in C00496 if necessary.

**Related topics:**

- Digital input terminals (📖 385)
- Using DI1(6) and DI2(7) as frequency inputs (📖 389)

### 6.4.2 HTL encoder at DI1/DI2

**Note!**

At the digital terminals DI1 and DI2, only encoders with HTL level can be used.

In spite of the selected operating mode without encoder feedback, the actual speed value (C00051) is calculated if an encoder is connected and "1: Encoder signal FrqIn12" is selected in C00495.

#### Encoder evaluation method DigIn12

Depending on the encoder used at the digital inputs DI1 and DI2, the following table specifies which evaluation method should be selected in C00496:

| Selection in C00496 | Encoder evaluation method |
|---|---|
| 0: High-resolution encoder | High-precision procedure for high-resolution encoders (>=512 increments)<br>• Method for speed measurement with automatic scan time setting (0.5 ... 500 ms).<br>• Evaluation with automatic scanning time minimisation for an optimum dynamic performance.<br>• Particularly suited for high-resolution encoders (≥ 1024 inc) with good signal quality, i.e.<br>  • good scanning ratio 1:1<br>  • exactly 90°-phase offset between track A and B (error ≤ ± 10°)<br>• Not suited for encoders with poor signal quality.<br>• Wiring according to EMC (e.g. motor and encoder cable shielding) is required! |
| 1: Low-resolution encoder (StateLine) | High-precision procedure for low-resolution encoders (<=128 increments)<br>• Exact method for speed measurement with automatic scanning time setting (0.5 ... 500 ms) for low-resolution encoders in the range of 4 .... 128 increments.<br>• Evaluation with automatic scanning time minimisation for an optimum dynamic performance.<br>• Method is also suited for encoders with poor signal quality, e.g. for encoders with high error rate in scanning ratio and phase offset.<br>• This method requires an equidistant period length per encoder increment.<br>• Wiring according to EMC (e.g. motor and encoder cable shielding) is required! |
| 2: Comb. encoder method (Lenze setting) | Combination of the first two procedures as a function of the speed (recommended procedure)<br>• For a high-precision speed measurement suited for encoders with an arbitrary number of increments (4 ... 1024 increments).<br>• Low input frequencies at the encoder inputs:<br>The method is used for low-resolution encoders.<br>• High input frequencies at the encoder inputs:<br>The method is used for high-resolution encoders.<br>• This method is suited for encoders with average to good signal quality.<br>• Evaluation with automatic scanning time minimisation for an optimum dynamic performance.<br>• This method requires an equidistant period length per encoder increment.<br>• Wiring according to EMC (e.g. motor and encoder cable shielding) is required! |
| 3: Edge-counting procedure | Simple edge counting procedure with adjustable scanning time (C00425)<br>• Speed measurement by means of the edges of tracks A and B measured per scanning interval.<br>• Integrated correction algorithm for EMC interference.<br>• Limited suitability for systems with unshielded encoder and/or motor cable.<br>• Limited suitability for encoders with poor signal quality, i.e. high error rate in scanning ratio and phase offset. |

**Tip!**

- We recommend to use the preset combined encoder method (C00496 = 2).
- Use one of the first three procedures (C00496 = 0, 1, or 2) for dynamic applications (e.g. operating mode: servo control).
- For dynamic speed control or positioning processes, use an HTL encoder with 1024 increments.

### Low speeds (except for edge counting)

For the first three methods (C00496 = 0, 1, or 2), the minimum speed that can be measured depends on the encoder resolution.

The quantisation error

- is independent of the encoder resolution,
- exclusively depends on the encoder quality (encoder errors).
- at least amounts to 0.5 rpm.

Internal arithmetic operations automatically maintain the minimally required value of the scanning time in order to achieve maximum dynamics.

| Encoder resolution (Number of increments) | Min. measurable speed in [rpm] |
|---|---|
| **8** | 16 |
| **16** | 8 |
| **32** | 4 |
| **64** | 2 |
| **128** | 1 |
| **256** | 0.5 |
| **≥ 512** | 0.25 |

### Low speeds with edge counting

The minimum speed that can be measured and the quantisation error of speed measurement in the edge-counting procedure (C00496 = 3) depend on the scanning time that can be set in C00425/1 and the encoder resolution.

Depending on accuracy and the requirements with regard to the dynamic performance, the respective scanning time must be selected and set in C00425/1:

| Encoder resolution (Number of increments) | Scanning time [ms] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **5** | **10** | **20** | **50** | **100** | **200** | **500** | **1000** |
| | Min. measurable speed in [rpm] | | | | | | | | | |
| **8** | 1875 | 938 | 375 | 188 | 93.8 | 37.5 | 18.8 | 9.4 | 3.8 | 1.9 |
| **16** | 938 | 469 | 188 | 94 | 46.9 | 18.8 | 9.4 | 4.7 | 1.9 | 0.9 |
| **32** | 469 | 234 | 94 | 46.9 | 23.4 | 9.4 | 4.7 | 2.3 | 0.9 | 0.5 |
| **64** | 234 | 117 | 46.9 | 23.4 | 11.7 | 4.7 | 2.3 | 1.2 | 0.5 | 0.2 |
| **128** | 117 | 58.6 | 23.4 | 11.7 | 5.9 | 2.3 | 1.2 | 0.6 | 0.2 | 0.12 |
| **256** | 58.6 | 29.3 | 11.7 | 5.9 | 2.9 | 1.2 | 0.6 | 0.3 | 0.12 | 0.06 |
| **512** | 29.3 | 14.6 | 5.9 | 2.9 | 1.5 | 0.6 | 0.3 | 0.15 | 0.06 | 0.03 |
| **1024** | 14.6 | 7.3 | 2.9 | 1.5 | 0.7 | 0.3 | 0.15 | 0.07 | 0.03 | 0.01 |

### 6.4.3 HTL encoder at DI6/DI7

**Note!**

Single-track evaluation of the digital DI6/DI7 terminals as speed feedback (C0115/2 = 1 or 3) is not possible. Hence, a single-track encoder cannot be used for speed control at the digital DI6 terminal!

#### Low speeds with edge counting

The speed measurement is evaluated at the digital terminals DI6/DI7 with the edge-counting procedure and a fixed scanning time that can be set in C00425/2.

The minimum speed that can be measured and the quantisation error of speed measurement in the edge-counting procedure depend on the scanning time that can be set in C00425/2 and the encoder resolution.

Depending on accuracy and the requirements with regard to the dynamic performance, the respective scanning time must be selected and set in C00425/2:

| Encoder resolution (Number of increments) | Scanning time [ms] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 5 | 10 | 20 | 50 | 100 | 200 | 500 | 1000 |
| | Min. measurable speed in [rpm] | | | | | | | | | |
| **8** | 1875 | 938 | 375 | 188 | 93.8 | 37.5 | 18.8 | 9.4 | 3.8 | 1.9 |
| **16** | 938 | 469 | 188 | 94 | 46.9 | 18.8 | 9.4 | 4.7 | 1.9 | 0.9 |
| **32** | 469 | 234 | 94 | 46.9 | 23.4 | 9.4 | 4.7 | 2.3 | 0.9 | 0.5 |
| **64** | 234 | 117 | 46.9 | 23.4 | 11.7 | 4.7 | 2.3 | 1.2 | 0.5 | 0.2 |
| **128** | 117 | 58.6 | 23.4 | 11.7 | 5.9 | 2.3 | 1.2 | 0.6 | 0.2 | 0.12 |
| **256** | 58.6 | 29.3 | 11.7 | 5.9 | 2.9 | 1.2 | 0.6 | 0.3 | 0.12 | 0.06 |
| **512** | 29.3 | 14.6 | 5.9 | 2.9 | 1.5 | 0.6 | 0.3 | 0.15 | 0.06 | 0.03 |
| **1024** | 14.6 | 7.3 | 2.9 | 1.5 | 0.7 | 0.3 | 0.15 | 0.07 | 0.03 | 0.01 |

#### Maximum speeds with edge counting

Due to the lower input frequency (max. 10 kHz) compared to the terminals DI1/DI2, operation with maximum speed at the terminals DI6/DI7 is limited. ▸ Digital input terminals (📖 385)

| Encoder resolution (Number of increments) | Max. measurable speed in [rpm] |
|---|---|
| **8** | No restrictions |
| **16** | 37500 |
| **32** | 18750 |
| **64** | 9375 |
| **128** | 4688 |
| **256** | 2344 |
| **512** | 1172 |
| **1024** | 586 |

## 6.5 Pole position identification (PPI)

### Note!

Only required:

- For servo control (SC) with synchronous motor of a third-party manufacturer.
- For servo control (SC) with synchronous motor and use of incremental encoders (TTL or sin/cos encoders as well as multi-pole pair resolvers).
- After changes of the motor feedback system, e.g. encoder exchange.

For the sensorless control of synchronous motors (SLPSM), a pole position identification is not required.

### Note!

**Acceptance of the resolver pole position from a Servo Drive 9400**

The resolver pole position (C00926/1) cannot be simply accepted from a Servo Drive 9400 if the pole position (C58/1) considerably differs from -90 ° in the Servo Drive 9400.

- We always recommend a pole position identification (PLI).
- As an alternative, the following conversion has to be carried out for a transfer of the resolver pole position from a Servo Drive 9400 to the 8400 TopLine:
  Pole position$_{8400}$ (C00926/1) = - (pole position$_{9400_C58/1}$ + 180 °)

For the control of permanent-magnet synchronous machines, the pole position – the angle between the motor phase U and the field axis of the rotor – must be known.

- For Lenze motors with absolute value encoder or resolver, the pole position is already set correctly in C00926/1...2.
- When incremental encoders (TTL or sin/cos encoders) are used, a pole position identification (PPI) is always required after mains switching, even with Lenze motors.
- The inverter can also evaluate multi-pole-pair resolvers.
  - When the number of motor pole pairs is an integer multiple of the number of pole pairs of the resolver, a pole position identification must only be executed once.
  - When the number of motor pole pairs is no integer multiple of the number of pole pairs of the resolver, a pole position identification must be executed after every mains switching.

### Parameterisation dialog (cutout)




| Parameters | Info | Lenze setting |
|---|---|---|
| C00002/34 | Pole position identification (360°) | 0: Off / ready |

The **Identify pole position...** button or the "pole position identification (360°)" device command serve to detect the pole position for the motor encoder currently activated in C00495.

Detailed information on the process and execution of the pole position identification can be found in the following subchapter "Pole position identification 360°".

### 6.5.1 Pole position identification 360°

**Procedure for "pole position identification 360°"**

If all conditions are met, the motor is energised with a direct current corresponding to the lower of the following two values:

$\sqrt{2}$ · Rated device current
or
$\sqrt{2}$ · Rated motor current

- The rotor is aligned through the current flow. This is absolutely necessary for the procedure.
- To ensure that the torque-neutral axis is not accidentally energised and the rotor stops, a 45° current vector is (electrically) generated for a short instant and then (electrically) switched back to 0° (≡ phase U).
  - Then a DC current of the above-mentioned value could be measured in this motor phase.
- If a resolver or an optical encoder without absolute track is used, the difference between the preselected current angle and the mechanical rotor angle is determined. After this, the current vector is (electrically) turned by another 22.5° and the difference between current angle and rotor angle is determined once again.
  - The procedure is repeated 16 times. This corresponds to one electrical revolution. The machine rotates by 360° (mech.)/pole pair number.
  - Take the average value of the 16 measurements to compensate for asymmetries.

**Adjustment of the pole position identification 360°**

The pole position identification can be adjusted to the respective machine and the prevailing moments of inertia by means of the parameters described below:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00644/1 | PLI 360° traversing direction | right rotating field | |
| C00645/2 | PLI 360° max. error tolerance | 0.0 | ° |
| C00646/2 | PLI 360° current amplitude | 100 | % |
| C00647/2 | PLI 360° ramp time | 100 | % |

- The current amplitude can be adjusted proportionally in C00646/2.
  - For large machines and high mass inertia values or for linear direct drives, the current amplitude usually has to be increased.
  - The Lenze setting "100 %" corresponds to the smaller of the two following values:

$\sqrt{2}$ · Rated device current
or
$\sqrt{2}$ · Rated motor current

**Stop!**

If there is no temperature monitoring in the motor and/or the I2xt motor monitoring and the maximum current monitoring are not parameterised correctly, the motor may be permanently damaged when the current amplitude is set too high (e.g. to the maximum value)!

- Motor load monitoring (I2xt) (📖 301)
- Maximum current monitoring (📖 311)

**Note!**

If the current amplitude is set to > 100 % in C00646/2, the device utilisation (Ixt) monitoring and/or one of the motor monitoring functions may respond and cause the abort of the pole position identification.

- The ramp time can be adjusted proportionally in C00647/2.
  - For large machines and high mass inertia values, the ramp time usually has to be increased.
  - For small machines, a reduction of the ramp time can speed up the pole position identification process.
- In some situations it may be helpful to reverse the travel direction in C00644/2 for the pole position identification (e.g. for linear motor at the end stop).
- The "pole position identification 360°" procedure comprises a plausibility check. If the rotor position determined via the encoder system does not correspond to the controlled output position:
  - the pole position identification procedure is aborted.
  - the response parameterised in C00643/1 (Lenze setting: "Fault") is activated.
  - the error message "Id5: Pole position identification error" is entered into the logbook.
- The preset fault tolerance for the plausibility check can be changed in C00645/2.

**Execute pole position identification 360°**



## Danger!

The machine must not be braked or blocked during the pole position identification! For this reason, the pole position identification is not permitted for hanging loads!

During the pole position identification the rotor aligns itself. The motor shaft moves by max. one electrical revolution which causes the corresponding movement of the connected mechanical components!




## Stop!

Check the correct parameterisation of the max. motor current monitoring before carrying out the pole position identification to prevent the motor from being permanently damaged.

▸ Maximum current monitoring (📖 311)



## Note!

By means of controller inhibit, the procedure started can be cancelled anytime, if required, without carrying out a change in C00926.

If the pole position identification is aborted, the response parameterised in C00643/1 is activated (Lenze setting: "Fault").

- If this behaviour is not wanted, deactivate the monitoring by selecting "0: No response" in C00643/1.

**Preconditions for the execution**

- A synchronous motor has been selected.
- No other identification is active.
- No error has occurred.

**How to execute the pole position identification:**

1. Inhibit the inverter if it is enabled, e.g. via the C00002/16 device command or a LOW signal at the X5/RFR terminal.
2. Start the pole position identification with the device command C00002/34 = "1: On / start".
3. Inverter is enabled again.
   - The pole position identification starts.
   - The progress of the identification run can be seen in C00002/34.
   - The identification is completed if the "0: Off / ready" message is displayed in C00002/34.
4. Inhibit inverter again.

**After successful completion...**

...the controller is inhibited automatically and the pole position determined for the activated feedback system is set in the corresponding subcode of C00926.

- For a permanent acceptance of the identified pole position, the parameter set must be saved (C00002/11 = "1: On / start").
- The next controller inhibit and subsequent controller enable serve to cancel the controller inhibit automatically set by the procedure (e.g. by first executing the device command C00002/16 = "0: Off / ready" and then executing the device command C00002/16 = "1: On / start").

**In the event of an error**

If an error occurs during the procedure or the pulse inhibit gets active (e.g. due to short-time undervoltage), the procedure is terminated with controller inhibit without making a change in C00926.

If the machine was braked or blocked during the procedure, this will be recognised at the end of the measurement and no change is made in C00926.

- If the pole position identification is aborted:
  - the response parameterised in C00643/1 (Lenze setting: "Fault") is activated.
  - the error message "Id5: Pole position identification error" is entered into the logbook.

### 6.5.2 Pole position identification without motion

This function extension is available from version 02.00.00!

**Note!**

Function only possible with:
- Servo control (SC) for synchronous motors
- Sensorless control for synchronous motors (SLPSM)

From version 02.00.00, a pole position identification without motion is also possible in case of servo control and sensorless control.

- For sensorless control of synchronous motors (SLPSM), this function is already activated in the Lenze setting, i.e. with every controller enable the rotor displacement angle is identified and thus jerks in the machine after controller enable can be avoided.
  - In order to achieve the same behaviour as before, set bit 0 to "0" in C02874.
- For servo control (SC), this function can be activated for various events in C02874 via the bits 1 ... 3:
  - Bit 1: Pole position identification after mains connection
  - Bit 2: Pole position identification after controller enable
  - Bit 3: Pole position identification after error acknowledgement

**Note!**

- The "Pole position identification without motion" cannot completely replace the "pole position identification 360°"!
  - The electrical rotor displacement angle can only be electrically identified with an error of up to 10°. This inaccuracy can cause a worse torque accuracy and a worse energy efficiency.
- The identification takes motor-dependent 1 ... 15 ms. The setpoint enabling of the device is reduced by this time.
- In the Lenze setting, the function is preset so that in most of the cases no further settings have to be made.

**Tip!**

In case of servo control (SC), it is generally sufficient to activate this function only once after mains connection of the device (C02874: Bit 1 = "1").

Typical application case: A speed sensor with an unknown pole position is used (e.g. encoder). The pole position identification 360°, however, cannot be used since, e.g. the motor is locked, the application does not permit it or the identification takes too long.

**Short overview of the relevant parameters:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02874 | PLI without motion | 0x1 (bit coded) | |
| C02872 | PLI without motion: Adaptation of time duration | 0 | |
| C02875 | PLI without motion: Adaptation of ident angle | 0 | ° |
| C02870 | PLI without motion: Optimisation factor | - | % |
| C02871 | PLI without motion: Running time | - | ms |
| C02873 | PPI without motion: Identified rotor displacement angle | - | ° |
| Greyed out = display parameter | | | |

## Note!

In case of synchronous motors with a stator time constant < 1 ms, the pole position identification is not executed since the resulting test current pulse could exceed the permissible motor current.

- This, however, only affects very few synchronous motors with very low power (e.g. Lenze motor MDSKS-020-13-300 with rated power of 40 W).
- A non-executed pole position identification can be recognised by the display C02870 = 0 % and C02871 = 0 ms.

- The stator time constants can be calculated based on the following formula:

$$T_s[ms] = \frac{L_{ss}[mH]}{R_s[\Omega]}$$

$T_S$ = stator time constant
$L_{ss}$ = Motor stator leakage inductance (C00085)
$R_s$ = Motor stator resistance (C00084)

### Optimising the pole position identification



**Stop!**

In case of a too high setting in C02872, an impermissible motor current may occur during pole position identification. In this case, the "Fault" error response occurs and the "Id5: Pole position identification error" error message entered in the logbook.

In case of aconsiderably too high setting in C02872:

- The following current monitoring functions can be triggered:
  - oC7: Motor overcurrent
  - oC11: Clamp operation active
  - oC1: Power section - short circuit
- The optimisation degree "0 %" is displayed in C02870.
- The time "0 ms" is displayed in C02871.



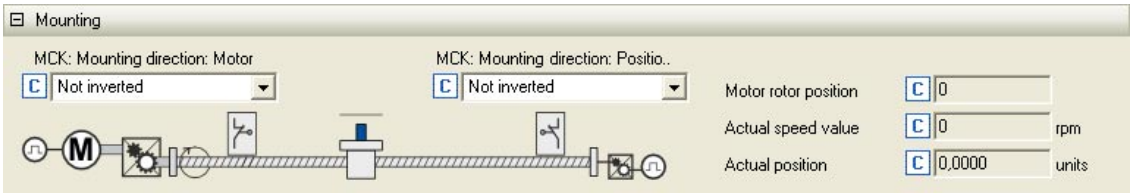
**How to optimise the pole position identification without motion:**

1. For optimisation, enable the controller for various rotor displacement angles.
2. After every controller enable, check the optimisation degree displayed in C02870.

   The pole position identification is set optimally if an optimisation degree in the range 70 ... 130 % is displayed in C02870 after every controller enable.
3. When the optimisation degree is > 130 %:
   Reduce the setting in C02872 step by step an execute controller enable for various rotor displacement angles until an optimisation degree < 130 % is displayed.
4. When the optimisation degree is < 70 %:
   Increase setting in C02872 step by step and execute controller enable for various rotor displacement angles until an optimisation degree > 70 % is displayed.
5. Optionally: C02875 serves to increase or reduce the identified electrical rotor displacement angle. This can prevent e.g. a reversing of the motor due to the accuracy of identification if applications require it.

## 6.6 Mounting

In case of an inverted mounting direction of the speed sensor and/or position encoder, the following parameters have to be adapted accordingly:



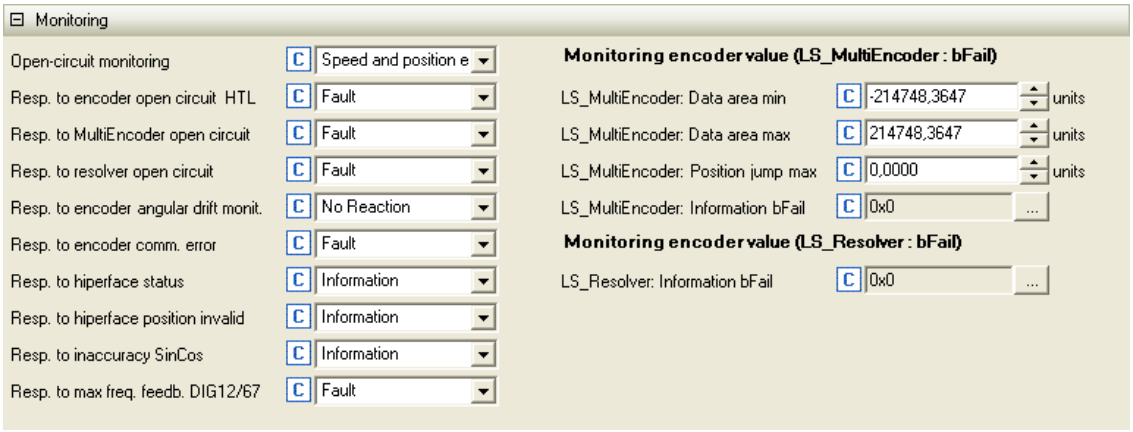

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01206/1 | MCK: Mounting direction: Motor | 0: Not inverted | |
| C01206/2 | MCK: Mounting direction: Position encoder | 0: Not inverted | |
| C00927 | Motor rotor position | - | ° |
| C00051 | MCTRL: Actual speed value | - | rpm |
| C01210/3 | MCK: Actual position | - | units |
| Greyed out = display parameter | | | |

**Related topics:**

- Machine parameters

## 6.7 Monitoring

### Parameterisation dialog (cutout)




| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00498 | Open-circuit monitoring | 0: Speed encoder and position encoder | |
| C00586 | Resp. open circuit HTL encoder | 1: Fault | |
| C00603/1 | Resp. to MultiEncoder open circuit | 1: Fault | |
| C00603/2 | Resp. to resolver open circuit | 1: Fault | |
| C00603/3 | Up to and including version 02.xx.xx:<br>Resp. to encoder comm. error | 1: Fault | |
| | From version 12.00.00:<br>Resp. to encoder pulse deviation | 0: No Reaction | |
| C00603/4 | Up to and including version 02.xx.xx:<br>Resp. to encoder pulse deviation | 1: Fault | |
| | From version 12.00.00:<br>Resp. to encoder comm. error | 1: Fault | |
| C00603/5 | Resp. to Hiperface status | 6: Information | |
| C00603/6 | Resp. to invalid Hiperface position | 6: Information | |
| C00603/7 | Resp. to SinCos inaccuracy | 6: Information | |
| C00607 | Resp. to max. freq. feedb. DIG12/67 | 1: Fault | |
| Monitoring of encoder value (LS_MultiEncoder: bFail) | | | |
| C01112/2 | LS_MultiEncoder: Data area min | -214748.3647 | units |
| C01112/3 | LS_MultiEncoder: Data area max | 214748.3647 | units |
| C01112/4 | LS_MultiEncoder: Max. position jump | 0.0000 | units |
| C00451/1 | LS_MultiEncoder: Information bFail | - | |
| Monitoring of encoder value (LS_Resolver: bFail) | | | |
| C00452/1 | LS_Resolver: Information bFail | - | |
| Greyed out = display parameter | | | |

### 6.7.1 Resolver/multi-encoder open-circuit monitoring

In the Lenze setting, the resolver cable and/or encoder cable are monitored with regard to open circuit as a function of the selected speed and position encoder.

**Danger!**

- For safety reasons, always select "Fault" (Lenze setting) as a response for the (open-circuit) monitoring of the resolver/encoder!
- To avoid the injection of interference when an encoder is being used, only use shielded motor and encoder cables!

**Tip!**

We recommend the Lenze setting "0: Speed sensor and position encoder" for open-circuit monitoring (C00498). Further possible settings make sense if the respective encoder is not used as speed sensor or position encoder but for other tasks in the application. Moreover, a selective switch-off of the monitoring is possible if other settings are made.

| Open-circuit monitoring (C00498) | Function |
|---|---|
| **0: Speed encoder and position encoder** (Lenze setting) | Open-circuit monitoring is active for the speed encoder selected in C00495 and the position encoder selected in C00490.<br>• Depending on whether a speed encoder or position encoder has been selected, the monitoring mode for the resolver and/or encoder is activated.<br>• If no speed encoder and position encoder have been selected, open-circuit monitoring is deactivated. |
| 1: Resolver only | Open-circuit monitoring is only active for the resolver, independent on the selection of the speed encoder and position encoder.<br>• Open-circuit monitoring for the encoder is deactivated. |
| 2: Encoder only | Open-circuit monitoring is only active for the encoder, independent on the selection of the speed encoder and position encoder.<br>• Open-circuit monitoring for the resolver is deactivated. |
| 3: Resolver and encoder | Open-circuit monitoring is active for the resolver and encoder, independent on the selection of the speed encoder and position encoder. |

#### When does the open-circuit monitoring system respond?

| Resolver | Multi encoder |
|---|---|
| • If there is an open circuit in the encoder cable.<br>• If the impedance of the resolver is too high.<br>• In the case of interference injections (EMC interferences). | • If there is an open circuit in the encoder cable. |

#### Response to open circuit

| Resolver | Multi encoder |
|---|---|
| The error response set in C00603/2 is triggered (Lenze setting: "0: No Reaction").<br>Logbook entry: "Sd2: Open circuit - resolver" | The error response set in C00603/1 is triggered (Lenze setting: "0: No Reaction").<br>Logbook entry: "Sd4: Open circuit - MultiEncoder" |

### 6.7.2 Open-circuit monitoring HTL encoder

Open-circuit monitoring is available for the HTL encoder cable.

**Danger!**

For safety reasons, always select "Fault" (Lenze setting) as a response for the (open-circuit) monitoring of the HTL encoder!

**Note!**

If you do not use an HTL encoder, deactivate the monitoring (C00586 = "0: No Reaction").

**When does the open-circuit monitoring system respond?**

The open-circuit monitoring will trigger if

- an open circuit occurs in the encoder cable.
- an extreme overload (e.g. blocked motor shaft) occurs during the start-up phase of the motor.
- highly dynamic reversion of the motor occurs.
- the motor still rotates at high speed in the "SC servo control ASM" mode when the controller enabled.

**Which measured values lead to an actuation of the open-circuit monitoring system?**

The following measured values checked for plausibility lead to an actuation of the open-circuit monitoring system:

1. If the total deviation between actual speed and setpoint speed is higher than f = 40 Hz for a time > 0.1 s .
2. If the actual speed detected is f = 0 Hz or n = 0 rpm and the $I_{max}$ controller or the torque limitation for servo control (SC) is active for t ≥ 0.1 s.
3. If the sign of the injected frequency and the actual speed is not the same, the $I_{max}$ controller is active and this status is active for 0.1 s. Usually this is the case when A/B tracks are reversed.

**Response to open circuit**

- If the open-circuit monitoring is tripped:
  - The error response set in C00586 is activated (Lenze setting: "Fault").
  - The "Sd3: Open circuit - feedback system" is entered into the Logbook.
  - The *bMctrlEncoderComFault* status output of the LS_DeviceMonitor SB is set to TRUE.
- A setting of C00586 = "0: No Reaction" deactivates the monitoring.

### 6.7.3 Encoder angular drift monitoring

This function extension is available from version 12.00.00!

**Note!**

The encoder angular drift monitoring is implemented for encoders without absolute information.

In the Lenze setting (C00603/3 = "0: No response"), the angular drift monitoring of the encoder is not activated!

The encoder angular drift monitoring monitors a possible deviation of the real encoder angle from the angle calculated by counting increments in the encoder evaluation.

- If a deviation higher than 45° (electrical) is recognised when monitoring is activated:
  - The error response set in C00603/3 is triggered.
  - The "Sd8: Encoder angular drift monit." error message is entered into the logbook.
  - The "Reference known" status of the "Homing" basic drive function is reset (if this status was set before)

**Tip!**

A deviation may occur, for instance, by incorrect parameter setting of the encoder increments, by lines in the form of interferences caused by EMC or loss of lines caused by EMC.

#### Functional principle

For an encoder without absolute information, the number of incoming encoder lines between two zero pulses (one revolution) is monitored. This value must equal the encoder increments set in C00420/3 .

**Note!**

After mains switching, monitoring is only active after second incoming zero pulse since the first line difference to be used can only be calculated with the second and first zero pulse.

When the motor (and thus the encoder) is replaced, it is very likely that a angular drift error occurs within the first revolution after acknowledging the encoder error since the monitoring function cannot recognise that the encoder has been replaced.

Due to the type of monitoring, accuracy (distance of Z pulse to the detected position) can only be detected while the machine is running. As a permanently pending error would prevent the error cause to be removed, a detected error will be automatically reset by the inverter after a short time. In order that the motor does not continue to rotate in this case, an engaging error response (e.g. "Fault" or "TroubleQSP") has to be set. If an error is detected again, it is entered again into the logbook.

### 6.7.4 Encoder value monitoring

For detecting invalid encoder values, a value range can be defined via the parameter C01112/2 and C01112/3. When the encoder signal leaves the set value range, the *bFail* status signal of the SB LS_MultiEncoder is set to TRUE and the corresponding information is displayed bit-coded in C00451/1.

- **Example 1: SSI laser distance meter**
  The *bFail* status signal can be used to detect an interruption of the laser beam or a dirty mirror.
- **Example 2: SSI bar code scanner**
  In an application, an SSI bar code scanner is used to detect the workpiece ID and the workpiece position. A setting of sensible limit values serves to use the *bFail* status signal to determine whether the ID or the position of the SSI bar code scanner has been detected.

### 6.7.5 Motor temperature monitoring (KTY)

The motor temperature can also be detected and monitored via the resolver cable and/or the encoder cable.

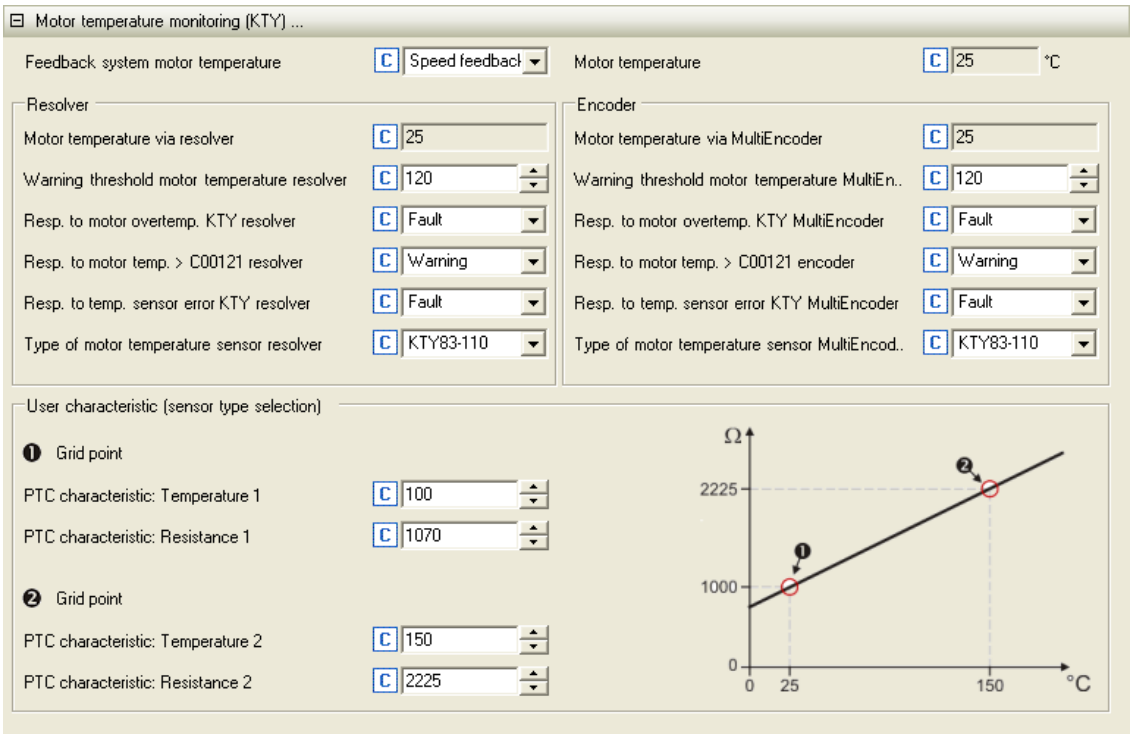
- The feedback system to be used for the motor temperature is selected in C01193.
- The respective motor temperature monitoring includes an early warning stage as well as an open-circuit monitoring for the thermal sensor.

**Note!**

In the Lenze setting of C00583/1...6, the motor temperature monitoring functions are switched on! These monitoring functions are only active with speed encoder selection "3: Multi encoder" or "4: Resolver" in C00495.

From version V02.00.00 with servo control (SC), the temperature compensation within the motor control is activated in the Lenze setting of C02878/1. However, temperature compensation is only active with speed sensor selection "3: Multi encoder" or "4: Resolver" in C00495 as well as error-free KTY temperature detection (display in C00063/1 ≠ 255 °C).

**Parameterisation dialog (cutout)**



| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01193 | Motor temp. feedback system | 0: Speed feedback | |
| C00063/1 | Motor temperature | - | °C |
| Resolver | | | |
| C00063/2 | Motor temperature via resolver | - | °C |
| C00121/1 | Warning threshold motor temperature resolver | 120 | °C |
| C00583/1 | Resp. to motor overtemp. KTY resolver | 1: Fault | |
| C00583/3 | Resp. to motor temp. > C00121 resolver | 5: Warning | |
| C00583/5 | Resp. to temp. sensor error KTY resolver | 1: Fault | |
| C01190/1 | Type of motor temperature sensor resolver | 0: KTY83-110 | °C |
| Multi encoder | | | |
| C00063/2 | Motor temperature via MultiEncoder | - | °C |
| C00121/2 | Warning threshold motor temperature MultiEncoder | 120 | °C |
| C00583/2 | Resp. to motor overtemp. KTY MultiEncoder | 1: Fault | |
| C00583/4 | Resp. to motor temp. > C00121 encoder | 5: Warning | |
| C00583/6 | Resp. to temp. sensor error KTY MultiEncoder | 1: Fault | |
| C01190/2 | Type of motor temperature sensor MultiEncoder | 0: KTY83-110 | °C |
| User characteristics (sensor type selection) | | | |
| C01191/1 | PTC characteristic: Temperature 1 | 100 | °C |
| C01191/2 | PTC characteristic: Temperature 2 | 150 | °C |
| C01192/1 | PTC characteristic: Resistance 1 | 1070 | Ohm |
| C01192/2 | PTC characteristic: Resistance 2 | 2225 | Ohm |
| Greyed out = display parameter | | | |

**Motor temperature monitoring (KTY) via resolver**

The sensor type has to be selected in C01190/1.

- If the winding temperature detected via the motor temperature sensor reaches the limit value set in C00121/1 (Lenze setting: 120 °):
  - An advance warning in the form of the error response set in C00583/3 is issued.
  - The "oH7: Motor temperature - resolver > C121" error message is entered into the logbook.
- If the fixed limit value of 150 °C is reached:
  - The error response set in C00583/1 is triggered.
  - The "oH9: Motor temperature - resolver" error message is entered into the logbook.
- If open circuit is detected for the motor temperature sensor:
  - The error response set in C00583/5 is triggered.
  - The "Sd6: Thermal detector error - resolver" error message is entered into the logbook.

**Motor temperature monitoring (KTY) via encoder**

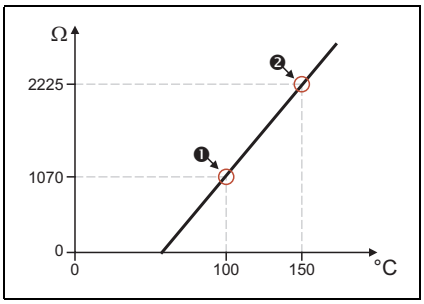
The sensor type has to be selected in C01190/2.

- If the winding temperature detected via the motor temperature sensor reaches the limit value set in C00121/2 (Lenze setting: 120 °):
  - An advance warning in the form of the error response set in C00583/4 is issued.
  - The "oH6: Motor temperature - MultiEncoder > C121" error message is entered into the logbook.
- If the fixed limit value of 150 °C is reached:
  - The error response set in C00583/2 is triggered.
  - The "oH12: Motor overtemperature - MultiEncoder" error message is entered into the logbook.
- If open circuit is detected for the motor temperature sensor:
  - The error response set in C00583/6 is triggered.
  - The "Sd12: Thermal detector error - MultiEncoder" error message is entered into the logbook.

**Specific characteristic for the motor temperature sensor**

If required, you can define and activate a special characteristic for the motor temperature sensor.

- The specific characteristic is defined on the basis of two grid points which must be set in C01191 and C01192. Those two points define a line which is extrapolated to the right and to the left.
- The specific characteristic can be activated by selecting "1: Spec. characteristic" in C01190/1 for the resolver and in C01190/2 for the encoder.
- In the Lenze setting, the specific characteristic is defined as follows:




[6-4] Lenze setting of the special characteristic

- Interpolation point❶
  - C01191/1 = 100 °C
  - C01192/1 = 1070 Ω
- Interpolation point❷
  - C01191/2 = 150 °C
  - C01192/2 = 2225 Ω

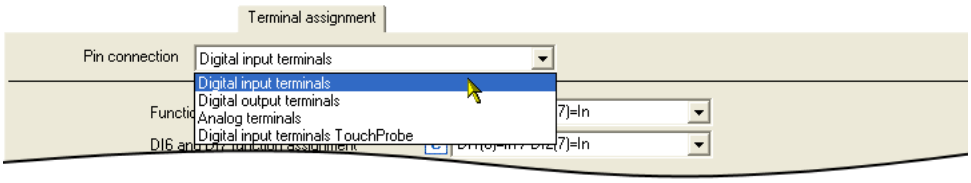
**Note!**

If a motor is selected from the motor catalogue, parameters C01190, C01191, and C01192 are overwritten!

# 7 I/O terminals

This chapter provides information on the function, possible parameter settings, and technical data of the input/output terminals of the inverter.

In the »Engineer«, the input and output terminals are parameterised on the **Terminal assignment** tab. To do this, go to the **Control terminals** list field and select the terminals that you wish to parameterise:




You can find further information in the respective subchapter:

- Digital input terminals (📖 385)
- Digital output terminals (📖 406)
- Analog terminals (📖 409)
- Touch probe detection (📖 419)

**Note!**

The input and output terminals of the inverter have already been functionally assigned in the default setting ("Lenze setting"). The preconfigured assignment depends on the technology application selected in C00005 and the control mode selected in C00007:

- **TA "Actuating drive speed":** Terminal assignment of the control modes (📖 449)
- **TA "Table positioning":** Terminal assignment of the control modes (📖 512)
- **TA "Switch-off positioning":** Terminal assignment of the control modes (📖 538)

Wiring diagram, assignment and electrical data of the input and output terminals can be found in the **hardware manual 8400** in the chapter "technical data". The hardware manual is stored in electronic form on the data carrier supplied with the 8400 inverter.

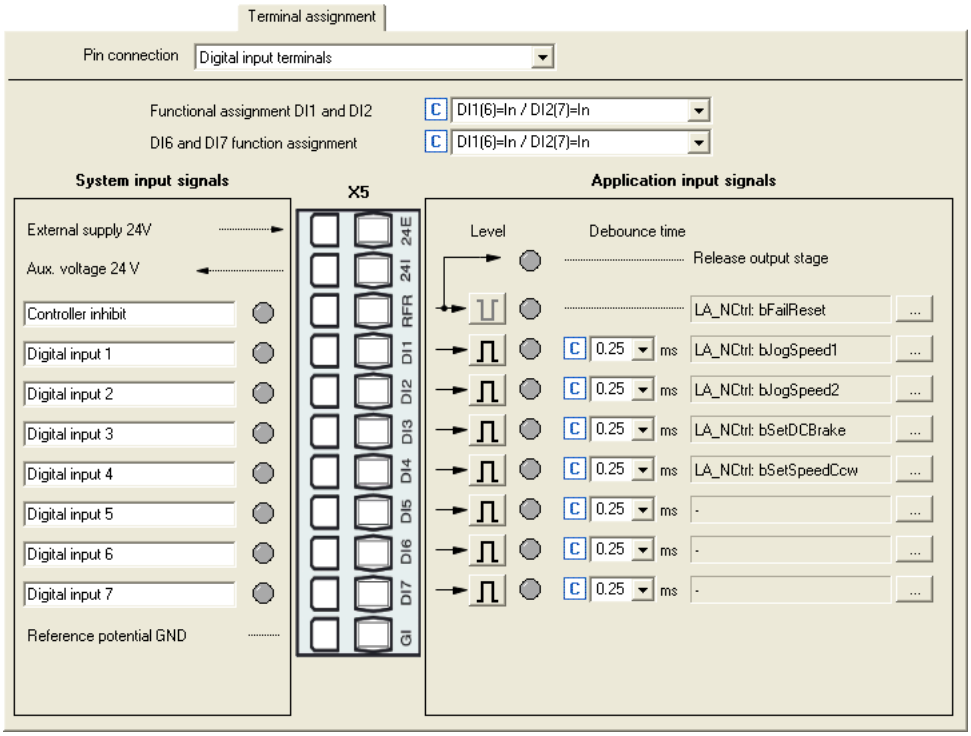
**Tip!**

How you can alter the preconfigured assignment of the input and output terminals is described in the chapter entitled "User-defined terminal assignment". (📖 429)

## 7.1 Digital input terminals

The inverter has 7 parameterisable input terminals (DI1 ... DI7) for detecting digital signals. The RFR control input for enabling the inverter is permanently connected to the device control unit.

**Parameterisation dialog in the »Engineer«:**




| Button | Function |
|---|---|
| ⎍ | Indicates the polarity of the input is HIGH active.<br>The polarity can be changed from HIGH active to LOW active by clicking this button. |
| ⎎ | Indicates that the polarity of the input is LOW active.<br>The polarity can be changed from LOW active to HIGH active by clicking this button. |
| ... | Open the parameterising dialog for assigning application inputs to the digital input.<br>▸ Changing the terminal assignment with the »Engineer« (📖 433) |

**Short overview of parameters for the digital input terminals:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00115/1 | Fct. DI 1/2 200kHz<br>▸ Change function assignment (📖 387) | 0: DI1(6)=In / DI2(7)=In | |
| C00115/2 | Fct. DI 6/7 10kHz<br>▸ Change function assignment (📖 387) | 0: DI1(6)=In / DI2(7)=In | |
| **Digital inputs DI1 ... DI7** | | | |
| C00114 | DigInX: Inversion | Bit coded | |
| C02830/1...7 | DI1...DI7: Debounce time | 1: 0.25 | |
| C00443/1 | DIx: Terminal level | - | |
| C00443/2 | DIx: Output level | - | |
| Greyed out = display parameter | | | |

**Related topics:**

- Touch probe detection (📖 419)
- User-defined terminal assignment (📖 429)

### 7.1.1 Change function assignment

The internal processing function of the digital input terminals DI1/DI2 and DI6/DI7 can be reconfigured in C00115 if necessary. In this way, these input terminals can alternatively be used as frequency or counting inputs in order to implement the following functions:

- Detection of the input frequency
- Detection and processing of two unipolar input frequencies to one bipolar frequency
- Counting of input pulses
- Evaluation of the speed feedback (HTL encoder) for the motor control (speed-controlled operation)

| C00115/1: Function assignment DI1 and DI2<br>C00115/2: Function assignment of DI6 and DI7 | | Function assignment | |
|---|---|---|---|
| | | DI1 / DI6 | DI2 / DI7 |
| **0** | **DI1(6)=In / DI2(7)=In** | **Digital input** | **Digital input** |
| 1 | DI1(6)=FreqIn / DI2(7)=In | Frequency input | Digital input |
| 2 | DI1(6)&DI2(7)=FreqIn (2-track) | Frequency input (2-track) | |
| 3 | DI1(6)=FreqIn / DI2(7)=direction | Frequency input (speed) | Frequency input (direction) |
| 4 | DI1(6)=CountIn / DI2(7)=In | Count input | Digital input |

**Note!**

- In the Lenze setting of C00115, the digital input terminals DI1/DI2 and DI6/DI7 have been configured as "normal" digital inputs.
- The digital input terminals DI3 ... DI5 are generally designed as "standard" digital inputs.
- Very high pulse frequencies can be measured at the DI1/DI2 and DI6/DI7 input terminals if the latter have been configured as frequency or counting inputs in C00115. Scanning is then carried out within less than µs instead of the otherwise usual scanning rate of 1 kHz (1 ms).

You can find detailed information on the respective function assignment in the following subchapters:

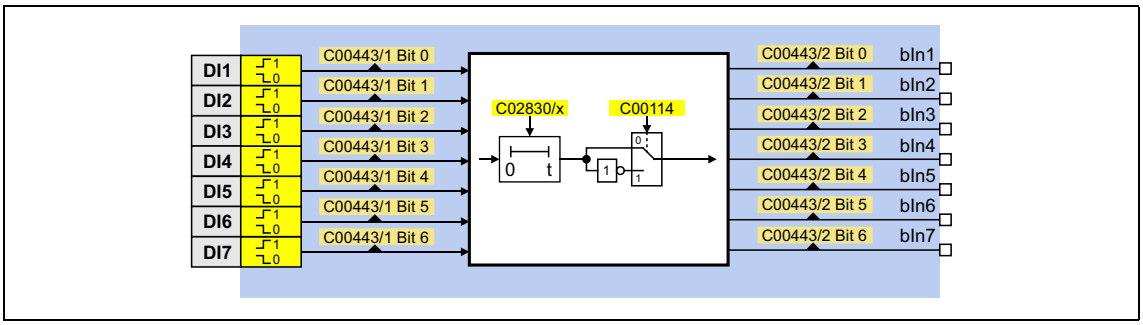
▸ Using DI1(6) and DI2(7) as digital inputs (📖 388)
▸ Using DI1(6) and DI2(7) as frequency inputs (📖 389)
▸ Using DI1(6) as counting input (📖 394)

### 7.1.1.1 Using DI1(6) and DI2(7) as digital inputs

**Function assignment 0: DI1(6)=In / DI2(7)=In**

With this setting in C00115 , the digital input terminals have been configured as "normal" digital inputs.

- For each digital input, the debounce time (C02830/1...7) and the terminal polarity (C00114) can be set individually.
- The current terminal level at the input of the internal processing function is shown in C00443/1 in bit-coded form.
- The output level for the application is shown in C00443/2 in bit-coded form.




**Internal interfaces to the application**

- Relevant outputs at the LS_DigitalInput system block:

| Out<br>DIS code \| data type | Value/meaning |
|---|---|
| bIn1 ... bIn7<br>C00443/2 \| BOOL | Digital input DI1 ... DI7 |

**Related topics:**

- Using DI1(6) and DI2(7) as frequency inputs (📖 389)
- Using DI1(6) as counting input (📖 394)
- Internal interfaces | System block "LS_DigitalInput" (📖 398)

### 7.1.1.2 Using DI1(6) and DI2(7) as frequency inputs

**General information on using the input terminals as frequency inputs**

The frequency inputs serve to detect HTL encoders with any number of increments and single-track and two-track signals. Single-track signals can be evaluated with or without rotation signal.

**Note!**

- Make sure that, when motor control with speed feedback is in use, the maximum input frequency of the respective input terminal is not exceeded.
  - DI1/DI2: $f_{max}$ = 100 kHz (from version 02.00.00: $f_{max}$ = 200 kHz)
  - DI6/DI7: $f_{max}$ = 10 kHz
- If the encoder signal is used as an actual speed value:
  Number of encoder pulses / revolution ≤ **8192!**

Example of DI6/DI7 (in accordance with the preceding note):

- Encoder increment: 512 pulses / motor revolution
- Reference speed (C00011): 1500 rpm
- Speed setpoint: 100 %

$$\text{Input frequency} = \frac{1500\ \text{rpm}}{60\ \text{s}} \times 512\ \text{pulses} = 12800\ \text{pulses/s} = 12.8\ \text{kHz}$$

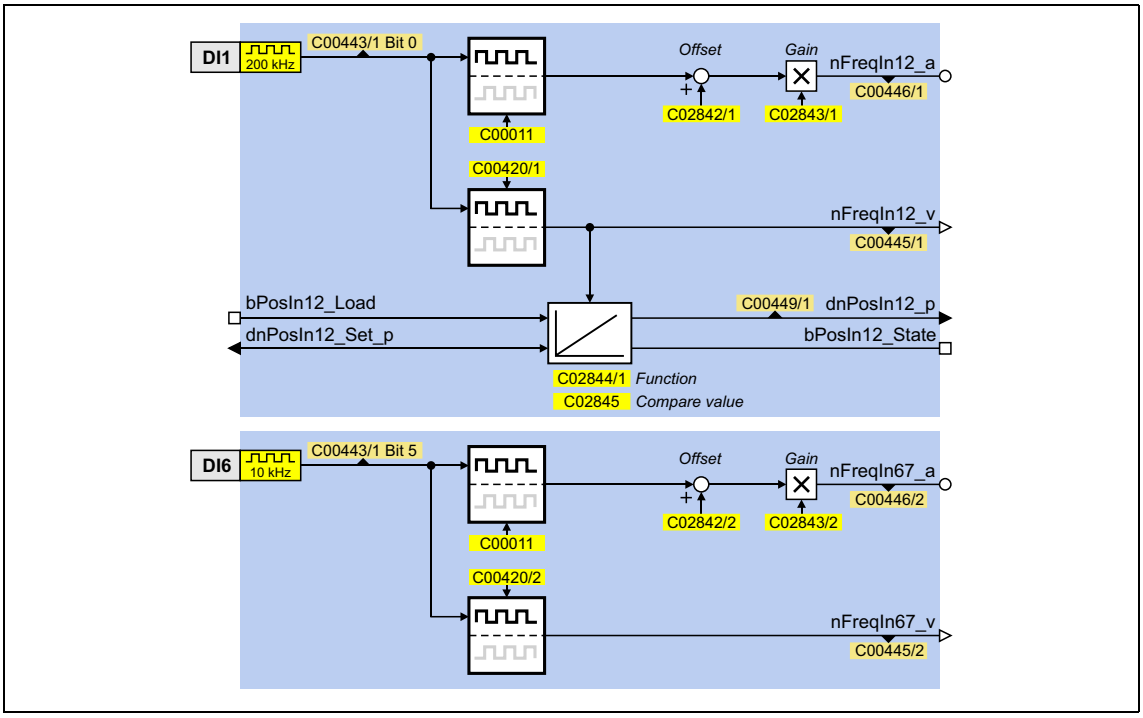
- Result: The speed or the number of increments is too high!

**Tip!**

The LS_DigitalInput system block can also provide the encoder position. Detailed information on this topic is provided in chapter "Output of the encoder position of the DI1/DI2 frequency input". (📖 401)

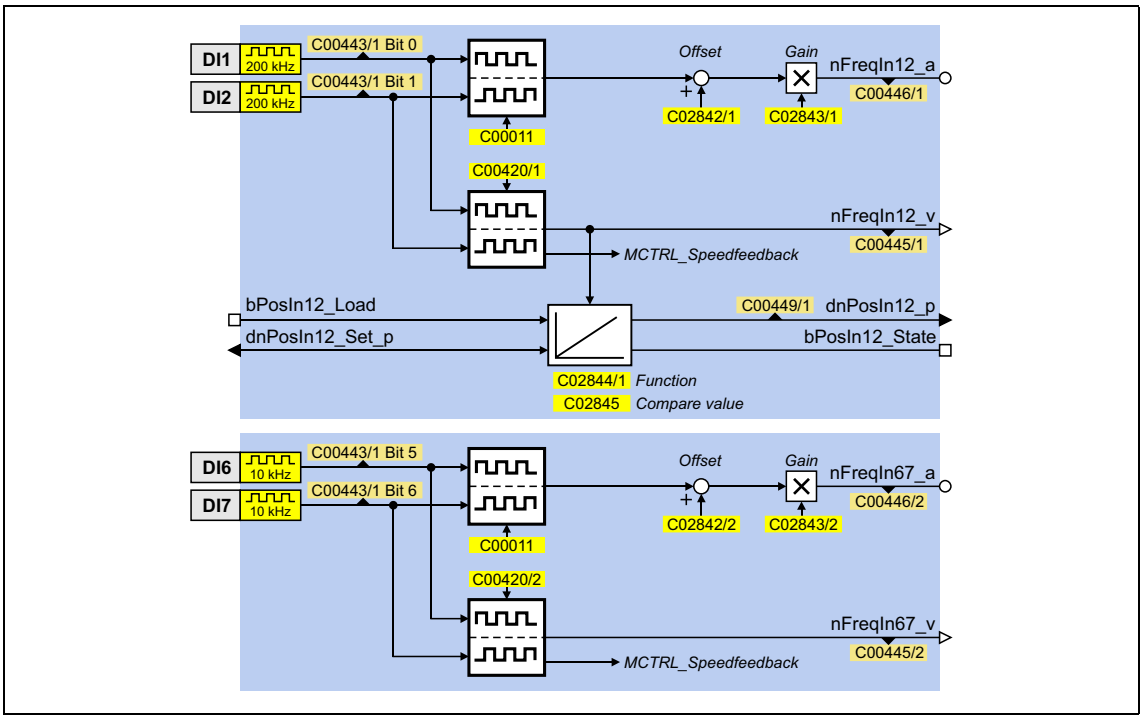
**Function assignment 1: DI1(6)=FreqIn / DI2(7)=In**

This setting in C00115 configures the input terminal DI1 or DI6 as frequency input. The input terminal DI2 or DI7 remains configured as "normal" digital input..



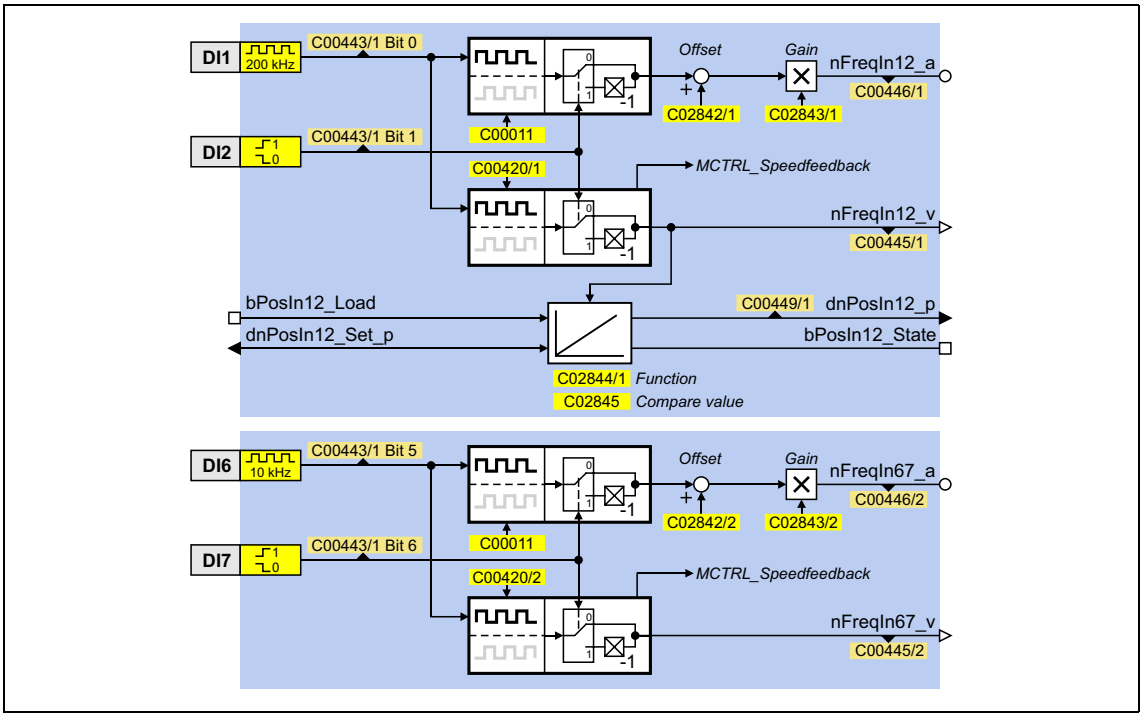

**Function assignment 2: DI1(6)&DI2(7)=FreqIn (2-track)**

This setting in C00115 can be used to connect a two-track encoder to the DI1/DI2 or DI6/DI7 terminals.



**Function assignment 3: DI1(6)=FreqIn / DI2(7)=Direction**

This setting in C00115 can be used to connect a single-track encoder to terminals DI1/DI2 or DI6/DI7. For this purpose, the rotation speed is evaluated via terminal DI1(6) and the direction of rotation of the encoder (LOW level ≡ CW direction of rotation) is evaluated via the DI(7) terminal.



**Short overview of the parameters for the frequency inputs:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00011 | Appl.: Reference speed | 1500 | rpm |
| **Frequency input DI1/DI2** | | | |
| C00115/1 | Fct. DI 1/2 200kHz | 0: DI1(6)=In / DI2(7)=In | |
| C00420/1 | Encoder increments at FreqIn12 | 128 | Incr./rev. |
| C02842/1 | FreqIn12: Offset | 0.00 | % |
| C02843/1 | FreqIn12: Gain | 100.00 | % |
| C02844/1 | PosIn12: Function | Loading with level | |
| C02845 | PosIn12: Comparison value | 0 | |
| C00443/1 | DIx: Terminal level | - | |
| C00445/1 | FreqIn12_nOut_v | - | Incr/ms |
| C00446/1 | FreqIn12_nOut_a | - | % |
| C00449/1 | FreqIn12_dnOut_p | - | Incr |
| **Frequency input DI6/DI7** | | | |
| C00115/2 | Fct. DI 6/7 10kHz | 0: DI1(6)=In / DI2(7)=In | |
| C00420/2 | Encoder increments at FreqIn67 | 128 | Incr./rev. |
| C02842/2 | FreqIn67: Offset | 0.00 | % |
| C02843/2 | FreqIn67: Gain | 100.00 | % |
| C00443/1 | DIx: Terminal level | - | |
| C00445/2 | FreqIn67_nOut_v | - | Incr/ms |
| C00446/2 | FreqIn67_nOut_a | - | % |
| Greyed out = display parameter | | | |

### Internal interfaces to the application

- Relevant inputs at the LS_DigitalInput system block:

| Input<br>Data type | Information/possible settings | |
|---|---|---|
| **Frequency input DI1/DI2** | | |
| bPosIn12_Load<br>BOOL | Load angle integrator with starting value and reset status signal | |
| | TRUE | Angle integrator is loaded with the value at *dnPosIn12_Set_p* and *bPosIn12_State* is reset to FALSE. |
| dnPosIn12_Set_p<br>DINT | Starting value for angle integrator | |

- Relevant outputs at the LS_DigitalInput system block:

| Out<br>Data type | Value/meaning | |
|---|---|---|
| **Frequency input DI1/DI2** | | |
| nFreqIn12_a<br>C00446/1 \| INT | Output frequency as scaled analog signal in [%] | |
| nFreqIn12_v<br>C00445/1 \| INT | Output frequency as speed signal in [inc/ms] | |
| dnPosIn12_p<br>DINT | Angle output signal<br>• 65536 [incr.] ≡ 1 encoder revolution<br>• Overflow is possible (display via *bPosIn12_State*) | |
| bPosIn12_State<br>BOOL | Status signal "Overflow occurred/distance processed"<br>• Status signal can be reset via *bPosIn12_Load*. | |
| | TRUE | Overflow has occurred or distance is processed. |
| **Frequency input DI6/DI7** | | |
| nFreqIn67_a<br>C00446/2 \| INT | Output frequency as scaled analog signal in [%] | |
| nFreqIn67_v<br>C00445/2 \| INT | Output frequency as speed signal in [inc/ms] | |

### Related topics:

- Output of the encoder position of the DI1/DI2 frequency input (📖 401)
- Using DI1(6) and DI2(7) as digital inputs (📖 388)
- Using DI1(6) as counting input (📖 394)
- Internal interfaces | System block "LS_DigitalInput" (📖 398)

### 7.1.1.3 Using DI1(6) as counting input

#### General information on use as a counting input

The counting input is used for counting fast edges. A 32-bit counter counts from a parameterisable starting value up to a parameterisable comparison value and then outputs a corresponding status signal.

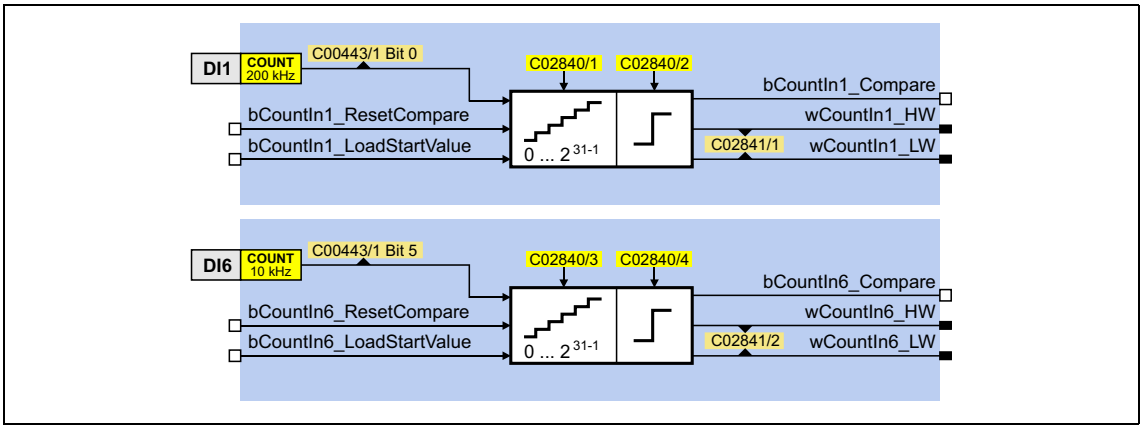
- Possible counting range: 0 ... $2^{31}$ - 1 (0 ... 2147483647)

**Note!**

- The starting value must be have been set so that it is smaller than the comparison value. Otherwise, the counter will be kept at the starting value because the condition "Count value ≥ Comparison value" has been satisfied.
- Note the maximum input frequency of the respective input terminal:
  - DI1: $f_{max}$ = 200 kHz
  - DI6: $f_{max}$ = 10 kHz

#### Function assignment 4: DI1(6)=CountIn / DI2(7)=In

This setting in C00115 configures the input terminal DI1 or DI6 as counting input. The input terminal DI2 or DI7 remains configured as "normal" digital input..



**Short overview of parameters for the counting inputs:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| **Counting input DI1** | | | |
| C00115/1 | Fct. DI 1/2 200kHz | 0: DI1(6)=In / DI2(7)=In | |
| C00621/3 | LS_DigitalInput: bCountIn1_Reset | 0: Not connected | |
| C00621/4 | LS_DigitalInput: bCountIn1_LoadStartValue | 0: Not connected | |
| C02840/1 | CountIn1: Starting value | 0 | incr |
| C02840/2 | CountIn1: Comparison value | 65535 | incr |
| C02841/1 | CountIn1: Counter content | - | incr |
| C00443/1 | DIx: Terminal level | - | |
| **Counting input DI6** | | | |
| C00115/2 | Fct. DI 6/7 10kHz | 0: DI1(6)=In / DI2(7)=In | |
| C00621/97 | LS_DigitalInput: bCountIn6_Reset | 0: Not connected | |
| C00621/98 | LS_DigitalInput: bCountIn6_LoadStartValue | 0: Not connected | |
| C02840/3 | CountIn6: Starting value | 0 | incr |
| C02840/4 | CountIn6: Comparison value | 65535 | incr |
| C02841/2 | CountIn6: Counter content | - | incr |
| C00443/1 | DIx: Terminal level | - | |

Greyed out = display parameter

### Internal interfaces to the application

- Relevant inputs at the LS_DigitalInput system block:

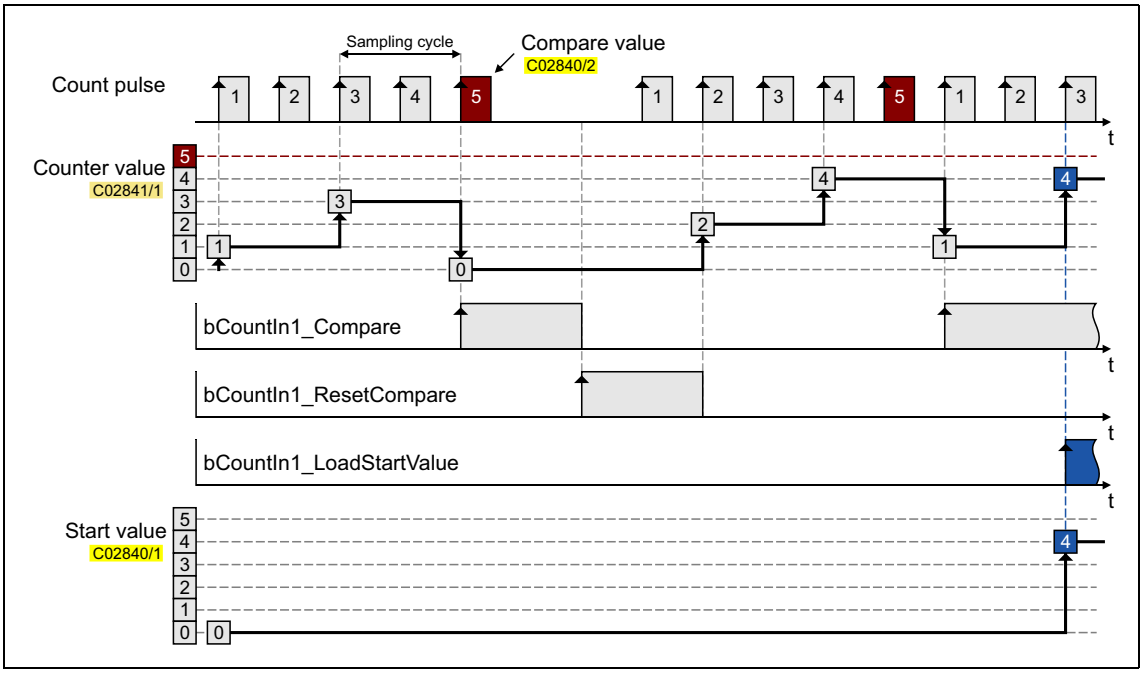
| Designator<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| **Counting input DI1** | | |
| bCountIn1_ResetCompare<br>BOOL | Reset status signal "Comparison value reached" | |
| | FALSE↗TRUE | The output *bCountIn1_Compare* is reset to FALSE. |
| bCountIn1_LoadStartValue<br>BOOL | Load starting value into counter | |
| | FALSE↗TRUE | The starting value set in C02840/1 is accepted as the current count value. |
| **Counting input DI6** | | |
| bCountIn6_ResetCompare<br>BOOL | Reset status signal "Comparison value reached" | |
| | FALSE↗TRUE | The output *bCountIn6_Compare* is reset to FALSE. |
| bCountIn6_LoadStartValue<br>BOOL | Load starting value into counter | |
| | FALSE↗TRUE | The starting value set in C02840/3 is accepted as the current count value. |

- Relevant outputs at the LS_DigitalInput system block:

| Designator<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| **Counting input DI1** | | |
| bCountIn1_Compare<br>BOOL | Status signal "Comparison value reached" | |
| | FALSE | Current count value < comparison value (C02840/2) |
| | TRUE | Current count value ≥ comparison value (C02840/2) |
| wCountIn1_HW<br>wCountIn1_LW<br>C02841/1 \| WORD | Current count value<br>• Output as High and Low word (without sign)<br>• Possible counting range: 0 ... $2^{31}$ - 1 | |
| **Counting input DI6** | | |
| bCountIn6_Compare<br>BOOL | Status signal "Comparison value reached" | |
| | FALSE | Current count value < comparison value (C02840/4) |
| | TRUE | Current count value ≥ comparison value (C02840/4) |
| wCountIn6_HW<br>wCountIn6_LW<br>C02841/2 \| WORD | Current count value<br>• Output as High and Low word (without sign)<br>• Possible counting range: 0 ... $2^{31}$ - 1 | |

### Counting behaviour

The following temporal characteristic shows the counting process depending on the signals of the interfaces described before:




[7-1] Transient characteristic of a quick counter block, sampling cycle = 1 ms

- The counter starts with the parameterised starting value.
- If the comparison value is reached or exceeded:
  - The counter jumps back to its starting value.
  - The output *bCount1(6)_Compare* is set to TRUE.
- If there is a FALSE-TRUE edge is at the input *bCountIn1(6)_ResetCompare*, the output *bCountIn1(6)_Compare* can be reset to FALSE.
- If there is a FALSE-TRUE edge at the input *bCountIn1(6)_LoadStartValue*, the current counter content can be reset to the parameterised starting value.

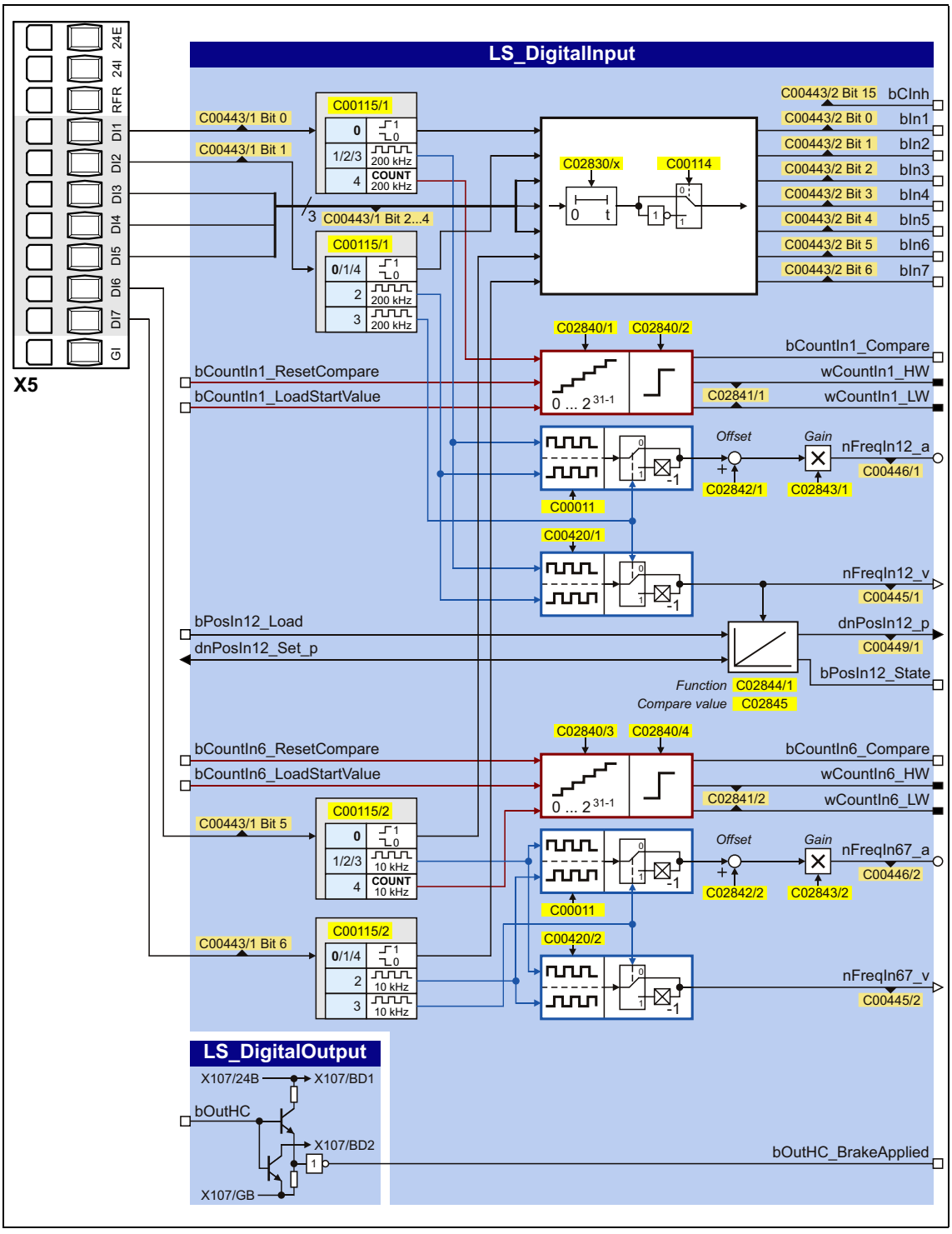
**Related topics:**


- Using DI1(6) and DI2(7) as digital inputs (📖 388)
- Using DI1(6) and DI2(7) as frequency inputs (📖 389)
- Internal interfaces | System block "LS_DigitalInput" (📖 398)

### 7.1.2 Internal interfaces | System block "LS_DigitalInput"

The system block **LS_DigitalInput** maps the digital input terminals in the FB editor.

- The internal processing function of the digital DI1/2 and DI6/7 input terminals can be reconfigured in C00115 if necessary. These input terminals can then be alternatively used as frequency inputs or counting inputs.
- The DI3 ... DI5 input terminals are basically designed as "normal" digital inputs.



### Inputs

| Designator<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| **Counting input DI1** | ▸ Using DI1(6) as counting input | |
| bCountIn1_ResetCompare<br>BOOL | Reset status signal "Comparison value reached" | |
| | FALSE↗TRUE | The output *bCountIn1_Compare* is reset to FALSE. |
| bCountIn1_LoadStartValue<br>BOOL | Load starting value into counter | |
| | FALSE↗TRUE | The starting value set in C02840/1 is accepted as the current count value. |
| **Frequency input DI1/DI2** | ▸ Output of the encoder position of the DI1/DI2 frequency input | |
| bPosIn12_Load<br>BOOL | Load angle integrator with starting value and reset status signal | |
| | TRUE | Angle integrator is loaded with the value at *dnPosIn12_Set_p* and *bPosIn12_State* is reset to FALSE. |
| dnPosIn12_Set_p<br>DINT | Starting value for angle integrator | |
| **Counting input DI6** | ▸ Using DI1(6) as counting input | |
| bCountIn6_ResetCompare<br>BOOL | Reset status signal "Comparison value reached" | |
| | FALSE↗TRUE | The output *bCountIn6_Compare* is reset to FALSE. |
| bCountIn6_LoadStartValue<br>BOOL | Load starting value into counter | |
| | FALSE↗TRUE | The starting value set in C02840/3 is accepted as the current count value. |

### Outputs

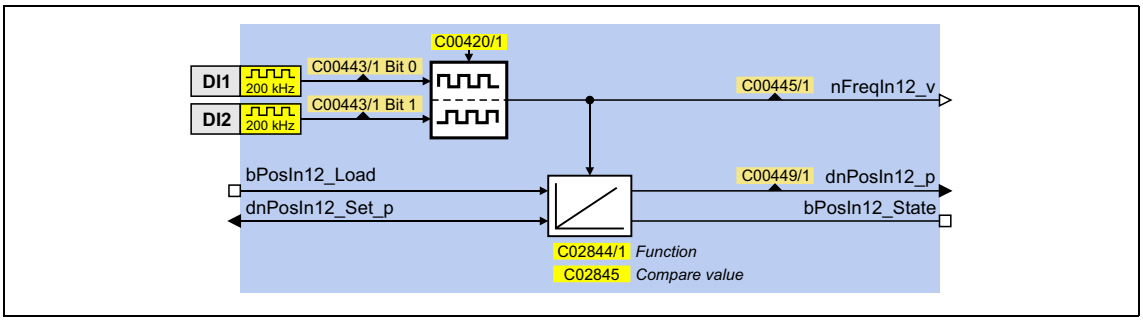
| Designator<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| bCInh<br>C00443/2 \| BOOL | RFR digital input (controller enable) | |
| **Digital inputs DI1 ... DI7** | ▸ Using DI1(6) and DI2(7) as digital inputs | |
| bIn1 ... bIn7<br>C00443/2 \| BOOL | Digital input DI1 ... DI7 | |
| **Counting input DI1** | ▸ Using DI1(6) as counting input | |
| bCountIn1_Compare<br>BOOL | Status signal "Comparison value reached" | |
| | FALSE | Current count value < comparison value (C02840/2) |
| | TRUE | Current count value ≥ comparison value (C02840/2) |
| wCountIn1_HW<br>wCountIn1_LW<br>C02841/1 \| WORD | Current count value<br>• Output as High and Low word (without sign)<br>• Possible counting range: 0 ... $2^{31}$ - 1 | |
| **Frequency input DI1/DI2** | ▸ Using DI1(6) and DI2(7) as frequency inputs | |
| nFreqIn12_a<br>C00446/1 \| INT | Output frequency as scaled analog signal in [%] | |
| nFreqIn12_v<br>C00445/1 \| INT | Output frequency as speed signal in [inc/ms] | |
| dnPosIn12_p<br>DINT | Angle output signal<br>• 65536 [incr.] ≡ 1 encoder revolution<br>• Overflow is possible (display via *bPosIn12_State*) | |
| bPosIn12_State<br>BOOL | Status signal "Overflow occurred/distance processed"<br>• Status signal can be reset via *bPosIn12_Load*. | |
| | TRUE | Overflow has occurred or distance is processed. |

| Designator<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| **Counting input DI6** | ▸ Using DI1(6) as counting input | |
| bCountIn1_Compare<br>BOOL | Status signal "Comparison value reached" | |
| | FALSE | Current count value < comparison value (C02840/4) |
| | TRUE | Current count value ≥ comparison value (C02840/4) |
| wCountIn6_HW<br>wCountIn6_LW<br>C02841/2 \| WORD | Current count value<br>• Output as High and Low word (without sign)<br>• Possible counting range: 0 ... $2^{31}$ - 1 | |
| **Frequency input DI6/DI7** | ▸ Using DI1(6) and DI2(7) as frequency inputs | |
| nFreqIn67_a<br>C00446/2 \| INT | Output frequency as scaled analog signal in [%] | |
| nFreqIn67_v<br>C00445/2 \| INT | Output frequency as speed signal in [inc/ms] | |
| **High current output** | ▸ Switching status of the motor holding brake at the high current output | |
| bOutHC_BrakeApplied<br>BOOL<br>(from version 15.00.00) | Switching status of a motor holding brake connected to the high current output<br>• The inverted signal logic of this status signal corresponds to the evaluation of a real brake contact which provides the FALSE state in case of a released holding brake and the TRUE state in case of a closed holding brake. | |
| | FALSE | High current output is controlled if an external 24-V supply is available (holding brake is released). |
| | TRUE | High current output is not controlled (holding brake is closed) or external 24-V supply is not available. |

### 7.1.2.1 Output of the encoder position of the DI1/DI2 frequency input

The LS_DigitalInput system block includes an integrator for providing the encoder position.

- The integrator can take max. ±32000 encoder revolutions.
- The starting position can be loaded via inputs.
- The internal function can be set via parameters.
- In addition to the encoder position, the "Overflow occurred/distance processed" status signal is provided.




#### Inputs

| Designator<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| bPosIn12_Load<br>BOOL | Load angle integrator with starting value and reset status signal | |
| | TRUE | Angle integrator is loaded with the value at *dnPosIn12_Set_p* and *bPosIn12_State* is reset to FALSE. |
| dnPosIn12_Set_p<br>DINT | Starting value for angle integrator | |

#### Outputs

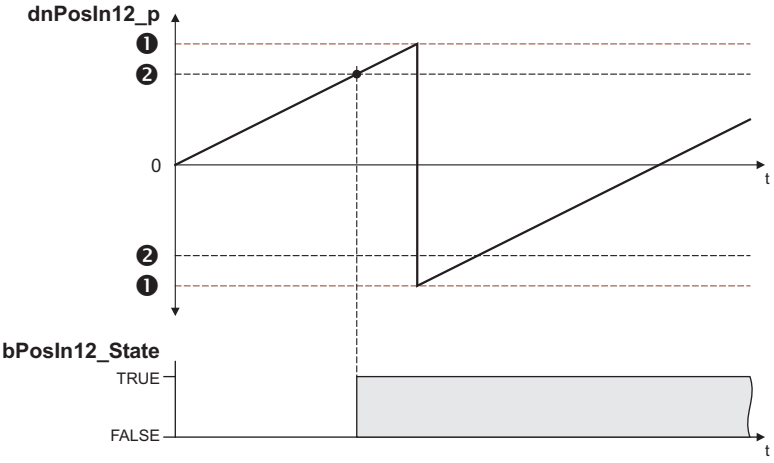
| Designator<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| dnPosIn12_p<br>DINT | Angle output signal<br>• 65536 [incr.] ≡ 1 encoder revolution<br>• Overflow is possible (display via *bPosIn12_State*) | |
| bPosIn12_State<br>BOOL | Status signal "Overflow occurred/distance processed"<br>• Status signal can be reset via *bPosIn12_Load*. | |
| | TRUE | Overflow has occurred or distance is processed. |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C02844/1 | | | | Function |
| | 0 | Loading with level | | Load integrator with TRUE level at the *bPosIn12_Load* input (Lenze setting). |
| | 1 | Loading with edge | | Load integrator with FALSE/TRUE edge at the *bPosIn12_Load* input. |
| | 2 | Loading with level + reset | | Load integrator when reaching the comparison value or with TRUE level at the *bPosIn12_Load* input. |
| C02845 | 0 | | 2000000000 | Comparison value<br>• Is valid for both the positive and the negative value range.<br>• Lenze setting: 0 |

### Function at constant input value

Selection: C02844/1 = "0: Loading with level" or "1: Loading with edge"




❶ Overflow at ±2147483648 incr. (±32767 encoder revolutions)
❷ C02845: Comparison value (applies to both positive and negative value range)

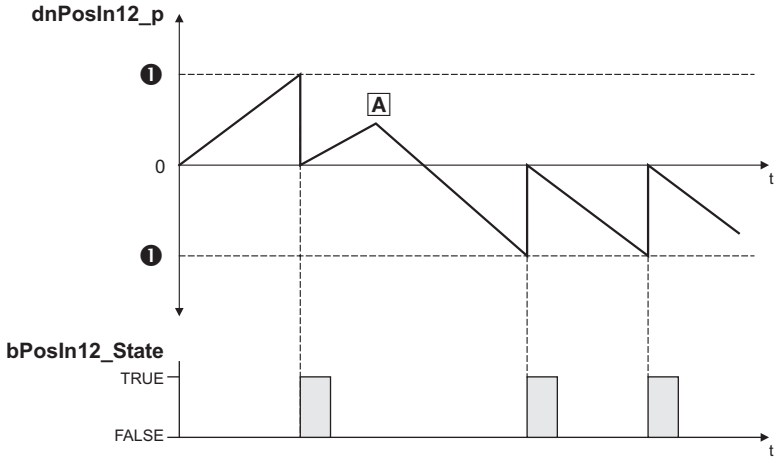
[7-2] Switching performance if the overflow is in the positive direction

- If "0: Loading with level" is selected in C02844/1, the *bPosIn12_Load* input is status-controlled: In case of a TRUE signal, the integrator is loaded with the value at *dnPosIn12_Set_p* and the *bPosIn12_State* output is set to FALSE.
- If "1: Loading with edge" is selected in C02844/1, the *bPosIn12_Load* input is edge-controlled: In case of a FALSE/TRUE edge, the integrator is loaded with the value at *dnPosIn12_Set_p* and then immediately continues to integrate, the *bPosIn12_State* output is set to FALSE.
- A positive *nFreqIn12_v* signal is incremented (the counter content is increased with every cycle).
- A negative *nFreqIn12_v* signal is decremented (the counter content is reduced with every cycle).

- *dnPosIn12_p* provides the counter content of the bipolar integrator.
  - If the counter content exceeds a value of +32767 encoder revolutions (corresponds to +2147483647 incr.), an overflow occurs and the counting process continues at a value of -32768 encoder revolutions.
  - If the counter content falls below a value of -32768 encoder revolutions (corresponds to -2147483648 incr.), an overflow occurs and the counting process starts at a value of +32767 encoder revolutions.
- *bPosIn12_State* is set to TRUE if the comparison value set in C02845 has been reached.

**Function at input value with sign reversal**

Selection: C02844/1 = "2: Loading with level + reset"




Ⓐ Sign reversal of the value at *nFreqIn12_v*
❶ C02845: Comparison value (applies to both positive and negative value range)

[7-3] Switching performance if the input signal changes signs

- If "2: Loading with level + reset" is selected in C02844/1, the *bPosIn12_Load* input is status-controlled: In case of a TRUE signal, the integrator is loaded with the value at *dnPosIn12_Set_p* and the *bPosIn12_State* output is set to FALSE.
- A positive *nFreqIn12_v* signal is incremented (the counter content is increased with every cycle).
- A negative *nFreqIn12_v* signal is decremented (the counter content is reduced with every cycle).
- *dnPosIn12_p* provides the counter content of the bipolar integrator.
  - If the positive counter content is higher than the comparison value set in C02845, the comparison value will be subtracted from the counter content, and *bPosIn12_State* will be set to TRUE for one task cycle.
  - If the negative counter content is lower than the comparison value set in C02845, the comparison value will be added to the counter content, and *bPosIn12_State* will be set to TRUE for one task cycle.

**Calculation of the output signal**

The output value at *dnPosIn12_p* is calculated as per the formula below:

$$\text{dnPosIn12_p [incr.]} = \text{nFreqIn12_v [rpm]} \cdot \text{t [s]} \cdot 65535\ \text{[incr./rev.]}$$

t = integration time
16384 ≡ 15000 rpm
1 ≡ 1 incr.

**Example**

You want to determine the counter content of the integrator at a certain speed at the input and a certain integration time **t**.

Given values:

- *nFreqIn12_v* = 1000 rpm ≈ integer value 1092
- Integration time t = 10 s
- Starting value of the integrator = 0

Solution:

- Conversion of the *nFreqIn12_v* input signal:

$$1000\ \text{rpm} = \frac{1000\ \text{rev.}}{60\ \text{s}}$$

- Calculation of the output value:

$$\text{dnPosIn12_p} = \frac{1000\ \text{rev.}}{60\ \text{s}} \cdot 10\ \text{s} \cdot \frac{65535\ \text{incr.}}{\text{Rev.}} = 10922666\ \text{incr.}$$

### 7.1.2.2 Switching status of the motor holding brake at the high current output

This function extension is available from version 15.00.00!

For a simple monitoring of the switching status of a motor holding brake connected to the high current output, the *bOutHC_BrakeApplied* status signal can be used.

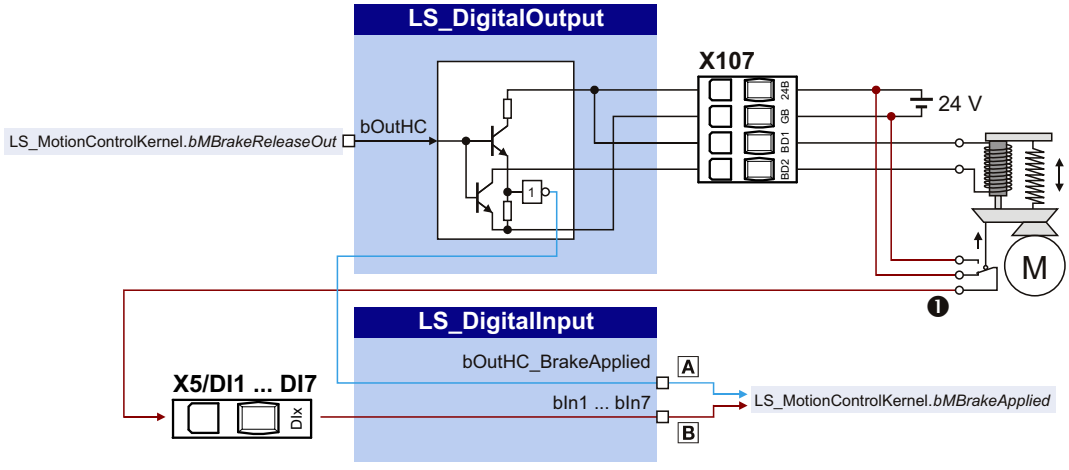
| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOutHC_BrakeApplied<br>BOOL | Switching status of a motor holding brake connected to the high current output<br>• The inverted signal logic of this status signal corresponds to the evaluation of a real brake contact which provides the FALSE state in case of a released holding brake and the TRUE state in case of a closed holding brake. | |
| | FALSE | High current output is controlled if an external 24-V supply is available (holding brake is released). |
| | TRUE | High current output is not controlled (holding brake is closed) or external 24-V supply is not available.<br>In the closed state, an open-circuit monitoring is available so that a brake status error is created in case of an open circuit. |

**Required steps for a simple monitoring of the switching state:**

1. Link the *bOutHC_BrakeApplied* status signal with the *bMBrakeApplied* input of the SB LS_MotionControlKernel.
2. Activate feedback monitoring of the holding brake control: Set bit 5 in C02582 to "1".

**Connection principle**

In the following, a connection principle for controlling a motor holding brake via the high current output is described with the two options for feedback monitoring:




Options for feedback monitoring:
A Use of the internal status signal *bOutHC_BrakeApplied*
B Evaluation of the switching contact of the holding via a free digital input
❶ Wire-break detection with a closed holding brake

[7-4] Connection principle motor holding brake

## 7.2 Digital output terminals

The inverter has

- three parameterisable output terminals (DO1 ... DO3) for outputting digital signals
- a relay output (terminal strip X101),
- a (high-current) output for controlling a brake (terminal strip X107).

### Note!

Initialisation behaviour:

- After mains switching up to the start of the application, the digital outputs remain set to FALSE.

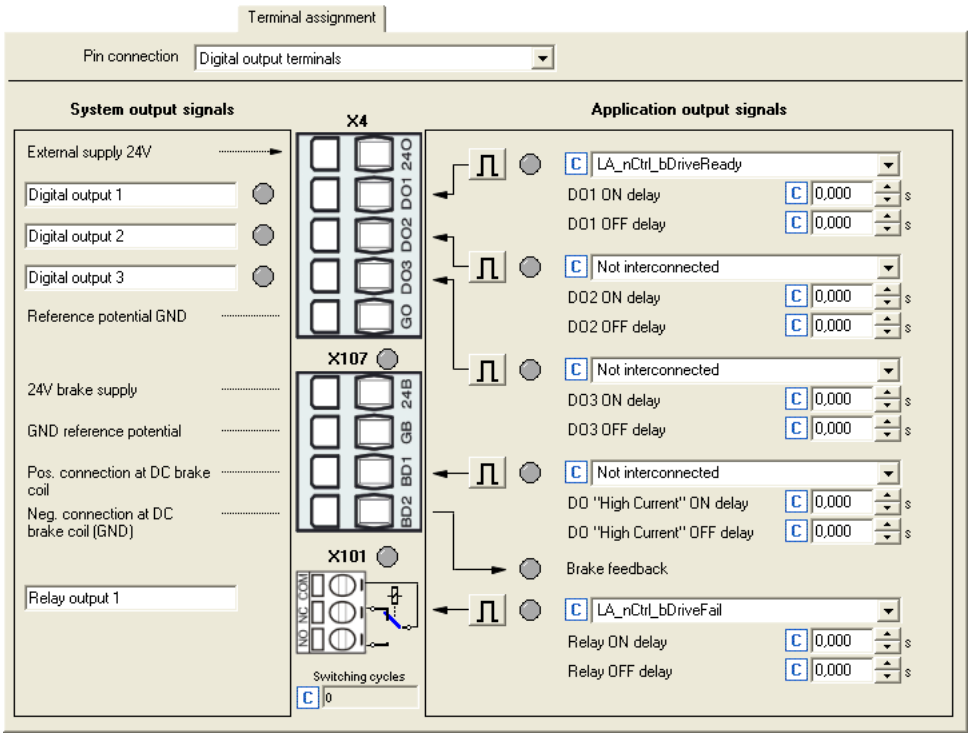
Exception handling:

- In case of a critical exception in the application (e.g. reset), the digital outputs are set to FALSE considering the terminal polarity parameterised in C00118.

Switching cycle diagnostics of the relay:

- A reference for evaluating the wear limit can be obtained via the number of switching cycles of the relay displayed in C00177/2.

**Parameterisation dialog in the »Engineer«:**




| Button | Function |
|---|---|
| ⎍ | Indicates that the polarity of the output is HIGH active.<br>The polarity can be changed from HIGH active to LOW active by clicking this button. |
| ⎎ | Indicates that the polarity of the output is LOW active.<br>The polarity can be changed from LOW active to HIGH active by clicking this button. |

**Short overview of parameters for the digital output terminals:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| **Digital outputs DO1 ... DO3** | | | |
| C00118 | DigOutX: Inversion | Bit coded | |
| C00423/3 | DO1 ON delay | 0.000 | s |
| C00423/4 | DO1 OFF delay | 0.000 | s |
| C00423/5 | DO2 ON delay | 0.000 | s |
| C00423/6 | DO2 OFF delay | 0.000 | s |
| C00423/7 | DO3 ON delay | 0.000 | s |
| C00423/8 | DO3 OFF delay | 0.000 | s |
| C00444/1 | DOx: Input level | - | |
| C00444/2 | DOx: Terminal level | - | |
| **High current output** | | | |
| C00423/9 | DO "High Current" ON delay | 0.000 | s |
| C00423/10 | DO "High Current" OFF delay | 0.000 | s |
| C00117 | Status of brake output BD | - | |
| **Relay output** | | | |
| C00423/1 | Relay ON delay | 0.000 | s |
| C00423/2 | Relay OFF delay | 0.000 | s |
| **Digital outputs - terminal configuration** | | | |
| C00621/1 | LS_DigitalOutput:bRelay | 1001: LA_nCtrl_bDriveFail | |
| C00621/2 | LS_DigitalOutput:bOut1 | 1000: LA_nCtrl_bDriveReady | |
| C00621/99 | LS_DigitalOutput: bOut2 | 0: Not connected | |
| C00621/100 | LS_DigitalOutput: bOut3 | 0: Not connected | |
| C00621/101 | LS_DigitalOutput: bOut HighCurrent | 0: Not connected | |
| Greyed out = display parameter | | | |

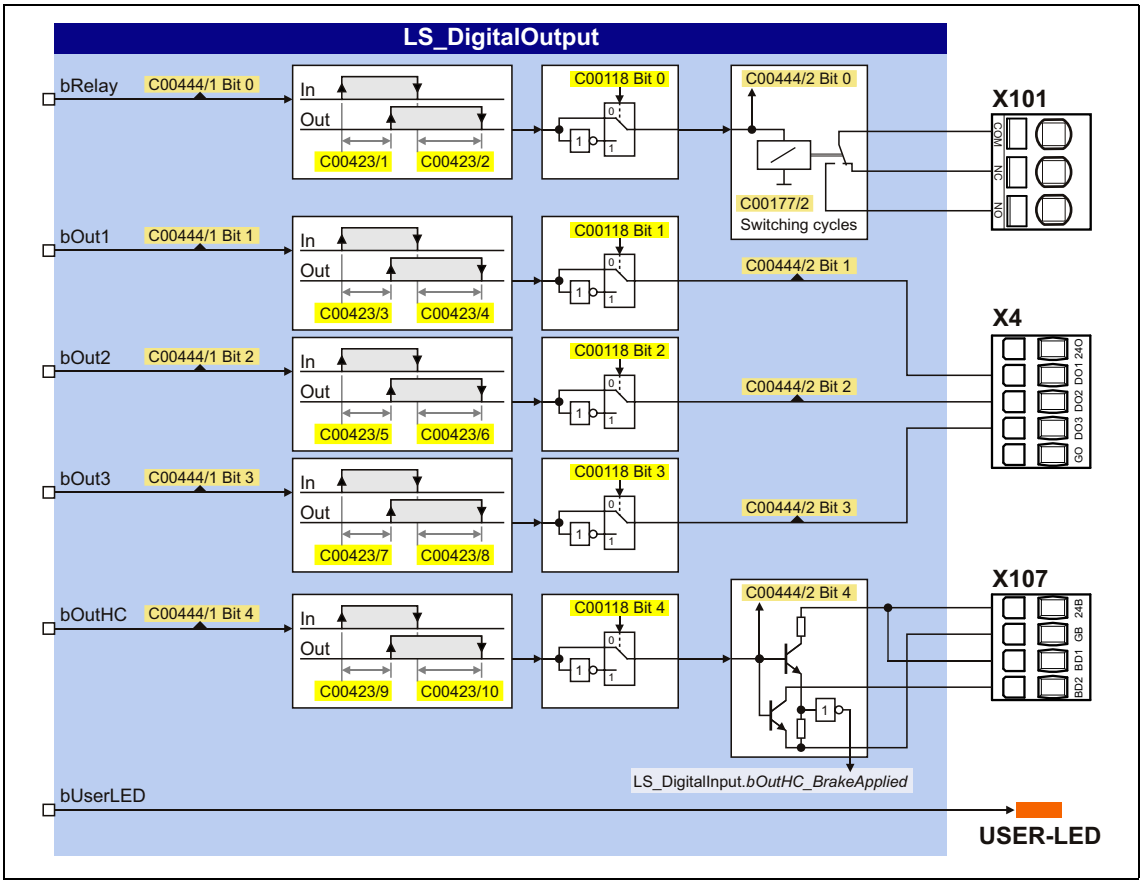
**Related topics:**

- Configuring exception handling of the output terminals (📖 428)
- User-defined terminal assignment (📖 429)

### 7.2.1 Internal interfaces | System block "LS_DigitalOutput"

The **LS_DigitalOutput** system block maps the digital output terminals in the FB editor.

- From version 12.00.00 onwards, the orange USER LED at the front of the inverter can be controlled with any digital process signal via the *bUserLED* input.




| Input<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| bRelay<br>C00444/1 \| BOOL | Relay output, potential-free two-way switch | |
| bOut1 ... bOut3<br>C00444/1 \| BOOL | Digital output DO1 ... DO3 | |
| bOutHC<br>C00444/1 \| BOOL | Output for brake control | |
| bUserLED<br>C00444/1 \| BOOL<br>(from version 12.00.00) | Control of the USER LED at the front of the inverter | |
| | FALSE | USER LED off |
| | TRUE | USER LED on |

## 7.3 Analog terminals

The analog input terminals together with the analog output terminals are located on the X3 plug connector.

### Analog input terminals

The inverter has four analog input terminals for detecting two current signals and two voltage signals:

- Voltage signals in the ± 10 V range
  The voltage signal can be e.g. an analog speed setpoint or the signal of an external sensor (temperature, pressure, etc.).
- Current signals in the 0/+ 4 ... + 20 mA range
  For open-circuit monitoring, the current signal can be evaluated with regard to "Life Zero" or "Dead Zero":
  - 0 ... 20 mA, without open-circuit monitoring
  - 4 ... 20 mA, with open-circuit monitoring

**Note!**

To avoid undefined states, free input terminals of the inverter must be assigned as well, e.g. by applying 0 V to the terminal.

### Analog output terminals

The inverter has four analog output terminals.

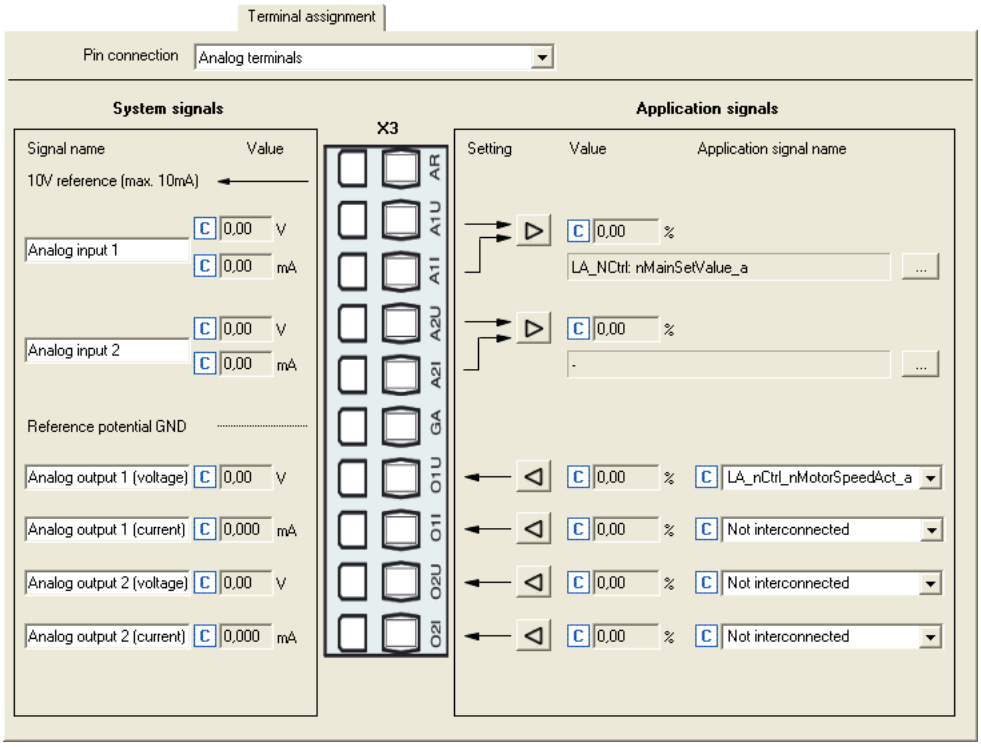
- Two output terminals output an analog current signal (O1I, O2I)
- Two output terminals output an analog voltage signal (O1U, O2U)

**Note!**

Initialisation behaviour:
- After mains switching up to the start of the application, the analog outputs remain set to 0 V.

Exception handling:
- In case of a critical exception in the application (e.g. reset), the analog outputs are set to 0 V.

**Parameterisation dialog in the »Engineer«:**




| Button | Function |
|---|---|
| ▷ | Parameterising analog input (📖 412) |
| ◁ | Parameterising analog output (📖 416) |
| ... | Open the parameterising dialog for assigning application inputs to the analog input. ▸ Changing the terminal assignment with the »Engineer« (📖 433) |

### Short overview of parameters for the analog terminals:

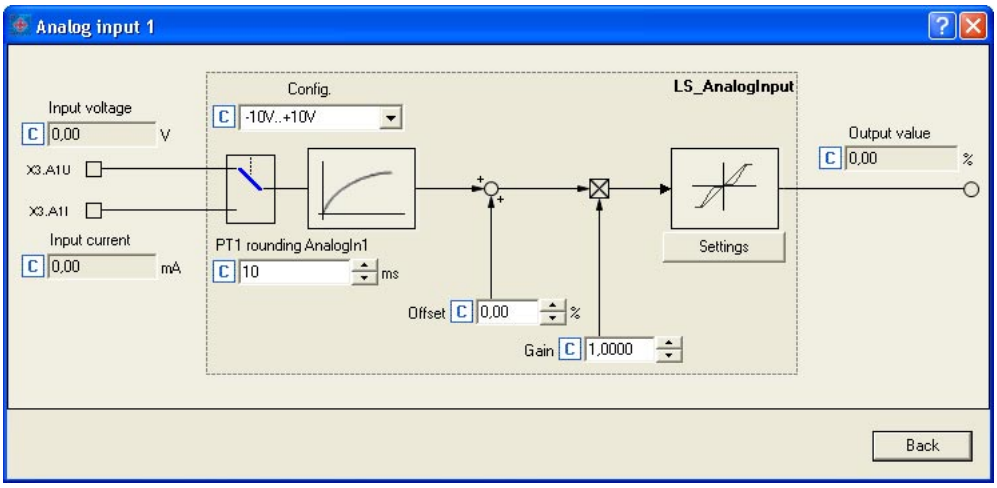
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| **Analog input 1** | | | |
| C00028/1 | AIN1: Input voltage | - | V |
| C00029/1 | AIN1: Input current | - | mA |
| C00033/1 | AIN1: Output value (to application) | - | % |
| **Analog input 2** | | | |
| C00028/2 | AIN2: Input voltage | - | V |
| C00029/2 | AIN2: Input current | - | mA |
| C00033/2 | AIN2: Output value (to application) | - | % |
| **Analog output 1** | | | |
| C00439/1 | O1U: Input value (from application) | - | % |
| C00439/3 | O1I: Input value (from application) | - | % |
| C00436/1 | O1U: Voltage | - | V |
| C00437/1 | O1I: Current | - | mA |
| **Analog output 2** | | | |
| C00439/2 | O2U: Input value (from application) | - | % |
| C00439/4 | O2I: Input value (from application) | - | % |
| C00436/2 | O2U: Voltage | - | V |
| C00437/2 | O2I: Current | - | mA |
| **Analog outputs - terminal assignment** | | | |
| C00620/1 | LS_AnalogOutput: nOut1_a (V) | 1003: LA_nCtrl_nMotorSpeedAct_a | |
| C00620/39 | LS_AnalogOutput: nOut1_a (I) | 0: Not connected | |
| C00620/38 | LS_AnalogOutput: nOut2_a (V) | 0: Not connected | |
| C00620/40 | LS_AnalogOutput: nOut2_a (I) | 0: Not connected | |
| Greyed out = display parameter | | | |

### Related topics:

- Configuring exception handling of the output terminals (📖 428)
- User-defined terminal assignment (📖 429)

### 7.3.1 Parameterising analog input

By clicking on the ▷ button on the **Terminal assignment** tab, you reach the parameterising dialog for the corresponding analog input:




**Short overview of parameters for the analog inputs:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| **Analog input 1** | | | |
| C00034/1 | AIN1: Config. | 0: -10V..+10V | |
| C00026/1 | AIN1: Offset | 0.00 | % |
| C00027/1 | AIN1: Gain | 1.0000 | |
| C00028/1 | AIN1: Input voltage | - | V |
| C00029/1 | AIN1: Input current | - | mA |
| C00033/1 | AIN1: Output value (to application) | - | % |
| C00440/1 | PT1 rounding AnalogIn1 | 10 | ms |
| C00598/1 | Resp. to open circuit AIN1 | 3: TroubleQuickStop | |
| **Analog input 2** | | | |
| C00034/2 | AIN2: Config. | 0: -10V..+10V | |
| C00026/2 | AIN2: Offset | 0.00 | % |
| C00027/2 | AIN2: Gain | 1.0000 | |
| C00028/2 | AIN2: Input voltage | - | V |
| C00029/2 | AIN2: Input current | - | mA |
| C00033/2 | AIN2: Output value (to application) | - | % |
| C00440/2 | PT1 rounding AnalogIn2 | 10 | ms |
| C00598/2 | Resp. to open circuit AIN2 | 3: TroubleQuickStop | |
| Greyed out = display parameter | | | |

### Using current input A1I/A2I

In the Lenze setting, voltage signals in the range of ± 10 V are evaluated via the A1U and A2U input terminals. If current signals are detected via the A1I or A2I input terminals instead, the selection "1: 0...20mA" or "2: 4...20mA" is to be set in C00034.



**Tip!**

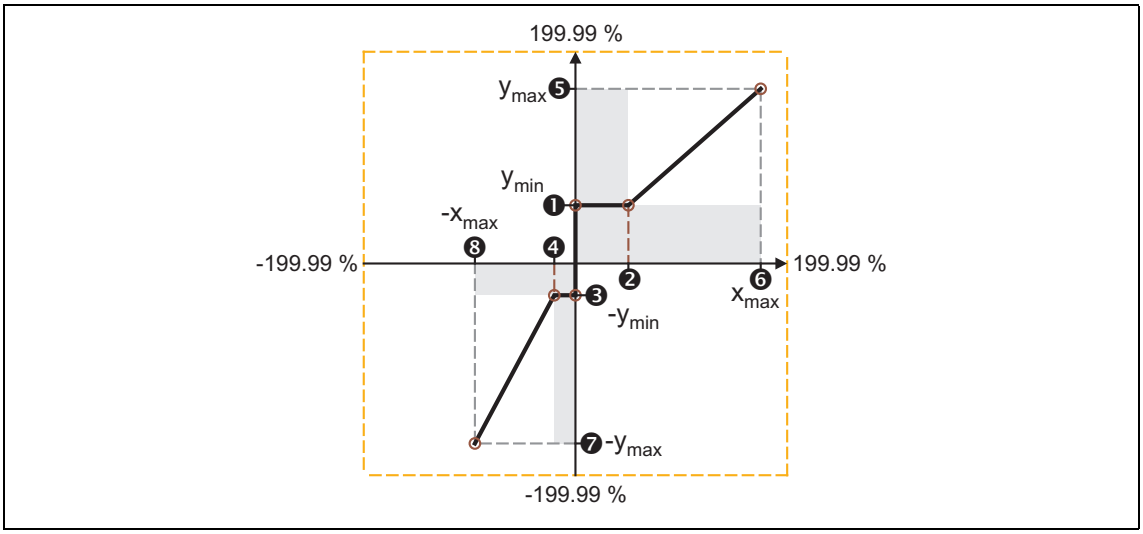
By selecting "2: 4...20mA", you can implement a 4 ...20 mA current loop, e.g. for stipulation of the speed setpoint.

### Open-circuit monitoring

In the case of configuration as a 4 ... 20 mA current loop, the fault response set in C00598 takes place in the event of a wire breakage (Lenze setting: "TroubleQuickStop").

### 7.3.1.1 Signal adaptation by means of characteristic

According to the illustration below, an individual characteristic can be parameterised for the analog inputs via the subcodes of C00010 and C00020 to provide different slopes and a dead band. Here, the input signal corresponds to the X axis and the output signal corresponds to the Y axis:




[7-5] Characteristic for analog inputs

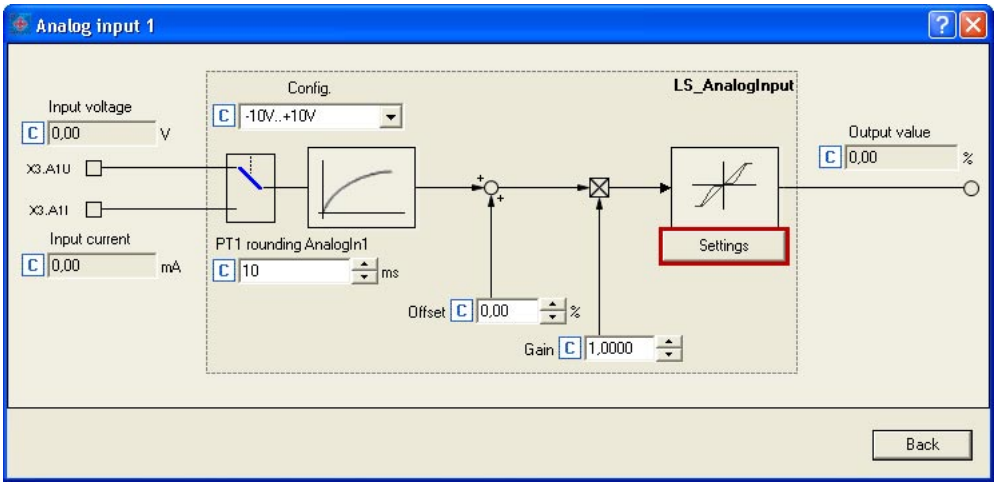
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00010/1 | ❶ AIN1: (+y0) = min | 0.00 | % |
| C00010/2 | ❷ AIN1: (+x0) = Dead band | 1.00 | % |
| C00010/3 | ❸ AIN1: (-y0) = (-min) | 0.00 | % |
| C00010/4 | ❹ AIN1: (-x0) = (-Dead band) | 1.00 | % |
| C00010/5 | ❺ AIN1: (+ymax) | 199.99 | % |
| C00010/6 | ❻ AIN1: (+xmax) | 199.99 | % |
| C00010/7 | ❼ AIN1: (-ymax) | 199.99 | % |
| C00010/8 | ❽ AIN1: (-xmax) | 199.99 | % |
| C00020/1 | ❶ AIN2: (+y0) = min | 0.00 | % |
| C00020/2 | ❷ AIN2: (+x0) = Dead band | 1.00 | % |
| C00020/3 | ❸ AIN2: (-y0) = (-min) | 0.00 | % |
| C00020/4 | ❹ AIN2: (-x0) = (-Dead band) | 1.00 | % |
| C00020/5 | ❺ AIN2: (+ymax) | 199.99 | % |
| C00020/6 | ❻ AIN2: (+xmax) | 199.99 | % |
| C00020/7 | ❼ AIN2: (-ymax) | 199.99 | % |
| C00020/8 | ❽ AIN2: (-xmax) | 199.99 | % |

In the »Engineer«, there is a parameterising dialog for entering the characteristic. This dialog also displays the set characteristic graphically.

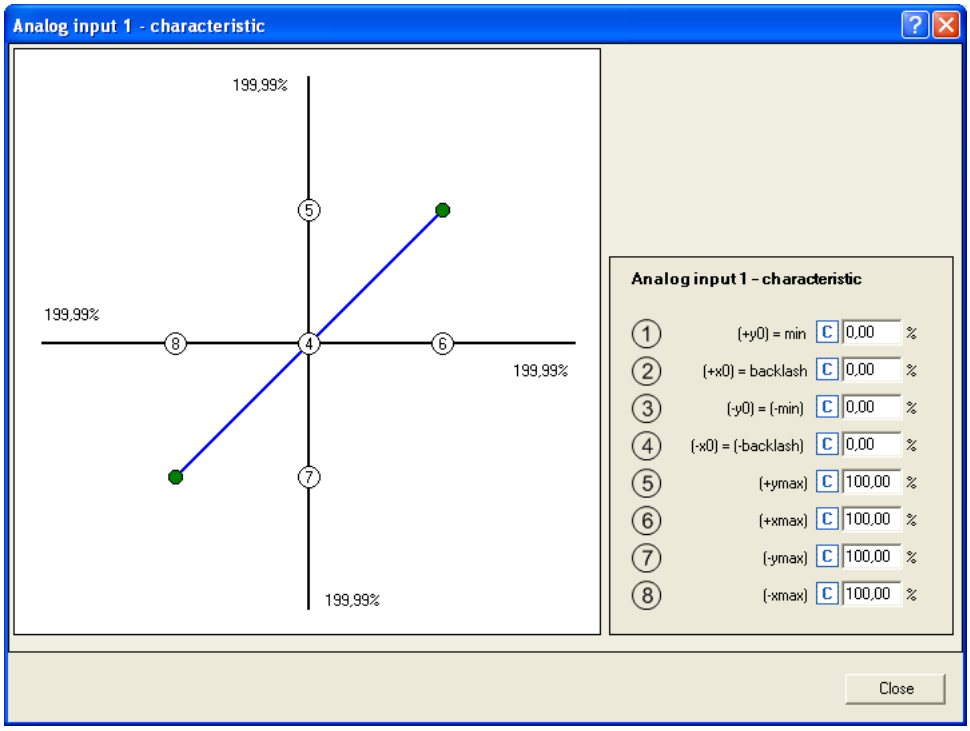
**Proceed as follows to open the dialog for parameterising the characteristic:**

1. Go to the **Terminal assignment** tab and select the "Analog terminals" entry in the **Control connections** list field.

2. Click on the ▷ button for the analog input in order to open the *Analog input* dialog.

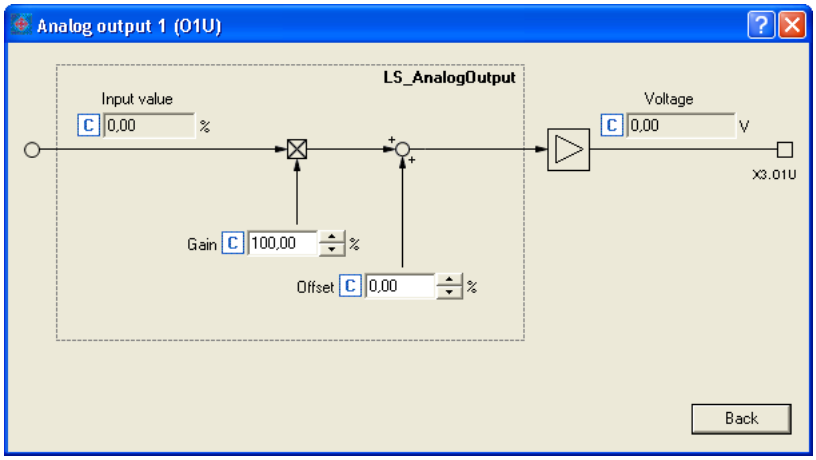



3. Click on the **Settings** button in order to open the *Analog input - Characteristic* dialog box:



### 7.3.2 Parameterising analog output

By clicking on the  button on the **Terminal assignment** tab, you can open the parameterising dialog for the corresponding analog output (here: O1U):

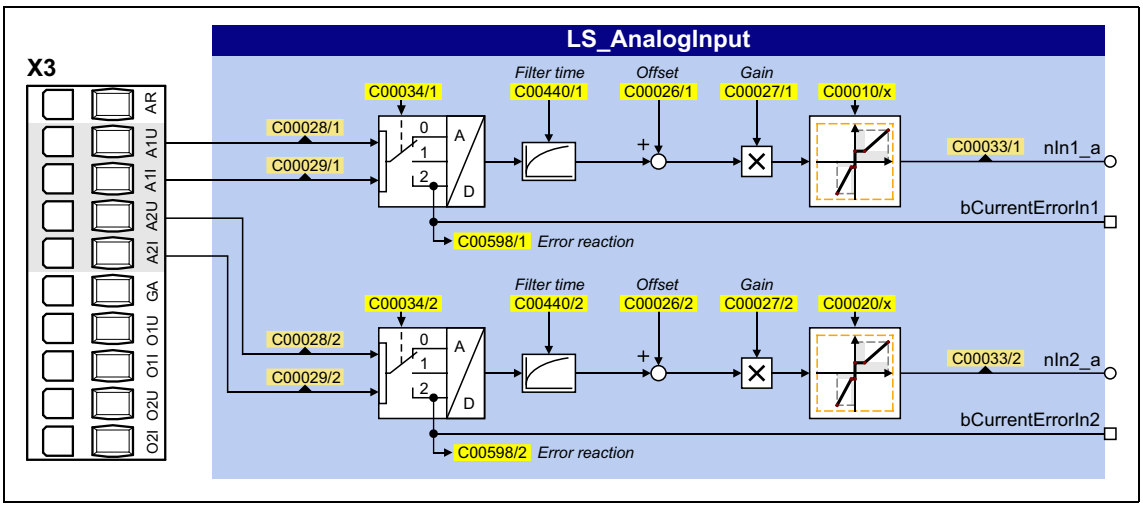



**Short overview of parameters for the analog outputs:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| **Analog output 1** | | | |
| C00434/1 | O1U: Gain | 100.00 | % |
| C00435/1 | O1U: Offset | 0.00 | % |
| C00439/1 | O1U: Input value (from application) | - | % |
| C00439/3 | O1I: Input value (from application) | - | % |
| C00436/1 | O1U: Voltage | - | V |
| C00437/1 | O1I: Current | - | mA |
| **Analog output 2** | | | |
| C00434/2 | O2U: Gain | 100.00 | % |
| C00435/2 | O2U: Offset | 0.00 | % |
| C00439/2 | O2U: Input value (from application) | - | % |
| C00439/4 | O2I: Input value (from application) | - | % |
| C00436/2 | O2U: Voltage | - | V |
| C00437/2 | O2I: Current | - | mA |
| Greyed out = display parameter | | | |

### 7.3.3 Internal interfaces | System block "LS_AnalogInput"

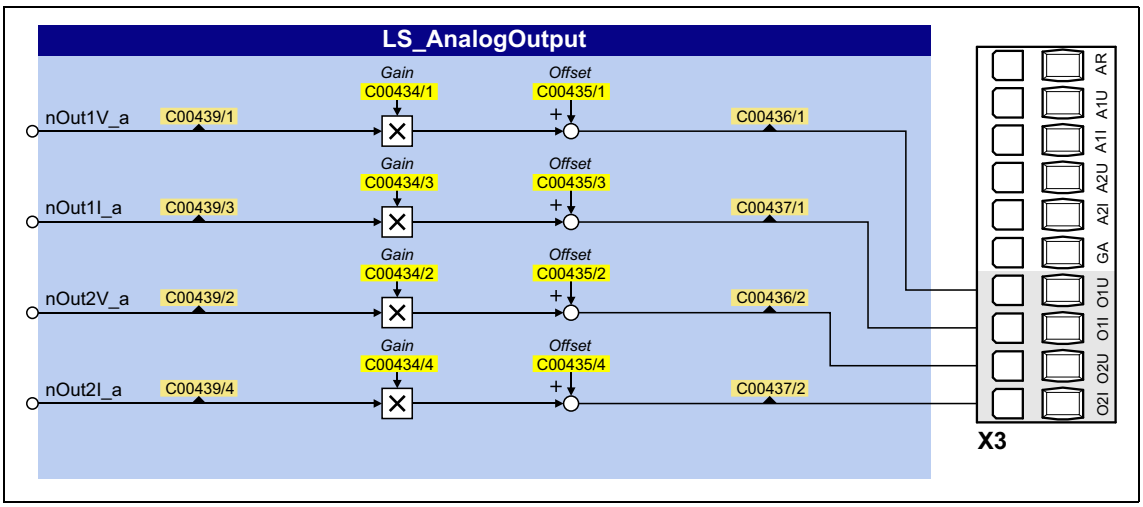
The **LS_AnalogInput** system block maps the analog inputs in the FB editor.




| Out<br>DIS code \| data type | Value/meaning |
|---|---|
| nIn1_a<br>C00033/1 \| INT | Analog input 1<br>• Scaling:<br>$\pm 2^{14} \equiv \pm 10$ V for use as voltage input<br>$+2^{14} \equiv +20$ mA for use as current input |
| bCurrentErrorIn1<br>BOOL | Status signal "Current input error"<br>• Only when analog input 1 is used as current input.<br>• Application: Cable-breakage monitoring of the 4 ...20 mA circuit. |
| | TRUE \| $\lvert I_{AIN1} \rvert < 4$ mA |
| nIn2_a<br>C00033/2 \| INT | Analog input 2<br>• Scaling:<br>$\pm 2^{14} \equiv \pm 10$ V for use as voltage input<br>$+2^{14} \equiv +20$ mA for use as current input |
| bCurrentErrorIn2<br>BOOL | Status signal "Current input error"<br>• Only when analog input 2 is used as current input.<br>• Application: Cable-breakage monitoring of the 4 ...20 mA circuit. |
| | TRUE \| $\lvert I_{AIN2} \rvert < 4$ mA |

### 7.3.4 Internal interfaces | System block "LS_AnalogOutput"

The **LS_AnalogInput** system block maps the analog outputs in the FB editor.




| Input<br>DIS code \| data type | Information/possible settings |
|---|---|
| nOut1V_a<br>C00439/1 \| INT | Analog output 1 (voltage)<br>• Scaling: $2^{14} \equiv 16384 \equiv 10$ V |
| nOut1I_a<br>C00439/3 \| INT | Analog output 1 (current)<br>• Scaling: $2^{14} \equiv 16384 \equiv 20$ mA |
| nOut2V_a<br>C00439/2 \| INT | Analog output 2 (voltage)<br>• Scaling: $2^{14} \equiv 16384 \equiv 10$ V |
| nOut2I_a<br>C00439/4 \| INT | Analog output 2 (current)<br>• Scaling: $2^{14} \equiv 16384 \equiv 20$ mA |

## 7.4 Touch probe detection

"Touch probe" (TP) in general means quick detection of a position by a quick sensor if the pulse duration of the sensor signal is too short to be detected by a "normal" digital input (scanning time: 1 ms).

Moreover, touch probe requires exact detection of the position. Here, the difference in position between the 1 ms position detection and the touch probe signal is considered as correction value.

**Applications:**

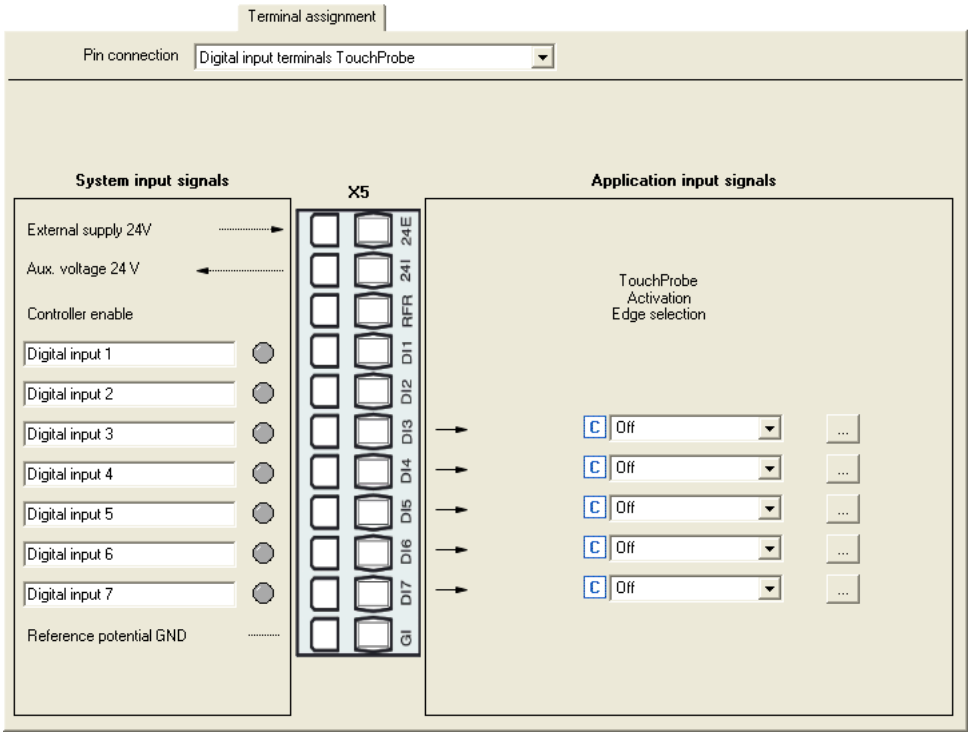
- Precise approach of a position at a previously detected signal mark
- High-precision traversing of a section starting at a previously detected signal mark
- Safe detection of signal marks with very short signal edges
- Measurement of a high-precision position at a previously detected signal mark
- Homing to touch probe (homing signal)
- Relative residual path positioning (traversing of a section starting at signal mark)
- Absolute positioning to target position, activated by a safely detected signal mark
- Position measurement of a distance between 2 signal marks

### Overview of touch probe signal sources

The 8400 TopLine provides the following signal sources for touch probe detection which can be configured independently:

| Signal source | Edge sensitivity | Applications |
|---|---|---|
| Digital input DI3<br>...<br>Digital input DI7 | Rising edge, falling edge, rising and falling edge (parameterisable) | Homing<br>Residual path positioning<br>Position measurement<br>Freely interconnectable |
| Z-track encoder<br>(from version 12.00.00) | | Homing |
| Z-track resolver<br>(from version 12.00.00) | | Homing |

**Parameterisation dialog in the »Engineer«:**



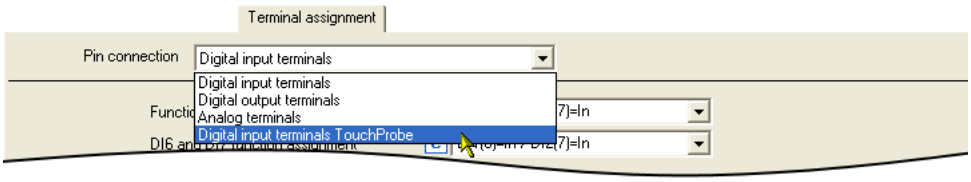

| Button | Function |
|---|---|
| ... | Open the parameterisation dialog for the selected TP signal source.<br>▸ Parameter setting (📖 421) |

| Parameters | Info | Lenze setting |
|---|---|---|
| C02810/3...7 | TPDigIn3 ... TPDigIn7: Edge selection | 0: Off |

### 7.4.1 Parameter setting

Proceed as follows to open the parameterisation dialog for setting a TP signal source:

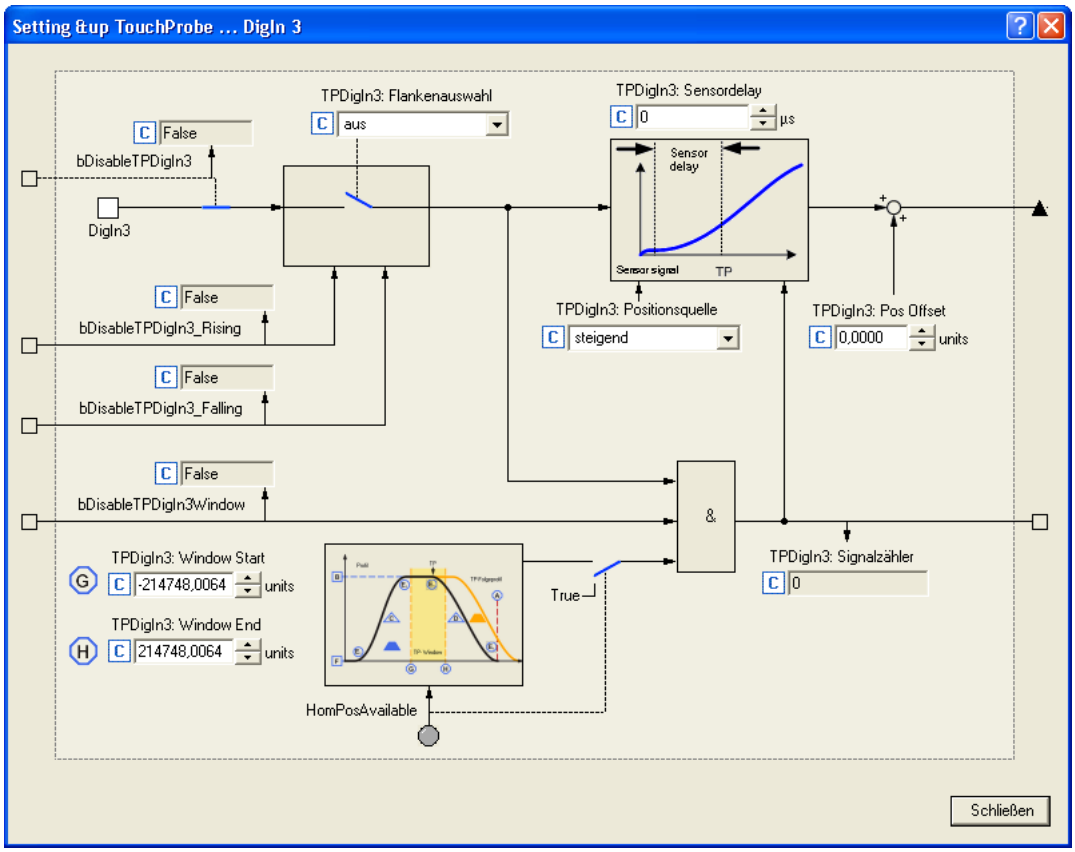
1. Go to the **Terminal Assignment** tab and select "Digital input terminals TouchProbe" in the **Control connections** list field:




2. Click the ... button for the digital input that is to be set for the touch probe detection.

**Tip!**

If the "Table positioning" technology application is used, the following parameterisation dialog can be opened for the selected TP signal source in the parameterisation dialogs for the Profile entry and the basic "Homing" function via the **Set up touch probe...** button:



**Short overview of the relevant parameters:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02810/3...7 | TPDigIn3 ... TPDigIn7: Edge selection | 0: Off | |
| C02811/3...7 | TPDigIn3 ... TPDigIn7: Sensor delay | 0 | µs |
| C02812/3...7 | TPDigIn3 ... TPDigIn7: Pos Offset | 0.0000 | units |
| C02813/1...3 | TPDigIn3 ... TPDigIn5: Window start | -214748.3647 | units |
| C02814/1...3 | TPDigIn3 ... TPDigIn5: Window end | 214748.3647 | units |
| C02815/3...7 | TPDigIn3 ... TPDigIn7: Position source | 0: Position encoder actual value | |
| C02816/3...7 | TPDigIn3 ... TPDigIn7: Signal counter | - | |
| C02817/3...7 | TPDigIn3 ... TPDigIn7: TP position | - | units |
| Greyed out = display parameter | | | |

### Edge selection

Select which edge the corresponding input is to respond to.

- In the case of signal sources DI3 ... DI5, the edge sensitivity can also be dynamically changed via inputs at the LS_TouchProbe SB.

### Sensor delay

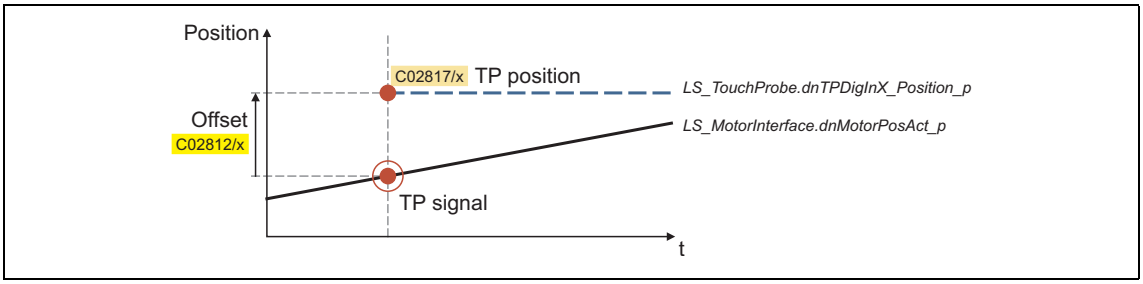
This setting serves to compensate for a sensor delay of the touch probe sensor, if any.

- Typical values for laser photoelectric barriers are e.g. 300 µs.
- Delay of the digital inputs for the 8400 device series:
  - 5 µs for a rising edge
  - 25 µs for a falling edge
- Internal automatic compensation for device-internal signal delay due to encoder zero pulse.

### Pos Offset

Use this setting to add an offset to the position value measured by touch probe. This may be required if the touch probe sensor has assumed a disadvantageous position on the machine. By adding an offset the touch probe sensor can be moved to a position which is more convenient to the application.

The offset value given in [units] is added to the TP position value which depends on the position source selected in C02815/x. The offset value influences the display of the TP position (C02817/x), home position values internally derived from the touch probe and the respective output *dnTpDigInX_Position_p* at the SB LS_TouchProbe. For this purpose, the offset value is converted internally from [units] to [increments].



**Window start / end**

Via the two parameters Window start (C02813/x) and Window end (C02814/x), acceptance windows can be set for the DI3 ... DI5 signal sources in which the touch probe signal is accepted.

- The acceptance windows can be dynamically activated via the inputs at the LS_TouchProbe SB.
- If the actual position is outside the acceptance window, touch probe is automatically deactivated.
- If both window limits are set to "±214748.3647", the acceptance window does not have any effect.
- To properly use this function, the drive needs to know the home position (zero position).

**Position source**

Selection of the position signal source to be measured with touch probe. This usually is the actual position of the motor / encoder position encoder.

**Note!**

When the touch probe functionality is used in the operating modes "Positioning" and "Homing":

Make sure that the position source of the respective TP signal in C02815/x is set to "0: position encoder actual value". Otherwise, no TP correction will take place.

### 7.4.1.1 Z-track encoder as a touch probe signal source

From version 12.00.00 onwards, the Z-track of a connected encoder is also supported as a touch probe signal source.

**Short overview of the relevant parameters:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02810/8 | TPEncoderZtrack: Edge selection | 0: Off | |
| C02811/8 | TPEncoderZTrack: Sensor delay | 0 | µs |
| C02812/8 | TPEncoderZTrack: Pos offset | 0.0000 | units |
| C02813/4 | TPEncoderZTrack: Window start | -214748.3647 | units |
| C02814/4 | TPEncoderZTrack: Window end | 214748.3647 | units |
| C02815/8 | TPEncoderZTrack: Position source | 0: Position encoder actual value | |
| C02816/8 | TPEncoderZTrack: Signal counter | - | |
| C02817/8 | TPEncoderZTrack: TP position | - | units |
| Greyed out = display parameter | | | |

**Related topics:**

- Multi-Encoder (X8) (📖 332)

### 7.4.1.2 Z-track resolver as a touch probe signal source

From version 12.00.00 onwards, the Z-track of a connected resolver is also supported as a touch probe signal source.

**Short overview of the relevant parameters:**

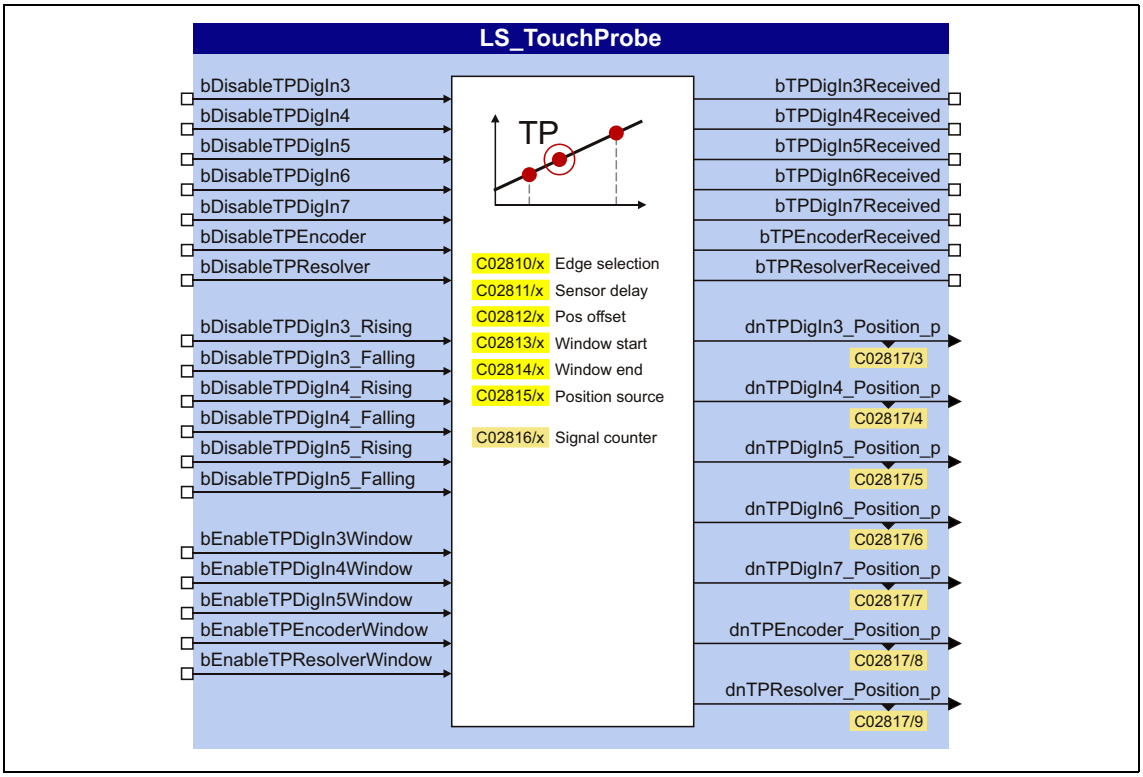
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02810/9 | TPResolverZtrack: Edge selection | 0: Off | |
| C02811/9 | TPResolverZTrack: Sensor delay | 0 | µs |
| C02812/9 | TPResolverZTrack: Pos offset | 0.0000 | units |
| C02813/5 | TPResolverZTrack: Window start | -214748.3647 | units |
| C02814/5 | TPResolverZTrack: Window end | 214748.3647 | units |
| C02815/9 | TPResolverZTrack: Position source | 0: Position encoder actual value | |
| C02816/9 | TPResolverZTrack: Signal counter | - | |
| C02817/9 | TPResolverZTrack: TP position | - | units |
| Greyed out = display parameter | | | |

**Related topics:**

- Resolver (X7) (📖 326)

### 7.4.2 Internal interfaces | System block "LS_TouchProbe"

The **LS_TouchProbe** system block provides the internal interfaces for touch probe detection in the function block editor:




#### Inputs

| Input<br>Data type | Information/possible settings | |
|---|---|---|
| bDisableTPDigIn3...7<br>BOOL | DI3 ... DI7: Dynamically deactivate TP function | |
| | TRUE | TP function is deactivated. |
| bDisableTPEncoder<br>BOOL<br>(from version 12.00.00) | Z-track of encoder: Dynamically deactivate TP function | |
| | TRUE | TP function is deactivated. |
| bDisableTPResolver<br>BOOL<br>(from version 12.00.00) | Z-track of resolver: Dynamically deactivate TP function | |
| | TRUE | TP function is deactivated. |
| bDisableTPDigIn3...5_Rising<br>BOOL | DI3 ... DI5: Dynamically deactivate detection of rising edges | |
| | TRUE | Detection of rising edges is deactivated. |
| bDisableTPDigIn3...5_Falling<br>BOOL | DI3 ... DI5: Dynamically deactivate detection of falling edges | |
| | TRUE | Detection of falling edges is deactivated. |
| bEnableTPDigIn3...5Window<br>BOOL | DI3 ... DI5: Activate acceptance window | |
| | TRUE | Acceptance window function is active:<br>• If the actual position is outside the acceptance window, whose starting position is set in C02813/1...3 and whose end position is set in C02814/1...3, touch probe is automatically deactivated. |
| bEnableTPEncoderWindow<br>BOOL<br>(from version 12.00.00) | Z-track of encoder: Activate acceptance window | |
| | TRUE | Acceptance window function is active:<br>• If the actual position is outside the acceptance window, whose starting position is set in C02813/4 and whose end position is set in C02814/4, touch probe is automatically deactivated. |

| Input<br>Data type | Information/possible settings | |
|---|---|---|
| bEnableTPResolverWindow<br>BOOL<br>(from version 12.00.00) | Z-track of resolver: Activate acceptance window | |
| | TRUE | Acceptance window function is active:<br>• If the actual position is outside the acceptance window, whose starting position is set in C02813/5 and whose end position is set in C02814/5, touch probe is automatically deactivated. |

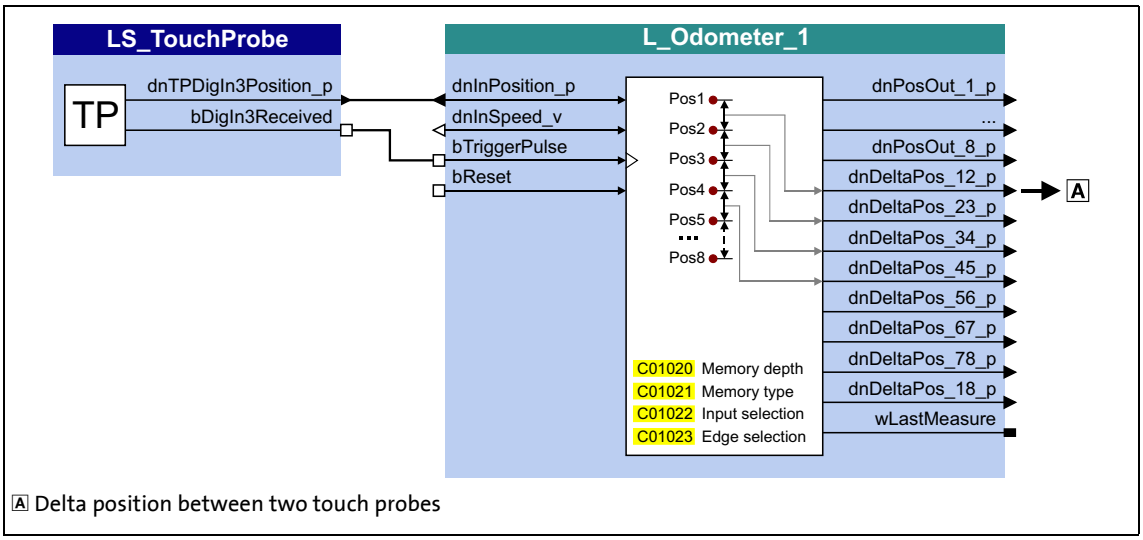
### Outputs

| Out<br>Data type | Value/meaning | |
|---|---|---|
| bTPDigIn3...7Received<br>BOOL | TRUE | DI3 ... DI7: Touch probe received.<br>• The signal is only pending for one task cycle (1 ms). |
| bTPEncoderReceived<br>BOOL<br>(from version 12.00.00) | TRUE | Z-track of encoder: Touch probe received.<br>• The signal is only pending for one task cycle (1 ms). |
| bTPResolverReceived<br>BOOL<br>(from version 12.00.00) | TRUE | Z-track of resolver: Touch probe received.<br>• The signal is only pending for one task cycle (1 ms). |
| dnTPDigIn3...7_Position_p<br>DINT | DI3 ... DI7: Position measured by touch probe in [increments] | |
| dnTPEncoder_Position_p<br>DINT<br>(from version 12.00.00) | Z track of encoder: Position measured by touch probe in [increments] | |
| dnTPResolver_Position_p<br>DINT<br>(from version 12.00.00) | Z track of resolver: Position measured by touch probe in [increments] | |

### 7.4.2.1 Application example: "Position measurement"

The touch probe function can be combined with the L_Odometer FB for a position measurement. This FB is able to save position signals to a ring buffer and detect differences between two position signals.

In the wiring below, digital input DI3 is used to connect the touch probe sensor. For the sake of clarity, irrelevant inputs and outputs of the LS_TouchProbe SB are masked out.




[7-6] Wiring for position measurement

## 7.5 Configuring exception handling of the output terminals

Exception handling for the analog and digital output terminals in the event of an error can be set via decoupling configuration and decoupling values.

- Bit coded selection is carried out in C00441 for the analog output terminals, defining the events that will trigger decoupling.
- Bit coded selection is carried out in C00447 for the digital output terminals, defining the events that will trigger decoupling.

| Bit | Event |
|---|---|
| Bit 0 ☐ | SafeTorqueOff |
| Bit 1 ☐ | ReadyToSwitchOn |
| Bit 2 ☐ | SwitchedOn |
| Bit 3 ☐ | Reserved |
| Bit 4 ☐ | Trouble |
| Bit 5 ☐ | Fault |
| Bit 6 ☐ | Reserved |
| Bit 7 ☐ | Reserved |
| Bit 8 ☐ | Reserved |
| Bit 9 ☐ | Fail CAN_Management |
| Bit 10 ☐ | Reserved |
| Bit 11 ☐ | Reserved |
| Bit 12 ☐ | Reserved |
| Bit 13 ☐ | Reserved |
| Bit 14 ☐ | Reserved |
| Bit 15 ☐ | Reserved |

Finally, the following parameters define the value/status that the output terminals are to have when they are decoupled:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00442/1 | AOut1_U: Decoupling value | 0.00 | % |
| C00442/2 | AOut2_U: Decoupling value | 0.00 | % |
| C00442/3 | AOut1_I: Decoupling value | 0.00 | % |
| C00442/4 | AOut2_I: Decoupling value | 0.00 | % |
| C00448 | DigOut decoupling value | Bit coded | |

**Related topics:**

- Configuring exception handling of the CAN PDOs (📖 806)

## 7.6 User-defined terminal assignment

In order to individually adapt the preconfigured assignment of the input/output terminals to your application, you can choose one of the following procedures:

A. In the »Engineer«:
   - Change the terminal assignment on the **Terminal assignment** tab.
   - Change the signal assignment on the **Application Parameters** tab, on the dialog level *Overview* → *Signal flow*.
   - Change the interconnections in the FB editor (on the I/O level).

B. In the »Engineer« or with the keypad:
   - Change the parameters for signal configuration in the parameters list.

**Note!**

If you change the preconfigured assignment of the input/output terminals, the terminal assignment will be a user-defined one. In C00007, control mode "0: Interconnection changed" will be shown.

**Tip!**

First of all, select a Lenze configuration useful for the purpose at hand by going to C00005 and selecting a technology application that matches your drive task and then going to C00007 and selecting an appropriate control mode. You will then have an application for which there is a signal flow, logical block links and terminal assignment.

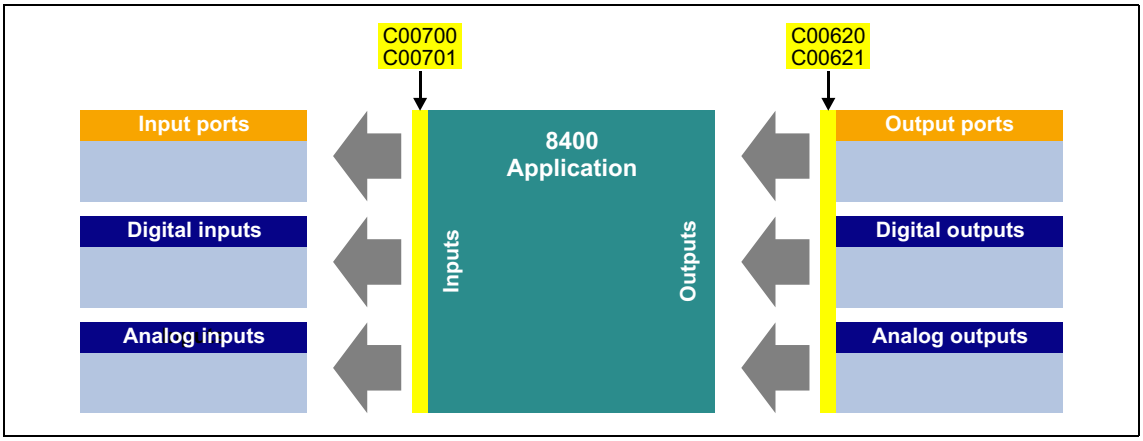
We recommend using the »Engineer« for the implementation of comprehensive user-defined drive solutions.

### 7.6.1 Source-destination principle

The I/O configuration of the input and output signals is carried out according to the source/destination principle:

- A connection always has a direction and therefore always has a source and a target.
- The inputs signals of the technology application are logically linked to the outputs of system blocks which represent the device input terminals.
- The inputs of system blocks that represent the device output terminals are logically linked to output signals of the technology application.

The following graphic illustrates the source/destination principle:




[7-7] Source-destination principle

Note the following:

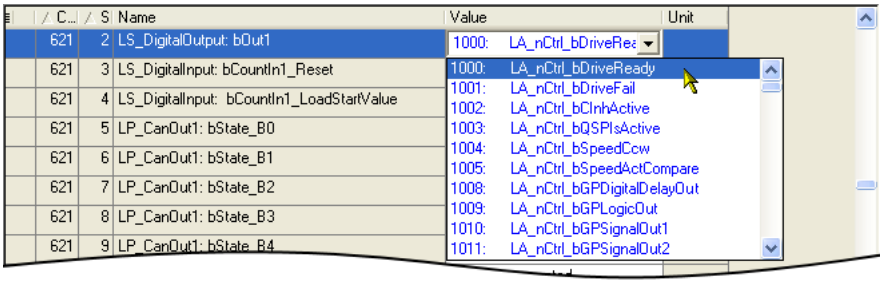
- An equipment input terminal can be logically linked to several inputs of the application block.
- Every input of the application block can only be logically linked to one input signal.
- An output of the application block can be logically linked to several device output terminals.

### 7.6.2 Changing the terminal assignment with the keypad

You can reconfigure the preconfigured terminal assignment with the keypad (and with the »Engineer«) by means of so-called configuration parameters.

- Each configuration parameter represents a signal input of a system block or application block.
- Each configuration parameter contains a selection list with output signals of the same type of data.
- Logical linking is thus carried out by selecting the output signal for the corresponding signal input.

In the following example, digital output 1 (**LS_DigitalOutput**.*bOut1* input) is logically linked to the status signal "Drive ready" (*LA_nCtrl_bDriveReady* output signal):




#### Configuration parameters for the analog and digital output terminals

The preconfigured assignment of the analog and digital output terminals can be altered by means of the subcodes of C00620 and C00621:

| Parameters | Info | Lenze setting |
|---|---|---|
| **Analog outputs - terminal assignment** | | |
| C00620/1 | LS_AnalogOutput: nOut1_a (V) | 1003: LA_nCtrl_nMotorSpeedAct_a |
| C00620/39 | LS_AnalogOutput: nOut1_a (I) | 0: Not connected |
| C00620/38 | LS_AnalogOutput: nOut2_a (V) | 0: Not connected |
| C00620/40 | LS_AnalogOutput: nOut2_a (I) | 0: Not connected |
| **Digital outputs - terminal assignment** | | |
| C00621/1 | LS_DigitalOutput:bRelay | 1001: LA_nCtrl_bDriveFail |
| C00621/2 | LS_DigitalOutput:bOut1 | 1000: LA_nCtrl_bDriveReady |
| C00621/99 | LS_DigitalOutput: bOut2 | 0: Not connected |
| C00621/100 | LS_DigitalOutput: bOut3 | 0: Not connected |
| C00621/101 | LS_DigitalOutput: bOut HighCurrent | 0: Not connected |

Other subcodes (not shown here) allow the configuration of input signals of different system blocks and port blocks.

### Configuration parameters for the inputs of the technology application

The following parameters can be used to change the preconfigured assignment of the application inputs:

| Parameters | Info |
|---|---|
| **TA "Actuating drive speed":** Configuration parameters (📖 462) | |
| C00700/x | Analog connection list |
| C00701/x | Digital connection list |
| **TA "Table positioning":** Configuration parameters (📖 524) | |
| C00710/x | Analog connection list |
| C00711/x | Digital connection list |
| **TA "Switch-off positioning":** Configuration parameters (📖 551) | |
| C00760/x | Analog connection list |
| C00761/x | Digital connection list |

### Example

Task: Starting from the preset technology application "Actuating drive speed" and the "Terminals 0" control mode, the DI2 digital input is to be used for choosing an alternative acceleration/deceleration time for the main setpoint instead of for choosing the fixed setpoint 2/3. To do this, the DI2 digital input is not to be linked to the *bJogSpeed2* input but to the *bJogRamp1* input of the application module.

Procedure:

1. Use the keypad to go to the menu level **Applications → Actuating drive speed (conf.)**. This menu level contains all the configuration parameters of the "Actuating drive speed" technology application". ▸ Configuration parameters (📖 462)
2. Navigate to the configuration parameter LA_NCtrl: bJogSpeed2 (C00701/10) which represents the logical signal link of the application input *bJogSpeed2*.
3. Change the setting of C00701/10:
   Change selection "16001: DigIn_bIn2" in selection "0: Not interconnected".
4. Navigate to the configuration parameter LA_NCtrl: bJogRamp1 (C00701/13) which represents the logical signal link of the application input *bJogRamp1*.
5. Change the setting of C00701/13:
   Change selection "0: Not interconnected" in selection "16001: DigIn_bIn2".



**Tip!**

The example shows that, for each input of the application block, the associated configuration parameter (C00700/x or C00701/x) is only allowed to contain one source that you enter.

### 7.6.3 Changing the terminal assignment with the »Engineer«

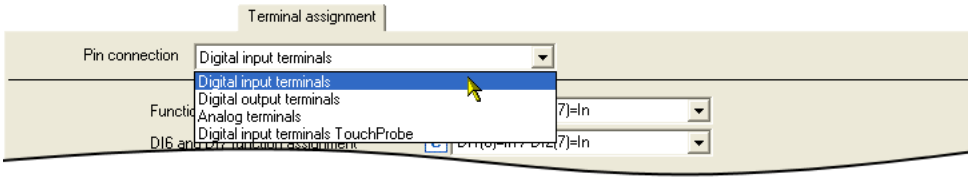
Whereas the configuration parameters referred to have to be parameterised with the keypad, implementation in the »Engineer« is much easier due to the availability of the corresponding dialogs. The following task illustrates the respective procedure.

Task: Starting from the preset technology application "Actuating drive speed" and the "Terminals 0" control mode, the DI2 digital input is to be used for choosing an alternative acceleration/deceleration time for the main setpoint instead of for choosing the fixed setpoint 2/3. To do this, the DI2 digital input is not to be linked to the *bJogSpeed2* input but to the *bJogRamp1* input of the application module.

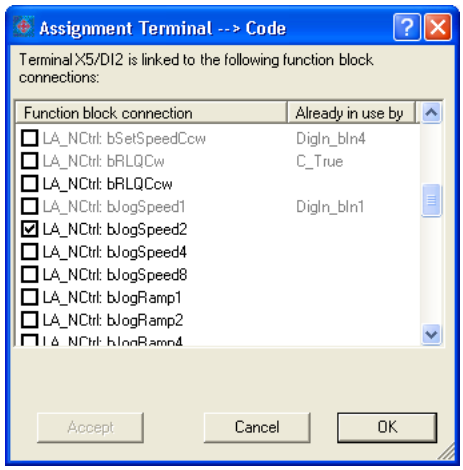
**Possibility 1: Change terminal assignment by means of the Terminal Assignment tab**

Procedure:

1. Go to the **Terminal Assignment** tab and select "Digital input terminals" in the **Control connections** list field:




2. Click on the ... button for the DI2 terminal in order to open the dialog box *Assignment Terminal --> Function block*.
   - In the list field, all block inputs that are currently logically linked to digital input DI2 are marked with a checkmark:

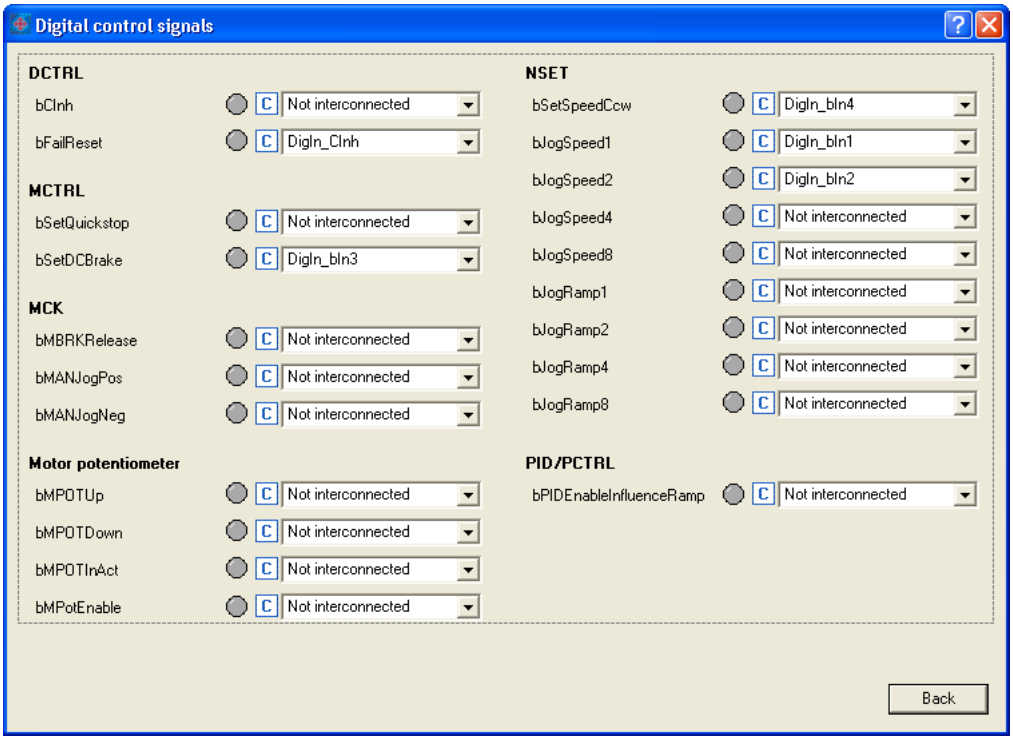



3. Remove checkmark for the connection **LA_NCtrl: bJogSpeed2** in order to cancel the existing logical link.
4. Set checkmark for connection **LA_NCtrl: bJogRamp1** in order to logically link this application input to digital input DI2.

**Possibility 2: Change terminal assignment by means of the signal flow shown**

Procedure:

1. Go to the **Application parameters** tab.
2. Go to the **Application Parameters** tab and click on the **Signal flow** button in order to change to the dialog level *Overview* ➞ *Signal flow*.
3. On the dialog level *Overview* ➞ *Signal flow*, click on the **Digital control signals** button in order to open the *Digital control signals* dialog box:

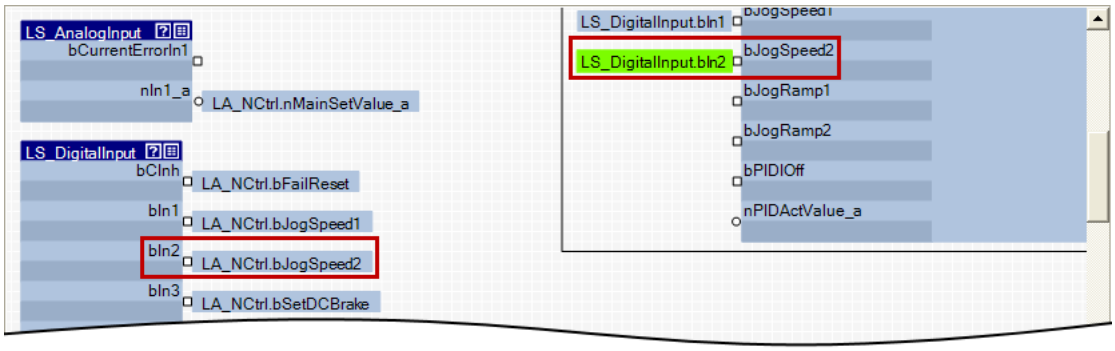



4. In the **bJogSpeed2** list field, set the selection "0: Not interconnected".
5. In the **bJogRamp1** list field, set the selection "16001: DigIn_bIn2".
6. Click on the **Back** button in order to close the dialog box again.

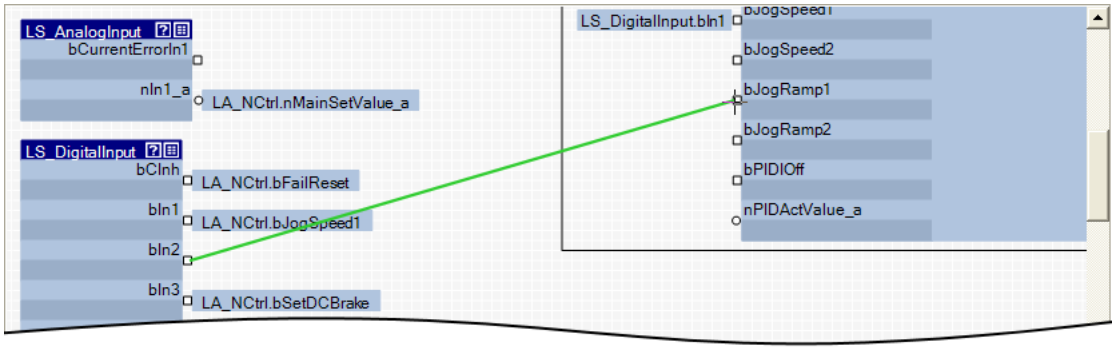
**Possibility 3: Change terminal assignment with the FB editor**

Procedure:

1. Go to the **FB Editor** tab.
2. Delete the existing interconnection from **LS_DigitalInput**.*bIn2* to **LA_NCtrl**.*bJogSpeed2*:




3. Establish a new interconnection from **LS_DigitalInput**.*bIn2* to **LA_NCtrl**.*bJogRamp1*:




**Tip!**

You can find detailed information on how to use the FB editor of the »Engineer« in the main chapter entitled "Working with the FB Editor". (📖 1373)

# 8 Technology applications

This chapter describes the handling and the functional range of the technology applications available for the 8400 TopLine inverter.



**Technology application "Actuating drive speed"**

This technology application preset in C00005 serves to solve speed-controlled drive tasks, e.g. conveyor drives (interconnected), extruders, test benches, vibrators, travelling drives, presses, machine tools, dosing systems.

▸ TA "Actuating drive speed" (📖 438)



**Technology application "actuating drive speed (AC Drive profile)"**

This technology application available from version 13.00.00 provides a speed and torque control by means of "AC Drive Profile". The fieldbuses EtherNet/IP™ and system bus (CANopen) are supported.

▸ TA "actuating drive speed (AC Drive Profile)" (📖 465)



**Technology application "Table positioning"**

This technology application serves to solve position-controlled drive tasks which are normally controlled by a higher-level control via a fieldbus, e.g. transport facilities, rotary tables, storage and retrieval units, feed drives, metering units, hoists.

**Note:** This TA requires an external sequence control!
From version 12.00.00, the FB L_Sequencer_1 can also be used as internal sequence control. This FB processes a positioning program based on a sequence table.

▸ TA "Table positioning" (📖 498)



**"Switch-off positioning" technology application**

This technology application is used to solve speed-controlled drive tasks which require a pre-switch off or stopping at certain positions, e.g. roller conveyors and conveying belts. The pre-switch off is implemented by connecting switch-off sensors.

▸ TA "Switch-off positioning" (📖 527)

**Note!**

Please note that the "StateLine", "HighLine" and "TopLine" device types differ with regard to the number, functional range, and flexibility of the technology applications offered.

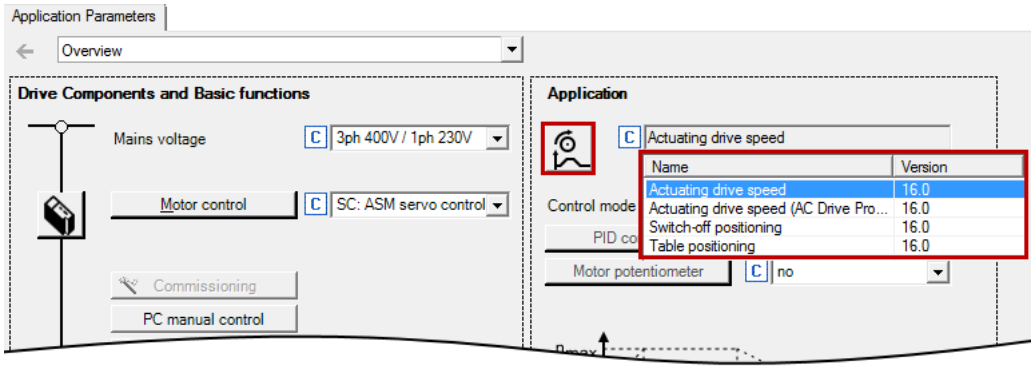
**Related topics:**

▸ Integrated technology applications (📖 33)
▸ Commissioning (📖 55)

## 8.1 Selection of the technology application and the control mode

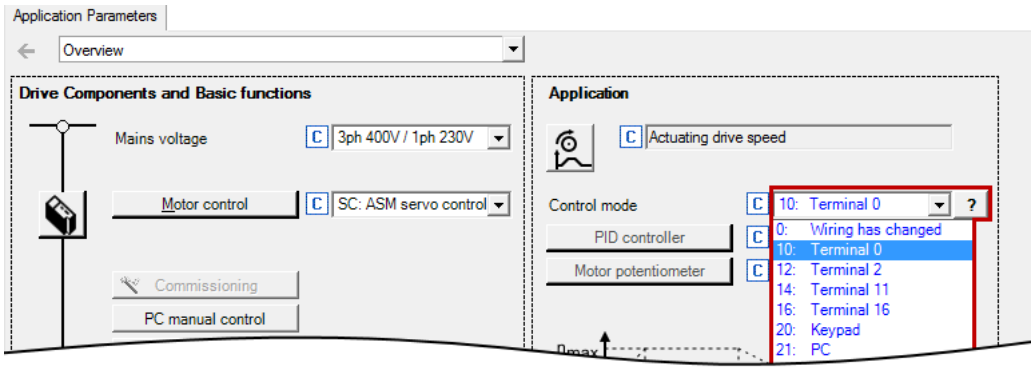
The technology application to be used is selected in C00005.

- You can select the technology application in the »Engineer« on the **Application parameter** tab via the **Application** list field:




Different control modes can be selected for every application in C00007. By selecting the control mode you set the way by which the technology application should be controlled, e.g. via terminals or via a fieldbus. The interconnection of the input/output terminals and ports shown in the FB editor in the I/O level changes accordingly.

- You can select the control mode in the »Engineer« on the **Application parameter** tab via the **Control mode** list field:




**Tip!**

You can infer the pre-configured assignment of the input/output terminals and ports for each control mode from the description of the corresponding technology application:

**TA "Actuating drive speed":** Terminal assignment of the control modes (📖 449)

**TA "Table positioning":** Terminal assignment of the control modes (📖 512)

**TA "Switch-off positioning":** Terminal assignment of the control modes (📖 538)

Detailed information on the individual configuration of the input/output terminals can be found in the description of the I/O terminals in the subchapter"User-defined terminal assignment". (📖 429)

## 8.2 TA "Actuating drive speed"

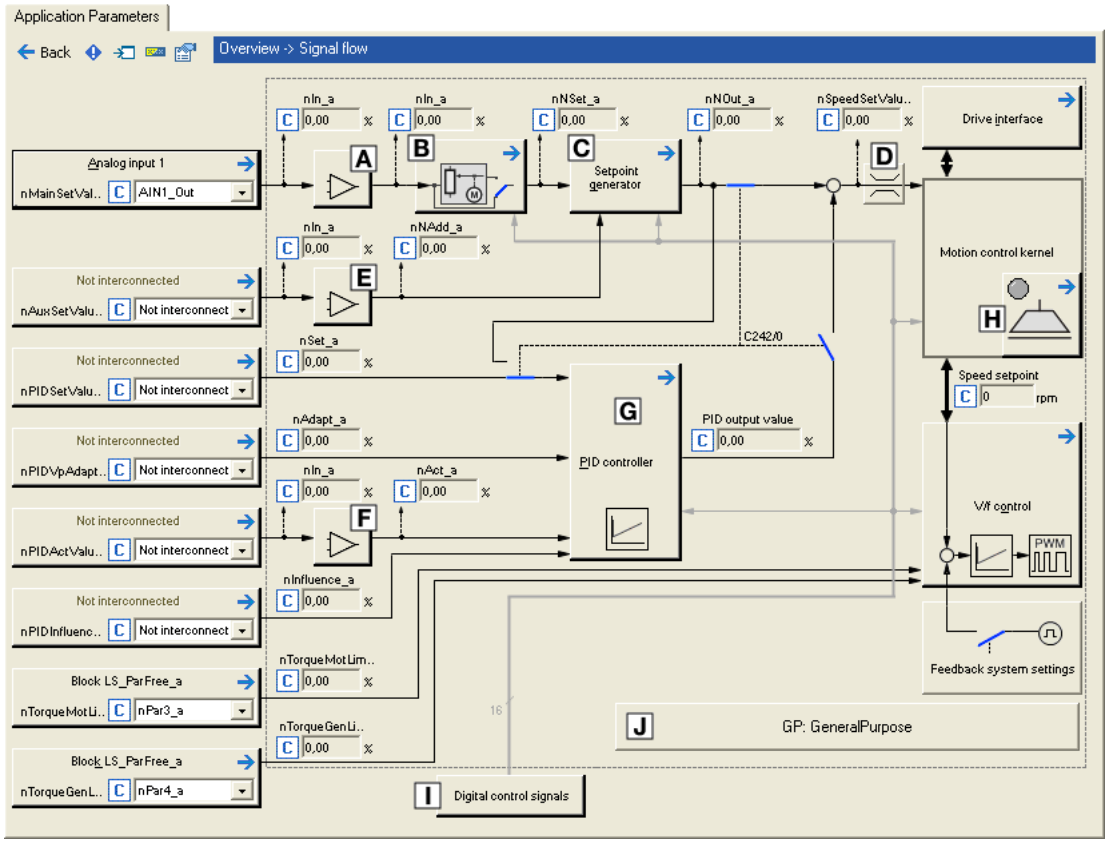
**Product features**

- Pre-configured control modes for terminals and bus control (with predefined process data connection to the fieldbus)
- Free configuration of input and output signals
- Offset, gain, and negation of main setpoint, additional setpoint, actual process controller value
- Up to 15 fixed setpoints for speed and ramp time
- Adjustable setpoint ramp times
- Freely selectable, variable ramp shape
- Automatic holding brake control
- Quick stop (QSP) with adjustable ramp time
- Motor potentiometer function
- Process controller
- Load monitoring *(in preparation)*
- Integrated, freely available "GeneralPurpose" functions:
  Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop
- Interface to the safety module (optional)
- Integration of encoder feedback

**Related topics:**

▸ Commissioning of the "Actuating drive speed" technology application (📖 62)

## 8.2.1 Basic signal flow




[8-1] Signal flow of the technology application "Actuating drive speed"

- A Main speed setpoint offset and gain (L_OffsetGainP_1)
- B Motor potentiometer function (L_MPot_1)
- C Setpoint generator (L_NSet_1)
- D Speed setpoint input limitation
- E Additional speed setpoint offset and gain (L_OffsetGainP_2)
- F Actual speed/sensor value offset and gain (L_OffsetGainP_3)
- G Process controller (L_PCTRL_1)
- H Holding brake control
- I Terminal assignment & display of digital control signals
- J Integrated disposable "GeneralPurpose" functions: Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop

### Selection of the main speed setpoint

The main speed setpoint is selected in the Lenze setting via the analog input 1.

- Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal adjustment of a setpoint encoder.
- Scaling: 16384 ≡ 100 % reference speed (C00011)
- The main setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.
- Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.
- For a detailed functional description see the L_NSet FB.

### Motor potentiometer function

Alternatively, the main speed setpoint can be generated via a motor potentiometer function.

- In the Lenze setting, the motor potentiometer function is deactivated.
- Activation is possible via C00806 or via the *bMPotEnable* input.
- The behaviour of the motor potentiometer during switch-on of the drive system can be selected in C00805.
- For a detailed functional description see the L_MPot FB.

### Optional selection of an additional speed setpoint

You can optionally select an additional speed setpoint (e.g. as a correcting signal).

- The additional speed setpoint can be linked arithmetically with the main speed setpoint behind the ramp function generator.
- You must set the setpoint arithmetic to "1: NOut = NSet + NAdd" in C00190 in order to activate the additional speed setpoint.
- Offset and gain of this input signal can be set in C00697 and C00671 for a simple signal adjustment of a setpoint encoder.
- Scaling: 16384 ≡ 100 % reference speed (C00011)
- The acceleration and deceleration time for the additional speed setpoint can be set in C00220 and C00221.
- For a detailed functional description see the L_NSet FB.

**Tip!**

In the case of a grinding machine, the additional speed setpoint can, for instance, be used to control a constant circumferential speed while the grinding disk diameter is reduced.

### 8.2.2 Internal interfaces | application block "LA_NCtrl"

**Note!**

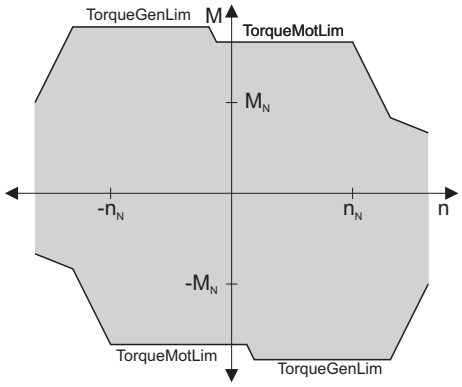
The connectors greyed out in the following table are hidden in the function block editor in the Lenze setting.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wCANDriveControl<br>WORD | Control word via system bus (CAN) for device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. | |
| wMCIDriveControl<br>WORD | Control word via communication module (e.g. PROFIBUS) for device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. | |
| wSMControl<br>WORD | Interface to the optional safety system.<br>• Setting control bit 0 ("SafeStop1") in this control word causes e.g. the automatic deceleration of the drive to standstill within this application (in the **Motion Control Kernel**).<br>• See the subchapter "Interface to safety system" of the chapter on basic drive functions for a detailed description of the individual control bits. | |
| bCInh<br>BOOL | Enable/inhibit inverter | |
| | FALSE | Enable inverter: The inverter switches to the "OperationEnabled" device status if no other source for controller inhibit is active.<br>• C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit. |
| | TRUE | Inhibit inverter (controller inhibit): The inverter switches to the "SwitchedOn" device status. |
| bFailReset<br>BOOL | Reset error message<br>In the Lenze setting this input is connected to the digital input controller enable so that a possibly existing error message is reset together with the controller enable (if the cause for the fault is eliminated). | |
| | TRUE | The current fault is reset, if the cause for the fault is eliminated.<br>• If the fault still exists, the error status remains unchanged. |
| bSetQuickstop<br>BOOL | Activate quick stop (QSP)<br>• Also see device command "Activate/deactivate quick stop". | |
| | TRUE | Activate quick stop<br>• Motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0).<br>• The motor is kept at a standstill during closed-loop operation.<br>• A pulse inhibit is set if the auto-DCB function has been activated via C00019. |
| | FALSE | Deactivate quick stop<br>• The quick stop is deactivated if no other source for the quick stop is active.<br>• C00159 displays a bit code of active sources/causes for the quick stop. |

| Designator<br>Data type | Information/possible settings |
|---|---|
| bSetDCBrake<br>BOOL | Manual DC-injection braking (DCB)<br>• Detailed information on DC-injection braking is provided in the motor control chapter, subchapter "DC-injection braking". |
| | **Note!**<br>Holding braking is not possible when this braking mode is used!<br>Use the basic "Holding brake control" function for controlling the holding brake with a low rate of wear. |
| | FALSE: Deactivate DC-injection braking. |
| | TRUE: Activate DC-injection braking, i.e. the drive is brought to a standstill by means of DC-injection braking.<br>• The braking effect stops when the rotor is at standstill.<br>• After the hold time (C00107) has expired, the controller sets the pulse inhibit. |
| bRFG_Stop<br>BOOL | Ramp function generator: Maintain the current value of the main setpoint integrator<br>• The speed, for instance, of a running ramp process is immediately kept constant when *bRFG_Stop* is activated. At the same time, the acceleration/deceleration jumps to the value "0".<br>• For a detailed functional description see the L_NSet FB. |
| | TRUE: The current value of the main setpoint integrator is held. |
| bRFG_0<br>BOOL | Ramp function generator: Lead the main setpoint integrator to "0" within the current Ti times<br>• For a detailed functional description see the L_NSet FB. |
| | TRUE: The current value of the main setpoint integrator is led to "0" within the Ti time set. |
| nVoltageAdd_a<br>INT | Additive voltage impression<br>• An additional setpoint for the motor voltage can be specified via this process input.<br>• If there are, for instance, different loads at the motor output end, it is possible to apply a voltage boost at the starting time.<br>• If the value is negative, the voltage is reduced.<br>• Scaling: 16384 ≡ 1000 V |
| | **Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! |
| nBoost_a<br>INT | Additional setpoint for the motor voltage at speed = 0<br>• The entire voltage-frequency characteristic is provided with an offset.<br>• Scaling: 16384 ≡ 1000 V |
| | **Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! |
| nPWMAngleOffset<br>INT | Additional offset for the electrical angle of rotation<br>• If a torque is connected, e.g. dynamic acceleration processes can be generated.<br>• Scaling: ±32767 ≡ ±180 ° angle of rotation |

| Designator<br>Data type | Information/possible settings |
|---|---|
| nTorqueMotLim_a<br>nTorqueGenLim_a<br>INT | Torque limitation in motor mode and in generator mode<br>• These input signals are directly transferred to the motor control to limit the inverter's maximum torque in motor and generator mode.<br>• The drive cannot output a higher torque in motor/generator mode than set here.<br>• The applied values (any polarity) are internally interpreted as absolute values.<br>• If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.<br>• If sensorless vector control (SLVC) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| | Torque limits in motor and generator mode:<br>  <br> |
| bSetSpeedCcw<br>BOOL | Change of direction of rotation<br>• For instance if a motor or gearbox is fixed laterally reversed to a machine part, but the setpoint selection should still be executed for the positive direction of rotation. |
| | FALSE: Clockwise rotation (Cw) |
| | TRUE: Direction of rotation to the left (Ccw) |
| bRLQCw<br>BOOL | Activate clockwise rotation (fail-safe)<br>• For a detailed functional description see the L_RLQ FB. |
| | FALSE: Quick stop |
| | TRUE: CW rotation |
| bRLQCcw<br>BOOL | Activate counter-clockwise rotation (fail-safe)<br>• For a detailed functional description see the L_RLQ FB. |
| | FALSE: Quick stop |
| | TRUE: Counter-clockwise rotation |
| nMainSetValue_a<br>INT | Main speed setpoint<br>• Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The main setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.<br>• Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.<br>• For a detailed functional description see the L_NSet FB. |

| Designator<br>Data type | Information/possible settings |
|---|---|
| nAuxSetValue_a<br>INT | Additional speed setpoint<br>• The additional speed setpoint can be linked arithmetically with the main speed setpoint behind the ramp function generator.<br>• You must set the setpoint arithmetic to "1: NOut = NSet + NAdd" in C00190 in order to activate the additional speed setpoint.<br>• Offset and gain of this input signal can be set in C00697 and C00671 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The acceleration and deceleration time for the additional speed setpoint can be set in C00220 and C00221.<br>• For a detailed functional description see the L_NSet FB. |
| bJogSpeed1<br>bJogSpeed2<br>BOOL | Inputs for overriding fixed setpoints (JOG setpoints) for the main setpoint<br>• A fixed setpoint for the setpoint generator can be activated instead of the main setpoint via these selection inputs.<br>• The four selection inputs are binary coded, therefore 15 fixed setpoints can be selected.<br>• In the case of binary coded selection "0" (all inputs = FALSE or not assigned), main setpoint *nMainSetValue_a* is active.<br>• The selection of the fixed setpoints is carried out in C00039/1...15 in [%] based on the reference speed (C00011).<br>• For a detailed functional description see the L_NSet FB. |
| bJogSpeed4<br>bJogSpeed8<br>BOOL | |
| bJogRamp1<br>bJogRamp2<br>BOOL | Selection inputs for alternative acceleration/deceleration times for the main setpoint<br>• The four selection inputs are binary coded, therefore 15 alternative acceleration/deceleration times can be selected.<br>• For main setpoint *nMainSetValue_a*, the set acceleration time (C00012) and deceleration time (C00013) are active in the case of the binary coded selection "0" (all inputs = FALSE or not assigned).<br>• Alternative acceleration times are selected in C00101/1...15.<br>• The selection of the alternative deceleration times is carried out in C00103/1...15.<br>• For a detailed functional description see the L_NSet FB. |
| bJogRamp4<br>bJogRamp8<br>BOOL | |
| **Motor potentiometer**<br>Alternatively to the input signal *nMainSetValue_a*, the main setpoint can also be generated by a motor potentiometer function.<br>• In the Lenze setting, the motor potentiometer function is deactivated.<br>• Activation is possible via C00806 or via the *bMPotEnable* input.<br>• The behaviour of the motor potentiometer during switch-on of the drive system can be selected in C00805.<br>• For a detailed functional description see the L_MPot FB. | |
| bMPotEnable<br>BOOL | Activating the motor potentiometer function<br>• This input and C00806 are OR'd. |
| | TRUE: The motor potentiometer function is active; the speed setpoint can be changed via the *bMPotUp* and *bMPotDown* control inputs. |
| bMPotUp<br>BOOL | Increasing the speed setpoint |
| | TRUE: Approach the upper speed limit value set in C00800 with the acceleration time set in C00802. |
| bMPotInAct<br>BOOL | Activating the inactive function |
| | TRUE: The speed setpoint behaves according to the inactive function set in C00804.<br>• In the Lenze setting, the speed setpoint is maintained. |
| bMPotDown<br>BOOL | Decreasing the speed setpoint |
| | TRUE: Approach the lower speed limit value set In C00801 with the deceleration time set in C00803. |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| **Process controller**<br>• In the Lenze setting, the process controller is deactivated.<br>• The activation is executed by selecting the operating mode in C00242.<br>• For a detailed functional description see FB L_PCTRL. | | |
| bPIDEnableInfluenceRamp<br>BOOL | Activate ramp for influencing factor | |
| | FALSE | Influencing factor of the PID controller is ramped down to "0". |
| | TRUE | Influencing factor of the PID controller is ramped up to the value *nPIDInfluence_a*. |
| bPIDIOff<br>BOOL | Switch off the I-component of the process controller<br>• In conjunction with the operating mode set in C00242 (Lenze setting: "Off"). | |
| | TRUE | I-component of the process controller is switched off. |
| nPIDVpAdapt_a<br>INT | Adaptation of gain Vp set in C00222 in percent<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Changes can be done online. | |
| nPIDSetValue_a<br>INT | Sensor and process setpoint for operating modes 2, 4 and 5<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |
| nPIDActValue_a<br>INT | Speed or actual sensor value (actual process value)<br>• Offset and gain for this input signal can be set in C00698 and C00672.<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |
| nPIDInfluence_a<br>INT | Limitation of the influencing factor in percent<br>• The influence factor of the PID controller can be limited to a certain value (-199.99% ... + 199.99%) via *nPIDInfluence_a*.<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |
| **MCK basic functions** | | |
| bMBrakeRelease<br>BOOL | Holding brake control: Release/apply brake<br>• In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off"). | |
| | FALSE | Apply brake.<br>• During automatic operation, the internal brake logic controls the brake. |
| | TRUE | Release brake manually (forced release).<br>• **Note!** The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If a controller inhibit has been set by the brake control, it will be deactivated.<br>• In semi-automatic operation, the brake is released including feedforward control. |

| Designator<br>Data type | Information/possible settings |
|---|---|
| **GP: GeneralPurpose**<br>The following inputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | |
| bGPFree1 ... bGPFree2<br>BOOL<br>(from version 12.00.00) | Free inputs for digital signals<br>• Digital signals can be transferred from the I/O level to the application level via these inputs. |
| nGPAnalogSwitchIn1_a<br>nGPAnalogSwitchIn2_a<br>INT | Analog switch: Input signals<br>• The input signal selected via the selection input *bGPAnalogSwitchSet* is output at output *nGPAnalogSwitchOut_a*. |
| bGPAnalogSwitchSet<br>BOOL | Analog switch: Selection input<br>FALSE: *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn1_a*<br>TRUE: *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn2_a* |
| nGPArithmetikIn1_a<br>nGPArithmetikIn2_a<br>INT | Arithmetic: Input signals<br>• The arithmetic function is selected in C00338.<br>• The result is output at output *nGPArithmetikOut_a*. |
| nGPMulDivIn_a<br>INT | Multiplication/Division: Input signal<br>• The factor for the multiplication can be set in C00699/1 (numerator) and C00699/2 (denominator).<br>• The result is output at output *nGPMulDivOut_a*. |
| bGPDigitalDelayIn<br>BOOL | Binary delay element: Input signal<br>• The on-delay can be set in C00720/1.<br>• The off-delay can be set in C00720/2.<br>• The time-delayed input signal is output at output *bGPDigitalDelayOut*. |
| bGPLogicIn1<br>bGPLogicIn2<br>bGPLogicIn3<br>BOOL | Binary logic: Input signals<br>• The logic operation is selected in C00820.<br>• The result is output at output *bGPLogicOut*. |
| nGPCompareIn1_a<br>nGPCompareIn2_a<br>INT | Analog comparison: Input signals<br>• The comparison operation is selected in C00680.<br>• Hysteresis and window size can be set in C00680 and C00682.<br>• If the comparison statement is true, the output *bGPCompareOut* will be set to TRUE. |
| bGPDFlipFlop_InD<br>bGPDFlipFlop_InClk<br>bGPDFlipFlop_InClr<br>BOOL | D-FlipFlop: Input signals<br>• Data, clock and reset input |
| **Free inputs**<br>The following inputs can freely be interconnected on the application level.<br>The signals can be transferred from the I/O level to the application level via these inputs. | |
| bFreeIn1 ... bFreeIn8<br>BOOL | Free inputs for digital signals |
| wFreeIn1 ... wFreeIn4<br>WORD | Free inputs for 16-bit signals |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| wDriveControlStatus<br>WORD | Status word of the inverter<br>• The status word contains information on the currents status of the inverter.<br>• See the "wDeviceStatusWord status word" subchapter of the chapter on device control for a detailed description of the bit assignment. | |
| wStateDetermFailNoLow<br>WORD | Display of the status determining error (LOW word) | |
| wStateDetermFailNoHigh<br>WORD | Display of the status determining error (HIGH word) | |
| bDriveFail<br>BOOL | TRUE | Inverter in error status.<br>• "Fault" device status is active. |
| bDriveReady<br>BOOL | TRUE | Inverter is ready for operation<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the inverter is still inhibited by the user (controller inhibit). |
| bCInhActive<br>BOOL | TRUE | Controller inhibit is active. |
| bQSPIsActive<br>BOOL | TRUE | Quick stop is active. |
| bSpeedCcw<br>BOOL | Current direction of rotation | |
| | FALSE | Clockwise rotation (Cw) |
| | TRUE | Direction of rotation to the left (Ccw) |
| bSpeedActCompare<br>BOOL | Result of the speed comparison (detection of speed=0) | |
| | TRUE | During open-loop operation:<br>Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>Actual speed value < Comparison value (C00024) |
| bOverLoadActive<br>BOOL | In preparation (output is not interconnected on the application level) | |
| bUnderLoadActive<br>BOOL | In preparation (output is not interconnected on the application level) | |
| bImaxActive<br>BOOL | "Current setpoint inside the limitation" status signal | |
| | TRUE | The current setpoint is internally limited (the inverter operates at the maximum current limit). |
| bSpeedSetReached<br>BOOL | Status signal "setpoint = 0" | |
| | TRUE | Speed setpoint from the ramp function generator = 0 |
| bSpeedActEqSet<br>BOOL | TRUE | Actual speed value = speed setpoint |
| nMotorCurrent_a<br>INT | Current stator current/effective motor current<br>• Scaling: 16384 ≡ 100 % $I_{max_mot}$ (C00022) | |
| nMotorSpeedSet_a<br>INT | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorSpeedAct_a<br>INT | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorTorqueAct_a<br>INT | Actual torque<br>• In the "VFC (+encoder)" operating mode of the motor control, this value is determined from the current motor current and corresponds to the actual torque only by approximation.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |

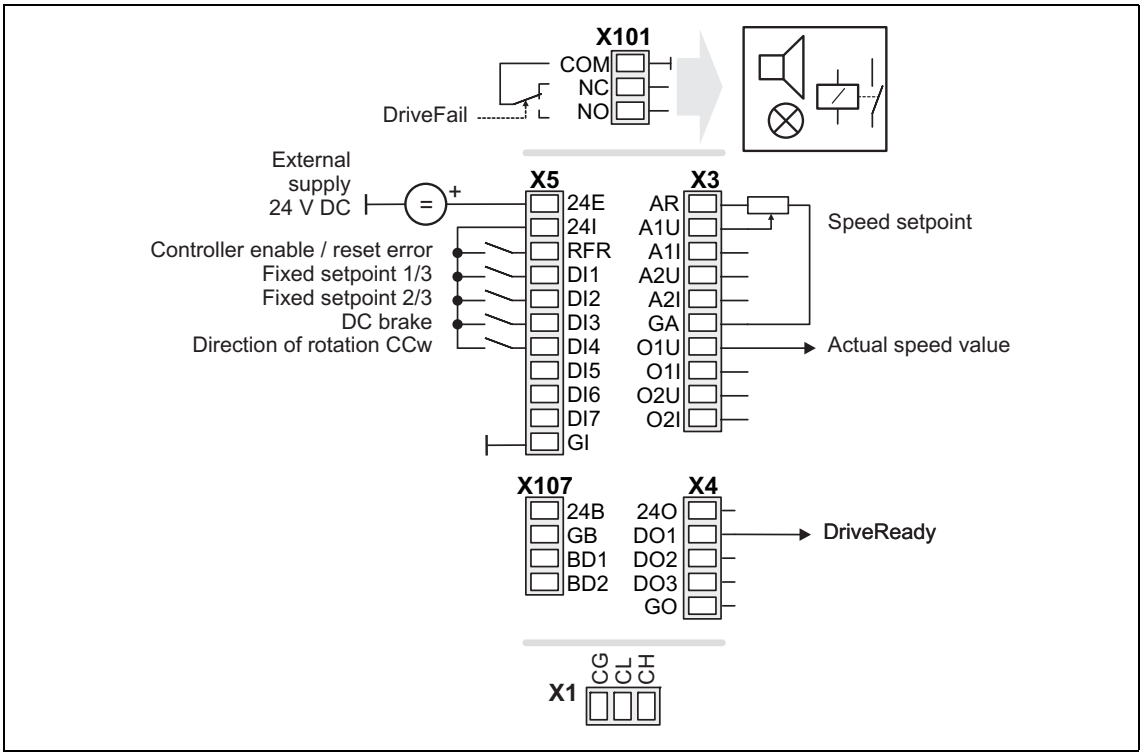
| Designator<br>Data type | Value/meaning |
|---|---|
| nDCVoltage_a<br>INT | Actual DC-bus voltage<br>• Scaling: 16384 ≡ 1000 V |
| nMotorVoltage_a<br>INT | Current motor voltage/inverter output voltage<br>• Scaling: 16384 ≡ 1000 V |
| **MCK basic functions** | |
| bMBrakeReleaseOut<br>BOOL | Holding brake control: Trigger signal for the holding brake control switching element via a digital output<br>• Use bit 0 in C02582 to activate inverted switching element triggering. |
| | FALSE: Apply brake. |
| | TRUE: Release brake. |
| bMBrakeReleased<br>BOOL | Holding brake control: "Brake released" considering the brake release time<br>• When the holding brake is triggered to close, *bMBrakeReleased* is immediately set to FALSE even if the brake closing time has not yet elapsed! |
| | TRUE: Brake released (after the brake release time has expired). |
| **GP: GeneralPurpose**<br>The following outputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | |
| nGPAnalogSwitchInOut_a<br>INT | Analog switch: Output signal |
| nGPArithmetikOut_a<br>INT | Arithmetic: Output signal |
| nGPMulDivOut_a<br>INT | Multiplication/Division: Output signal |
| bGPDigitalDelayOut<br>BOOL | Binary delay element: Output signal |
| bGPLogicOut<br>BOOL | Binary logic: Output signal |
| bGPCompareOut<br>BOOL | Analog comparison: Output signal |
| bGPSignalOut1<br>...<br>bGPSignalOut4<br>BOOL | Binary signal monitor: Output signals<br>• The signal sources to be output are selected in C00411/1...4.<br>• A bit coded inversion of the output signals can be parameterised in C00412. |
| nGPSignalOut1_a<br>...<br>nGPSignalOut4_a<br>BOOL | Analog signal monitor: Output signals<br>• The signal sources to be output are selected in C00410/1...4.<br>• Gain and offset for each output signal can be parameterised in C00413/1...8. |
| bGPDFlipFlop_Out<br>BOOL | D-FlipFlop: Output signal |
| bGPDFlipFlop_NegOut<br>BOOL | D-FlipFlop: Negated output signal |
| **Free outputs**<br>The following outputs can freely be interconnected on the application level.<br>The signals from the application level can be transferred to the I/O level via these outputs. | |
| bFreeOut1 ... bFreeOut8<br>BOOL | Free outputs for digital signals |
| wFreeOut1 ... wFreeOut4<br>WORD | Free outputs for 16-bit signals |

## 8.2.3 Terminal assignment of the control modes

The following comparison provides information about which inputs/outputs of the application block **LA_NCtrl** are interconnected to the digital and analog input/output terminals of the inverter in the different control modes.

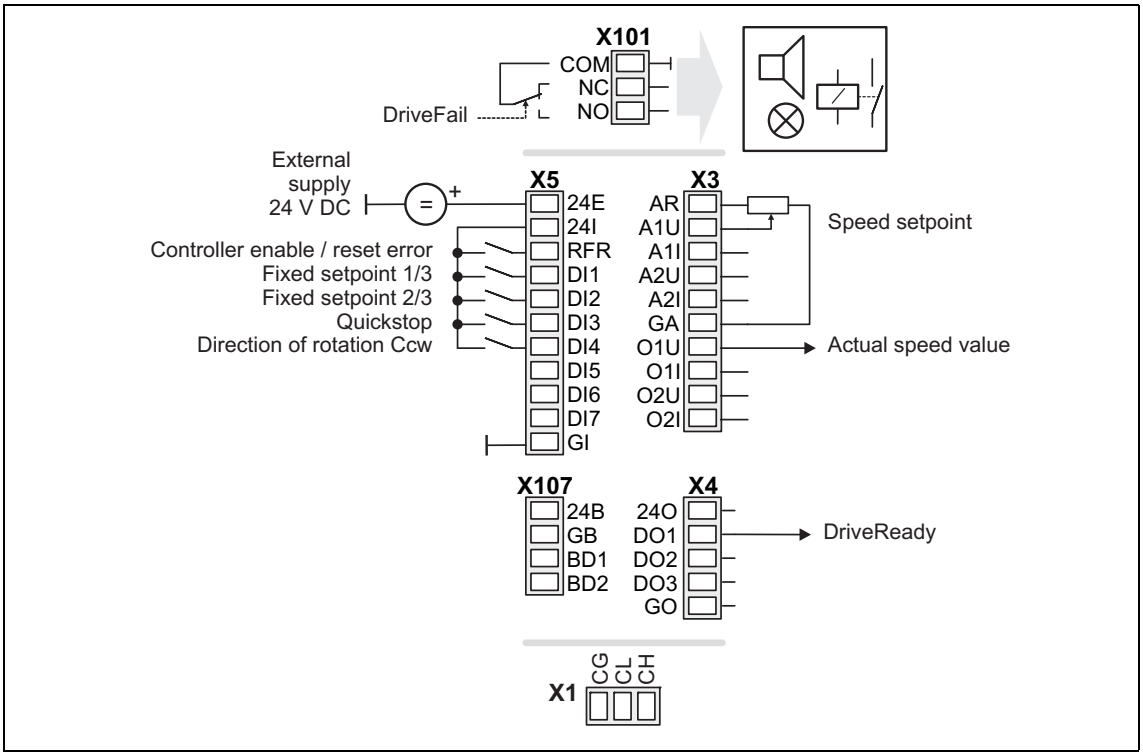
| | Control mode (C00007) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 10: Terminals 0 | 12: Terminals 2 | 14: Terminals 11 | 16: Terminal 16 | 20: Keypad | 21: PC | 30: CAN | 40: MCI |
| **Digital input terminals** | | | | | | | | |
| **X5/RFR** | **Controller enable / Reset of error message**<br>bFailReset | | | | | | | |
| **X5/DI1** | **Fixed setpoint 1/3**<br>bJogSpeed1 | | **Change of direction of rotation**<br>bSetSpeedCcw | **Fixed setpoint 1/3**<br>bJogSpeed1 | - | - | **Quick stop**<br>bSetQuickstop | |
| **X5/DI2** | **Fixed setpoint 2/3**<br>bJogSpeed2 | | **Activate manual DC-injection braking (DCB)**<br>bSetDCBrake | **Fixed setpoint 2/3**<br>bJogSpeed2 | - | - | - | - |
| **X5/DI3** | **Activate manual DC-injection braking (DCB)**<br>bSetDCBrake | **Quick stop**<br>bSetQuickstop | **Motor potentiometer: Increase speed**<br>bMPotUp | **CW rotation quick stop**<br>bRLQCw | - | - | - | - |
| **X5/DI4** | **Change of direction of rotation**<br>bSetSpeedCcw | | **Motor potentiometer: Decrease speed**<br>bMPotDown | **CCW rotation quick stop**<br>bRLQCcw | - | - | - | - |
| **X5/DI5 ... DI7** | - | - | - | - | - | - | - | - |
| **Analog input terminals** | | | | | | | | |
| **X3/A1U, A1I** | **Main speed setpoint**<br>nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) | | | | - | - | **Additional speed setpoint**<br>nAuxSetValue_a<br>10 V ≡ 100 % reference speed (C00011) | |
| **X3/A2U, A2I** | - | - | - | - | - | - | - | - |
| **Digital output terminals** | | | | | | | | |
| **X4/DO1** | **Status "Drive is ready"**<br>bDriveReady | | | | | | | |
| **X4/DO2 ... DO3** | - | - | - | - | - | - | - | - |
| **X107/BD1, BD2** | - | - | - | - | - | - | - | - |
| **X101/COM, NO** | **Status "Error is pending"**<br>bDriveFail | | | | | | | |
| **Analog output terminals** | | | | | | | | |
| **X3/O1U** | **Actual speed value**<br>nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) | | | | | | | |
| **X3/O1I** | - | - | - | - | - | - | - | - |
| **X3/O2U** | | | | | | | | |
| **X3/O2I** | | | | | | | | |

### 8.2.3.1 Terminals 0



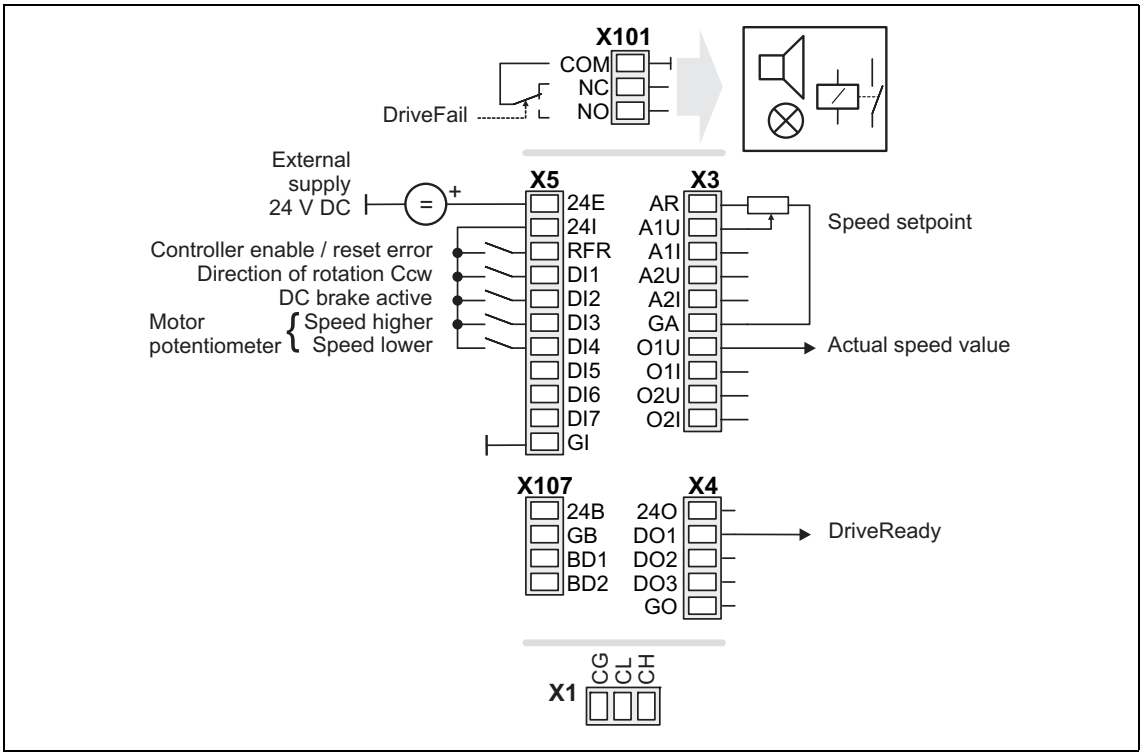

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | LA_NCtrl.nMainSetValue_a * |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 | | **X3/A1I** | - |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 | | **X3/A2U** | - |
| **X5/DI3** | LA_NCtrl.bSetDCBrake | | **X3/A2I** | - |
| **X5/DI4** | LA_NCtrl.bSetSpeedCcw | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.2.3.2 Terminals 2



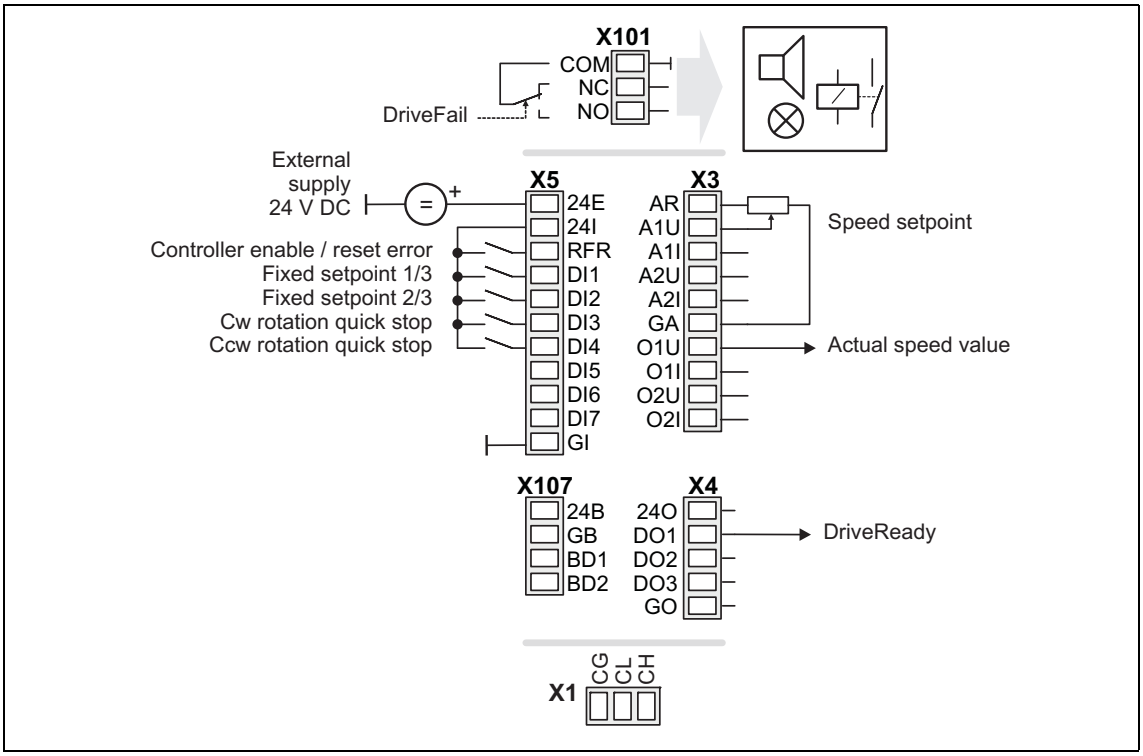

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | LA_NCtrl.nMainSetValue_a * |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 | | **X3/A1I** | - |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 | | **X3/A2U** | - |
| **X5/DI3** | LA_NCtrl.bSetQuickstop | | **X3/A2I** | - |
| **X5/DI4** | LA_NCtrl.bSetSpeedCcw | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.2.3.3 Terminals 11



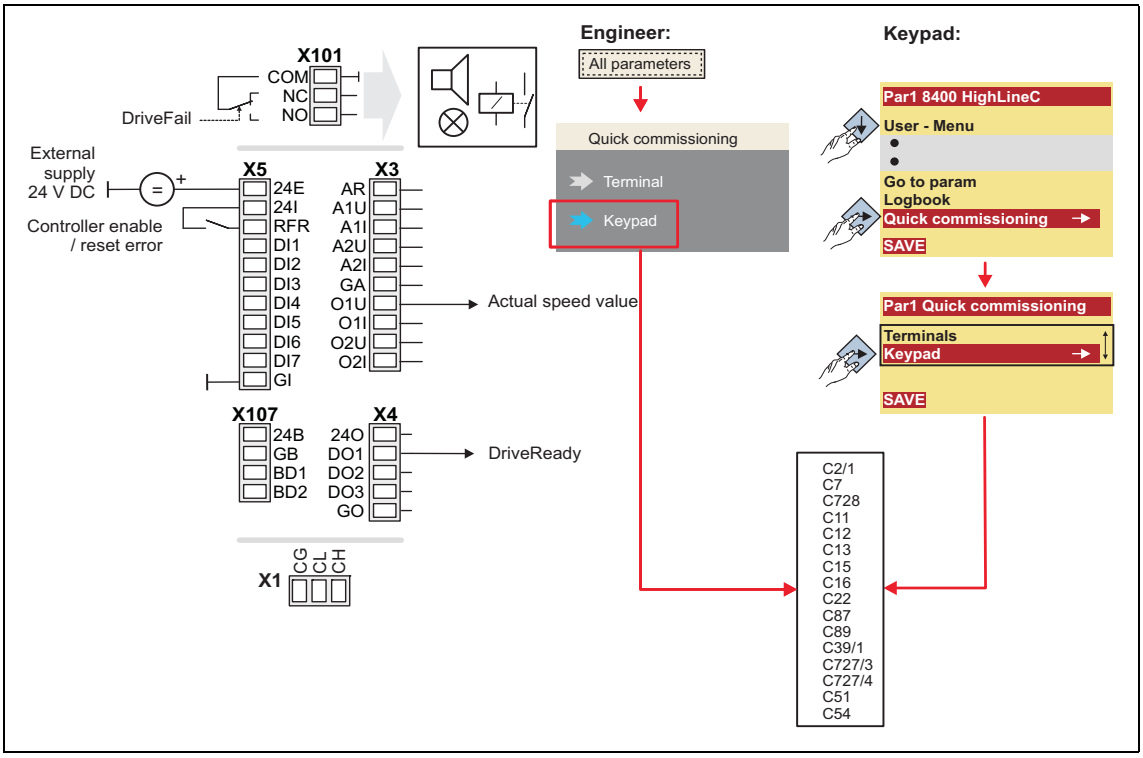

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | LA_NCtrl.nMainSetValue_a * |
| **X5/DI1** | LA_NCtrl.bSetSpeedCcw | | **X3/A1I** | - |
| **X5/DI2** | LA_NCtrl.bSetDCBrake | | **X3/A2U** | - |
| **X5/DI3** | LA_NCtrl.bMPotUp | | **X3/A2I** | - |
| **X5/DI4** | LA_NCtrl.bMPotDown | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.2.3.4 Terminal 16



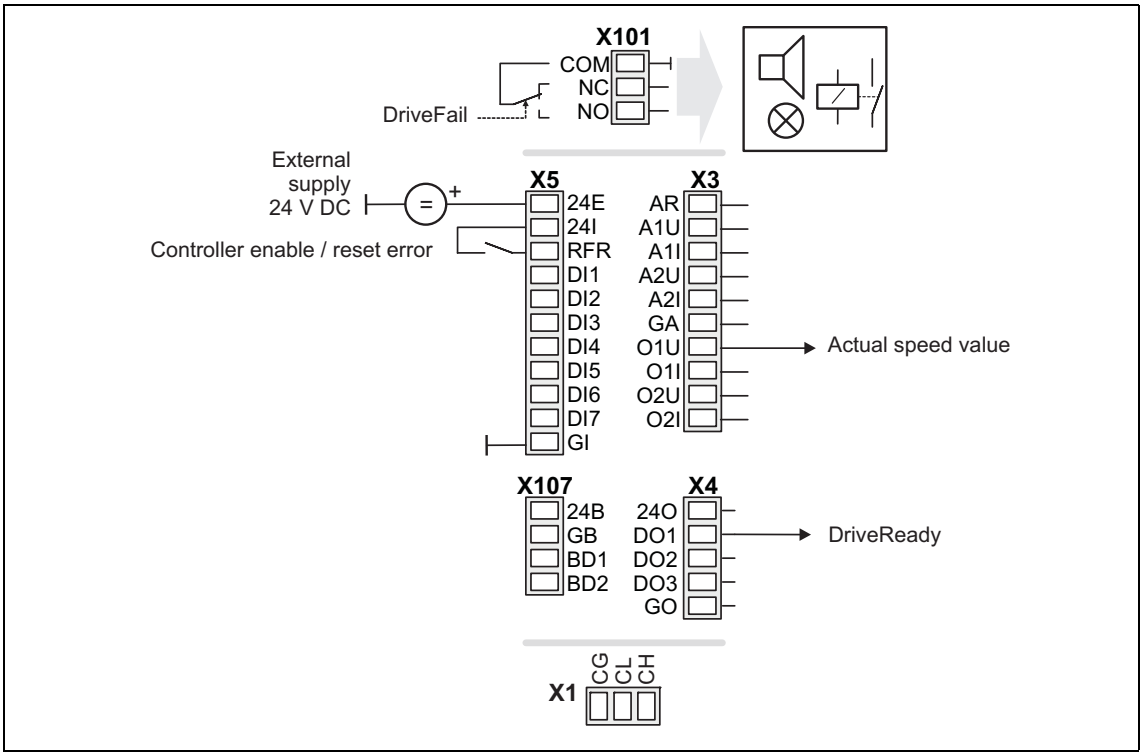

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | LA_NCtrl.nMainSetValue_a * |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 | | **X3/A1I** | - |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 | | **X3/A2U** | - |
| **X5/DI3** | LA_NCtrl. bRLQCw | | **X3/A2I** | - |
| **X5/DI4** | LA_NCtrl. bRLQCcw | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.2.3.5 Keypad



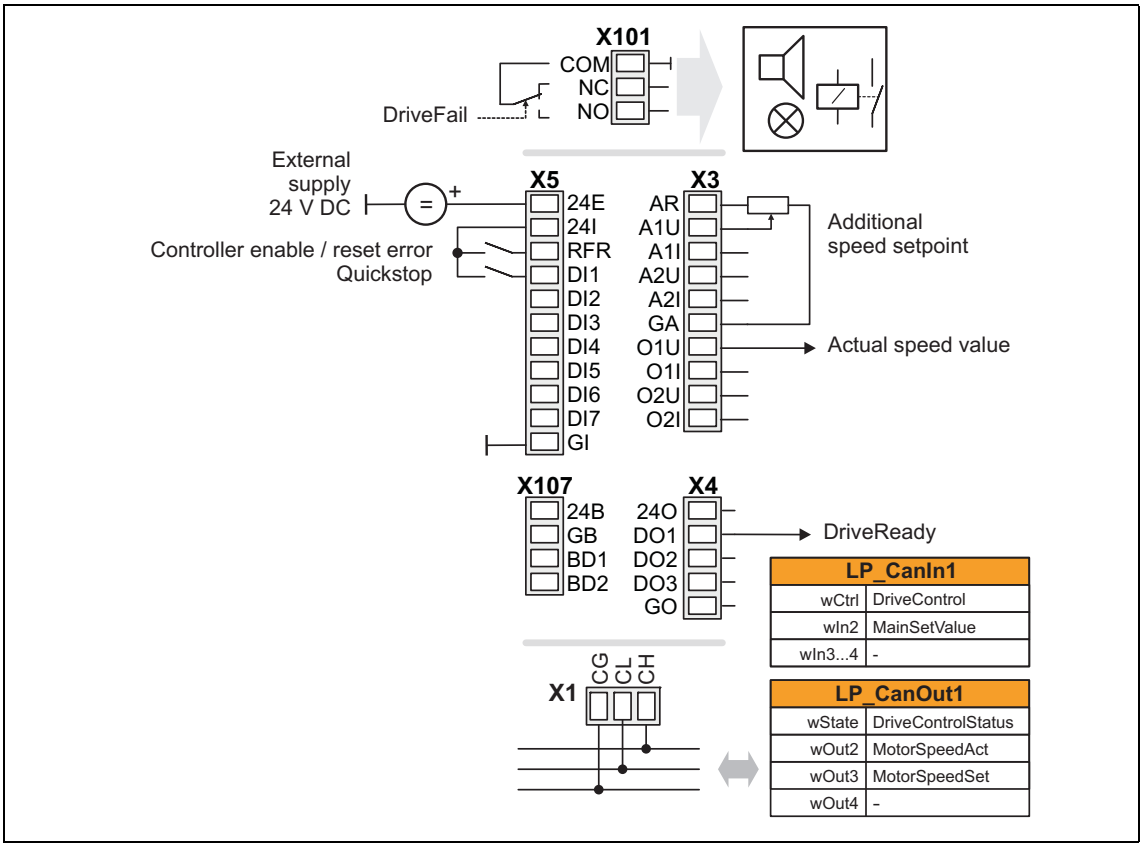

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | - |
| **X5/DI1** | - | | **X3/A1I** | - |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | - | | **X3/A2I** | - |
| **X5/DI4** | - | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.2.3.6 PC




| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | - |
| **X5/DI1** | - | | **X3/A1I** | - |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | - | | **X3/A2I** | - |
| **X5/DI4** | - | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.2.3.7 CAN



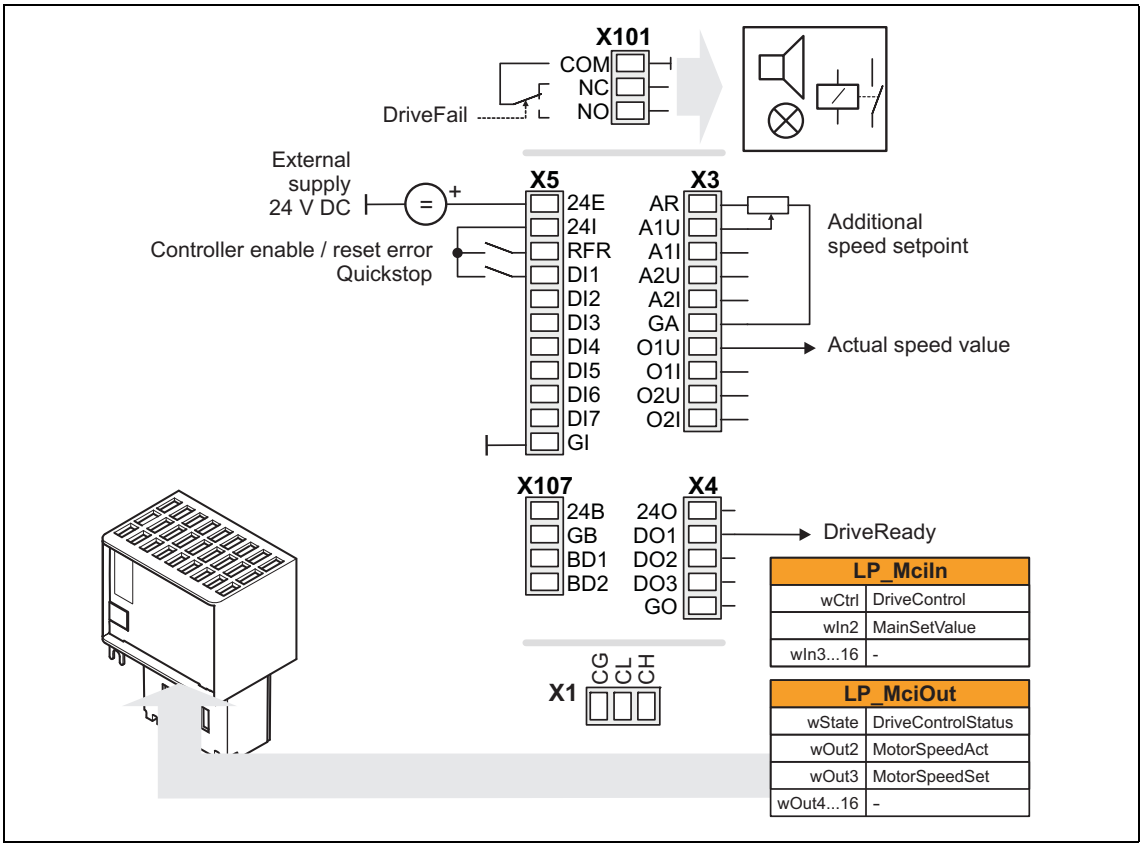

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| X101/NC-NO | LA_NCtrl.bDriveFail | | | |
| | | | | |
| X5/RFR | LA_NCtrl.bFailReset | | X3/A1U | LA_NCtrl.nAuxSetValue_a * |
| X5/DI1 | LA_NCtrl.bSetQuickStop | | X3/A1I | - |
| X5/DI2 | - | | X3/A2U | - |
| X5/DI3 | - | | X3/A2I | - |
| X5/DI4 | - | | X3/O1U | LA_NCtrl.NMotorSpeedAct_a * |
| X5/DI5 | - | | X3/O1I | - |
| X5/DI6 | - | | X3/O2U | - |
| X5/DI7 | - | | X3/O2I | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| X107/BD1 | - | | X4/DO1 | LA_NCtrl.bDriveReady |
| X107/BD2 | - | | X4/DO2 | - |
| | | | X4/DO3 | - |

▸ Process data assignment for fieldbus communication (📖 458)

**Note!**

- You must set the setpoint arithmetic in C00190 to "1: NOut = NSet + NAdd" so that the additional speed setpoint selected via the analog input A1U has an additive effect.
- The "manual jog" function via digital terminals is being prepared!

### 8.2.3.8 MCI




| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| X101/NC-NO | LA_NCtrl.bDriveFail | | | |
| | | | | |
| X5/RFR | LA_NCtrl.bFailReset | | X3/A1U | LA_NCtrl.nAuxSetValue_a * |
| X5/DI1 | LA_NCtrl.bSetQuickStop | | X3/A1I | - |
| X5/DI2 | - | | X3/A2U | - |
| X5/DI3 | - | | X3/A2I | - |
| X5/DI4 | - | | X3/O1U | LA_NCtrl.NMotorSpeedAct_a * |
| X5/DI5 | - | | X3/O1I | - |
| X5/DI6 | - | | X3/O2U | - |
| X5/DI7 | - | | X3/O2I | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| X107/BD1 | - | | X4/DO1 | LA_NCtrl.bDriveReady |
| X107/BD2 | - | | X4/DO2 | - |
| | | | X4/DO3 | - |

▸ Process data assignment for fieldbus communication (📖 458)

**Note!**

- You must set the setpoint arithmetic in C00190 to "1: NOut = NSet + NAdd" so that the additional speed setpoint selected via the analog input A1U has an additive effect.
- The "manual jog" function via digital terminals is being prepared!

### 8.2.4 Process data assignment for fieldbus communication

The fieldbus communication is connected (preconfigured) to the previously selected technology application by selecting the corresponding control mode in C00007:

- "30: CAN" for the connection to the system bus (CAN)
- "40: MCI" for the connection to a plugged-on communication module (e.g. PROFIBUS)

The assignment of the process data words depends only on the application, not on the bus system used:

| Input words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControl | Control word<br>• For bit assignment see the table below. |
| Word 2 | MainSetValue | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| Word 3 | - | Not preconfigured |
| Word 4 | - | Not preconfigured |
| Words 5 ... 16 | - | Not preconfigured<br>• Only available in control mode "40: MCI". |

| Control word | Name | Function |
|---|---|---|
| Bit 0 | SwitchOn | 1 ≡ Change to the "SwitchedOn" device status<br>• This bit must be set in the CAN/MCI control word to ensure that the device changes to the "SwitchedOn" device status after mains connection without the need for a master control specifying this bit via fieldbus.<br>• If control via a bus system is not wanted (e.g. in the case of control via terminals), the *wDriveCtrl* output signal of the LS_ParFix system block can be connected to the control word inputs. |
| Bit 1 | DisableVoltage | 1 ≡ Inhibit inverter control (pulse inhibit) |
| Bit 2 | SetQuickStop | 1 ≡ Activate quick stop (QSP).<br>▸ Activate/deactivate quick stop (📖 111) |
| Bit 3 | EnableOperation | 1 ≡ Enable inverter (RFR)<br>• If control via terminals is performed, this bit must be set both in the CAN control word and in the MCI control word. Otherwise, the controller is inhibited.<br>▸ Enable/inhibit inverter (📖 110) |
| Bit 4 | ModeSpecific_1 | Reserved (currently not assigned) |
| Bit 5 | ModeSpecific_2 | |
| Bit 6 | ModeSpecific_3 | |
| Bit 7 | ResetFault | 1 ≡ Reset fault (trip reset)<br>• Acknowledge error message (if the error cause has been eliminated).<br>▸ Reset error (📖 114) |
| Bit 8 | SetHalt | 1 ≡ Activate stop function<br>• Stop drive via stopping ramp (in preparation). |
| Bit 9 | reserved_1 | Reserved (currently not assigned) |
| Bit 10 | reserved_2 | |
| Bit 11 | SetDCBrake | 1 ≡ Activate DC-injection braking<br>▸ Manual DC-injection braking (DCB) (📖 275) |
| Bit 12 | JogSpeed1 | Activation of fixed speed 1 ... 3 |
| Bit 13 | JogSpeed2 | |

| Control word | Name | Function |
|---|---|---|
| Bit 14 | SetFail | 1 ≡ Set error (trip set) |
| Bit 15 | SetSpeedCcw | 0 ≡ Direction of rotation to the right (Cw)<br>1 ≡ Direction of rotation to the left (Ccw) |

| Output words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControlStatus | Status word<br>• For bit assignment see the table below. |
| Word 2 | MotorSpeedAct | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| Word 3 | MotorSpeedSet | Resulting overall setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| Word 4 | - | Not preconfigured |
| Words 5 ... 16 | - | Not preconfigured<br>• Only available in control mode "40: MCI". |

| Status word | Name | Status |
|---|---|---|
| Bit 0 | FreeStatusBit0 | Free status bit 0 (not assigned, freely assignable) |
| Bit 1 | PowerDisabled | 1 ≡ Inverter control inhibited (pulse inhibit is active) |
| Bit 2 | FreeStatusBit2 | Free status bit 2 (not assigned, freely assignable) |
| Bit 3 | FreeStatusBit3 | Free status bit 3 (not assigned, freely assignable) |
| Bit 4 | FreeStatusBit4 | Free status bit 4 (not assigned, freely assignable) |
| Bit 5 | FreeStatusBit5 | Free status bit 5 (not assigned, freely assignable) |
| Bit 6 | ActSpeedIsZero | During open-loop operation:<br>1 ≡ Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>1 ≡ Actual speed value < Comparison value (C00024) |
| Bit 7 | ControllerInhibit | 1 ≡ Inverter is inhibited (controller inhibit is active) |
| Bit 8 | StatusCodeBit0 | Bit coded display of the active device status<br>▸ Device state machine and device states (see table [4-1]) |
| Bit 9 | StatusCodeBit1 | |
| Bit 10 | StatusCodeBit2 | |
| Bit 11 | StatusCodeBit3 | |
| Bit 12 | Warning | 1 ≡ A warning is indicated |
| Bit 13 | Trouble | 1 ≡ Inverter is in the "Trouble" device status<br>• E.g. if an overvoltage has occurred. |
| Bit 14 | FreeStatusBit14 | Free status bit 14 (not assigned, freely assignable) |
| Bit 15 | FreeStatusBit15 | Free status bit 15 (not assigned, freely assignable) |

### 8.2.5 Setting parameters (short overview)

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00012 | Accel. time - main setpoint | 2.000 | s |
| C00013 | Decel. time - main setpoint | 2.000 | s |
| C00019 | Auto-DCB: Threshold | 3 | rpm |
| C00024 | LS_DriveInterface: bNActCompare | 0.00 | % |
| C00036 | DCB braking: Current | 50.00 | % |
| C00039/1 | Fixed setpoint 1 | 40.00 | % |
| C00039/2 | Fixed setpoint 2 | 60.00 | % |
| C00039/3 | Fixed setpoint 3 | 80.00 | % |
| C00039/4...15 | Fixed setpoint 4 ... 15 | 0.00 | % |
| C00101/1...15 | Add. accel. time 1 ... 15 | 0.000 | s |
| C00103/1...15 | Add. decel. time 1 ... 15 | 0.000 | s |
| C00105 | Decel. time - quick stop | 2.000 | s |
| C00106 | Auto-DCB: Hold time | 0.500 | s |
| C00107 | DCB braking: Hold time | 999.000 | s |
| C00134 | L_NSet_1: Ramp smoothing | 0: Off | |
| C00182 | L_NSet_1: S-ramp time PT1 | 20.00 | s |
| C00190 | L_NSet_1: Setpoint arithmetic | 0: Out = Set | |
| C00220 | L_NSet_1: Acceleration time - add. setpoint | 0.000 | s |
| C00221 | L_NSet_1: Deceleration time - add. setpoint | 0.000 | s |
| C00222 | L_PCTRL_1: Vp | 1.0 | |
| C00223 | L_PCTRL_1: Tn | 400 | ms |
| C00224 | L_PCTRL_1: Kd | 0.0 | |
| C00225 | L_PCTRL_1: MaxLimit | 199.99 | % |
| C00226 | L_PCTRL_1: MinLimit | -199.99 | % |
| C00227 | L_PCTRL_1: Acceleration time | 0.010 | s |
| C00228 | L_PCTRL_1: Deceleration time | 0.010 | s |
| C00233 | L_PCTRL_1: Root function | 0: Off | |
| C00241 | L_NSet_1: Hyst. NSet reached | 0.50 | % |
| C00242 | Operating mode of process controller | 0: Off | |
| C00243 | L_PCTRL_1: Accel. time influence | 5.000 | s |
| C00244 | L_PCTRL_1: Deceleration time influence | 5.000 | s |
| C00632/1 | L_NSet_1: Blocking speed 1 max | 0.00 | % |
| C00632/2 | L_NSet_1: Blocking speed 2 max | 0.00 | % |
| C00632/3 | L_NSet_1: Blocking speed 3 max | 0.00 | % |
| C00633/1 | L_NSet_1: Blocking speed 1 min | 0.00 | % |
| C00633/2 | L_NSet_1: Blocking speed 2 min | 0.00 | % |
| C00633/3 | L_NSet_1: Blocking speed 3 min | 0.00 | % |
| C00635 | L_NSet_1: nMaxLimit | 199.99 | % |
| C00636 | L_NSet_1: nMinLimit | -199.99 | % |
| C00670 | L_OffsetGainP_1: Gain | 1.0000 | |
| C00671 | L_OffsetGainP_2: Gain | 1.0000 | |
| C00672 | L_OffsetGainP_3: Gain | 1.0000 | |

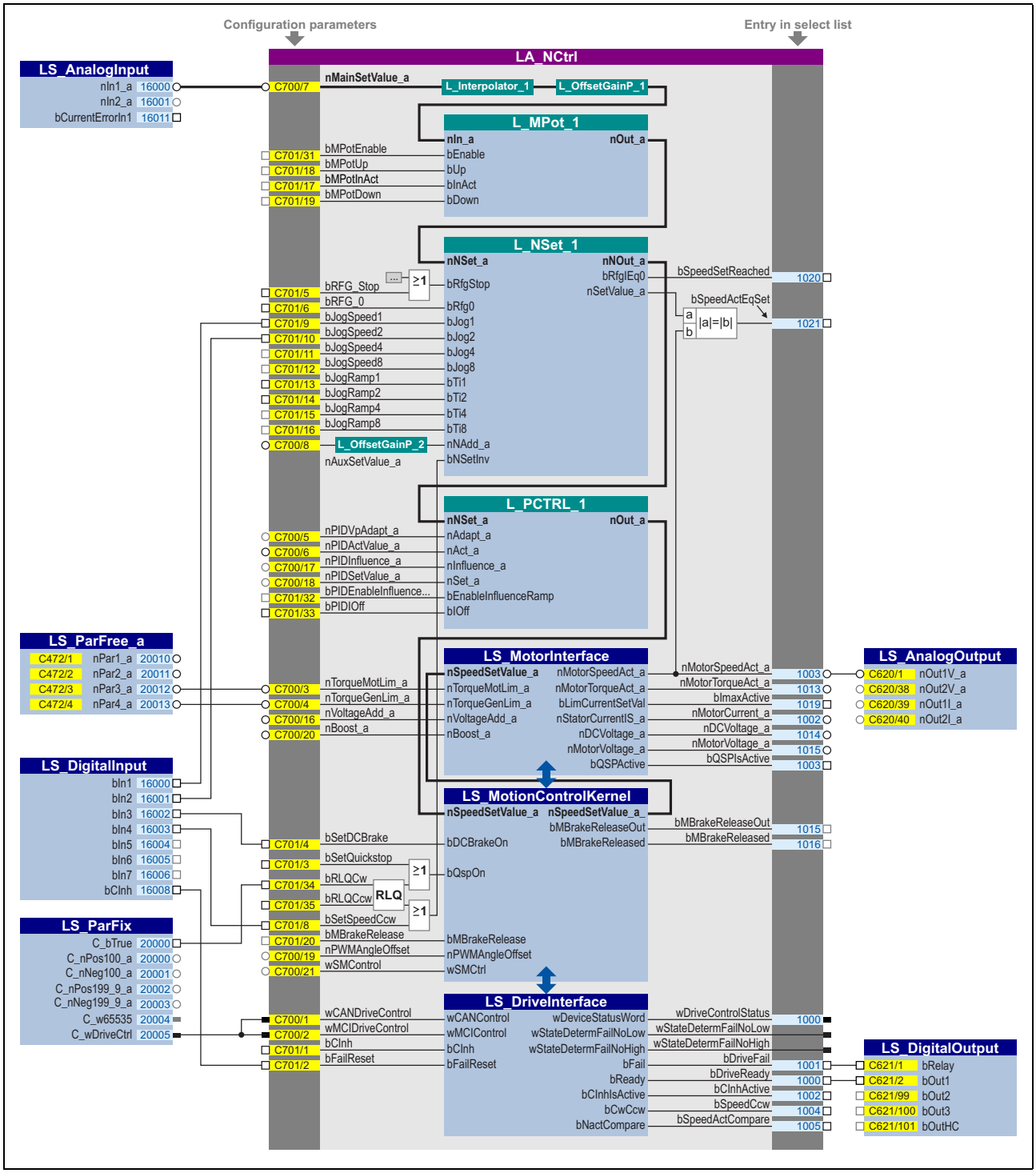
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00696 | L_OffsetGainP_1: Offset | 0.00 | % |
| C00697 | L_OffsetGainP_2: Offset | 0.00 | % |
| C00698 | L_OffsetGainP_3: Offset | 0.00 | % |
| C00800 | L_MPot_1: Upper limit | 100.00 | % |
| C00801 | L_MPot_1: Lower limit | -100.00 | % |
| C00802 | L_MPot_1: Acceleration time | 10.0 | s |
| C00803 | L_MPot_1: Deceleration time | 10.0 | s |
| C00804 | L_MPot_1: Inactive fct. | 0: Retain value | |
| C00805 | L_MPot_1: Init fct. | 0: Load last value | |
| C00806 | Use of motor potentiometer | 0: No | |

**Related topics:**

- "GeneralPurpose" functions (📖 554)

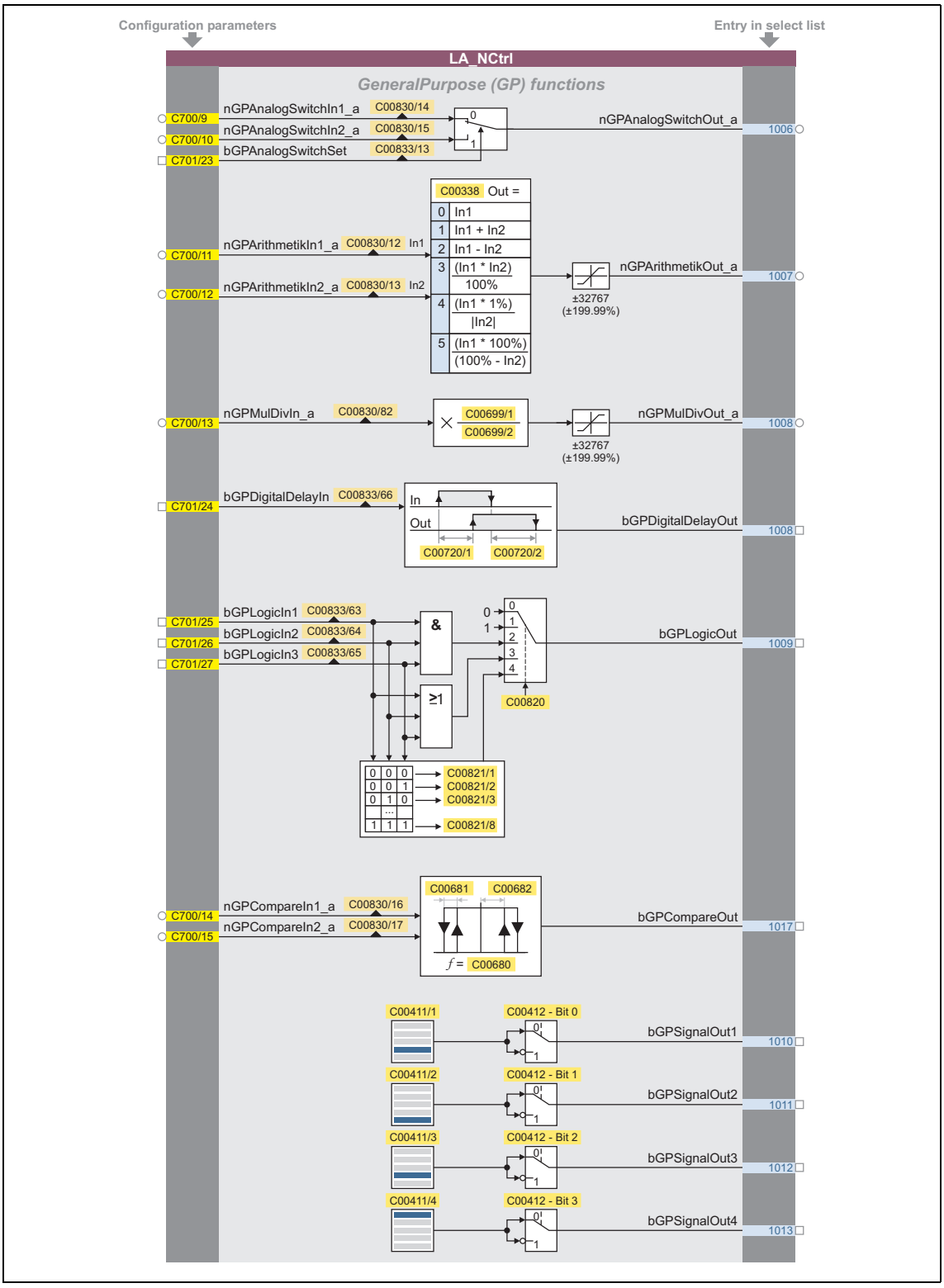
### 8.2.6 Configuration parameters

If required, the subcodes of C00700 and C00701 serve to change the pre-configured assignment of the application inputs:

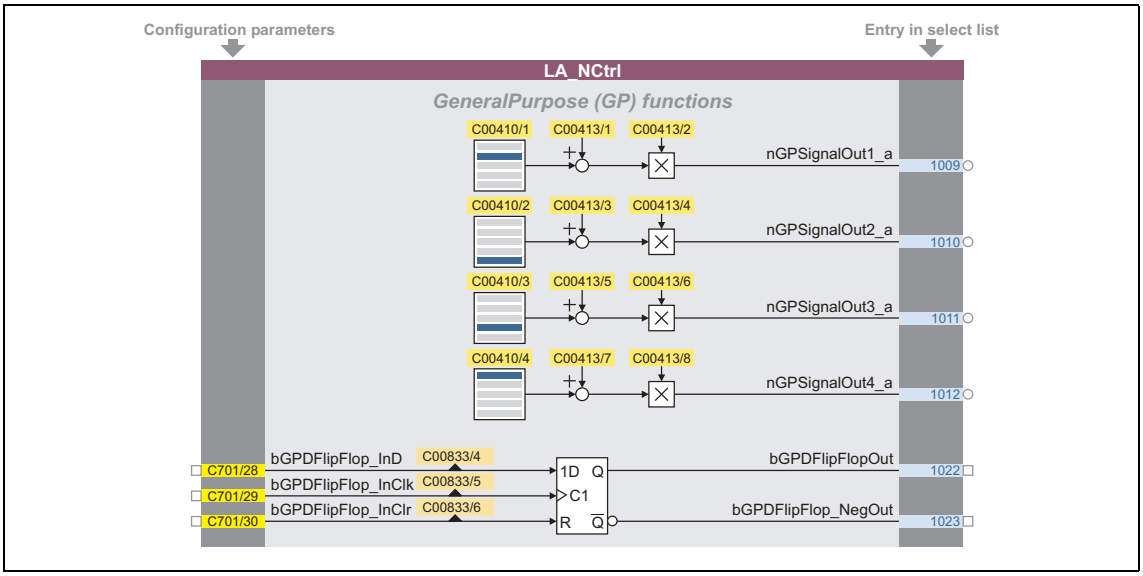



[8-2] Pre-assignment of the "Actuating drive speed" application in the "Terminals 0" control mode

### Configuration parameters for "GeneralPurpose" functions



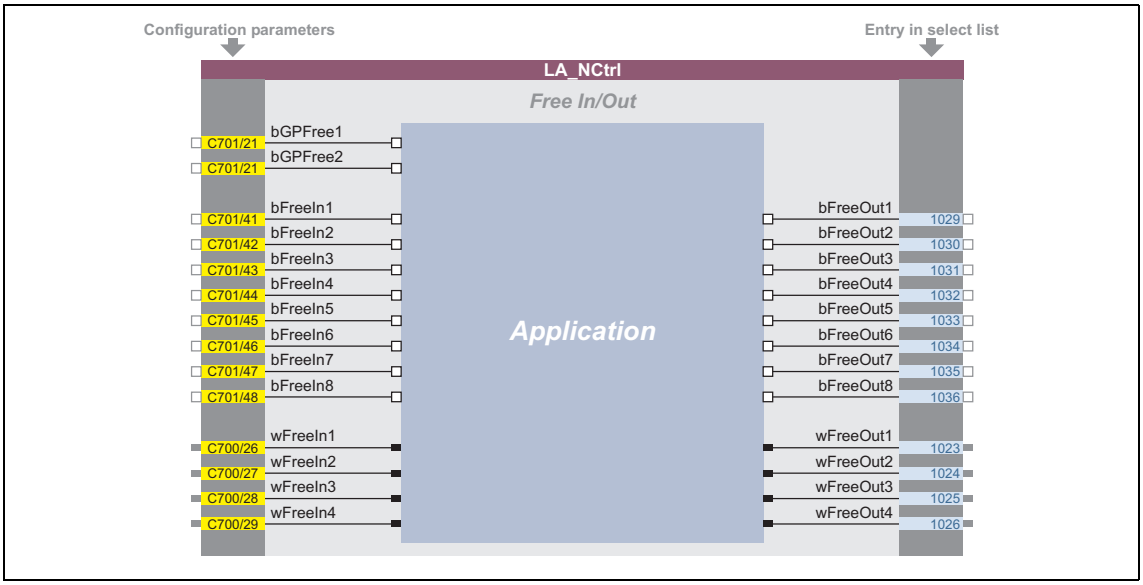

[8-3] "GeneralPurpose" functions




[8-4] "GeneralPurpose" functions (continuation)

### Free inputs and outputs

These inputs can be freely interconnected in the application level. They can be used to transfer signals from the I/O level to the application level and vice versa.




[8-5] Free inputs/outputs

**Related topics:**

- User-defined terminal assignment (📖 429)
- "GeneralPurpose" functions (📖 554)

## 8.3 TA "actuating drive speed (AC Drive Profile)"

This technology application is available from version 13.00.00!

The EtherNet/IP™ communication module supports the "AC Drive Profile".

When the inverter is provided with an EtherNet/IP™ communication module and the control is to be carried out by means of "AC Drive Profile" via EtherNet/IP™, make the following settings:

1. Set the application "1100: Actuating drive speed (AC Drive Profile)" in C00005.
2. Set the "40: MCI" control mode in C00007.
   - The process data word received by the master control is then interpreted by the application as "AC Drive Profile" control word.
   - When the control is carried out via the system bus (CANopen), set the "30: CAN" control mode instead in C00007.

**Product features**

- Pre-configured control modes for terminals and bus control (with predefined process data connection to the fieldbus)
- Free configuration of input and output signals
- adjustable offset, gain and negation of the speed setpoint
- Up to 15 fixed setpoints for speed and ramp time
- Adjustable setpoint ramp times
- Freely selectable, variable ramp shape
- Automatic holding brake control
- Quick stop (QSP) with adjustable ramp time
- Motor potentiometer function (optional)
- Process controller (optional)
- Integrated, freely available "GeneralPurpose" functions:
  Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop
- Interface to the safety module (optional)
- Integration of encoder feedback



**Note!**

In contrast to the "actuating drive speed" standard application, this application makes use of the *nAuxSetValue_a* input to determine a local speed setpoint (when NetRef=0). For this reason, the *nNAdd_a* input at the L_NSet_1 setpoint generator for determining an additional speed setpoint is not connected in the Lenze setting.

### 8.3.1 I/O assemblies

For the data exchange, the technology application supports the assembly output object instance 23 (0x17) and assembly input object instance 73 (0x49) defined by the ODVA (Open DeviceNet Vendor Association).

| Instance 23 (0x17): Extended Speed and Torque Control Output | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bytes** | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| **0** | | Net Ref | NetCtrl | | | Fault Reset | Run Rev | Run Fwd |
| **1** | | | | | | | | |
| **2** | Speed Reference (Low Byte) | | | | | | | |
| **3** | Speed Reference (High Byte) | | | | | | | |
| **4** | Torque Reference (Low Byte) | | | | | | | |
| **5** | Torque Reference (High Byte) | | | | | | | |

| Instance 73 (0x49): Extended Speed and Torque Control Input | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bytes** | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| **0** | At Reference | RefFrom Net | CtrlFrom Net | Ready | Running2 (Rev) | Running1 (Fwd) | Warning | Faulted |
| **1** | Drive State | | | | | | | |
| **2** | Speed Actual (Low Byte) | | | | | | | |
| **3** | Speed Actual (High Byte) | | | | | | | |
| **4** | Torque Actual (Low Byte) | | | | | | | |
| **5** | Torque Actual (High Byte) | | | | | | | |

Detailed information on the data transfer and "AC Drive Profile" can be found in the E84AYCEO communication manual (EtherNet/IP™).

**Tip!**

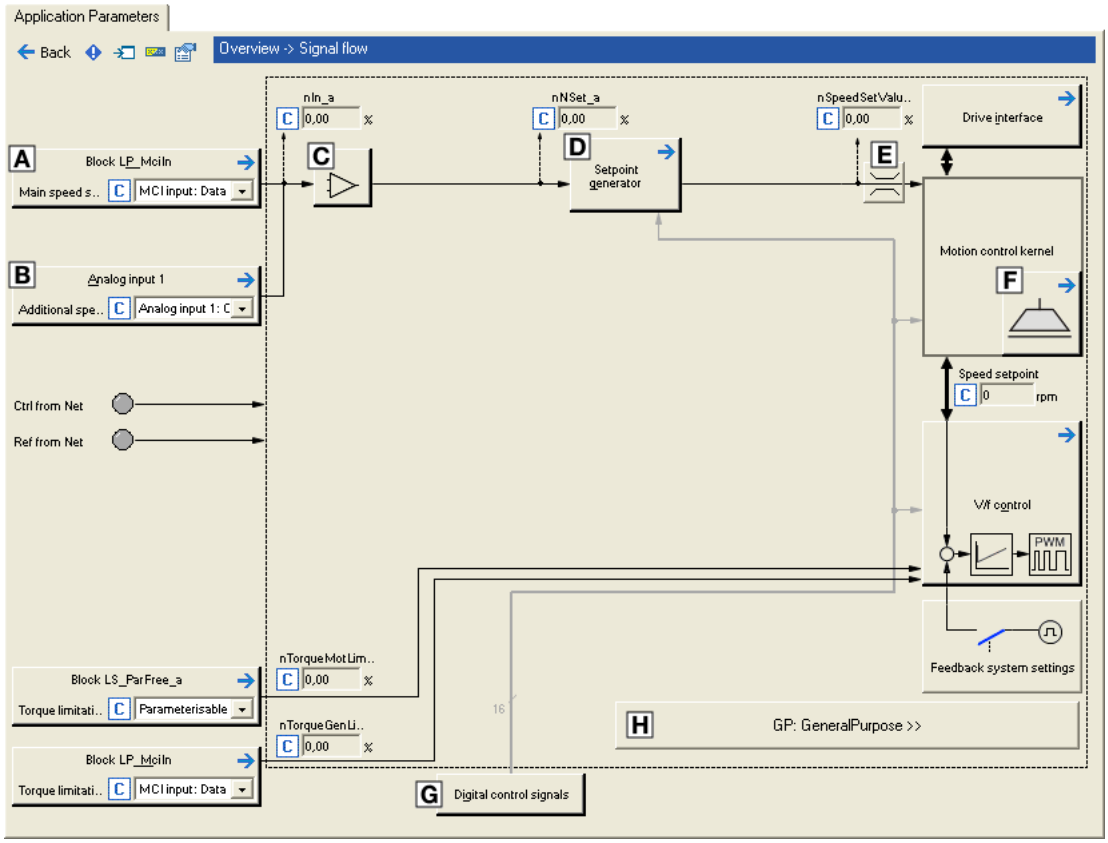
Detailed information on EtherNet/IP™ can be found at the web page of the ODVA (Open DeviceNet Vendor Association) user organisation:

http://www.odva.org

**Related topics:**

- Process data assignment for fieldbus communication
- Run/Stop event
- Scaling of speed and torque values

## 8.3.2 Basic signal flow




[8-6] Signal flow of the "actuating drive speed (AC Drive Profile)" technology application

- A Configuration of the signal source for setpoint selection via fieldbus (NetRef=1)
- B Configuration of the signal source for local setpoint selection (NetRef=0)
- C Offset and gain for speed setpoint (L_OffsetGainP_1)
- D Setpoint generator (L_NSet_1)
- E Speed setpoint input limitation
- F Holding brake control
- G Terminal assignment & display of digital control signals
- H Integrated disposable "GeneralPurpose" functions: Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop

### Definition of the speed setpoint

The speed setpoint is usually defined as process date via the fieldbus. If set accordingly, the speed setpoint can also be determined locally (e.g. via the analog input 1). Depending on the selection, either only the *nMainSetValue_a* application input or only the application input *nAuxSetValue_a* is effective. The following table shows the relationships:

| Fieldbus used | Control mode (C00007) | Speed reference | AC Drive Profile control word (*wMCIDriveControl*) | effective input (at LA_NCtrl) |
|---|---|---|---|---|
| EtherNet/IP | 40: MCI | via fieldbus (data word 2) | Bit 6 ("NetRef") = 1 | *nMainSetValue_a* |
| | | local* | Bit 6 ("NetRef") = 0 | *nAuxSetValue_a* |
| CANopen | 30: CAN | via fieldbus (data word 2) | Bit 6 ("NetRef") = 1 | *nMainSetValue_a* |
| | | local* | Bit 6 ("NetRef") = 0 | *nAuxSetValue_a* |
| - | 10: Terminals 0<br>12: Terminals 2<br>14: Terminals 11<br>16: Terminals 16 | local* | 9 ≡ 0x0009<br>• Bit 0, SwitchOn = TRUE<br>• Bit 3, EnableOperation = TRUE<br>• All others: FALSE | *nAuxSetValue_a* |
| | 20: Keypad | C00728/3 | | |
| | 21: PC | C00472/1 | | |

* The local setpoint is selected in the Lenze setting via the analog input 1

### Scaling of the speed setpoint

If the setpoint is determined via fieldbus (NetRef=1), the setpoint applied to the *nMainSetValue_a* application input is evaluated with the speed scaling set in C01353/1 (AC Drive Attribute 22). ▸ Scaling of speed and torque values.

If the setpoint is determined locally (e.g. via the analog input 1), the setpoint applied to the *nAuxSetValue_a* application input is scaled as follows:

16384 ≡ 100 % reference speed (C00011).

### Offset & gain

Offset and gain of the speed signal can be set in C00696 and C00670 for a simple signal adjustment of a setpoint encoder.

### Ramp function generator

The setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.

- Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.
- For a detailed functional description see the L_NSet FB.

### Selection of the direction of rotation

- In case of control via fieldbus via bit 0 "Run Forward" and bit 1 "Run Backward" of the AC Drive Profile control word.
- In case of local control via the *bSetSpeedCcw* application input or by selecting negative fixed setpoints.

### Selection of the torque setpoint in torque mode

In torque mode", the *nTorqueGenLim_a* application input has the function of the torque setpoint selection.

The torque setpoint is usually defined as process date via the fieldbus. If set accordingly, the torque setpoint can also be determined locally (e.g. via the analog input 1).

For the local selection of the torque setpoint, the same application input (*nAuxSetValue_a*) is used as for the local selection of the speed setpoint. In "torque mode" however, the *nAuxSetValue_a* input is internally connected to the *nTorqueGenLim_a* input. In this case, the speed setpoint is internally permanently set to "100 %".

The relationships are shown in the following table:

| Fieldbus used | Control mode (C00007) | Torque reference | AC Drive Profile control word (*wMCIDriveControl*) | effective input (at LA_NCtrl) |
|---|---|---|---|---|
| EtherNet/IP | 40: MCI | via fieldbus (data word 3) | Bit 6 ("NetRef") = 1 | *nTorqueGenLim_a* |
| | | local* | Bit 6 ("NetRef") = 0 | *nAuxSetValue_a* |
| CANopen | 30: CAN | via fieldbus (data word 3) | Bit 6 ("NetRef") = 1 | *nTorqueGenLim_a* |
| | | local* | Bit 6 ("NetRef") = 0 | *nAuxSetValue_a* |
| - | 10: Terminals 0<br>12: Terminals 2<br>14: Terminals 11<br>16: Terminals 16 | local* | 9 ≡ 0x0009<br>• Bit 0, SwitchOn = TRUE<br>• Bit 3, EnableOperation = TRUE<br>• All others: FALSE | *nAuxSetValue_a* |
| | 20: Keypad | C00728/3 | | |
| | 21: PC | C00472/1 | | |

* The local setpoint is selected in the Lenze setting via the analog input 1

### 8.3.3 Internal interfaces | application block "LA_NCtrl"

**Note!**

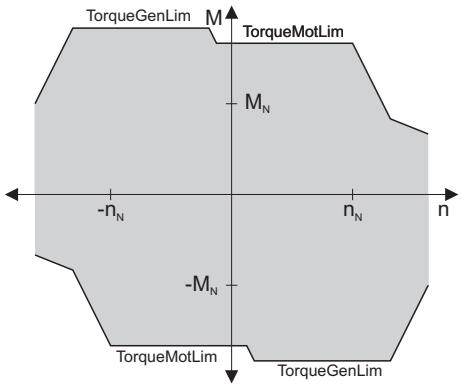
The connectors greyed out in the following table are hidden in the function block editor in the Lenze setting.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wCANDriveControl<br>WORD | Input for CAN control word<br>• Is not used in this configuration and is thus set to the permanent value "9" in the Lenze setting (SwitchOn = TRUE and EnableOperation = TRUE).<br>• For the evaluation of the AC Drive Profile control word received via fieldbus, the *wMCIDriveControl* input is used in the control modes 30: CAN" and "40: MCI". | |
| wMCIDriveControl<br>WORD | Input for the AC Drive Profile control word received via fieldbus<br>• The AC Drive Profile control word operates the inverter in compliance with the assembly output object instances 20 ... 23. For this purpose, the control bits are evaluated and lead to a corresponding modification of the control signals *bFailReset*, *bRFG_0* and *bSetSpeedCcw* which results in the AC Drive-specific behaviour.<br>• See the "Process data assignment for fieldbus communication" subchapter for a detailed description of the individual control bits.<br>• Display parameter: C01351/1 | |
| wSMControl<br>WORD | Interface to the optional safety system.<br>• Setting control bit 0 ("SafeStop1") in this control word causes e.g. the automatic deceleration of the drive to standstill within this application (in the **Motion Control Kernel**).<br>• See the subchapter "Interface to safety system" of the chapter on basic drive functions for a detailed description of the individual control bits. | |
| bCInh<br>BOOL | Enable/inhibit inverter | |
| | FALSE | Enable inverter: The inverter switches to the "OperationEnabled" device status if no other source for controller inhibit is active.<br>• C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit. |
| | TRUE | Inhibit inverter (controller inhibit): The inverter switches to the "SwitchedOn" device status. |
| bFailReset<br>BOOL | Reset error message<br>• In the Lenze setting this input is connected to the digital input controller enable so that a possibly existing error message is reset together with the controller enable (if the cause for the fault is eliminated).<br>• **In case of control via fieldbus (NetCtrl=1):** This input is OR'd with bit 2 ("fault reset") of the AC Drive Profile control word. | |
| | TRUE | The current fault is reset, if the cause for the fault is eliminated.<br>• If the fault still exists, the error status remains unchanged. |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bSetQuickstop<br>BOOL | Activate quick stop (QSP)<br>• Also see device command "Activate/deactivate quick stop". | |
| | TRUE | Activate quick stop<br>• Motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0).<br>• The motor is kept at a standstill during closed-loop operation.<br>• A pulse inhibit is set if the auto-DCB function has been activated via C00019. |
| | FALSE | Deactivate quick stop<br>• The quick stop is deactivated if no other source for the quick stop is active.<br>• C00159 displays a bit code of active sources/causes for the quick stop. |
| bSetDCBrake<br>BOOL | Manual DC-injection braking (DCB)<br>• Detailed information on DC-injection braking is provided in the motor control chapter, subchapter "DC-injection braking". | |
| | **Note!**<br>Holding braking is not possible when this braking mode is used!<br>Use the basic "Holding brake control" function for controlling the holding brake with a low rate of wear. | |
| | FALSE | Deactivate DC-injection braking. |
| | TRUE | Activate DC-injection braking, i.e. the drive is brought to a standstill by means of DC-injection braking.<br>• The braking effect stops when the rotor is at standstill.<br>• After the hold time (C00107) has expired, the controller sets the pulse inhibit. |
| bRFG_Stop<br>BOOL | Ramp function generator: Maintain the current value of the main setpoint integrator<br>• The speed, for instance, of a running ramp process is immediately kept constant when *bRFG_Stop* is activated. At the same time, the acceleration/deceleration jumps to the value "0".<br>• For a detailed functional description see the L_NSet FB. | |
| | TRUE | The current value of the main setpoint integrator is held. |
| bRFG_0<br>BOOL | Ramp function generator: Lead the main setpoint integrator to "0" within the current Ti times<br>• **In case of control via fieldbus (NetCtrl=1):** If a stop is triggered via bit 0 ("Run Forward") and bit 1 ("Run Backward") of the AC Drive Profile control word, this signal is internally set to TRUE and thus the drive is braked to standstill.<br>• For a detailed functional description see the L_NSet FB. | |
| | TRUE | The current value of the main setpoint integrator is led to "0" within the Ti time set. |
| nVoltageAdd_a<br>INT | Additive voltage impression<br>• An additional setpoint for the motor voltage can be specified via this process input.<br>• If there are, for instance, different loads at the motor output end, it is possible to apply a voltage boost at the starting time.<br>• If the value is negative, the voltage is reduced.<br>• Scaling: 16384 ≡ 1000 V | |
| | **Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! | |

| Designator<br>Data type | Information/possible settings |
|---|---|
| nBoost_a<br>INT | Additional setpoint for the motor voltage at speed = 0<br>• The entire voltage-frequency characteristic is provided with an offset.<br>• Scaling: 16384 ≡ 1000 V |
| | **Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! |
| nPWMAngleOffset<br>INT | Additional offset for the electrical angle of rotation<br>• If a torque is connected, e.g. dynamic acceleration processes can be generated.<br>• Scaling: ±32767 ≡ ±180 ° angle of rotation |
| nTorqueMotLim_a<br>nTorqueGenLim_a<br>INT | Torque limitation in motor mode and in generator mode (speed mode) or torque setpoint selection (torque mode) |
| | **When "speed mode" is set, the following applies:**<br>• The torque limitation in motor mode and generator mode are determined via *nTorqueMotLim_a*.<br>• The *nTorqueGenLim_a* input is not effective. |
| | **When "torque mode" is set, the following applies:**<br>• The torque setpoint is determined via *nTorqueGenLim_a*.<br>• The *nTorqueMotLim_a* input is not effective. |
| | More notes for both inputs:<br>• These input signals are directly transferred to the motor control to limit the inverter's maximum torque in motor and generator mode.<br>• The drive cannot output a higher torque in motor/generator mode than set here.<br>• The applied values (any polarity) are internally interpreted as absolute values.<br>• If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.<br>• If sensorless vector control (SLVC) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| | Torque limits in motor and generator mode:<br>  |
| bSetSpeedCcw<br>BOOL | Change of direction of rotation<br>• For instance if a motor or gearbox is fixed laterally reversed to a machine part, but the setpoint selection should still be executed for the positive direction of rotation.<br>• **Input only effective in case of local control (NetCtrl=0)** |
| | FALSE: Clockwise rotation (Cw) |
| | TRUE: Direction of rotation to the left (Ccw) |
| bRLQCw<br>BOOL | Activate clockwise rotation (fail-safe)<br>• For a detailed functional description see the L_RLQ FB. |
| | FALSE: Quick stop |
| | TRUE: CW rotation |

| Designator<br>Data type | Information/possible settings |
|---|---|
| bRLQCcw<br>BOOL | Activate counter-clockwise rotation (fail-safe)<br>• **Input only effective in case of local control (NetCtrl=0)**<br>• For a detailed functional description see the L_RLQ FB. |
| | FALSE: Quick stop |
| | TRUE: Counter-clockwise rotation |
| nMainSetValue_a<br>INT | Speed setpoint via fieldbus in [rpm]<br>• **Input only effective in case of setpoint selection via fieldbus (NetRef=1)**<br>• This input is evaluated with the speed scaling set in C01353/1 (AC Drive Attribute 22). ▸ Scaling of speed and torque values<br>• An absolute value is created internally (the sign has no meaning).<br>• Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal adjustment of a setpoint encoder.<br>• The setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.<br>• Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.<br>• For a detailed functional description see the L_NSet FB. |
| nAuxSetValue_a<br>INT | Local speed setpoint (speed mode) or torque setpoint (torque mode)<br>• **Input only effective in case of local setpoint selection (NetRef=0)** |
| | **When "speed mode" is set, the following applies:**<br>• The input value is interpreted as speed setpoint.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal adjustment of a setpoint encoder.<br>• The setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.<br>• Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.<br>• For a detailed functional description see the L_NSet FB. |
| | **When "torque mode" is set, the following applies:**<br>• The input value is interpreted as torque setpoint.<br>(Input *nAuxSetValue_a* is internally connected to input *nTorqueGenLim_a*).<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• The speed setpoint is internally permanently set to "100 %". |
| bJogSpeed1<br>bJogSpeed2<br>BOOL | Selection inputs for override fixed setpoints (JOG setpoints)<br>• **Inputs only effective in case of local setpoint selection (NetRef=0)**<br>• These selection inputs can be used to activate a fixed setpoint for the setpoint generator instead of the setpoint applied to the *nAuxSetValue_a* input.<br>• The four selection inputs are binary coded, therefore 15 fixed setpoints can be selected.<br>• In case of binary coded selection "0" (all inputs = FALSE or not assigned), the main setpoint applied to the *nAuxSetValue_a* input is active.<br>• The selection of the fixed setpoints is carried out in C00039/1...15 in [%] based on the reference speed (C00011).<br>• For a detailed functional description see the L_NSet FB. |
| bJogSpeed4<br>bJogSpeed8<br>BOOL | |
| bJogRamp1<br>bJogRamp2<br>BOOL | Selection inputs for alternative acceleration/deceleration times<br>• The four selection inputs are binary coded, therefore 15 alternative acceleration/deceleration times can be selected.<br>• In case of binary selection "0" (all inputs = FALSE or not assigned), the acceleration time (C00012) and deceleration time (C00013) set for the main setpoint are active.<br>• Alternative acceleration times are selected in C00101/1...15.<br>• The selection of the alternative deceleration times is carried out in C00103/1...15.<br>• For a detailed functional description see the L_NSet FB. |
| bJogRamp4<br>bJogRamp8<br>BOOL | |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| **Motor potentiometer**<br>Alternatively to the input signal *nMainSetValue_a* (or *nAuxSetValue_a* in case of local setpoint selection), the speed setpoint can also be generated via a motor potentiometer function.<br>• In the Lenze setting, the motor potentiometer function is deactivated.<br>• Activation is possible via C00806 or via the *bMPotEnable* input.<br>• The behaviour of the motor potentiometer during switch-on of the drive system can be selected in C00805.<br>• For a detailed functional description see the L_MPot FB. | | |
| bMPotEnable<br>BOOL | Activating the motor potentiometer function<br>• This input and C00806 are OR'd. | |
| | TRUE | The motor potentiometer function is active; the speed setpoint can be changed via the *bMPotUp* and *bMPotDown* control inputs. |
| bMPotUp<br>BOOL | Increasing the speed setpoint | |
| | TRUE | Approach the upper speed limit value set in C00800 with the acceleration time set in C00802. |
| bMPotInAct<br>BOOL | Activating the inactive function | |
| | TRUE | The speed setpoint behaves according to the inactive function set in C00804.<br>• In the Lenze setting, the speed setpoint is maintained. |
| bMPotDown<br>BOOL | Decreasing the speed setpoint | |
| | TRUE | Approach the lower speed limit value set In C00801 with the deceleration time set in C00803. |
| **Process controller**<br>• In the Lenze setting, the process controller is deactivated.<br>• The activation is executed by selecting the operating mode in C00242.<br>• For a detailed functional description see FB L_PCTRL. | | |
| bPIDEnableInfluenceRamp<br>BOOL | Activate ramp for influencing factor | |
| | FALSE | Influencing factor of the PID controller is ramped down to "0". |
| | TRUE | Influencing factor of the PID controller is ramped up to the value *nPIDInfluence_a*. |
| bPIDIOff<br>BOOL | Switch off the I-component of the process controller<br>• In conjunction with the operating mode set in C00242 (Lenze setting: "Off"). | |
| | TRUE | I-component of the process controller is switched off. |
| nPIDVpAdapt_a<br>INT | Adaptation of gain Vp set in C00222 in percent<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Changes can be done online. | |
| nPIDSetValue_a<br>INT | Sensor and process setpoint for operating modes 2, 4 and 5<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |
| nPIDActValue_a<br>INT | Speed or actual sensor value (actual process value)<br>• Offset and gain for this input signal can be set in C00698 and C00672.<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |
| nPIDInfluence_a<br>INT | Limitation of the influencing factor in percent<br>• The influence factor of the PID controller can be limited to a certain value (-199.99% ... + 199.99%) via *nPIDInfluence_a*.<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| **MCK basic functions** | | |
| bMBrakeRelease<br>BOOL | Holding brake control: Release/apply brake<br>• In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off"). | |
| | FALSE | Apply brake.<br>• During automatic operation, the internal brake logic controls the brake. |
| | TRUE | Release brake manually (forced release).<br>• **Note!** The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If a controller inhibit has been set by the brake control, it will be deactivated.<br>• In semi-automatic operation, the brake is released including feedforward control. |
| **GP: GeneralPurpose**<br>The following inputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | | |
| bGPFree1 ... bGPFree2<br>BOOL | Free inputs for digital signals<br>• Digital signals can be transferred from the I/O level to the application level via these inputs. | |
| nGPAnalogSwitchIn1_a<br>nGPAnalogSwitchIn2_a<br>INT | Analog switch: Input signals<br>• The input signal selected via the selection input *bGPAnalogSwitchSet* is output at output *nGPAnalogSwitchOut_a*. | |
| bGPAnalogSwitchSet<br>BOOL | Analog switch: Selection input | |
| | FALSE | *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn1_a* |
| | TRUE | *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn2_a* |
| nGPArithmetikIn1_a<br>nGPArithmetikIn2_a<br>INT | Arithmetic: Input signals<br>• The arithmetic function is selected in C00338.<br>• The result is output at output *nGPArithmetikOut_a*. | |
| nGPMulDivIn_a<br>INT | Multiplication/Division: Input signal<br>• The factor for the multiplication can be set in C00699/1 (numerator) and C00699/2 (denominator).<br>• The result is output at output *nGPMulDivOut_a*. | |
| bGPDigitalDelayIn<br>BOOL | Binary delay element: Input signal<br>• The on-delay can be set in C00720/1.<br>• The off-delay can be set in C00720/2.<br>• The time-delayed input signal is output at output *bGPDigitalDelayOut*. | |
| bGPLogicIn1<br>bGPLogicIn2<br>bGPLogicIn3<br>BOOL | Binary logic: Input signals<br>• The logic operation is selected in C00820.<br>• The result is output at output *bGPLogicOut*. | |
| nGPCompareIn1_a<br>nGPCompareIn2_a<br>INT | Analog comparison: Input signals<br>• The comparison operation is selected in C00680.<br>• Hysteresis and window size can be set in C00680 and C00682.<br>• If the comparison statement is true, the output *bGPCompareOut* will be set to TRUE. | |
| bGPDFlipFlop_InD<br>bGPDFlipFlop_InClk<br>bGPDFlipFlop_InClr<br>BOOL | D-FlipFlop: Input signals<br>• Data, clock and reset input | |

| Designator<br>Data type | Information/possible settings |
|---|---|
| **Free inputs**<br>The following inputs can freely be interconnected on the application level.<br>The signals can be transferred from the I/O level to the application level via these inputs. | |
| bFreeIn1 ... bFreeIn8<br>BOOL | Free inputs for digital signals |
| wFreeIn1 ... wFreeIn4<br>WORD | Free inputs for 16-bit signals |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| wDriveControlStatus<br>WORD | AC Drive Profile status word<br>• The status word contains information on the currents status of the inverter.<br>• For a detailed description of the individual status bits, see subchapter entitled "Process data assignment for fieldbus communication.<br>• Display parameter: C01352/1 | |
| wStateDetermFailNoLow<br>WORD | Display of the status determining error (LOW word) | |
| wStateDetermFailNoHigh<br>WORD | Display of the status determining error (HIGH word) | |
| bDriveFail<br>BOOL | TRUE | Inverter in error status.<br>• "Fault" device status is active. |
| bDriveReady<br>BOOL | TRUE | Inverter is ready for operation<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the inverter is still inhibited by the user (controller inhibit). |
| bCInhActive<br>BOOL | TRUE | Controller inhibit is active. |
| bQSPIsActive<br>BOOL | TRUE | Quick stop is active. |
| bSpeedCcw<br>BOOL | Current direction of rotation | |
| | FALSE | Clockwise rotation (Cw) |
| | TRUE | Direction of rotation to the left (Ccw) |
| bSpeedActCompare<br>BOOL | Result of the speed comparison (detection of speed=0) | |
| | TRUE | During open-loop operation:<br>Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>Actual speed value < Comparison value (C00024) |
| bOverLoadActive<br>BOOL | In preparation (output is not interconnected on the application level) | |
| bUnderLoadActive<br>BOOL | In preparation (output is not interconnected on the application level) | |
| bImaxActive<br>BOOL | "Current setpoint inside the limitation" status signal | |
| | TRUE | The current setpoint is internally limited (the inverter operates at the maximum current limit). |
| bSpeedSetReached<br>BOOL | Status signal "setpoint = 0" | |
| | TRUE | Speed setpoint from the ramp function generator = 0 |
| bSpeedActEqSet<br>BOOL | TRUE | Actual speed value = speed setpoint |
| nMotorCurrent_a<br>INT | Current stator current/effective motor current<br>• Scaling: 16384 ≡ 100 % $I_{max_mot}$ (C00022) | |

| Designator<br>Data type | Value/meaning |
|---|---|
| nMotorSpeedSet_a<br>INT | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| nMotorSpeedAct_a<br>INT | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The absolute value is output (the sign has no meaning). |
| nMotorTorqueAct_a<br>INT | Actual torque<br>• In the "VFC (+encoder)" operating mode of the motor control, this value is determined from the current motor current and corresponds to the actual torque only by approximation.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• The absolute value is output (the sign has no meaning). |
| nDCVoltage_a<br>INT | Actual DC-bus voltage<br>• Scaling: 16384 ≡ 1000 V |
| nMotorVoltage_a<br>INT | Current motor voltage/inverter output voltage<br>• Scaling: 16384 ≡ 1000 V |
| **MCK basic functions** | |
| bMBrakeReleaseOut<br>BOOL | Holding brake control: Trigger signal for the holding brake control switching element via a digital output<br>• Use bit 0 in C02582 to activate inverted switching element triggering. |
| | FALSE: Apply brake. |
| | TRUE: Release brake. |
| bMBrakeReleased<br>BOOL | Holding brake control: "Brake released" considering the brake release time<br>• When the holding brake is triggered to close, *bMBrakeReleased* is immediately set to FALSE even if the brake closing time has not yet elapsed! |
| | TRUE: Brake released (after the brake release time has expired). |
| **GP: GeneralPurpose**<br>The following outputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | |
| nGPAnalogSwitchInOut_a<br>INT | Analog switch: Output signal |
| nGPArithmetikOut_a<br>INT | Arithmetic: Output signal |
| nGPMulDivOut_a<br>INT | Multiplication/Division: Output signal |
| bGPDigitalDelayOut<br>BOOL | Binary delay element: Output signal |
| bGPLogicOut<br>BOOL | Binary logic: Output signal |
| bGPCompareOut<br>BOOL | Analog comparison: Output signal |
| bGPSignalOut1<br>...<br>bGPSignalOut4<br>BOOL | Binary signal monitor: Output signals<br>• The signal sources to be output are selected in C00411/1...4.<br>• A bit coded inversion of the output signals can be parameterised in C00412. |
| nGPSignalOut1_a<br>...<br>nGPSignalOut4_a<br>BOOL | Analog signal monitor: Output signals<br>• The signal sources to be output are selected in C00410/1...4.<br>• Gain and offset for each output signal can be parameterised in C00413/1...8. |
| bGPDFlipFlop_Out<br>BOOL | D-FlipFlop: Output signal |
| bGPDFlipFlop_NegOut<br>BOOL | D-FlipFlop: Negated output signal |

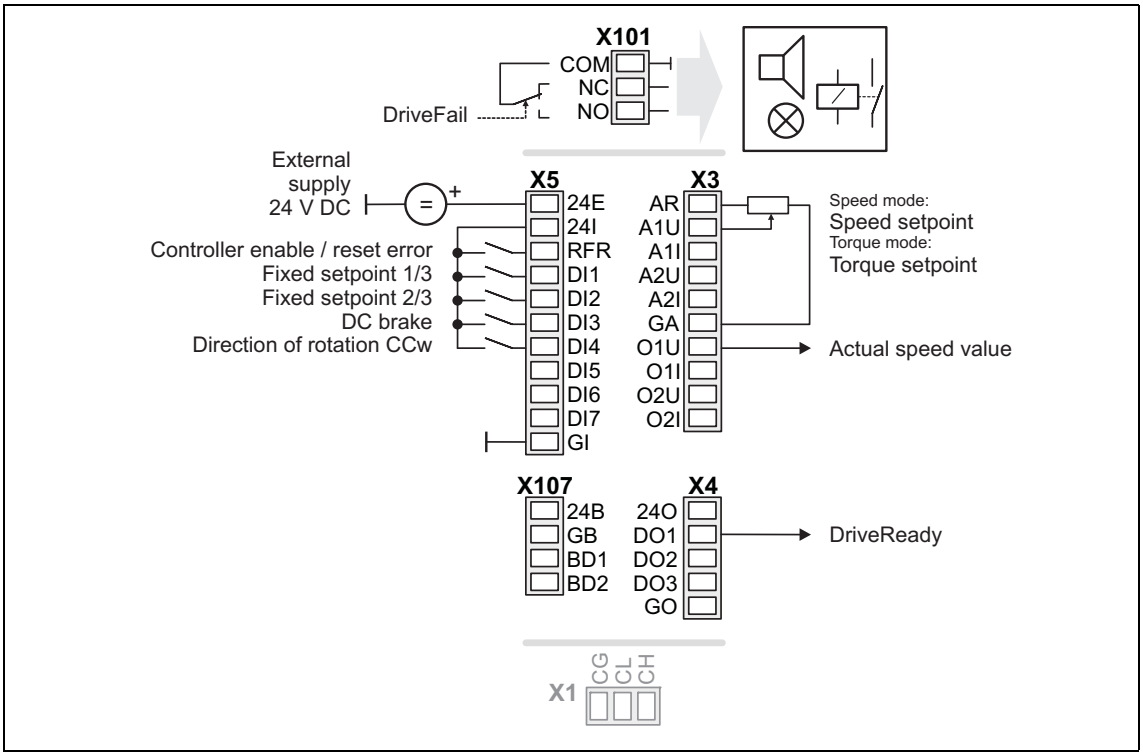
| Designator<br>Data type | Value/meaning |
|---|---|
| **Free outputs**<br>The following outputs can freely be interconnected on the application level.<br>The signals from the application level can be transferred to the I/O level via these outputs. | |
| bFreeOut1 ... bFreeOut8<br>BOOL | Free outputs for digital signals |
| wFreeOut1 ... wFreeOut4<br>WORD | Free outputs for 16-bit signals |

## 8.3.4 Terminal assignment of the control modes

The following comparison provides information about which inputs/outputs of the application block **LA_NCtrl** are interconnected to the digital and analog input/output terminals of the inverter in the different control modes.

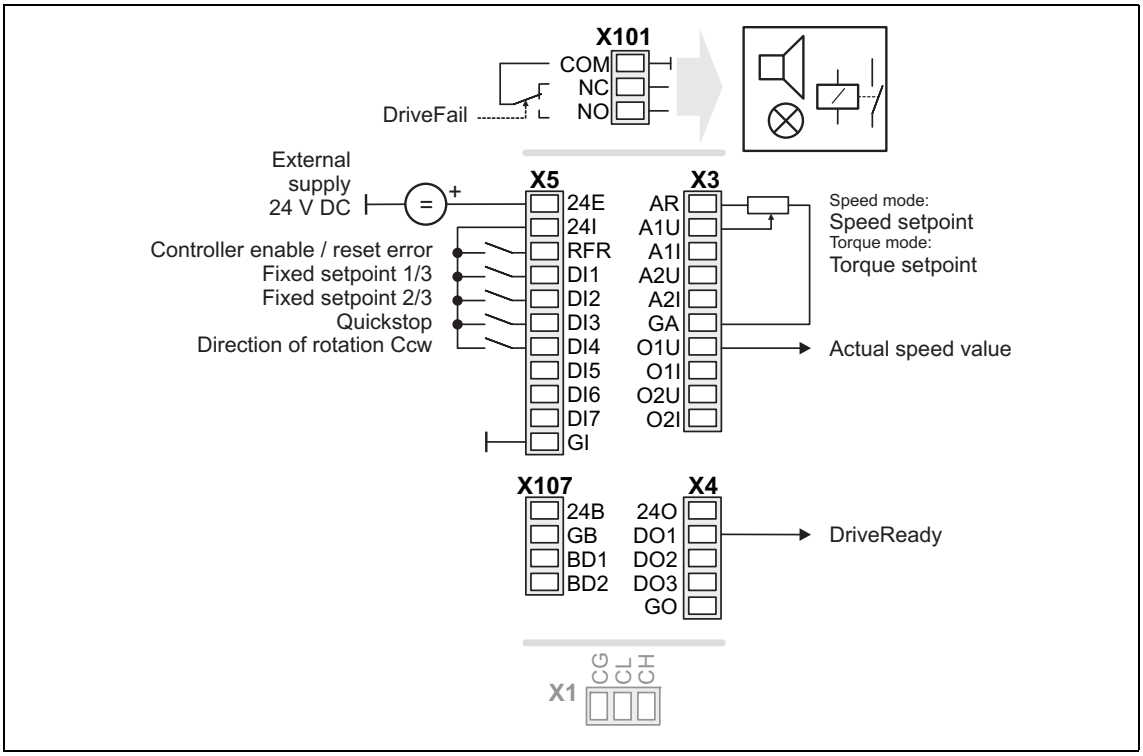
| | Control mode (C00007) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 10: Terminals 0 | 12: Terminals 2 | 14: Terminals 11 | 16: Terminal 16 | 20: Keypad | 21: PC | 30: CAN | 40: MCI |
| **Digital input terminals** | | | | | | | | |
| **X5/RFR** | **Controller enable / Reset of error message**<br>bFailReset | | | | | | | |
| **X5/DI1** | **Fixed setpoint 1/3**<br>bJogSpeed1 | | **Change of direction of rotation**<br>bSetSpeedCcw | **Fixed setpoint 1/3**<br>bJogSpeed1 | - | - | Only if NetCtrl=0:<br>**fixed setpoint 1/3**<br>bJogSpeed1 | |
| **X5/DI2** | **Fixed setpoint 2/3**<br>bJogSpeed2 | | **Activate manual DC-injection braking (DCB)**<br>bSetDCBrake | **Fixed setpoint 2/3**<br>bJogSpeed2 | - | - | Only if NetCtrl=0:<br>**fixed setpoint 2/3**<br>bJogSpeed2 | |
| **X5/DI3** | **Activate manual DC-injection braking (DCB)**<br>bSetDCBrake | **Quick stop**<br>bSetQuickstop | **Motor potentiometer: Increase speed**<br>bMPotUp | **CW rotation quick stop**<br>bRLQCw | - | - | Only if NetCtrl=0:<br>**Activate manual DC-injection braking (DCB)**<br>bSetDCBrake | |
| **X5/DI4** | **Change of direction of rotation**<br>bSetSpeedCcw | | **Motor potentiometer: Decrease speed**<br>bMPotDown | **CCW rotation quick stop**<br>bRLQCcw | - | - | Only if NetCtrl=0:<br>**change of direction of rotation**<br>bSetSpeedCcw | |
| **X5/DI5 ... DI7** | - | - | - | - | - | - | - | - |
| **Analog input terminals** | | | | | | | | |
| **X3/A1U, A1I** | **Local setpoint**<br>nAuxSetValue_a<br>Speed mode: 10 V ≡ 100 % reference speed (C00011)<br>Torque mode: 10 V ≡ 100 % $M_{max}$ (C00057); speed setpoint = 100 % (fixed) | | | | - | - | Only if NetRef=0: **Local setpoint**<br>nAuxSetValue_a<br>Speed mode: 10 V ≡ 100 % reference speed (C00011)<br>Torque mode: 10 V ≡ 100 % $M_{max}$ (C00057) | |
| **X3/A2U, A2I** | - | - | - | - | - | - | - | - |
| **Digital output terminals** | | | | | | | | |
| **X4/DO1** | **Status "Drive is ready"**<br>bDriveReady | | | | | | | |
| **X4/DO2 ... DO3** | - | - | - | - | - | - | - | - |
| **X107/BD1, BD2** | - | - | - | - | - | - | - | - |
| **X101/COM, NO** | **Status "Error is pending"**<br>bDriveFail | | | | | | | |
| **Analog output terminals** | | | | | | | | |
| **X3/O1U** | **Actual speed value**<br>nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) | | | | | | | |
| **X3/O1I** | - | - | - | - | - | - | - | - |
| **X3/O2U** | | | | | | | | |
| **X3/O2I** | | | | | | | | |

### 8.3.4.1 Terminals 0



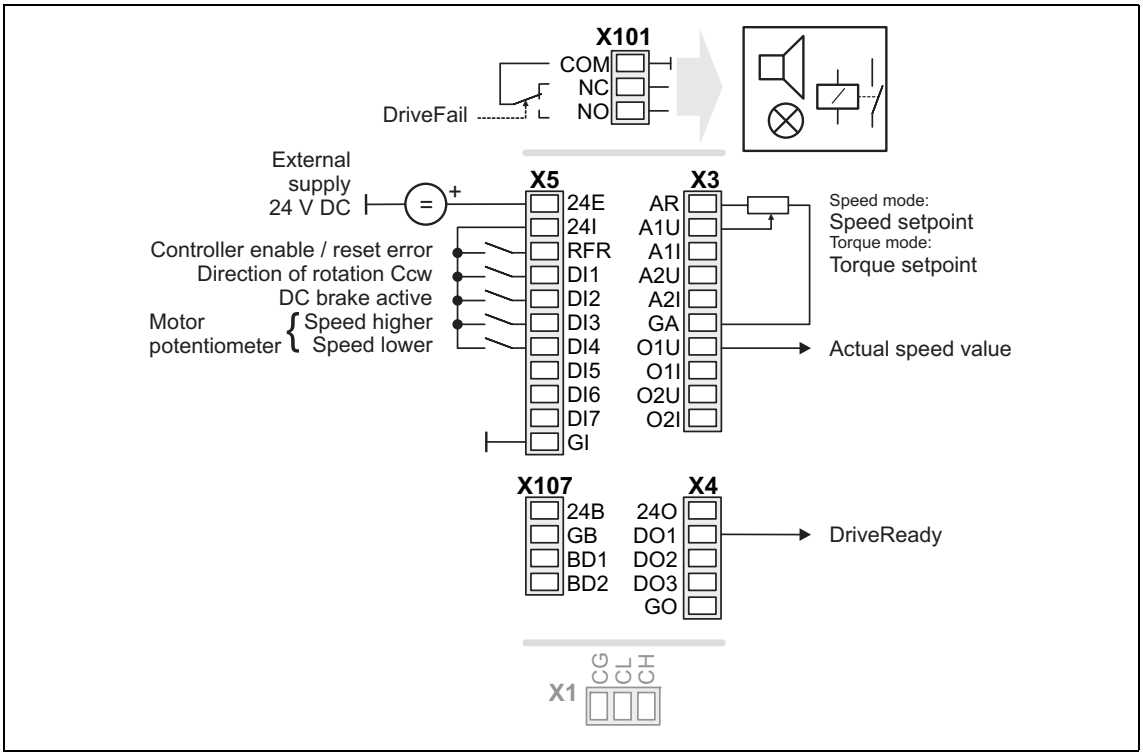

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | LA_NCtrl.nAuxSetValue_a * |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 | | **X3/A1I** | - |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 | | **X3/A2U** | - |
| **X5/DI3** | LA_NCtrl.bSetDCBrake | | **X3/A2I** | - |
| **X5/DI4** | LA_NCtrl.bSetSpeedCcw | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.3.4.2 Terminals 2



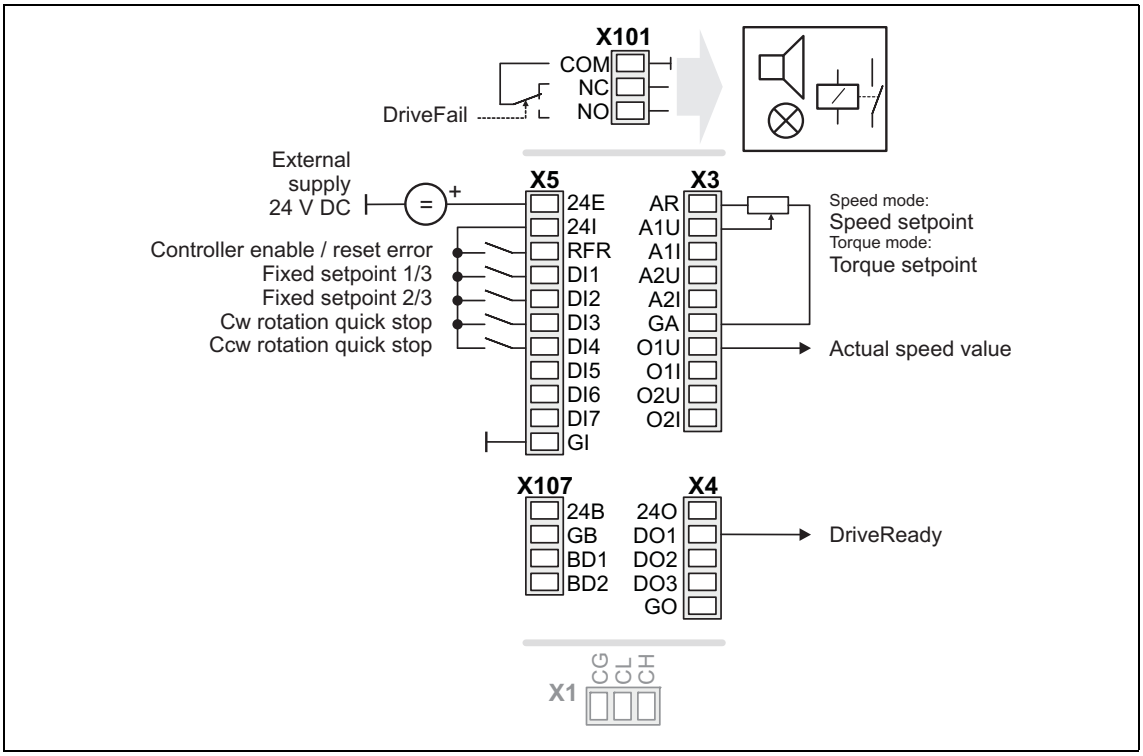

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | LA_NCtrl.nAuxSetValue_a * |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 | | **X3/A1I** | - |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 | | **X3/A2U** | - |
| **X5/DI3** | LA_NCtrl.bSetQuickstop | | **X3/A2I** | - |
| **X5/DI4** | LA_NCtrl.bSetSpeedCcw | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.3.4.3 Terminals 11



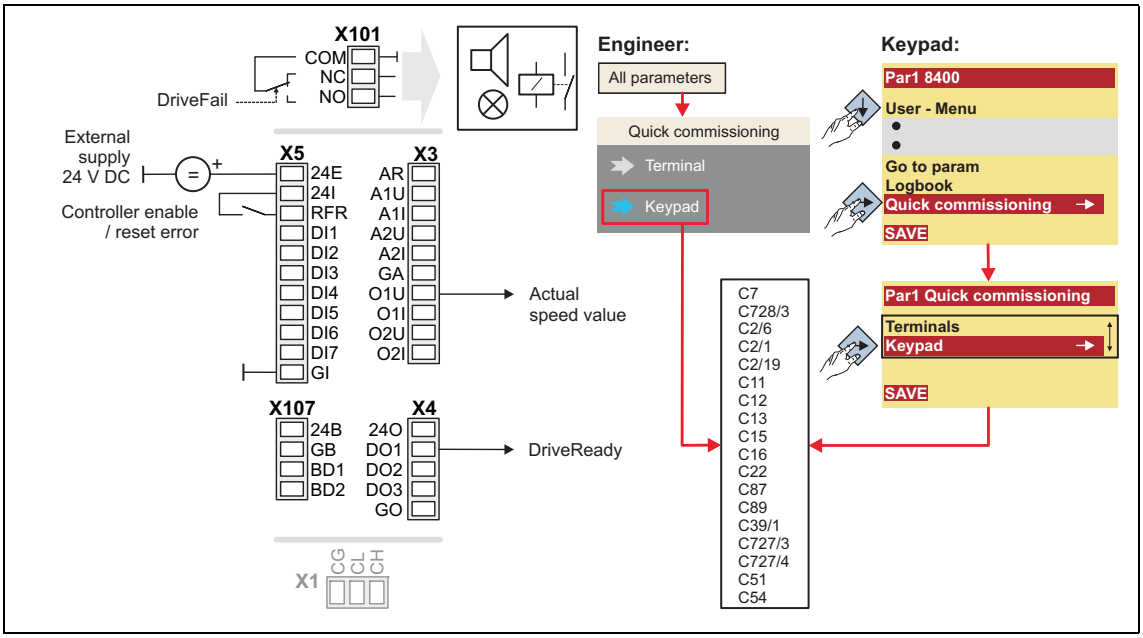

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | LA_NCtrl.nAuxSetValue_a * |
| **X5/DI1** | LA_NCtrl.bSetSpeedCcw | | **X3/A1I** | - |
| **X5/DI2** | LA_NCtrl.bSetDCBrake | | **X3/A2U** | - |
| **X5/DI3** | LA_NCtrl.bMPotUp | | **X3/A2I** | - |
| **X5/DI4** | LA_NCtrl.bMPotDown | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.3.4.4 Terminal 16



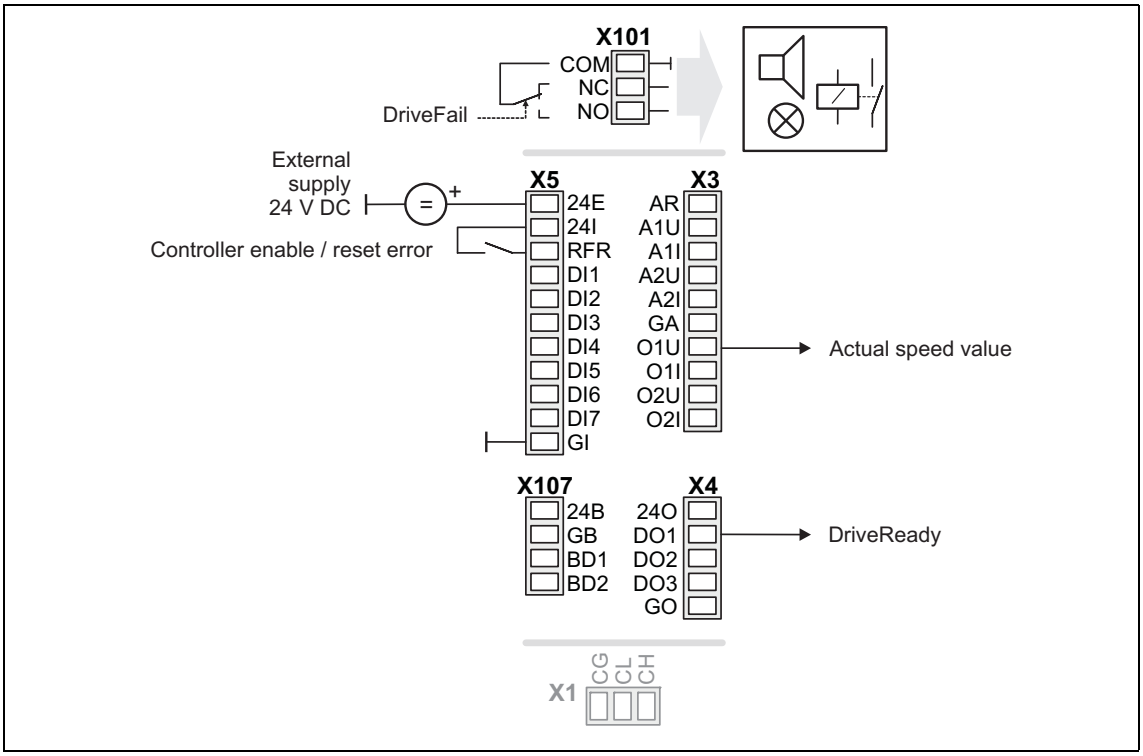

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | LA_NCtrl.nAuxSetValue_a * |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 | | **X3/A1I** | - |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 | | **X3/A2U** | - |
| **X5/DI3** | LA_NCtrl. bRLQCw | | **X3/A2I** | - |
| **X5/DI4** | LA_NCtrl. bRLQCcw | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.3.4.5 Keypad



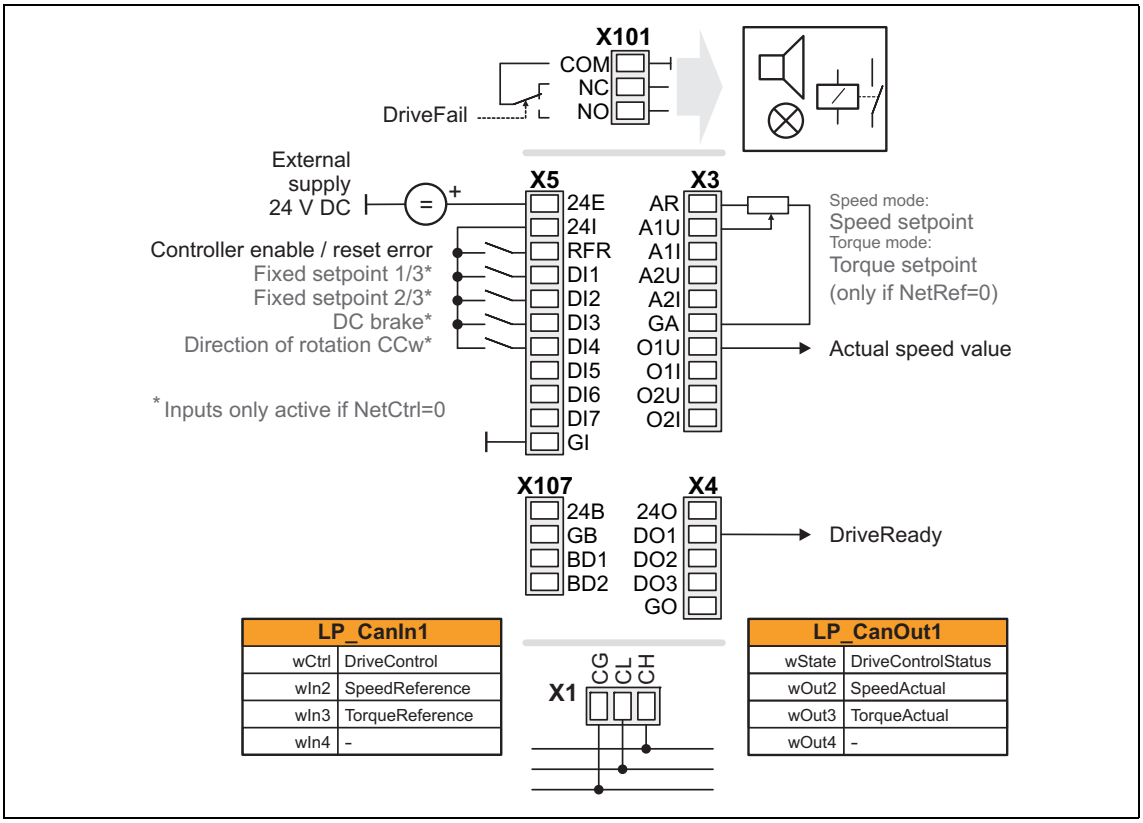

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | - |
| **X5/DI1** | - | | **X3/A1I** | - |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | - | | **X3/A2I** | - |
| **X5/DI4** | - | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.3.4.6 PC




| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | - |
| **X5/DI1** | - | | **X3/A1I** | - |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | - | | **X3/A2I** | - |
| **X5/DI4** | - | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

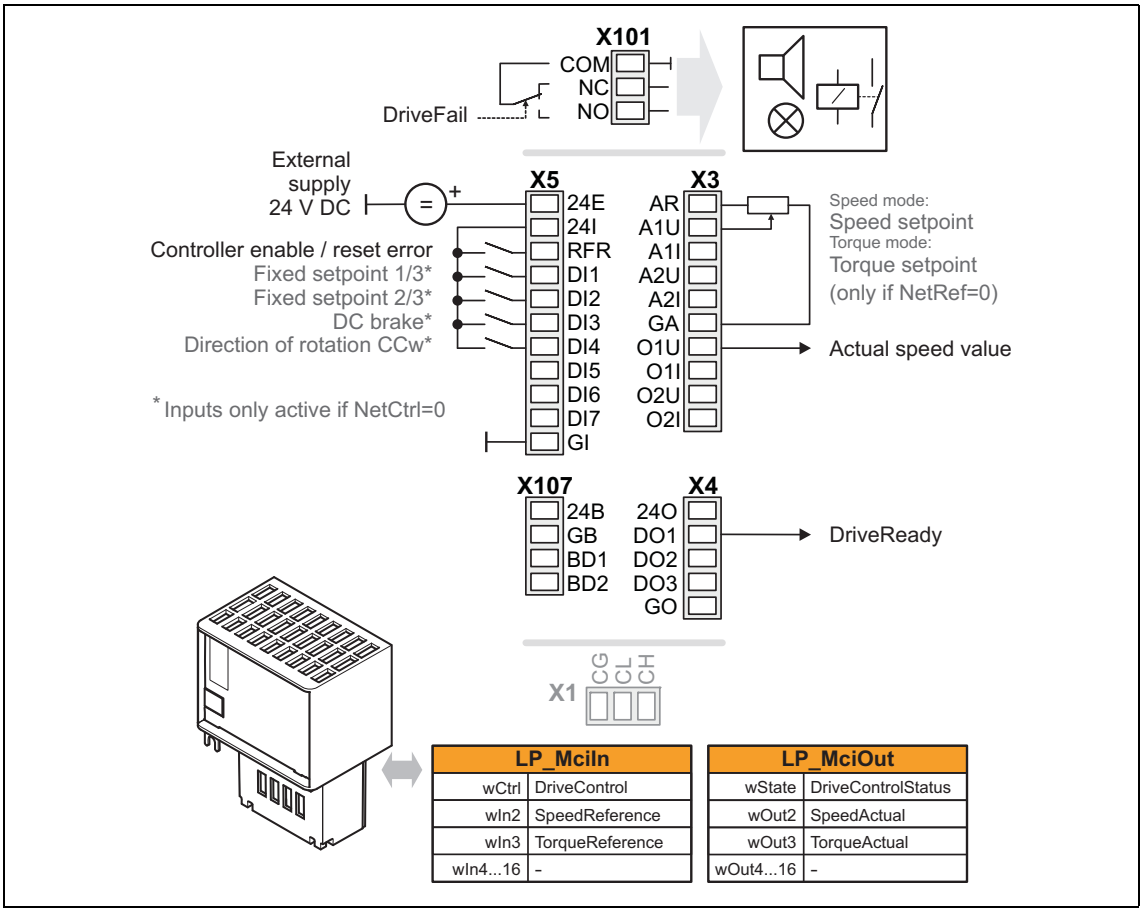
### 8.3.4.7 CAN




| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | LA_NCtrl.nAuxSetValue_a * |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 | | **X3/A1I** | - |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 | | **X3/A2U** | - |
| **X5/DI3** | LA_NCtrl.bSetDCBrake | | **X3/A2I** | - |
| **X5/DI4** | LA_NCtrl.bSetSpeedCcw | | **X3/O1U** | LA_NCtrl.NMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

▸ Process data assignment for fieldbus communication (📖 458)

### 8.3.4.8 MCI




| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_NCtrl.bFailReset | | **X3/A1U** | LA_NCtrl.nAuxSetValue_a * |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 | | **X3/A1I** | - |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 | | **X3/A2U** | - |
| **X5/DI3** | LA_NCtrl.bSetDCBrake | | **X3/A2I** | - |
| **X5/DI4** | LA_NCtrl.bSetSpeedCcw | | **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | * 10 V ≡ 100 % reference speed (C00011) |
| **X107/BD1** | - | | **X4/DO1** | LA_NCtrl.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

▸ Process data assignment for fieldbus communication (📖 488)

### 8.3.5 Process data assignment for fieldbus communication

The fieldbus communication is connected (preconfigured) to the previously selected technology application by selecting the corresponding control mode in C00007:

- "30: CAN" for the connection to the system bus (CAN)
- "40: MCI" for the connection to a plugged-in communication module (e.g. EtherNet/IP™)

The assignment of the process data words depends only on the application, not on the bus system used:

| Input words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControl | Control word<br>• For bit assignment see the table below.<br>• Display parameter: C01351/1 |
| Word 2 | SpeedReference | Speed setpoint in [rpm]<br>▸ Scaling of speed and torque values |
| Word 3 | TorqueReference | Torque setpoint in [Nm]<br>▸ Scaling of speed and torque values |
| Word 4 | - | Not preconfigured |
| Words 5 ... 16 | - | Not preconfigured<br>• Only available in control mode "40: MCI". |

| Control word | Name | Function |
|---|---|---|
| Bit 0 | Run Forward | Relationships between Run1 and Run2 and trigger events can be found in the chapter "Run/Stop event". |
| Bit 1 | Run Backward | |
| Bit 2 | Fault Reset | 0↗1 ≡ Reset error<br>0 ≡ No response |
| Bit 3 | Reserved | - |
| Bit 4 | Reserved | - |
| Bit 5 | NetCtrl | Run/Stop control:<br>0 ≡ via local setting in the device or terminal<br>1 ≡ via fieldbus (e.g. by the scanner) |
| Bit 6 | NetRef | Reference speed/reference torque:<br>0 ≡ via local setting in the device or terminal<br>1 ≡ via fieldbus (e.g. by the scanner) |
| Bit 7 ... 15 | Reserved | - |

| Output words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControlStatus | Status word<br>• For bit assignment see the table below.<br>• Display parameter: C01352/1 |
| Word 2 | SpeedActual | Actual speed value in [rpm]<br>▸ Scaling of speed and torque values |
| Word 3 | TorqueActual | Actual torque in [Nm]<br>▸ Scaling of speed and torque values |
| Word 4 | - | Not preconfigured |
| Words 5 ... 16 | - | Not preconfigured<br>• Only available in control mode "40: MCI". |

| Status word | Name | Status |
|---|---|---|
| Bit 0 | Faulted | 0 ≡ No errors<br>1 ≡ Errors have occurred |
| Bit 1 | Warning | 0 ≡ No warnings<br>1 ≡ Warnings have occurred |
| Bit 2 | Running1 (Fwd) | Relationships between Run1 and Run2 and trigger events can be found in the chapter "Run/Stop event". |
| Bit 3 | Running2 (Rev) | |
| Bit 4 | Ready | 0 ≡ Different status than in case of "1"<br>1 ≡ Ready or Enabled or Stopping |
| Bit 5 | Ctrl from Net | Run/Stop control:<br>0 ≡ via local setting in the device or terminal<br>1 ≡ via fieldbus (e.g. by the scanner) |
| Bit 6 | Ref from Net | Reference speed/reference torque:<br>0 ≡ via local setting in the device or terminal<br>1 ≡ via fieldbus (e.g. by the scanner) |
| Bit 7 | At Reference | 1 ≡ The drive currently runs with reference speed (speed mode) or reference torque (Torque mode) |
| Bit 8 | Drive State | The "Drive State" is coded as follows:<br>0: Manufacturer-specific (not used with 8400)<br>1: Start-up (drive initialisation)<br>2: Not_Ready (mains voltage switched off)<br>3: Ready (mains voltage switched-on)<br>4: Enabled (drive has received "Run" command)<br>5: Stopping (drive has received "Stop" command and is stopped)<br>6: Fault_Stop (drive is stopped due to an error)<br>7: Faulted (errors have occurred) |
| Bit 9 | Drive State | |
| Bit 10 | Drive State | |
| Bit 11 | Drive State | |
| Bit 12 | Drive State | |
| Bit 13 | Drive State | |
| Bit 14 | Drive State | |
| Bit 15 | Drive State | |

### 8.3.5.1 Run/Stop event

Relationships between Run1 and Run2:

| | Starter | | | | | Drive |
|---|---|---|---|---|---|---|
| | Contactor | Starter | Reverser | Speed | Soft start | |
| Run1 | Close | Run | RunFwd | RunLow | RunRamp1 | RunFwd |
| Run2 | No Action | No Action | RunRev | RunHigh | RunRamp2 | RunRev |

Run1 and Run2 trigger:

| Run1 | Run2 | Trigger event | Run type |
|---|---|---|---|
| 0 | 0 | Stop | No Action |
| 0 → 1 | 0 | Run | Run1 |
| 0 | 0 → 1 | Run | Run2 |
| 0 → 1 | 0 → 1 | No Action | No Action |
| 1 | 1 | No Action | No Action |
| 1 → 0 | 1 | Run | Run2 |
| 1 | 1 → 0 | Run | Run1 |

### 8.3.5.2 Scaling of speed and torque values

#### Scaling of the speed values

The speed setpoint is defined by the bus in [rpm]. The conversion then takes place in the inverter as all speed-related signals are processed to a reference variable in percent. An adjustable scaling factor serves to make an additional scaling.

**Equation for scaling the speed setpoint**

$$\text{Speed setpoint}_{\text{Application}} = \text{Speed setpoint}_{\text{Bus}}[\text{rpm}] \cdot \frac{16384}{\text{Reference speed [rpm]}} \cdot \frac{1}{2^{\text{Scaling factor}}}$$

| Parameters | Name | Description |
|---|---|---|
| C00011 | Appl.: Reference speed | Reference variable for speed-related signals |
| C01353/1 | ACDrive: Speed scaling | In the Lenze setting "0", no scaling takes place ($2^0 = 1$) |

For the output of the actual speed value to the bus, the following conversion is made:

**Equation for scaling the actual speed value**

$$\text{Actual speed value}_{\text{Bus}}[\text{rpm}] = \text{Actual speed value}_{\text{Application}} \cdot \frac{\text{Reference speed [rpm]}}{16384} \cdot 2^{\text{Scaling factor}}$$

| Parameters | Name | Description |
|---|---|---|
| C00011 | Appl.: Reference speed | Reference variable for speed-related signals |
| C01353/1 | ACDrive: Speed scaling | In the Lenze setting "0", no scaling takes place ($2^0 = 1$) |

#### Scaling of the torque values

The torque setpoint is defined by the bus in [Nm]. The conversion then takes place in the inverter as all torque-related signals are processed to a reference variable in percent. An adjustable scaling factor serves to make an additional scaling.

**Equation for scaling the torque setpoint**

$$\text{Torque setpoint}_{\text{Application}} = \text{Torque setpoint}_{\text{Bus}}[\text{Nm}] \cdot \frac{16384 \cdot 100}{\text{Maximum torque [0.01 Nm]}} \cdot \frac{1}{2^{\text{Scaling factor}}}$$

| Parameters | Name | Description |
|---|---|---|
| C00057 | Maximum torque | Reference variable for torque-related signals |
| C01353/2 | ACDrive: Torque scaling | In the Lenze setting "0", no scaling takes place ($2^0 = 1$) |

For the output of the actual torque value to the bus, the following conversion is made:

**Equation for scaling the actual torque**

$$\text{Actual torque}_{\text{Bus}}[\text{Nm}] = \text{Actual torque}_{\text{Application}} \cdot \frac{\text{Maximum torque [0.01 Nm]}}{16384 \cdot 100} \cdot 2^{\text{Scaling factor}}$$

| Parameters | Name | Description |
|---|---|---|
| C00057 | Maximum torque | Reference variable for torque-related signals |
| C01353/2 | ACDrive: Torque scaling | In the Lenze setting "0", no scaling takes place ($2^0 = 1$) |

### 8.3.6 AC Drive Profile diagnostic parameters

In the **All parameters** tab, the parameters for diagnostic purposes listed in the following table are displayed in the **AC Drive Profile** category.

**Note!**

These parameter are set by the EtherNet/IP™ communication module and should not be written by the user.

| Parameters | Info | Lenze setting |
|---|---|---|
| C01350/1 | ACDrive: DriveMode | 1: Speed mode |
| C01351/1 | ACDrive: Control word | - |
| C01352/1 | ACDrive: Status word | - |
| C01353/1 | ACDrive: Speed scaling | 0 |
| C01353/2 | ACDrive: Torque scaling | 0 |
| Greyed out = display parameter | | |

### 8.3.7 Setting parameters (short overview)

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00012 | Accel. time - main setpoint | 2.000 | s |
| C00013 | Decel. time - main setpoint | 2.000 | s |
| C00019 | Auto-DCB: Threshold | 3 | rpm |
| C00024 | LS_DriveInterface: bNActCompare | 0.00 | % |
| C00036 | DCB braking: Current | 50.00 | % |
| C00039/1 | Fixed setpoint 1 | 40.00 | % |
| C00039/2 | Fixed setpoint 2 | 60.00 | % |
| C00039/3 | Fixed setpoint 3 | 80.00 | % |
| C00039/4...15 | Fixed setpoint 4 ... 15 | 0.00 | % |
| C00101/1...15 | Add. accel. time 1 ... 15 | 0.000 | s |
| C00103/1...15 | Add. decel. time 1 ... 15 | 0.000 | s |
| C00105 | Decel. time - quick stop | 2.000 | s |
| C00106 | Auto-DCB: Hold time | 0.500 | s |
| C00107 | DCB braking: Hold time | 999.000 | s |
| C00134 | L_NSet_1: Ramp smoothing | 0: Off | |
| C00182 | L_NSet_1: S-ramp time PT1 | 20.00 | s |
| C00190 | L_NSet_1: Setpoint arithmetic | 0: Out = Set | |
| C00220 | L_NSet_1: Acceleration time - add. setpoint | 0.000 | s |
| C00221 | L_NSet_1: Deceleration time - add. setpoint | 0.000 | s |
| C00222 | L_PCTRL_1: Vp | 1.0 | |
| C00223 | L_PCTRL_1: Tn | 400 | ms |
| C00224 | L_PCTRL_1: Kd | 0.0 | |

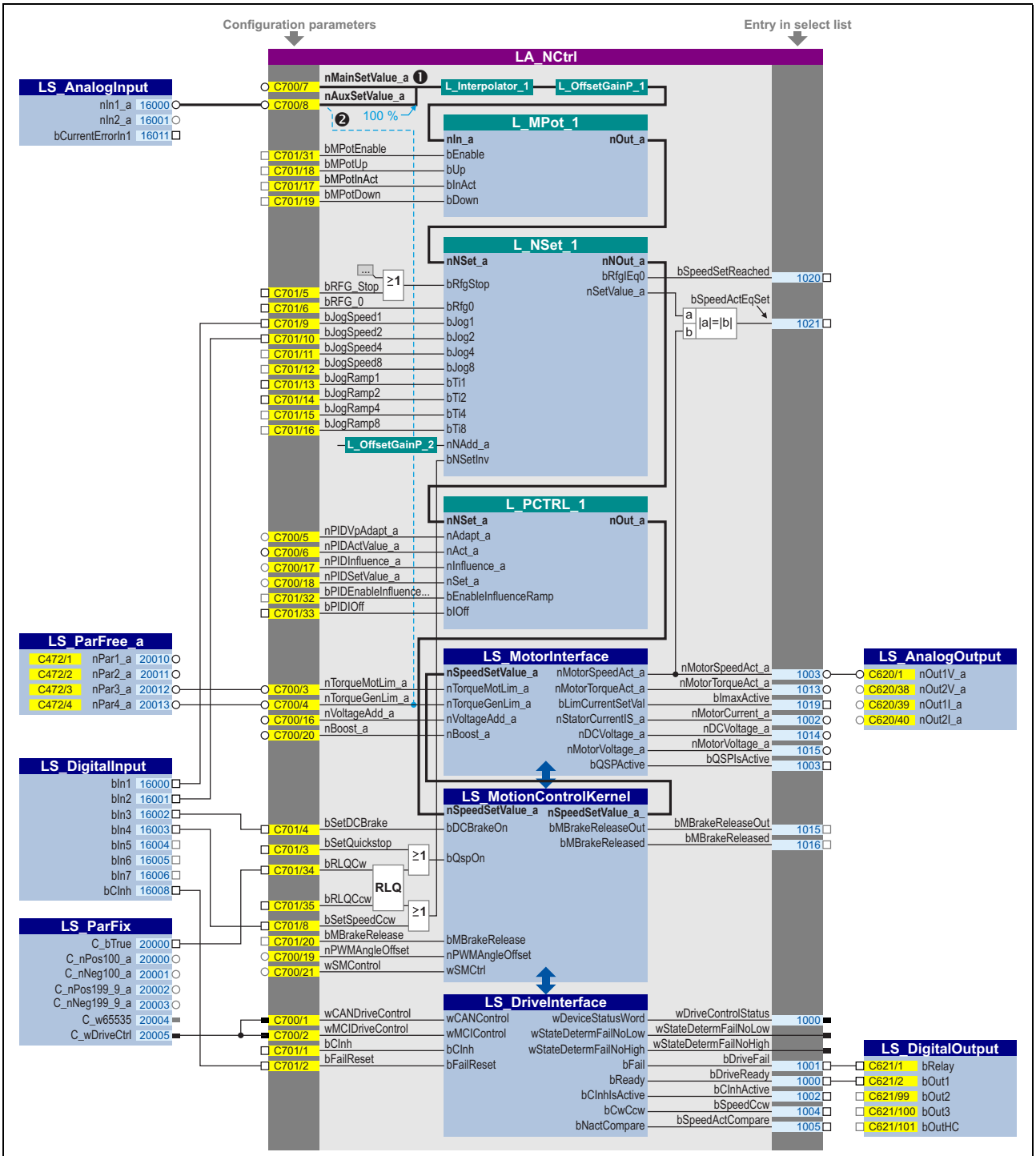
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00225 | L_PCTRL_1: MaxLimit | 199.99 | % |
| C00226 | L_PCTRL_1: MinLimit | -199.99 | % |
| C00227 | L_PCTRL_1: Acceleration time | 0.010 | s |
| C00228 | L_PCTRL_1: Deceleration time | 0.010 | s |
| C00233 | L_PCTRL_1: Root function | 0: Off | |
| C00241 | L_NSet_1: Hyst. NSet reached | 0.50 | % |
| C00242 | Operating mode of process controller | 0: Off | |
| C00243 | L_PCTRL_1: Accel. time influence | 5.000 | s |
| C00244 | L_PCTRL_1: Deceleration time influence | 5.000 | s |
| C00632/1 | L_NSet_1: Blocking speed 1 max | 0.00 | % |
| C00632/2 | L_NSet_1: Blocking speed 2 max | 0.00 | % |
| C00632/3 | L_NSet_1: Blocking speed 3 max | 0.00 | % |
| C00633/1 | L_NSet_1: Blocking speed 1 min | 0.00 | % |
| C00633/2 | L_NSet_1: Blocking speed 2 min | 0.00 | % |
| C00633/3 | L_NSet_1: Blocking speed 3 min | 0.00 | % |
| C00635 | L_NSet_1: nMaxLimit | 199.99 | % |
| C00636 | L_NSet_1: nMinLimit | -199.99 | % |
| C00670 | L_OffsetGainP_1: Gain | 1.0000 | |
| C00671 | L_OffsetGainP_2: Gain | 1.0000 | |
| C00672 | L_OffsetGainP_3: Gain | 1.0000 | |
| C00696 | L_OffsetGainP_1: Offset | 0.00 | % |
| C00697 | L_OffsetGainP_2: Offset | 0.00 | % |
| C00698 | L_OffsetGainP_3: Offset | 0.00 | % |
| C00800 | L_MPot_1: Upper limit | 100.00 | % |
| C00801 | L_MPot_1: Lower limit | -100.00 | % |
| C00802 | L_MPot_1: Acceleration time | 10.0 | s |
| C00803 | L_MPot_1: Deceleration time | 10.0 | s |
| C00804 | L_MPot_1: Inactive fct. | 0: Retain value | |
| C00805 | L_MPot_1: Init fct. | 0: Load last value | |
| C00806 | Use of motor potentiometer | 0: No | |

**Related topics:**

- "GeneralPurpose" functions (📖 554)

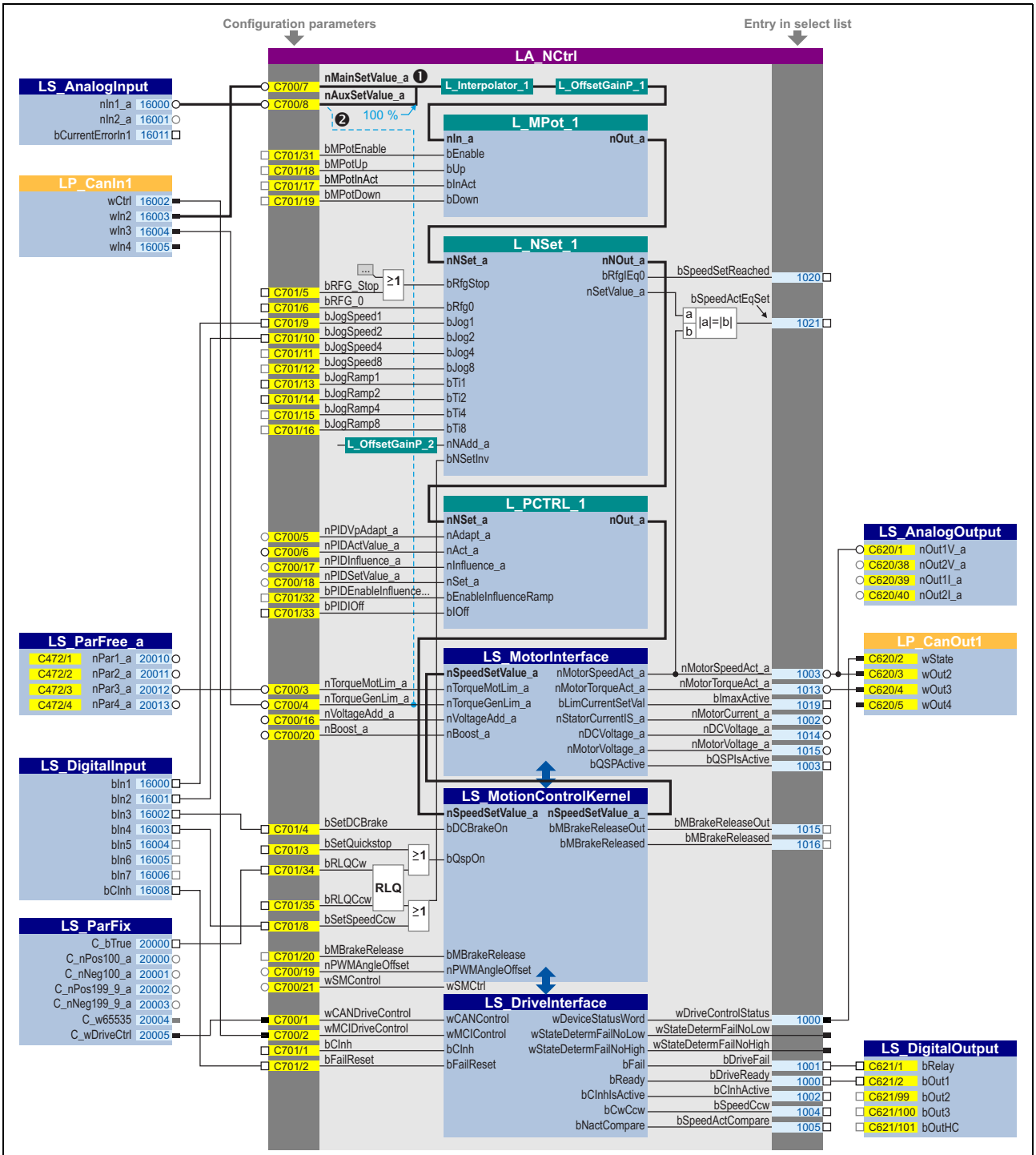
### 8.3.8 Configuration parameters

If required, the subcodes of C00700 and C00701 serve to change the pre-configured assignment of the application inputs:



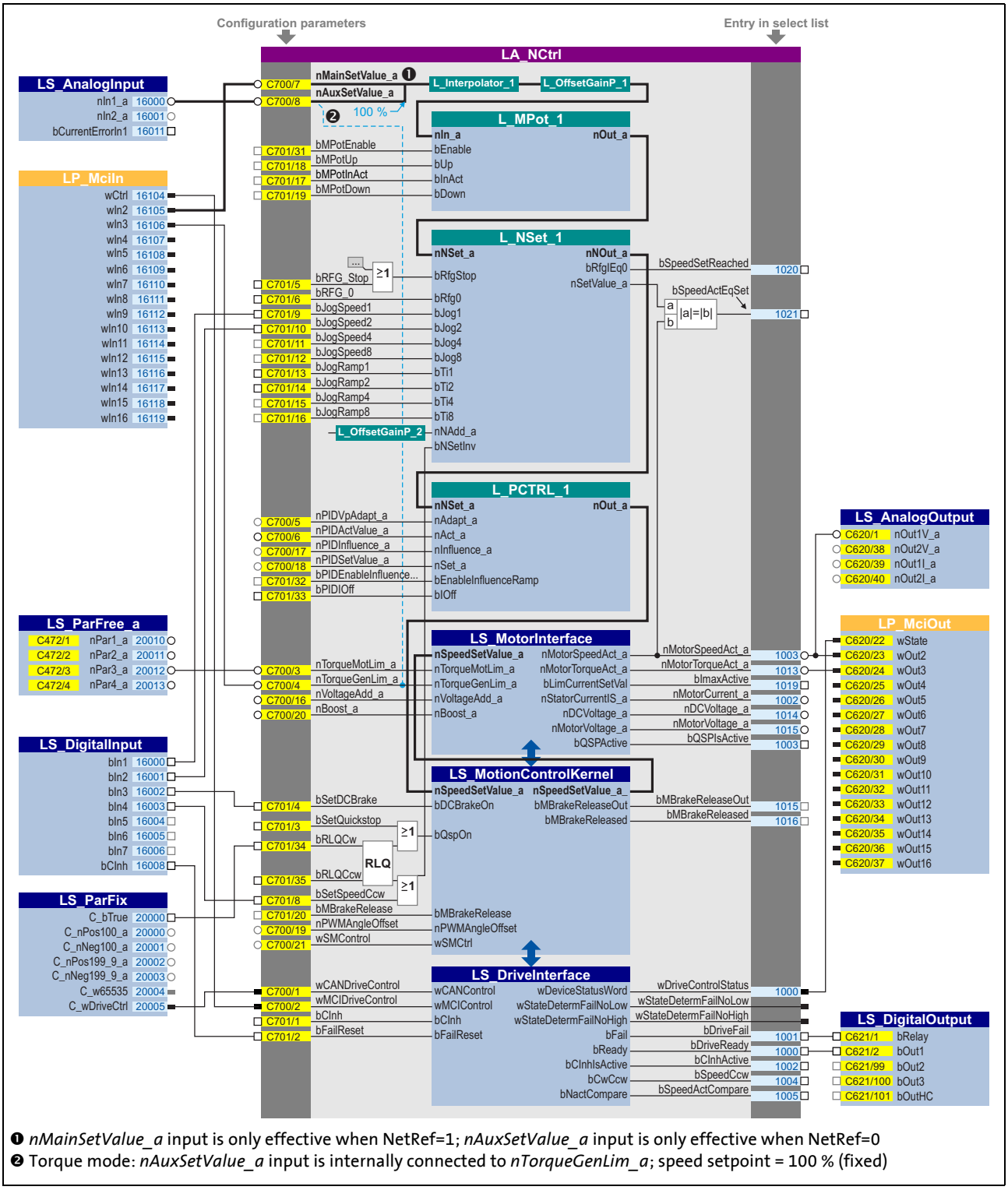

❶ *nMainSetValue_a* input is only effective when NetRef=1; *nAuxSetValue_a* input is only effective when NetRef=0
❷ Torque mode: *nAuxSetValue_a* input is internally connected to *nTorqueGenLim_a*; speed setpoint = 100 % (fixed)

[8-7] Pre-assignment of the "Actuating drive speed" (AC Drive Profile) application in the "Terminals 0" control mode




❶ *nMainSetValue_a* input is only effective when NetRef=1; *nAuxSetValue_a* input is only effective when NetRef=0
❷ Torque mode: *nAuxSetValue_a* input is internally connected to *nTorqueGenLim_a*; speed setpoint = 100 % (fixed)

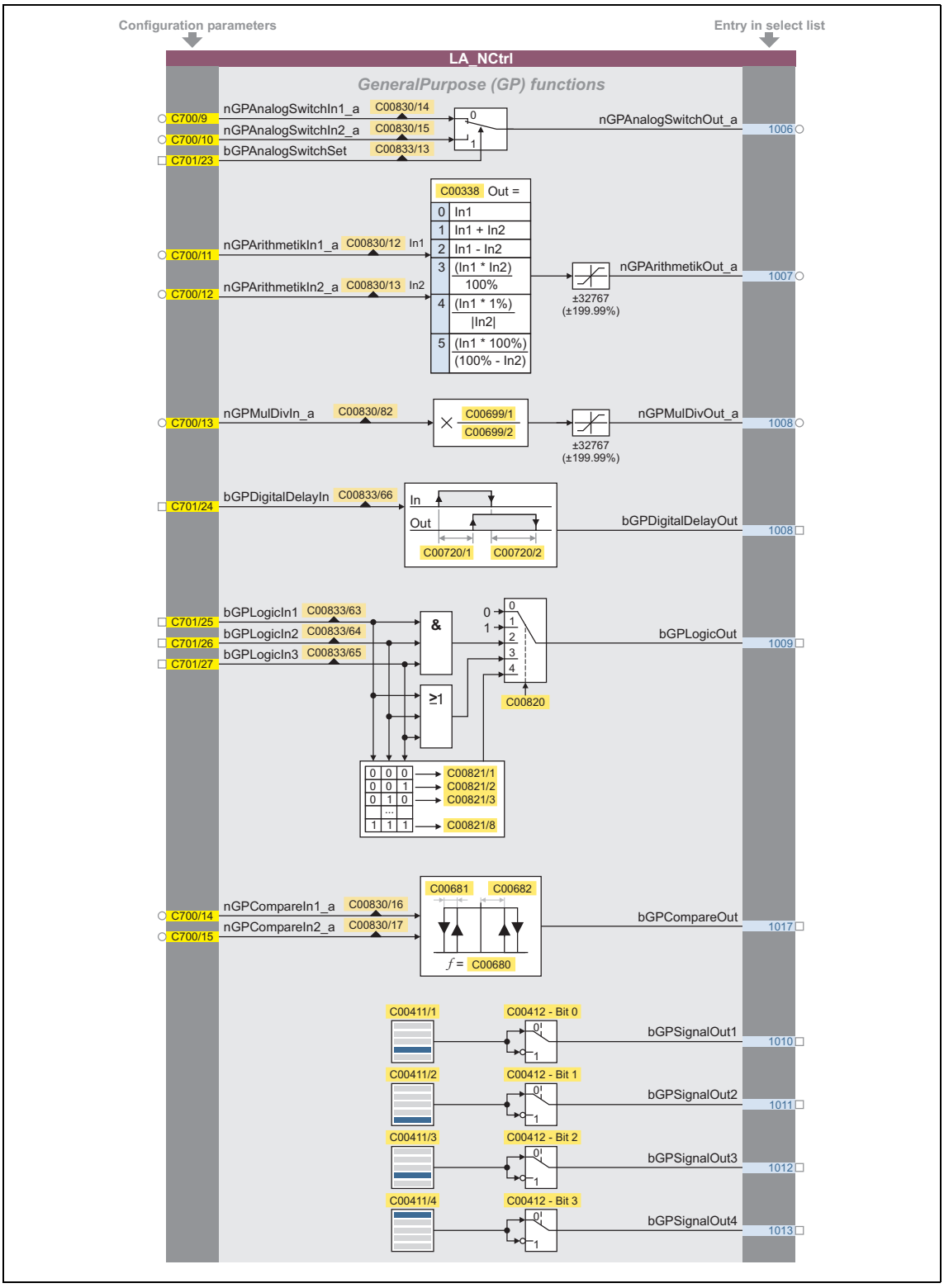
[8-8] Pre-assignment of the "Actuating drive speed" (AC Drive Profile) application in the "CAN " control mode



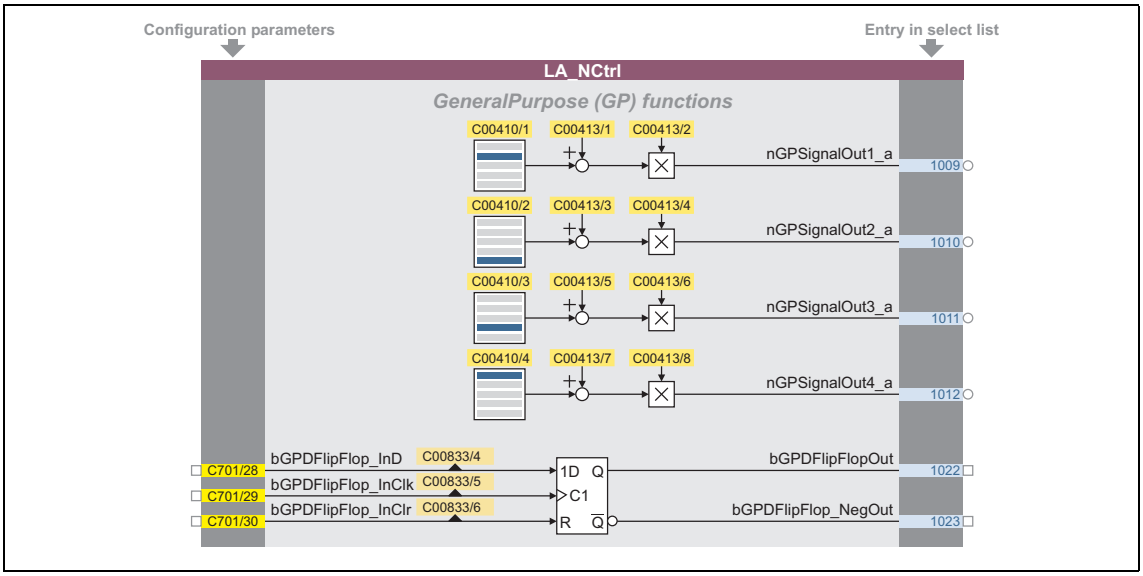

❶ *nMainSetValue_a* input is only effective when NetRef=1; *nAuxSetValue_a* input is only effective when NetRef=0
❷ Torque mode: *nAuxSetValue_a* input is internally connected to *nTorqueGenLim_a*; speed setpoint = 100 % (fixed)

[8-9] Pre-assignment of the "Actuating drive speed" (AC Drive Profile) application in the "MCI" control mode

### Configuration parameters for "GeneralPurpose" functions



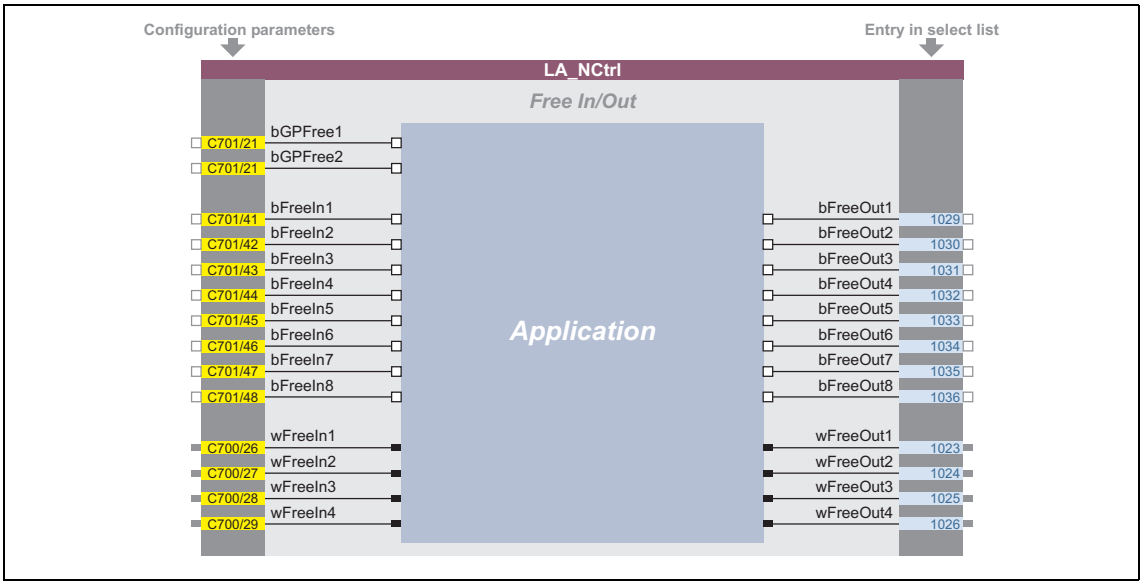

[8-10] "GeneralPurpose" functions




[8-11] "GeneralPurpose" functions (continuation)

### Free inputs and outputs

These inputs can be freely interconnected in the application level. They can be used to transfer signals from the I/O level to the application level and vice versa.




[8-12] Free inputs/outputs

### Related topics:

- User-defined terminal assignment (📖 429)
- "GeneralPurpose" functions (📖 554)

## 8.4 TA "Table positioning"

Numerous functions for the motion control of a single axis can be executed with the "Table positioning" technology application. For this, the technology application accesses the basic functions implemented in the **Motion Control Kernel** which are described in detail in the main chapter "Basic drive functions (MCK)". (📖 560)

- Manual jog
- Retracting from limit switches
- Reference setting/homing in 18 different modes (*in preparation*)
- Positioning (absolute, relative)
- Continuous travel (speed mode)
- Speed override
- Acceleration override
- Profile sequence block control
- Travel block restart within the profile (online profile start)
- Specification of target position via process signal interface (e.g. via fieldbus)

**Further functions**

- Up to 15 travel sets can be set via parameters
- Parameterisation of the profile data in physical units or relative values
- Adjustable torque limitation
- Freely selectable, variable ramp shape
- Motor potentiometer function
- Software limit position monitoring
- Following error monitoring (with static limits)
- Automatic holding brake control
- Quick stop (QSP) with adjustable ramp time
- Enable of individual functions via control word
- Status and diagnostic displays
- Operating mode changeover (manual jog, homing, speed follower, positioning)
- Integrated, freely available "GeneralPurpose" functions:
  Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-FlipFlop, counter
- From version 12.00.00, the application interconnection can be extended by an internal sequence control using the FB L_Sequencer_1 if required.

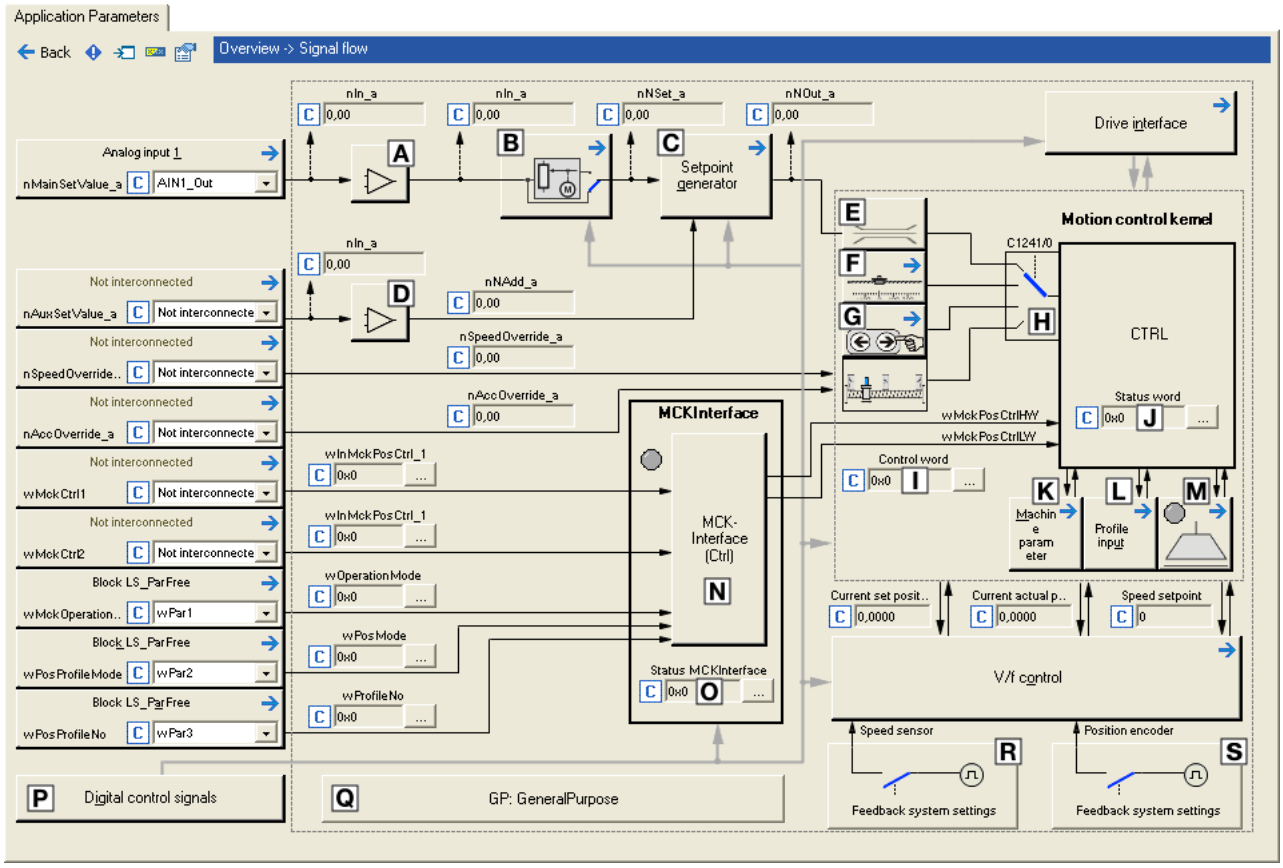
**Related topics:**

▸ Commissioning of the "Table positioning" technology application (📖 70)

### 8.4.1 Basic signal flow

The main element of the technology application is the **Motion Control Kernel**.

- The **Motion Control Kernel** is controlled via a 32-bit double control word. All motion profiles in the different operating modes can be controlled with this interface.
- Upstream to **the Motion Control Kernel** there is a so-called MCKInterface called which provides additional process inputs for the application in order to control the **Motion Control Kernel**.




[8-13] Signal flow of the table positioning

- A Offset and gain (L_OffsetGainP_1)
- B Motor potentiometer function (L_MPot_1)
- C Setpoint generator (L_NSet_1)
- D Offset and gain (L_OffsetGainP_2)
- E Speed setpoint input limitation
- F Basic function "Homing"
- G Basic function "Manual jog"
- H Operating mode changeover (is executed by the MCKInterface)
- I MCK control word
- J MCK status word
- K Selection of the machine parameters
- L Profile entry for the "Positioning" basic function.
- M Holding brake control
- N MCKInterface
- O Status word of the MCKInterface
- P Terminal assignment & display of digital control signals
- Q Integrated disposable "GeneralPurpose" functions: Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop

### 8.4.1.1 Possibilities for the position selection

The following options are available for the selection of different positions:

1. Changeover of profiles
2. Changing the profile position via parameter data (SDO)
3. Changing the profile position via process data (PDO)
   - The profile position is defined in [increments] or in the application unit [units].

| | Changeover of profiles | Changing the profile position via SDO | Changing the profile position via PDO | |
|---|---|---|---|---|
| **Number of positions** | max. 15 | n > 15 | n > 15 | |
| **Position selection** | Parameterisation in the profile | Selection via SDO | Selection via PDO | |
| **Unit** | units | units | Increments | units |
| **Remainder allowance** | in the drive | in the drive | in the PLC | in the drive |

**Note!**

Setting the machine parameters is a basic prerequisite for the operating modes "Homing", "Manual jog" and "Positioning".

The more precisely the machine parameters are set, the better the results of positioning!

▸ Machine parameters (📖 593)

**Possibility 1: Changeover of profiles**

A total of 15 profiles is available in the 8400 TopLine.

- A profile describes a motion task that can be converted into a rotary motion by the **Motion Control Kernel** in the "Positioning" operating mode.
- A detailed explanation of all profile parameters can be found in the subchapter "Profile entry". (📖 651) of the description of the "Positioning" basic drive function.
- The selection of the profile to be executed can either be executed as a data word via the input *wPosProfileNo* or binary coded via the inputs *bPosProfileNo_1* ... *bPosProfileNo_4*.
- The chosen profile is started via the process input *bPosExecute* of the FB L_MckCtrlInterface or the control bit "PosExecute" in the MCK control word.

**Possibility 2: Changing the profile position via parameter data (SDO)**

The parameter data channel of a fieldbus can also be used to change the position in a profile.

- The position of the profiles 1 ... 15 are stored in the subcodes 1 ... 15 of the code C01301.
- More than 15 positions are available via the profile changeover (as described under possibility 1).

**Possibility 3: Changing the profile position via process data (PDO)**

For selecting the position in [increments], the *dnPosProfilePosition_p* input at the application block is available.

**Note!**

For selecting the position in [units], the two inputs *wPosProfileUnitsLW* and *wPosProfileUnitsHW* at the application block are available.

- In the application level, the position in the FB L_MckCtrlInterface is converted from [units] to [increments] again and output to *dnPosSetOut_p*. For this purpose, the respective mode for position calculation must be selected in C01296/1.
- Since the conversion is not executed within one controller cycle, the data at the *dnPosSetOut_p* output can be out-of-date. The *bPosSetDataValid* output is set to TRUE when the conversion is completed and the position in increments is valid. Then, the position can be transmitted to the profile (see the following section).

The acceptance of the position into the currently selected profile is executed with a FALSE-TRUE edge at the *bPosSetProfilePosition* input.

- If bit 2 is set in C01297, the applied setpoint position is accepted into the currently selected profile and the profile is started directly when setting the process input *bPosExecute* to TRUE or setting the control bit "PosExecute" in the MCK control word.
- If bit 3 is set in C01297, the setpoint positions at the MCKInterface are automatically accepted into the profile with the applied profile number if a change of data is detected at the corresponding input for the setpoint position.
- The accepted position is stored in the code C01301/x in the application unit [units]. By reading out the code, you can check if the position was correctly accepted into the profile.

**Note!**

How to store changed profile parameters safely against mains failure in the memory module: Set C00002/11 = "1: On / Start".

### 8.4.2 Internal interfaces | application block "LA_TabPos"

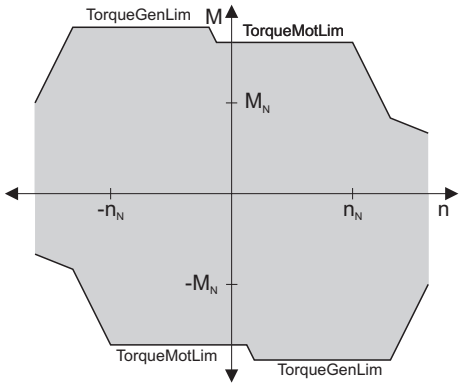
**Note!**

The connectors greyed out in the following table are hidden in the function block editor in the Lenze setting.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wCANDriveControl<br>WORD | Control word via system bus (CAN) for device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. | |
| wMCIDriveControl<br>WORD | Control word via communication module (e.g. PROFIBUS) for device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. | |
| wSMControl<br>WORD | Interface to the optional safety system.<br>• Setting control bit 0 ("SafeStop1") in this control word causes e.g. the automatic deceleration of the drive to standstill within this application (in the **Motion Control Kernel**).<br>• See the subchapter "Interface to safety system" of the chapter on basic drive functions for a detailed description of the individual control bits. | |
| bCInh<br>BOOL | Enable/inhibit inverter | |
| | FALSE | Enable inverter: The inverter switches to the "OperationEnabled" device status if no other source for controller inhibit is active.<br>• C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit. |
| | TRUE | Inhibit inverter (controller inhibit): The inverter switches to the "SwitchedOn" device status. |
| bFailReset<br>BOOL | Reset error message<br>In the Lenze setting this input is connected to the digital input controller enable so that a possibly existing error message is reset together with the controller enable (if the cause for the fault is eliminated). | |
| | TRUE | The current fault is reset, if the cause for the fault is eliminated.<br>• If the fault still exists, the error status remains unchanged. |
| bSetQuickstop<br>BOOL | Activate quick stop (QSP)<br>• Also see device command "Activate/deactivate quick stop". | |
| | TRUE | Activate quick stop<br>• Motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0).<br>• The motor is kept at a standstill during closed-loop operation.<br>• A pulse inhibit is set if the auto-DCB function has been activated via C00019. |
| | FALSE | Deactivate quick stop<br>• The quick stop is deactivated if no other source for the quick stop is active.<br>• C00159 displays a bit code of active sources/causes for the quick stop. |

| Designator<br>Data type | Information/possible settings |
|---|---|
| nTorqueMotLim_a<br>nTorqueGenLim_a<br>INT | Torque limitation in motor mode and in generator mode<br>• These input signals are directly transferred to the motor control to limit the inverter's maximum torque in motor and generator mode.<br>• The drive cannot output a higher torque in motor/generator mode than set here.<br>• The applied values (any polarity) are internally interpreted as absolute values.<br>• If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.<br>• If sensorless vector control (SLVC) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>**Note:**<br>Setting this input is ineffective in the reference modes 14 and 15 ("Homing to positive stop"). |
| | Torque limits in motor and generator mode:<br>  <br> |
| bSetSpeedCcw<br>BOOL | Change of direction of rotation<br>• For instance if a motor or gearbox is fixed laterally reversed to a machine part, but the setpoint selection should still be executed for the positive direction of rotation. |
| | FALSE: Clockwise rotation (Cw) |
| | TRUE: Direction of rotation to the left (Ccw) |
| nMainSetValue_a<br>INT | Main speed setpoint<br>• Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The main setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.<br>• Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.<br>• For a detailed functional description see the L_NSet FB. |
| nAuxSetValue_a<br>INT | Additional speed setpoint<br>• Offset and gain of this input signal can be set in C00697 and C00671 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The additional speed setpoint can be linked arithmetically with the main speed setpoint behind the ramp function generator.<br>• The additional speed setpoint can be shown via ramp times of a second ramp function generator.<br>• For a detailed functional description see the L_NSet FB. |
| bJogSpeed1<br>bJogSpeed2<br>BOOL | Inputs for overriding fixed setpoints (JOG setpoints) for the main setpoint<br>• Selection inputs are binary coded. |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bJogSpeed1<br>bJogSpeed2<br>BOOL | Inputs for overriding fixed setpoints (JOG setpoints) for the main setpoint<br>• A fixed setpoint for the setpoint generator can be activated instead of the main setpoint via these selection inputs.<br>• The two selection inputs are binary coded, therefore you can select 3 fixed setpoints.<br>• In the case of binary coded selection "0" (all inputs = FALSE or not assigned), main setpoint *nMainSetValue_a* is active.<br>• The selection of the fixed setpoints is carried out in C00039/1...3 in [%] based on the reference speed (C00011).<br>• For a detailed functional description see the L_NSet FB. | |
| **Motor potentiometer**<br>Alternatively to the input signal *nMainSetValue_a*, the main setpoint can also be generated by a motor potentiometer function.<br>• In the Lenze setting, the motor potentiometer function is deactivated.<br>• Activation is possible via C00806 or via the *bMPotEnable* input.<br>• The behaviour of the motor potentiometer during switch-on of the drive system can be selected in C00805.<br>• For a detailed functional description see the L_MPot FB. | | |
| bMPotEnable<br>BOOL | Activating the motor potentiometer function<br>• This input and C00806 are OR'd. | |
| | TRUE | The motor potentiometer function is active; the speed setpoint can be changed via the *bMPotUp* and *bMPotDown* control inputs. |
| bMPotUp<br>BOOL | Increasing the speed setpoint | |
| | TRUE | Approach the upper speed limit value set in C00800 with the acceleration time set in C00802. |
| bMPotDown<br>BOOL | Decreasing the speed setpoint | |
| | TRUE | Approach the lower speed limit value set In C00801 with the deceleration time set in C00803. |
| **MCK basic functions** | | |
| bMBrakeRelease<br>BOOL | Holding brake control: Release/apply brake<br>• In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off"). | |
| | FALSE | Apply brake.<br>• During automatic operation, the internal brake logic controls the brake. |
| | TRUE | Release brake manually (forced release).<br>• **Note!** The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If a controller inhibit has been set by the brake control, it will be deactivated.<br>• In semi-automatic operation, the brake is released including feedforward control. |
| wMckCtrl1<br>wMckCtrl2<br>WORD | Direct selection of MCK control words<br>• E.g. via a master control which has been connected to the fieldbus, too. For this purpose, the control word inputs can directly be connected to the **LP_McIIn** or **LP_CanIn** fieldbus interface.<br>• See the "MCK control word" chapter for a detailed description of the individual control bits. | |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wMckOperationMode<br>WORD | Selection of the operating mode of the **Motion Control Kernel**<br>• Only bit 0 ... bit 3 of *wMckOperationMode* is evaluated.<br>• If an invalid operating mode is selected, the response set in C00595/11 is activated (Lenze setting: "Warning").<br>• The current operating mode is displayed in C01243. | |
| | 0 | Speed follower |
| | 1 | Homing |
| | 2 | Manual jog |
| | 3 | Positioning |
| | 4 | Stop |
| | 5 | Position follower |
| | 6 ... 15 | Reserved for future extensions |
| bMckOperationMode_1<br>...<br>bMckOperationMode_8<br>BOOLBOOL | Binary-coded selection of the operating mode of the **Motion Control Kernel**<br>• See the "MCK control word" chapter for a detailed description of the individual control bits.<br>• If an invalid operating mode is selected, the response set in C00595/11 is activated (Lenze setting: "Warning").<br>• The current operating mode is displayed in C01243. | |
| bPosCtrlOn<br>BOOL | Position control/Angle control<br>• Pre-configured assignment: TRUE (position control active) | |
| | FALSE | Deactivate position control/angle control |
| | TRUE | Activate position control/angle control |
| nPosCtrlOutLimit_a<br>INT | Limitation of the position controller output<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• Pre-configured assignment: 100 % | |
| nPosCtrlPAdapt_a<br>INT | Adaptation of the position controller gain<br>• Scaling: 16384 ≡ 100 % Vp (C00254)<br>• Pre-configured assignment: 100 % | |
| bLimitSwitchPos<br>bLimitSwitchNeg<br>BOOL | Limit position monitoring: Inputs for positive/negative limit switch | |
| bReleaseLimitSwitch<br>BOOL | Manual jog: Retract operated limit switch | |
| | TRUE | Retract operated limit switch (in opposite direction) |
| bManJogPos<br>bManJogNeg<br>BOOL | Manual jog:<br>*bManJogPos* = TRUE: Manual jog right<br>*bManJogNeg* = TRUE: Manual jog left<br>Both inputs = TRUE: No change compared to previous state<br>Both inputs = FALSE: Stop manual jog | |
| bManJogExecute2ndVel<br>BOOL | Manual jog: Changeover to speed 2 | |
| | FALSE | Speed 1 (C01231/1) active |
| | TRUE | Speed 2 (C01231/2) active |
| bEnableSpeedOverride<br>BOOL | Speed override | |
| | TRUE | Activate speed override |
| nSpeedOverride_a<br>INT | Value for Speed override<br>• Percentage multiplier for the currently active speed.<br>• 16384 ≡ 100 % of the maximum traversing speed (display in C01211/1).<br>• If the override value is 0 %, the drive is brought to a standstill. | |
| bEnableAccOverride<br>BOOL | Acceleration override | |
| | TRUE | Activate acceleration override |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nAccOverride_a<br>INT | Value for Acceleration override<br>• Percentage multiplier for the currently active acceleration.<br>• 16384 ≡ 100 % of the parameterised acceleration of the corresponding operating mode.<br>• If the override value is 0 %, acceleration ceases. | |
| bHomeStartStop<br>BOOL | Homing: Start/stop homing | |
| | TRUE | Start homing |
| | TRUE↘FALSE | Stop homing |
| bHomeSetPosition<br>BOOL | Homing: Set home position | |
| | TRUE | Set home position |
| bHomeResetPosition<br>BOOL | Homing: Reset home position | |
| | TRUE | Reset the "Reference known" status |
| bHomeMark<br>BOOL | Homing: Input for reference switch (pre-switch off mark)<br>• This input responds to the FALSE status (fail-safe) and is to be connected to the corresponding digital input to which the reference switch is connected. | |
| bPosSetProfilePosition<br>BOOL | Position teaching: MCK setpoint position | |
| | TRUE | Teach MCK setpoint position into the selected profile. |
| bPosSetActualPosition<br>BOOL | Position teaching: Current position | |
| | TRUE | Teach current position into the selected profile. |
| bPosExecute<br>BOOL | Positioning: Start travelling | |
| | FALSE↗TRUE | Execute selected profile |
| bPosFinishTarget<br>BOOL | Positioning: Complete cancelled profile | |
| | FALSE↗TRUE | A positioning process previously cancelled, e.g. by *bPosStop* or due to a device error, is resumed by travelling to the original target. |
| bPosDisableFollowProfile<br>BOOL | Positioning: Do not execute sequence profile (switch-off profile linkage) | |
| | TRUE | Evaluation of the sequence profile number parameterised in C01307/1...15 for the selected profile is suppressed. |
| bPosStop<br>BOOL | Positioning: Cancel travelling | |
| | TRUE | Stop positioning<br>From version 02.00.00, more travel requests will be inhibited ("PosExecute" will be blocked). |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wPosProfileMode<br>WORD | Override of the positioning mode set in the profile data<br>• Via this input, an override of the positioning mode parameterised in C01300/1...15 for the selected profile is possible.<br>• The value set in C01300/1...15 is not overwritten in this case.<br>• Only bit 0 ... bit 3 of *wPosMode* are evaluated.<br>• If *wPosMode* = 0 is selected, the positioning mode set in C01300/1...15 is used. | |
| | 0 | Positioning mode = setting in C01300/1...15 |
| | 1 | Absolute (shortest path) |
| | 2 | Continuous |
| | 3 | Relative |
| | 4 | absolute (Cw) |
| | 5 | absolute (Ccw) |
| | 8 | Absolute (shortest path) to TP |
| | 9 | Continuous to TP |
| | 10 | Relative to TP |
| | 11 | Absolute (Cw) on TP |
| | 12 | Absolute (Ccw) on TP |
| | All other possible settings are reserved for future extensions! | |
| wPosProfileNo<br>WORD | Stipulation of the profile to be executed<br>• Optionally as a data word or binary coded. | |
| bPosProfileNo_1<br>...<br>bPosProfileNo_8<br>BOOL | | |
| dnPosProfilePosition_p<br>DINT | Selection of the target position in [increments]<br>• The mode for calculating the position is selected in C01296/1. | |
| wPosProfileUnitsLW<br>wPosProfileUnitsHW<br>WORD | Selection of the target position in [units]<br>• *wPosProfileUnitsLW* = Low word, *wPosProfileUnitsHW* = High word<br>• The mode for calculating the position is selected in C01296/1. | |
| **GP: GeneralPurpose**<br>The following inputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | | |
| nGPAnalogSwitchIn1_a<br>nGPAnalogSwitchIn2_a<br>INT | Analog switch: Input signals<br>• The input signal selected via the selection input *bGPAnalogSwitchSet* is output at output *nGPAnalogSwitchOut_a*. | |
| bGPAnalogSwitchSet<br>BOOL | Analog switch: Selection input | |
| | FALSE | *nGPAnalogSwitchOut_a* = *nGPAnalogSwitchIn1_a* |
| | TRUE | *nGPAnalogSwitchOut_a* = *nGPAnalogSwitchIn2_a* |
| nGPArithmetikIn1_a<br>nGPArithmetikIn2_a<br>INT | Arithmetic: Input signals<br>• The arithmetic function is selected in C00338.<br>• The result is output at output *nGPArithmetikOut_a*. | |
| nGPMulDivIn_a<br>INT | Multiplication/Division: Input signal<br>• The factor for the multiplication can be set in C00699/1 (numerator) and C00699/2 (denominator).<br>• The result is output at output *nGPMulDivOut_a*. | |
| bGPDigitalDelayIn<br>BOOL | Binary delay element: Input signal<br>• The on-delay can be set in C00720/1.<br>• The off-delay can be set in C00720/2.<br>• The time-delayed input signal is output at output *bGPDigitalDelayOut*. | |

| Designator<br>Data type | Information/possible settings |
|---|---|
| bGPLogicIn1<br>bGPLogicIn2<br>bGPLogicIn3<br>BOOL | Binary logic: Input signals<br>• The logic operation is selected in C00820.<br>• The result is output at output *bGPLogicOut*. |
| nGPCompareIn1_a<br>nGPCompareIn2_a<br>INT | Analog comparison: Input signals<br>• The comparison operation is selected in C00680.<br>• Hysteresis and window size can be set in C00680 and C00682.<br>• If the comparison statement is true, the output *bGPCompareOut* will be set to TRUE. |
| bGPDFlipFlop_InD<br>bGPDFlipFlop_InClk<br>bGPDFlipFlop_InClr<br>BOOL | D-FlipFlop: Input signals<br>• Data, clock and reset input |
| bGPCounter1ClkUp<br>BOOL | Counter: Clock input<br>• With each edge, the module counts up by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bGPCounter1ClkDown<br>BOOL | Counter: Clock input<br>• With each edge, the module counts down by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bGPCounter1Load<br>BOOL | Counter: Loading input<br>• The input has the highest priority.<br>TRUE: Accept the starting value *wGPCounter1LdVal*. |
| wGPCounter1LdVal<br>WORD | Counter: Starting value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |
| wGPCounter1CmpVal<br>WORD | Counter: Comparison value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |
| **Free inputs**<br>The following inputs can freely be interconnected on the application level.<br>The signals can be transferred from the I/O level to the application level via these inputs. | |
| bFreeIn1 ... bFreeIn8<br>BOOL | Free inputs for digital signals |
| wFreeIn1 ... wFreeIn4<br>WORD | Free inputs for 16-bit signals |
| dnFreeIn1_p ... dnFreeIn2_p<br>DINT | Free inputs for 32-bit signals |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wDriveControlStatus<br>WORD | Status word of the inverter<br>• The status word contains information on the currents status of the inverter.<br>• See the "wDeviceStatusWord status word" subchapter of the chapter on device control for a detailed description of the bit assignment. |
| wStateDetermFailNoLow<br>WORD | Display of the status determining error (LOW word) |
| wStateDetermFailNoHigh<br>WORD | Display of the status determining error (HIGH word) |
| bDriveFail<br>BOOL | TRUE: Inverter in error status.<br>• "Fault" device status is active. |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bDriveReady<br>BOOL | TRUE | Inverter is ready for operation.<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the inverter is still inhibited by the user (controller inhibit). |
| bCInhActive<br>BOOL | TRUE | Controller inhibit is active. |
| bQSPIsActive<br>BOOL | TRUE | Quick stop is active. |
| bSpeedCcw<br>BOOL | Current direction of rotation | |
| | FALSE | Clockwise rotation (Cw) |
| | TRUE | Direction of rotation to the left (Ccw) |
| bSpeedActCompare<br>BOOL | Result of the speed comparison (detection of speed=0) | |
| | TRUE | During open-loop operation:<br>Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>Actual speed value < Comparison value (C00024) |
| bImaxActive<br>BOOL | "Current setpoint inside the limitation" status signal | |
| | TRUE | The current setpoint is internally limited (the inverter operates at the maximum current limit). |
| bSpeedSetReached<br>BOOL | Status signal "setpoint = 0" | |
| | TRUE | Speed setpoint from the ramp function generator = 0 |
| nMotorCurrent_a<br>INT | Current stator current/effective motor current<br>• Scaling: 16384 ≡ 100 % $I_{max_mot}$ (C00022) | |
| nMotorSpeedSet_a<br>INT | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorSpeedAct_a<br>INT | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorTorqueAct_a<br>INT | Actual torque<br>• In the "VFC (+encoder)" operating mode of the motor control, this value is determined from the current motor current and corresponds to the actual torque only by approximation.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| nDCVoltage_a<br>INT | Actual DC-bus voltage<br>• Scaling: 16384 ≡ 1000 V | |
| nMotorVoltage_a<br>INT | Current motor voltage/inverter output voltage<br>• Scaling: 16384 ≡ 1000 V | |
| **MCK basic functions** | | |
| bMBrakeReleaseOut<br>BOOL | Holding brake control: Trigger signal for the holding brake control switching element via a digital output<br>• Use bit 0 in C02582 to activate inverted switching element triggering. | |
| | FALSE | Apply brake. |
| | TRUE | Release brake. |
| bMBrakeReleased<br>BOOL | Holding brake control: "Brake released" considering the brake release time<br>• When the holding brake is triggered to close, *bMBrakeReleased* is immediately set to FALSE even if the brake closing time has not yet elapsed! | |
| | TRUE | Brake released (after the brake release time has expired). |
| wMckState1<br>wMckState2<br>WORD | Output of the MCK status words<br>• For a detailed description of each status bit see chapter "MCK status word". | |
| wMckActOperationMode<br>WORD | Active setpoint-generating state of the MCK.<br>• Bit B0 ... B3 contain the information of the MCK status word.<br>• Bits B4 ... B15 are fixed at "0". | |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bHomeDone<br>BOOL | TRUE | Homing has been executed. |
| bHomePosAvailable<br>BOOL | TRUE | Home position is known. |
| bProfileDone<br>BOOL | TRUE | Target position from the profile has been approached. |
| bProfileBusy<br>BOOL | TRUE | Profile positioning is active. |
| bAccelerating<br>BOOL | TRUE | Acceleration phase active. |
| bConstantDuty<br>BOOL | TRUE | Constant phase active. |
| bDecelerating<br>BOOL | TRUE | Braking phase active. |
| bDwellTime<br>BOOL | TRUE | Settling in target position is active |
| bInTarget<br>BOOL | TRUE | Target position (actual value) is in the target window. |
| wActProfileNo<br>WORD | Current profile number<br>• Bit B0 ... B7 contain the information of the MCK status word.<br>• Bits B8 ... B15 are fixed at "0". | |
| wActPosMode<br>WORD | Current positioning mode<br>• Bit B0 ... B3 contain the information of the MCK status word.<br>• Bits B4 ... B15 are fixed at "0". | |
| dnTargetPos_p<br>DINT | Target position in [increments]<br>• 65535 ≡ 1 revolution of the motor shaft | |
| dnSetPos_p<br>DINT | Absolute position setpoint | |
| dnPosAct_p<br>DINT | Current position of the motor shaft in [increments] | |
| dnDeltaPosAct_p<br>DINT | Current following error in [increments]<br>• Following error = Difference between set position and actual position | |
| wPosOutUnitsLW<br>wPosOutUnitsHW<br>WORD | Output of the target position in [units]<br>• *wPosOutUnitsLW* = Low-Word, *wPosOutUnitsHW* = High-Word<br>• The mode for calculating the position is selected in C01296/2. | |

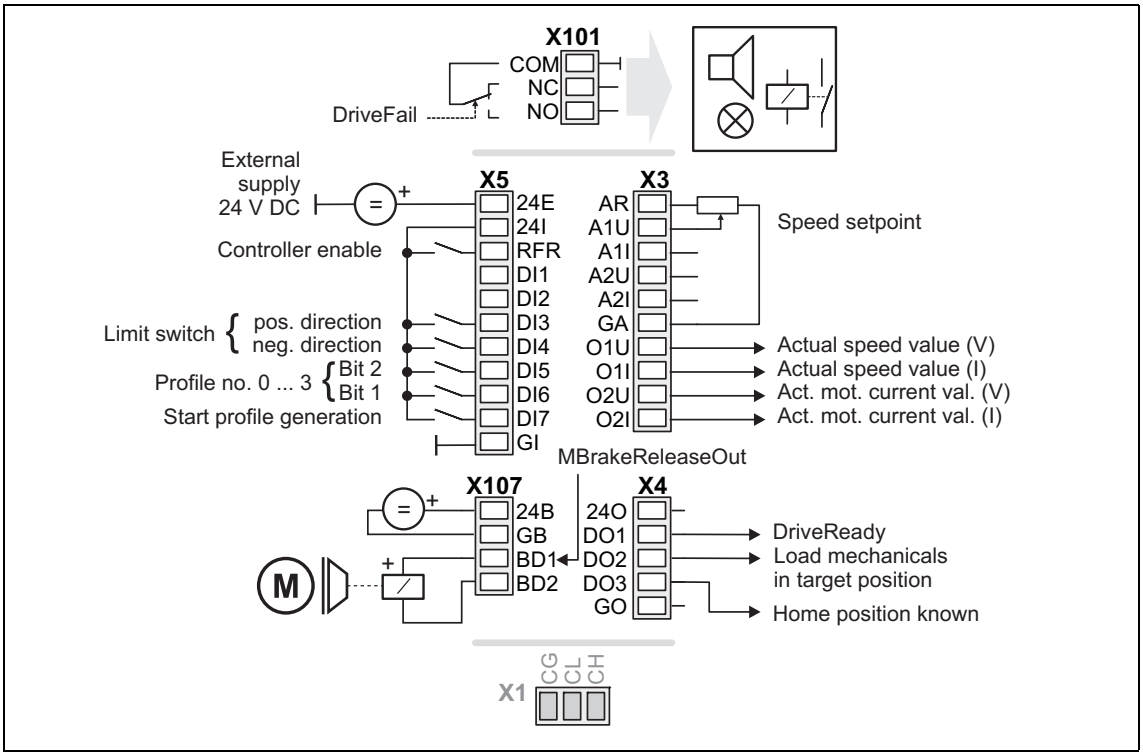
| Designator<br>Data type | Value/meaning |
|---|---|
| **GP: GeneralPurpose**<br>The following outputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | |
| nGPAnalogSwitchInOut_a<br>INT | Analog switch: Output signal |
| nGPArithmetikOut_a<br>INT | Arithmetic: Output signal |
| nGPMulDivOut_a<br>INT | Multiplication/Division: Output signal |
| bGPDigitalDelayOut<br>BOOL | Binary delay element: Output signal |
| bGPLogicOut<br>BOOL | Binary logic: Output signal |
| bGPCompareOut<br>BOOL | Analog comparison: Output signal |
| bGPSignalOut1<br>...<br>bGPSignalOut4<br>BOOL | Binary signal monitor: Output signals<br>• The signal sources to be output are selected in C00411/1...4.<br>• A bit coded inversion of the output signals can be parameterised in C00412. |
| nGPSignalOut1_a<br>...<br>nGPSignalOut4_a<br>BOOL | Analog signal monitor: Output signals<br>• The signal sources to be output are selected in C00410/1...4.<br>• Gain and offset for each output signal can be parameterised in C00413/1...8. |
| bGPDFlipFlop_Out<br>BOOL | D-FlipFlop: Output signal |
| bGPDFlipFlop_NegOut<br>BOOL | D-FlipFlop: Negated output signal |
| bGPCounter1Equal<br>BOOL | Counter: Status signal "Comparison value reached"<br>TRUE \| Comparison value *wGPCounter1CmpVal* reached. |
| wGPCounter1Out<br>WORD | Counter: Counter content<br>• Internal limitation to ± 32767<br>• The most significant bit determines the sign! |
| **Free outputs**<br>The following outputs can freely be interconnected on the application level.<br>The signals from the application level can be transferred to the I/O level via these outputs. | |
| bFreeOut1 ... bFreeOut8<br>BOOL | Free outputs for digital signals |
| wFreeOut1 ... wFreeOut4<br>WORD | Free outputs for 16-bit signals |
| dnFreeOut1_p<br>dnFreeOut2_p<br>WORD | Free outputs for 32-bit signals |

## 8.4.3 Terminal assignment of the control modes

The following comparison provides information about which inputs/outputs of the application block **LA_TabPos** are interconnected to the digital and analog input/output terminals of the inverter in the different control modes.

| | Control mode (C00007) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 10: Terminals 0 | 12: Terminals 2 | 14: Terminals 11 | 16: Terminal 16 | 20: Keypad | 21: PC | 30: CAN | 40: MCI |
| **Digital input terminals** | | | | | | | | |
| **X5/RFR** | **Controller enable / Reset of error message**<br>bFailReset | | | | | | | |
| **X5/DI1** | - | - | **Positive limit switch**<br>bLimitSwitchPos | - | - | - | - | - |
| **X5/DI2** | - | - | **Negative limit switch**<br>bLimitSwitchNeg | - | - | - | - | - |
| **X5/DI3** | **Positive limit switch**<br>bLimitSwitchPos | - | - | **Manual jog in positive direction**<br>bManJogPos | - | - | **Positive limit switch**<br>bLimitSwitchPos | |
| **X5/DI4** | **Negative limit switch**<br>bLimitSwitchNeg | **Positioning profile, array bit (valency 4)**<br>bPosProfileNo_4 | | **Manual jog in negative direction**<br>bManJogNeg | - | - | **Negative limit switch**<br>bLimitSwitchNeg | |
| **X5/DI5** | **Positioning profile, array bit (valency 2)**<br>bPosProfileNo_2 | | | | - | - | **Home position**<br>bHomeMark | |
| **X5/DI6** | **Positioning profile, array bit (valency 1)**<br>bPosProfileNo_1 | | | | - | - | **Quick stop**<br>bSetQuickstop | |
| **X5/DI7** | **Start of profile generation**<br>bPosExecute | | | | - | - | **Status word - bit 7**<br>LP_CanOut1: bState_B7 | **Status word - bit 7**<br>LP_MciOut: bState_B7 |
| **Analog input terminals** | | | | | | | | |
| **X3/A1U**<br>**X3/A1I** | **Main speed setpoint**<br>nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) | | | | - | - | **Main speed setpoint**<br>nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) | |
| **X3/A2U** | - | - | - | - | - | - | - | - |
| **X3/A2I** | - | - | - | - | - | - | - | - |
| **Digital output terminals** | | | | | | | | |
| **X4/DO1** | **Status "Drive is ready"** (bDriveReady) | | | | | | | |
| **X4/DO2** | **Status "Target position (actual value) is in the target window"** (bInTarget) | | | | | | | |
| **X4/DO3** | **Status "Home position is known"**(bHomePosAvailable) | | | | | | **Status "Maximum current limit"** (bImaxActive) | |
| **X107/BD1, BD2** | **Control of the holding brake** (bMBrakeReleaseOut) | | | | | | | |
| **X101/COM, NO** | **Status "Error is pending"** (bDriveFail) | | | | | | | |
| **Analog output terminals** | | | | | | | | |
| **X3/O1U**<br>**X3/O1I** | **Actual speed value**<br>nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) | | | | | | | |
| **X3/O2U**<br>**X3/O2I** | **Actual motor current**<br>nMotorCurrent_a<br>10V ≡ 100% of $I_{max_mot}$ (C00022) | | | | | | | |

### 8.4.3.1 Terminals 0

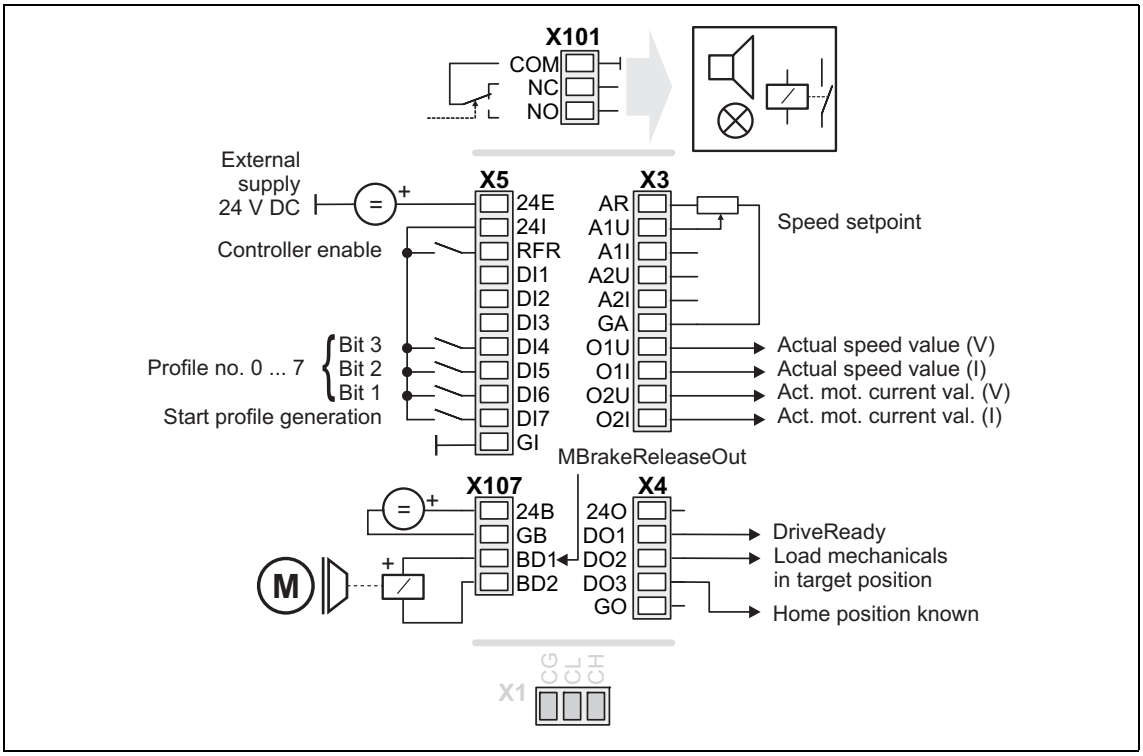



| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_TabPos.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_TabPos.bFailReset | | **X3/A1U** | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | - | | **X3/A1I** | |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | LA_TabPos.bLimitSwitchPos | | **X3/A2I** | - |
| **X5/DI4** | LA_TabPos.bLimitSwitchNeg | | **X3/O1U** | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI5** | LA_TabPos.bPosProfileNo_2 | | **X3/O1I** | |
| **X5/DI6** | LA_TabPos.bPosProfileNo_1 | | **X3/O2U** | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max_mot}$ (C00022) |
| **X5/DI7** | LA_TabPos.bPosExecute | | **X3/O2I** | |
| | | | | |
| **X107/BD1** | LA_TabPos.bMBrakeReleaseOut | | **X4/DO1** | LA_TabPos.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | LA_TabPos.bInTarget |
| | | | **X4/DO3** | LA_TabPos.bHomePosAvailable |

When the profile is defined, the operating mode in the Lenze setting is changed simultaneously:

| bPosProfileNo_2 (DI5) | bPosProfileNo_1 (DI6) | Selected profile | Activation of operating mode |
|---|---|---|---|
| FALSE | FALSE | 0 | Speed follower |
| FALSE | TRUE | 1 | Homing |
| TRUE | FALSE | 2 | Manual jog |
| TRUE | TRUE | 3 | Positioning |

### 8.4.3.2 Terminals 2

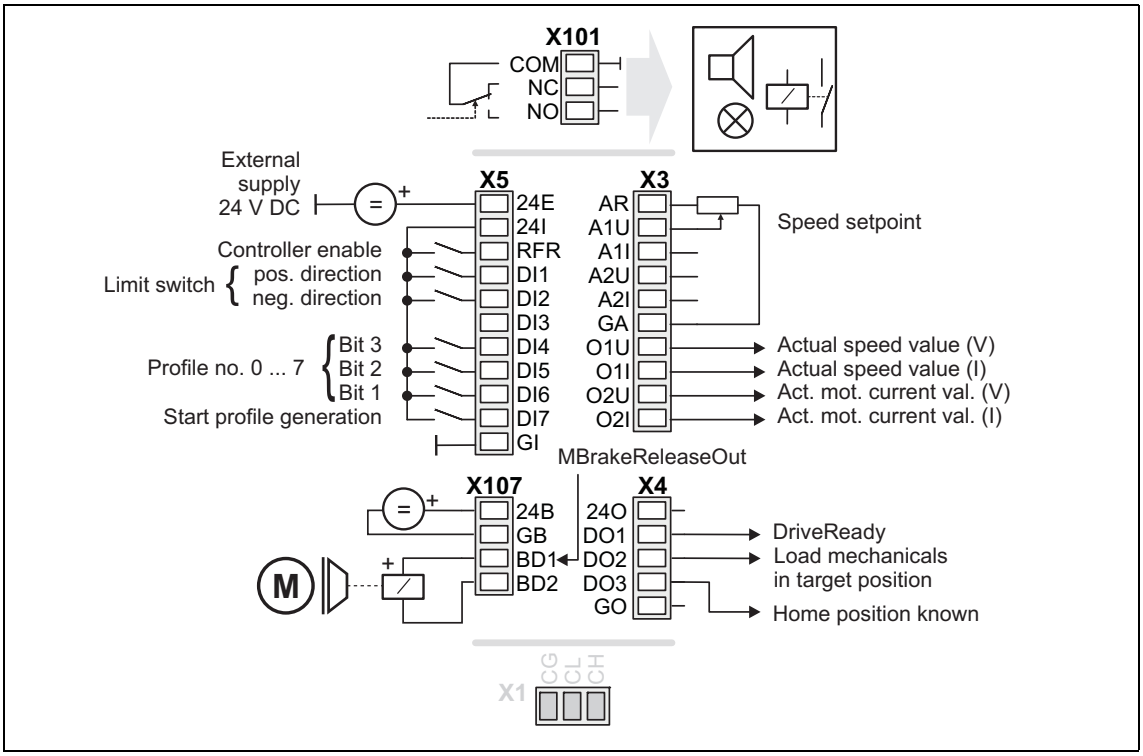



| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_TabPos.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_TabPos.bFailReset | | **X3/A1U** | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | - | | **X3/A1I** | |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | - | | **X3/A2I** | - |
| **X5/DI4** | LA_TabPos.bPosProfileNo_4 | | **X3/O1U** | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI5** | LA_TabPos.bPosProfileNo_2 | | **X3/O1I** | |
| **X5/DI6** | LA_TabPos.bPosProfileNo_1 | | **X3/O2U** | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max_mot}$ (C00022) |
| **X5/DI7** | LA_TabPos.bPosExecute | | **X3/O2I** | |
| | | | | |
| **X107/BD1** | LA_TabPos.bMBrakeReleaseOut | | **X4/DO1** | LA_TabPos.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | LA_TabPos.bInTarget |
| | | | **X4/DO3** | LA_TabPos.bHomePosAvailable |

When the profile is defined, the operating mode in the Lenze setting is changed simultaneously:

| bPosProfileNo_4 **(DI4)** | bPosProfileNo_2 **(DI5)** | bPosProfileNo_1 **(DI6)** | **Selected profile** | **Activation of operating mode** |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | 0 | Speed follower |
| FALSE | FALSE | TRUE | 1 | Homing |
| FALSE | TRUE | FALSE | 2 | Manual jog |
| FALSE | TRUE | TRUE | 3 | Positioning |
| ... | | | ... | |
| TRUE | TRUE | TRUE | 7 | |

### 8.4.3.3 Terminals 11

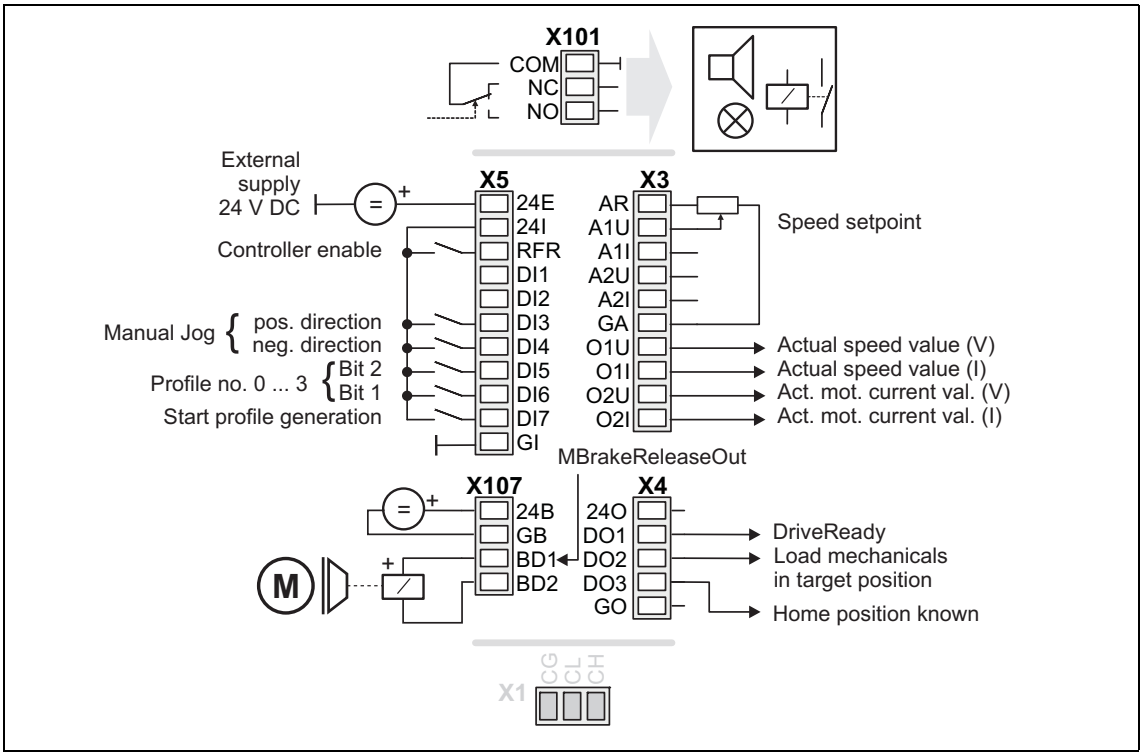



| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_TabPos.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_TabPos.bFailReset | | **X3/A1U** | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | LA_TabPos.bLimitSwitchPos | | **X3/A1I** | |
| **X5/DI2** | LA_TabPos.bLimitSwitchNeg | | **X3/A2U** | - |
| **X5/DI3** | - | | **X3/A2I** | - |
| **X5/DI4** | LA_TabPos.bPosProfileNo_4 | | **X3/O1U** | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI5** | LA_TabPos.bPosProfileNo_2 | | **X3/O1I** | |
| **X5/DI6** | LA_TabPos.bPosProfileNo_1 | | **X3/O2U** | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max_mot}$ (C00022) |
| **X5/DI7** | LA_TabPos.bPosExecute | | **X3/O2I** | |
| | | | | |
| **X107/BD1** | LA_TabPos.bMBrakeReleaseOut | | **X4/DO1** | LA_TabPos.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | LA_TabPos.bInTarget |
| | | | **X4/DO3** | LA_TabPos.bHomePosAvailable |

When the profile is defined, the operating mode in the Lenze setting is changed simultaneously:

| bPosProfileNo_4 **(DI4)** | bPosProfileNo_2 **(DI5)** | bPosProfileNo_1 **(DI6)** | **Selected profile** | **Activation of operating mode** |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | 0 | Speed follower |
| FALSE | FALSE | TRUE | 1 | Homing |
| FALSE | TRUE | FALSE | 2 | Manual jog |
| FALSE | TRUE | TRUE | 3 | Positioning |
| ... | | | ... | |
| TRUE | TRUE | TRUE | 7 | |

### 8.4.3.4 Terminal 16

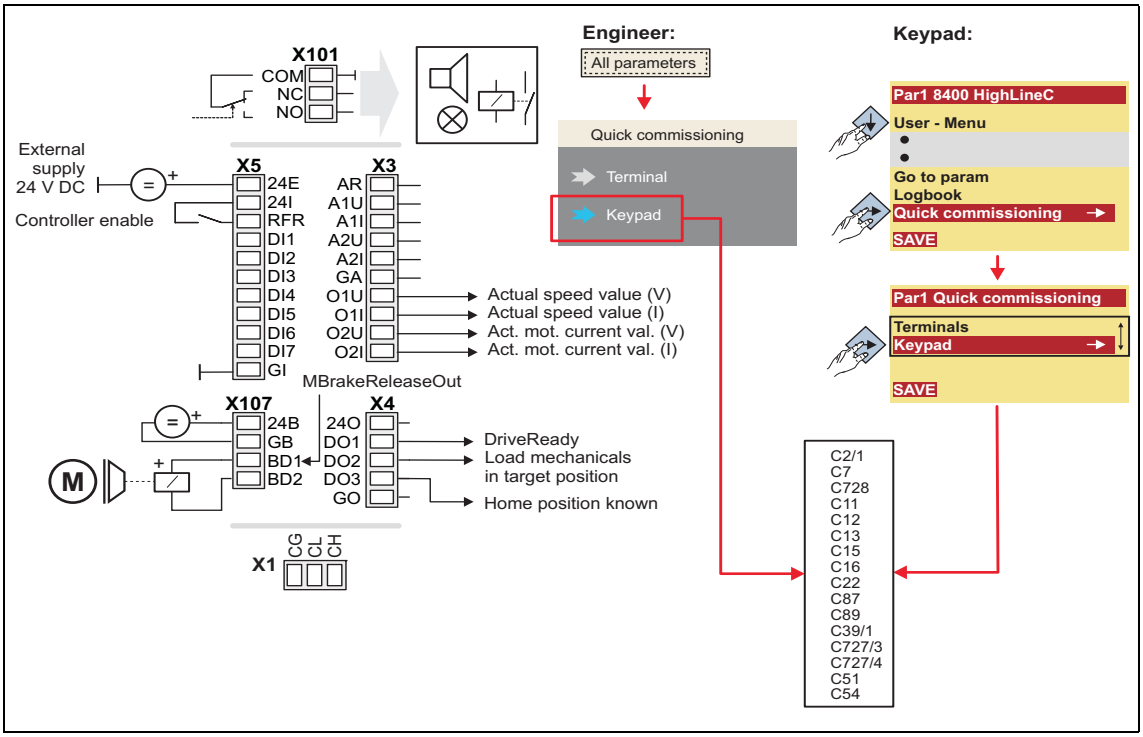



| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_TabPos.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_TabPos.bFailReset | | **X3/A1U** | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | - | | **X3/A1I** | |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | LA_TabPos.bManJogPos | | **X3/A2I** | - |
| **X5/DI4** | LA_TabPos.bManJogNeg | | **X3/O1U** | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI5** | LA_TabPos.bPosProfileNo_2 | | **X3/O1I** | |
| **X5/DI6** | LA_TabPos.bPosProfileNo_1 | | **X3/O2U** | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max_mot}$ (C00022) |
| **X5/DI7** | LA_TabPos.bPosExecute | | **X3/O2I** | |
| | | | | |
| **X107/BD1** | LA_TabPos.bMBrakeReleaseOut | | **X4/DO1** | LA_TabPos.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | LA_TabPos.bInTarget |
| | | | **X4/DO3** | LA_TabPos.bHomePosAvailable |

When the profile is defined, the operating mode in the Lenze setting is changed simultaneously:

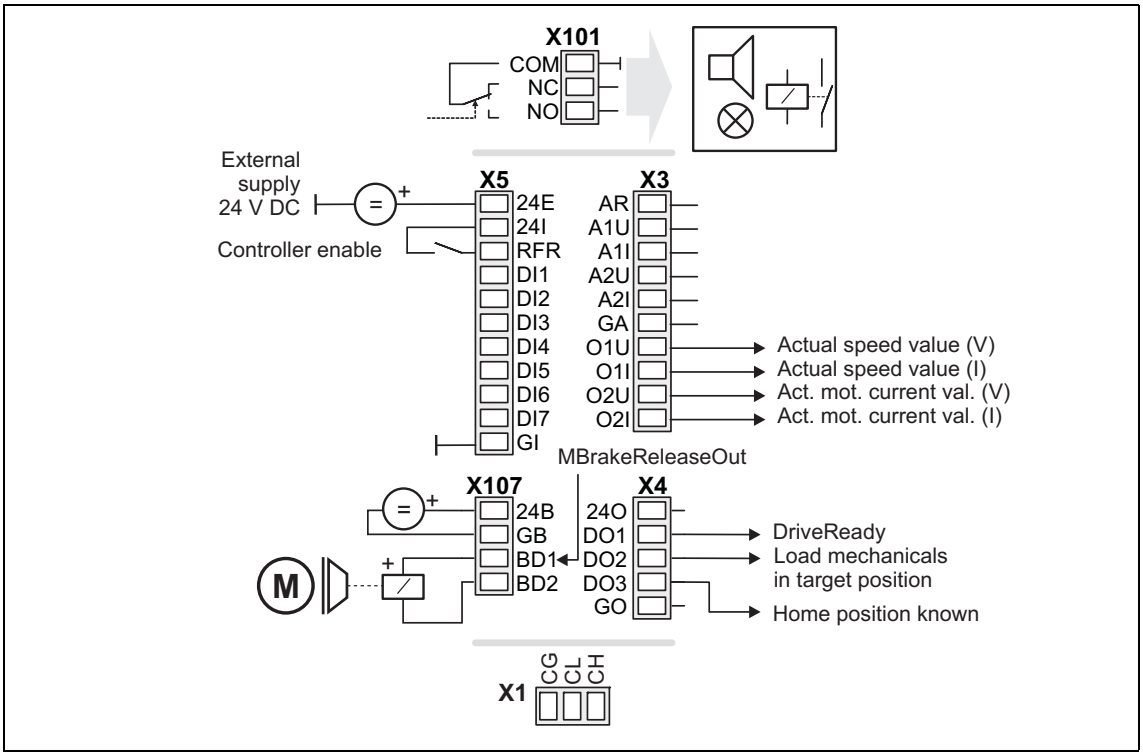
| bPosProfileNo_2 (DI5) | bPosProfileNo_1 (DI6) | Selected profile | Activation of operating mode |
|---|---|---|---|
| FALSE | FALSE | 0 | Speed follower |
| FALSE | TRUE | 1 | Homing |
| TRUE | FALSE | 2 | Manual jog |
| TRUE | TRUE | 3 | Positioning |

### 8.4.3.5 Keypad



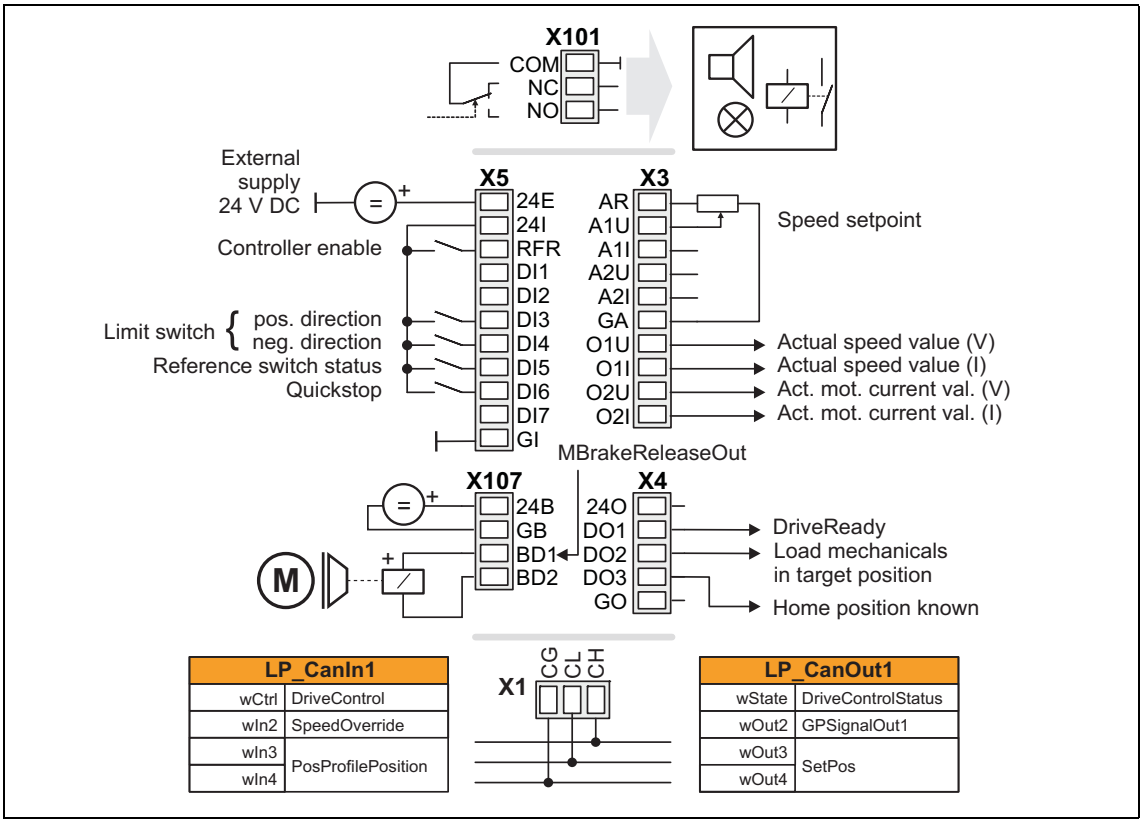

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_TabPos.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_TabPos.bFailReset | | **X3/A1U** | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | - | | **X3/A1I** | |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | - | | **X3/A2I** | - |
| **X5/DI4** | - | | **X3/O1U** | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI5** | - | | **X3/O1I** | |
| **X5/DI6** | - | | **X3/O2U** | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max_mot}$ (C00022) |
| **X5/DI7** | - | | **X3/O2I** | |
| | | | | |
| **X107/BD1** | LA_TabPos.bMBrakeReleaseOut | | **X4/DO1** | LA_TabPos.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | LA_TabPos.bInTarget |
| | | | **X4/DO3** | LA_TabPos.bHomePosAvailable |

### 8.4.3.6 PC




| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_TabPos.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_TabPos.bFailReset | | **X3/A1U** | - |
| **X5/DI1** | - | | **X3/A1I** | - |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | - | | **X3/A2I** | - |
| **X5/DI4** | - | | **X3/O1U** | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI5** | - | | **X3/O1I** | |
| **X5/DI6** | - | | **X3/O2U** | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max_mot}$ (C00022) |
| **X5/DI7** | - | | **X3/O2I** | |
| | | | | |
| **X107/BD1** | LA_TabPos.bMBrakeReleaseOut | | **X4/DO1** | LA_TabPos.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | LA_TabPos.bInTarget |
| | | | **X4/DO3** | LA_TabPos.bHomePosAvailable |

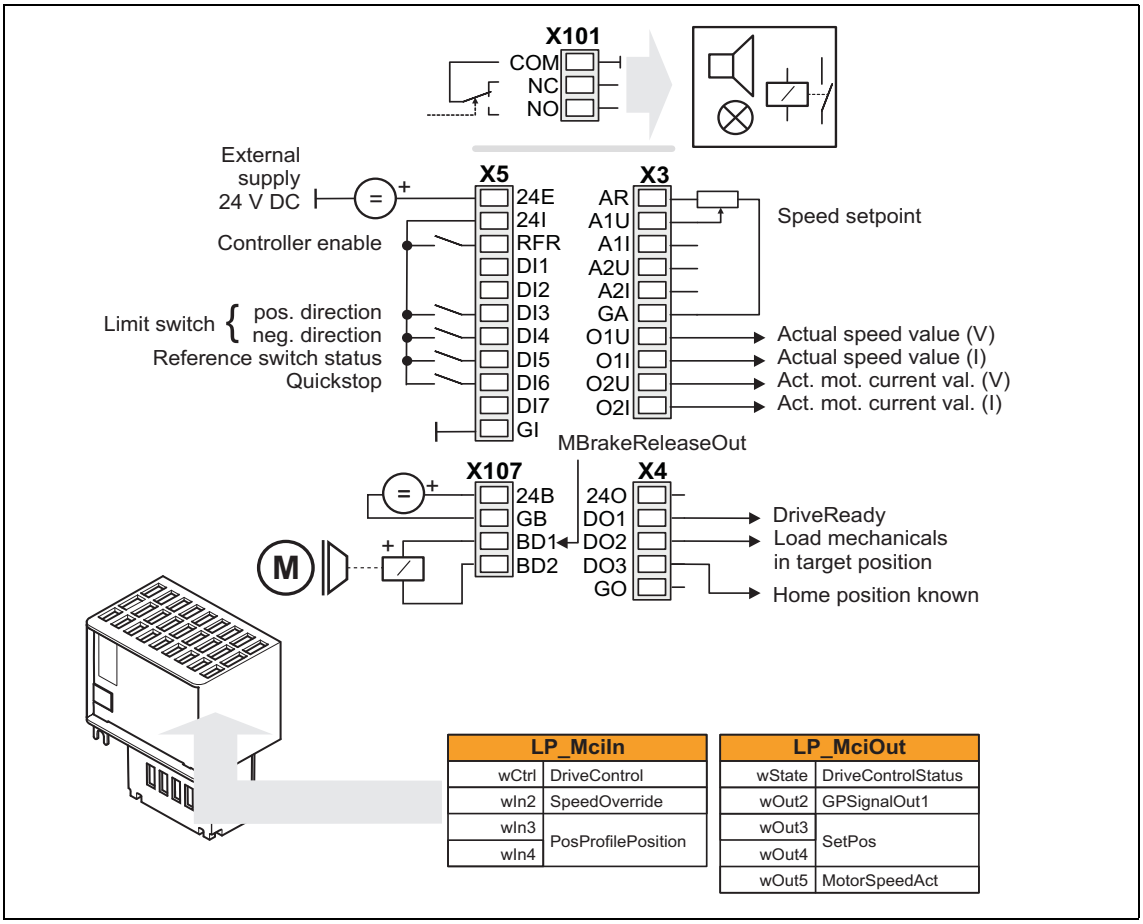
### 8.4.3.7 CAN




| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_TabPos.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_TabPos.bFailReset | | **X3/A1U** | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | - | | **X3/A1I** | |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | LA_TabPos.bLimitSwitchPos | | **X3/A2I** | - |
| **X5/DI4** | LA_TabPos.bLimitSwitchNeg | | **X3/O1U** | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI5** | LA_TabPos.bHomeMark | | **X3/O1I** | |
| **X5/DI6** | LA_TabPos.bSetQuickstop | | **X3/O2U** | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max_mot}$ (C00022) |
| **X5/DI7** | LP_CanOut1: bState_B7 | | **X3/O2I** | |
| | | | | |
| **X107/BD1** | LA_TabPos.bMBrakeReleaseOut | | **X4/DO1** | LA_TabPos.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | LA_TabPos.bInTarget |
| | | | **X4/DO3** | LA_TabPos.bImaxActive |

▸ Process data assignment for fieldbus communication (📖 521)

### 8.4.3.8 MCI




| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | LA_TabPos.bDriveFail | | | |
| | | | | |
| **X5/RFR** | LA_TabPos.bFailReset | | **X3/A1U** | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | - | | **X3/A1I** | |
| **X5/DI2** | - | | **X3/A2U** | - |
| **X5/DI3** | LA_TabPos.bLimitSwitchPos | | **X3/A2I** | - |
| **X5/DI4** | LA_TabPos.bLimitSwitchNeg | | **X3/O1U** | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI5** | LA_TabPos.bHomeMark | | **X3/O1I** | |
| **X5/DI6** | LA_TabPos.bSetQuickstop | | **X3/O2U** | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max_mot}$ (C00022) |
| **X5/DI7** | LP_MciOut: bState_B7 | | **X3/O2I** | |
| | | | | |
| **X107/BD1** | LA_TabPos.bMBrakeReleaseOut | | **X4/DO1** | LA_TabPos.bDriveReady |
| **X107/BD2** | - | | **X4/DO2** | LA_TabPos.bInTarget |
| | | | **X4/DO3** | LA_TabPos.bImaxActive |

▸ Process data assignment for fieldbus communication (📖 521)

### 8.4.4 Process data assignment for fieldbus communication

The fieldbus communication is connected (preconfigured) to the previously selected technology application by selecting the corresponding control mode in C00007:

- "30: CAN" for the connection to the system bus (CAN)
- "40: MCI" for the connection to a plugged-on communication module (e.g. PROFIBUS)

The assignment of the process data words depends only on the application, not on the bus system used:

| Input words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControl | Control word<br>• For bit assignment see the table below. |
| Word 2 | SpeedOverride | Value for speed override<br>• Percentage multiplier for the currently active speed.<br>• 16384 ≡ 100 % of the maximum traversing speed (display in C01211/1).<br>• Values > 16384 are ignored.<br>• If the override value is 0 %, the drive is brought to a standstill.<br>▸ Speed override (📖 670) |
| Word 3 and 4 | PosProfilePosition | Selection of the target position in [increments]<br>• The mode for calculating the position is selected in C01296/1.<br>• 65535 [Increments] ≡ 1 motor revolution<br>• The position resolution, i.e. how many increments form one unit, is displayed in C01205. |
| Words 5 ... 16 | - | Not preconfigured<br>• Only available in control mode "40: MCI". |

| Control word | Name | Function |
|---|---|---|
| Bit 0 | MckOperationMode_1 | Binary-coded selection of the operating mode of the **Motion Control Kernel**<br>• For a detailed description of the individual control bits, see chapter "MCK control word". (📖 569) |
| Bit 1 | MckOperationMode_2 | |
| Bit 2 | MckOperationMode_4 | |
| Bit 3 | CINH | 1 ≡ Inhibit inverter (controller inhibit): The inverter switches to the "SwitchedOn" device status.<br>▸ Enable/inhibit inverter (📖 110) |
| Bit 4 | PosProfileNo_1 | Selection of the profile number<br>▸ Stipulation of the profile to be executed (📖 661) |
| Bit 5 | PosProfileNo_2 | |
| Bit 6 | PosProfileNo_4 | |
| Bit 7 | PosProfileNo_8 | |
| Bit 8 | PosExecute | 1 ≡ Start travel job<br>▸ Positioning (📖 646) |
| Bit 9 | EnableSpeedOverride | 1 ≡ activate speed override<br>▸ Speed override (📖 670) |
| Bit 10 | HomeSetPosition | 1 ≡ Set home position<br>▸ Homing (📖 618) |
| Bit 11 | FailReset | 1 ≡ Reset error message<br>▸ Reset error message (📖 719) |
| Bit 12 | ManJogNeg | Manual jog in positive/negative direction<br>▸ Manual jog (📖 638) |
| Bit 13 | ManJogPos | |
| Bit 14 | - | Free control bit 14 (not assigned, freely assignable) |
| Bit 15 | - | Free control bit 15 (not assigned, freely assignable) |

| Output words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControlStatus | Status word<br>• For bit assignment see the table below. |
| Word 2 | GPSignalOut1 | Analog signal monitor: Output signal 1<br>• The selection of the signal source to output is executed in C00410/1.<br>• Gain and offset for the output signal can be parameterised in C00413/1 and C00413/2.<br>• For a detailed functional description see the L_SignalMonitor_a FB. |
| Word 3 and 4 | SetPos | Absolute position setpoint in [increments]<br>• 65535 [Increments] ≡ 1 motor revolution<br>• The position resolution, i.e. how many increments form one unit, is displayed in C01205. |
| Word 5 | MotorSpeedAct | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• Only available in control mode "40: MCI". |
| Word 6 ... 16 | - | Not preconfigured<br>• Only available in control mode "40: MCI". |

| Status word | Name | Status |
|---|---|---|
| Bit 0 | DriveFail | 1 ≡ Inverter in the error status<br>• "Fault" device status is active. |
| Bit 1 | GPSignalOut1 | Binary signal monitor: Output signals 1 & 2<br>• The signal sources to be output are selected in C00411/1...2.<br>• A bit coded inversion of the output signals can be parameterised in C00412.<br>• For a detailed functional description see FB L_SignalMonitor_b. |
| Bit 2 | GPSignalOut2 | |
| Bit 3 | CInhActive | 1 ≡ Controller inhibit is active |
| Bit 4 | DriveReady | 1 ≡ Inverter is ready for operation<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the inverter is still inhibited by the user (controller inhibit). |
| Bit 5 | DigitalInput5 | Signal from the digital input DI5 |
| Bit 6 | DigitalInput6 | Signal from the digital input DI6 |
| Bit 7 | DigitalInput7 | Signal from the digital input DI7 |
| Bit 8 | InTarget | 1 ≡ Target position (actual value) is in the target window |
| Bit 9 | ProfileBusy | 1 ≡ Profile positioning is active |
| Bit 10 | HomePosAvailable | 1 ≡ Home position is known |
| Bit 11 | SpeedActCompare | Result of the speed comparison (detection of speed=0)<br>• In case of the "Open loop" operation:<br>1 ≡ Speed setpoint < comparison value (C00024)<br>• For "Closed loop" operation:<br>1 ≡ actual speed value < comparison value (C00024) |
| Bit 12 | DigitalInput4 | Signal from the digital input DI4 |
| Bit 13 | DigitalInput3 | Signal from the digital input DI3 |
| Bit 14 | QSPIsActive | 1 ≡ Quick stop is active |
| Bit 15 | - | Free status bit 15 (not assigned, freely assignable) |

### 8.4.5 Setting parameters (short overview)

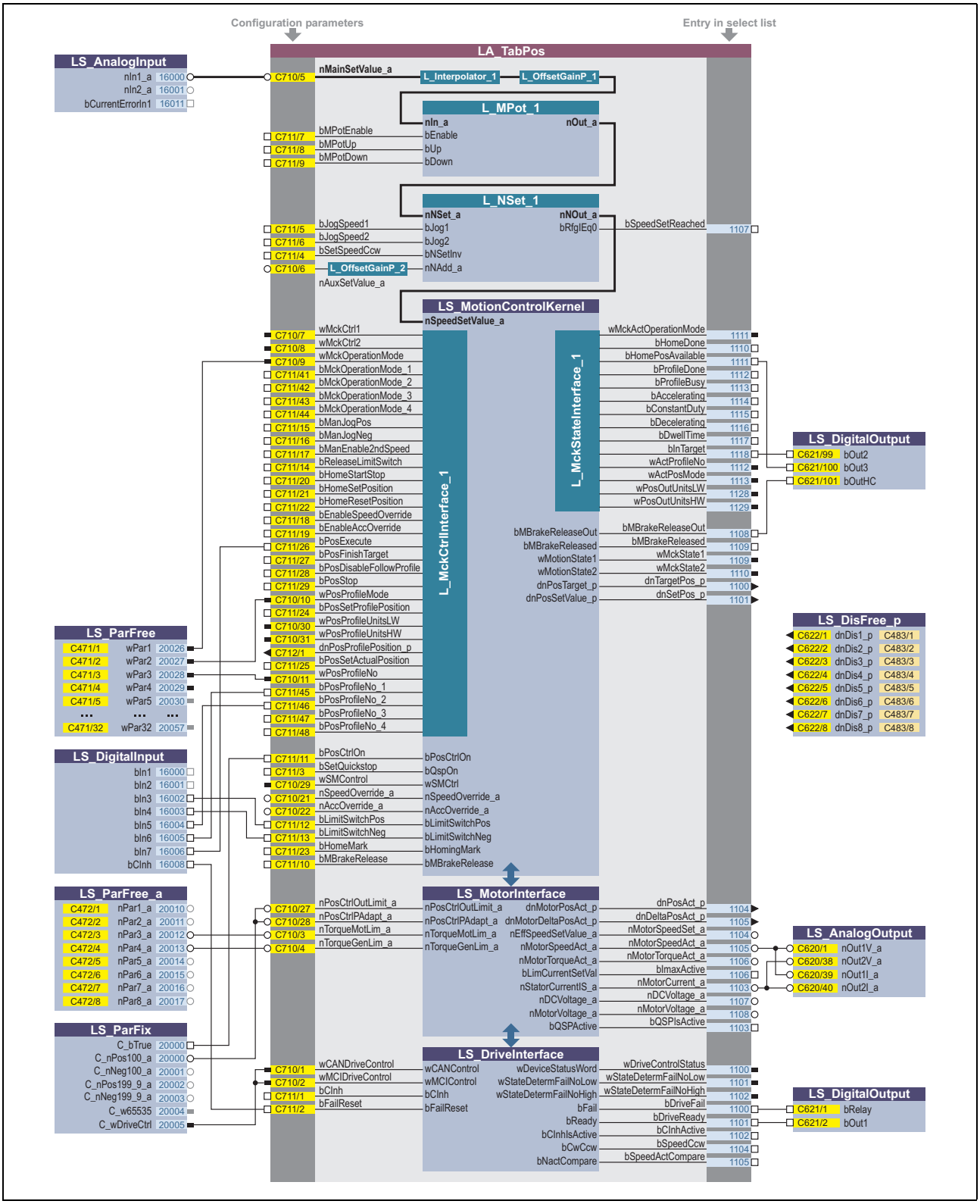
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00012 | Accel. time - main setpoint | 2.000 | s |
| C00013 | Decel. time - main setpoint | 2.000 | s |
| C00024 | Comparison value N_Act | 0.00 | % |
| C00039/1 | Fixed setpoint 1 | 40.00 | % |
| C00039/2 | Fixed setpoint 2 | 60.00 | % |
| C00182 | S-ramp time PT1 | 20.00 | s |
| C00190 | Setpoint arithmetic | 0: NOut = NSet | |
| C00220 | Accel. time - add. setpoint | 0.000 | s |
| C00221 | Decel. time - add. setpoint | 0.000 | s |
| C00632/1 | L_NSet_1: Blocking speed 1 max | 0.00 | % |
| C00632/2 | L_NSet_1: Blocking speed 2 max | 0.00 | % |
| C00632/3 | L_NSet_1: Blocking speed 3 max | 0.00 | % |
| C00633/1 | L_NSet_1: Blocking speed 1 min | 0.00 | % |
| C00633/2 | L_NSet_1: Blocking speed 2 min | 0.00 | % |
| C00633/3 | L_NSet_1: Blocking speed 3 min | 0.00 | % |
| C00635 | L_NSet_1: nMaxLimit | 199.99 | % |
| C00636 | L_NSet_1: nMinLimit | -199.99 | % |
| C00670 | L_OffsetGainP_1: Gain | 1.0000 | |
| C00671 | L_OffsetGainP_2: Gain | 1.0000 | |
| C00696 | L_OffsetGainP_1: Offset | 0.00 | % |
| C00697 | L_OffsetGainP_2: Offset | 0.00 | % |
| C00800 | L_MPot_1: Upper limit | 100.00 | % |
| C00801 | L_MPot_1: Lower limit | -100.00 | % |
| C00802 | L_MPot_1: Acceleration time | 10.0 | s |
| C00803 | L_MPot_1: Deceleration time | 10.0 | s |
| C00804 | L_MPot_1: Inactive fct. | 0: Retain value | |
| C00805 | L_MPot_1: Init fct. | 0: Load last value | |
| C00806 | L_MPot_1: Use | 0: No | |
| C01297 | Alternative function | Bit coded | |
| C01298/1 | MCK operating mode at profile no. 0 | 1: Follower | |
| C01298/2 | MCK operating mode at profile no. 1 | 2: Homing | |
| C01298/3 | MCK operating mode at profile no. 2 | 3: ManualJog | |
| C01298/4 | MCK operating mode at profile no. 3...15 | 4: Positioning | |
| C01299 | MCKI: Status MCKInterface | - | |

**Related topics:**

- "GeneralPurpose" functions (📖 554)

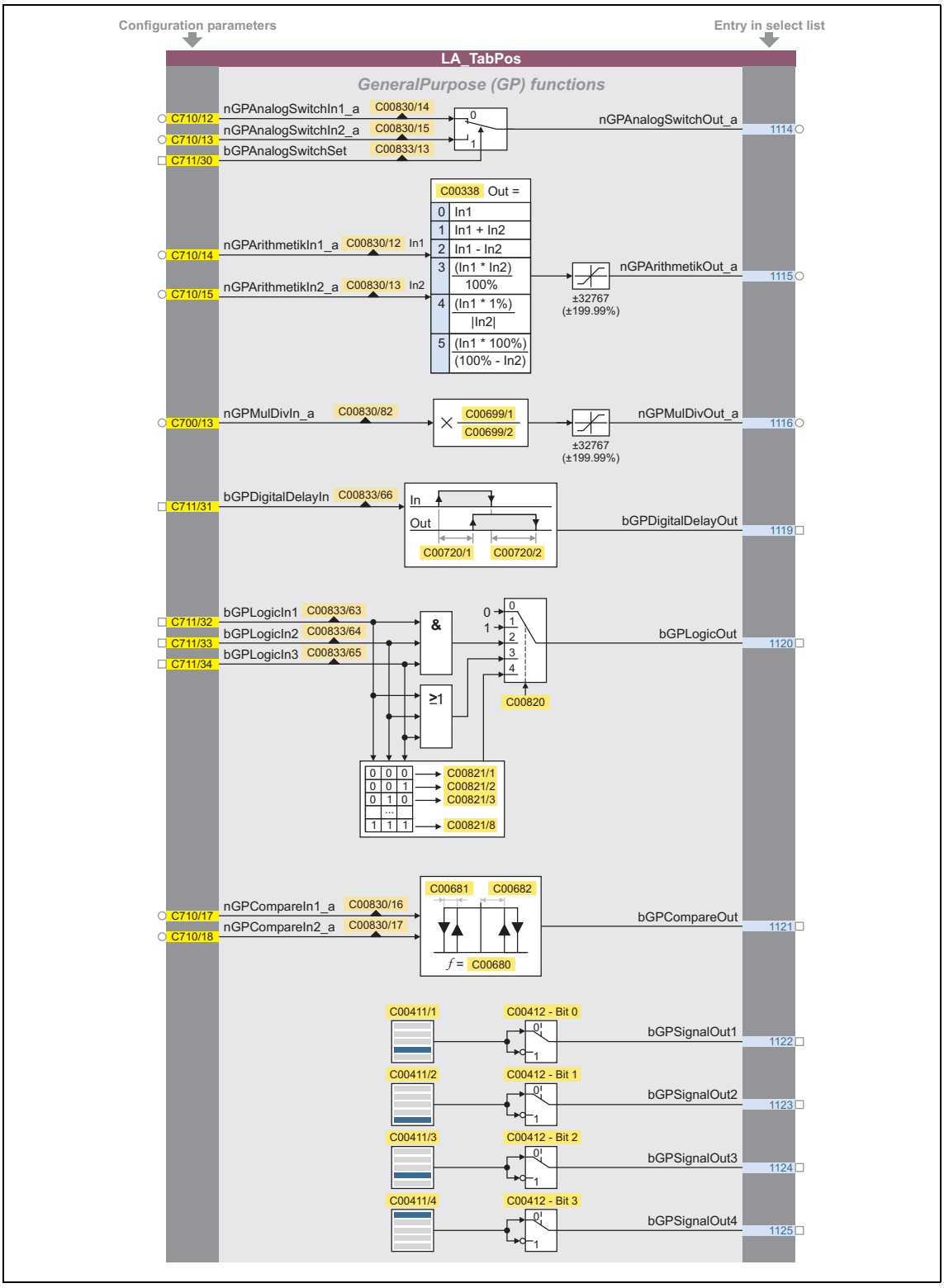
### 8.4.6 Configuration parameters

If required, the subcodes of C00710, C00711 and C00712 serve to change the pre-configured assignment of the application inputs:

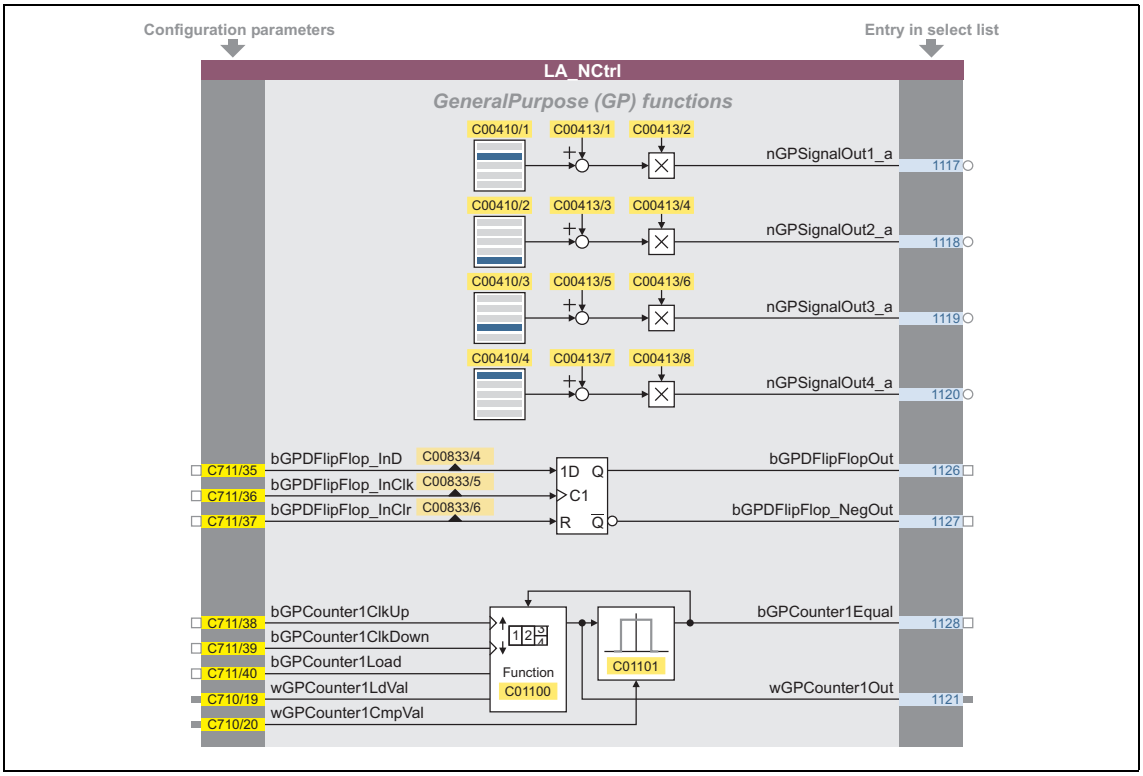



[8-14] Pre-assignment of the "Table positioning" application in the "Terminals 0" control mode

### Configuration parameters for "GeneralPurpose" functions



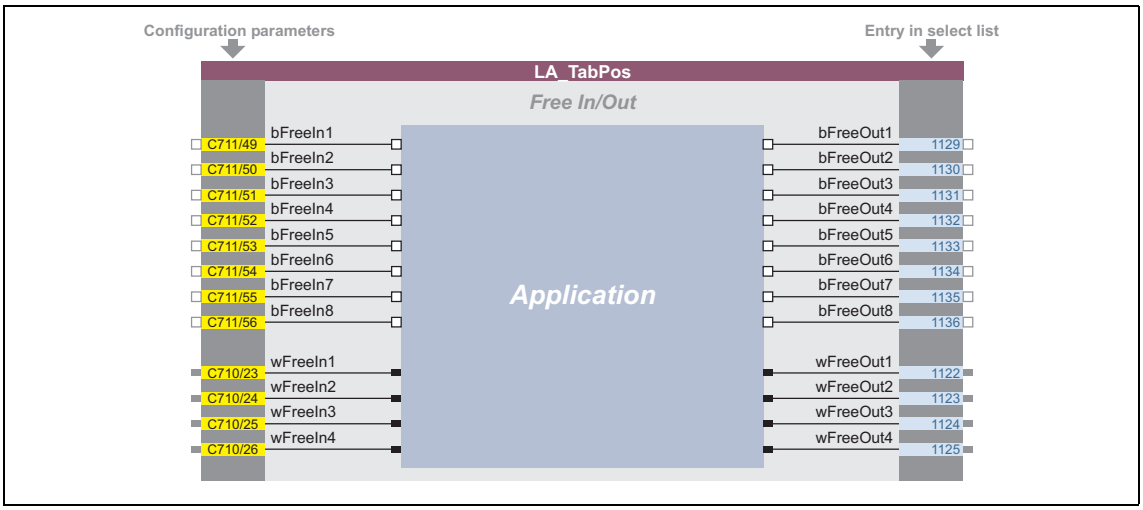

[8-15] "GeneralPurpose" functions




[8-16] "GeneralPurpose" functions (continuation)

### Free inputs and outputs

These inputs can be freely interconnected in the application level. They can be used to transfer signals from the I/O level to the application level and vice versa.




[8-17] Free inputs/outputs

### Related topics:

- User-defined terminal assignment (📖 429)
- "GeneralPurpose" functions (📖 554)

## 8.5 TA "Switch-off positioning"

The basic principle of this technology application is to travel to a switch-off sensor (e.g. a limit switch) in a speed-controlled manner and to stop as close as possible at this position. Unlike other positioning controls, the switch-off positioning neither has a position feedback nor calculates the path in advance. Thus, the accuracy that can be achieved depends on various factors such as the speed at which the switch-off sensor is advanced.

In addition, a pre-switch off can be implemented which requires a sufficient number of unassigned digital inputs on the inverter which can be used to connect other sensors for the additional stop positions. These sensors effect a reduction in speed before the last switch-off sensor is reached.

**Product features**

- Pre-configured control modes for terminals and bus control (with predefined process data connection to the fieldbus)
- Free configuration of input and output signals
- Offset, gain, and negation of main setpoint & additional setpoint
- Up to 15 fixed setpoints for speed and ramp time
- Adjustable setpoint ramp times
- Freely selectable, variable ramp shape
- Automatic holding brake control
- Quick stop (QSP) with adjustable ramp time
- Integrated, freely available "GeneralPurpose" functions:
  Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop
- Interface to the safety module (optional)
- Integration of encoder feedback
- Switch-off sensor management for the implementation of a pre-switch off

**Decision criteria**

| Criteria | Switch-off positioning with constant load | Switch-off positioning with variable load |
|---|---|---|
| **Operating mode** | V/f characteristic without speed sensor.<br>Alternatively for large breakaway torques: Use of a sensorless vector control (only applicable for horizontal movements). | |
| **Limit switch evaluation** | One limit switch is required per direction of movement.<br>When the limit switch is reached, the drive is brought to a standstill led by the deceleration ramp or the QSP ramp. | One limit switch and an initiator are required for fast/slow changeover per direction of movement.<br>When the initiator has been reached, the speed of the drive is reduced to a creeping speed (fixed setpoint 2) in a controlled way.<br>When the limit switch is reached, the drive is brought to a standstill led by the deceleration ramp or the QSP ramp. |

| Criteria | Switch-off positioning with constant load | Switch-off positioning with variable load |
|---|---|---|
| **Positioning accuracy at the motor shaft**<br>The positioning accuracy of the load depends, among other things, on the clearance and friction of the selected mechanics and has to be determined individually. | The ideal case is 5-10° at the motor shaft.<br>Consider the influence of the motor temperature.<br>In the case of a constant load, you can assume a good repeat accuracy during positioning. In the case of variable loads, you must take significant deviations into account. | 5-10° at the motor shaft. As the positioning is executed in a creeping speed, a good repeat accuracy is reached even for variable loads. |
| **Speed setting range** | 1 : 50, based on 50Hz and $M_n$ | 1 : 50, based on 50Hz and $M_n$ |
| **Typical applications** | Switch-off positioning with constant load, e.g. travelling drive, roll-up door. | Switch-off positioning with variable load, e.g. travelling drive, conveying belt, hoists approaching a stop position. |

### System limits and exclusion criteria

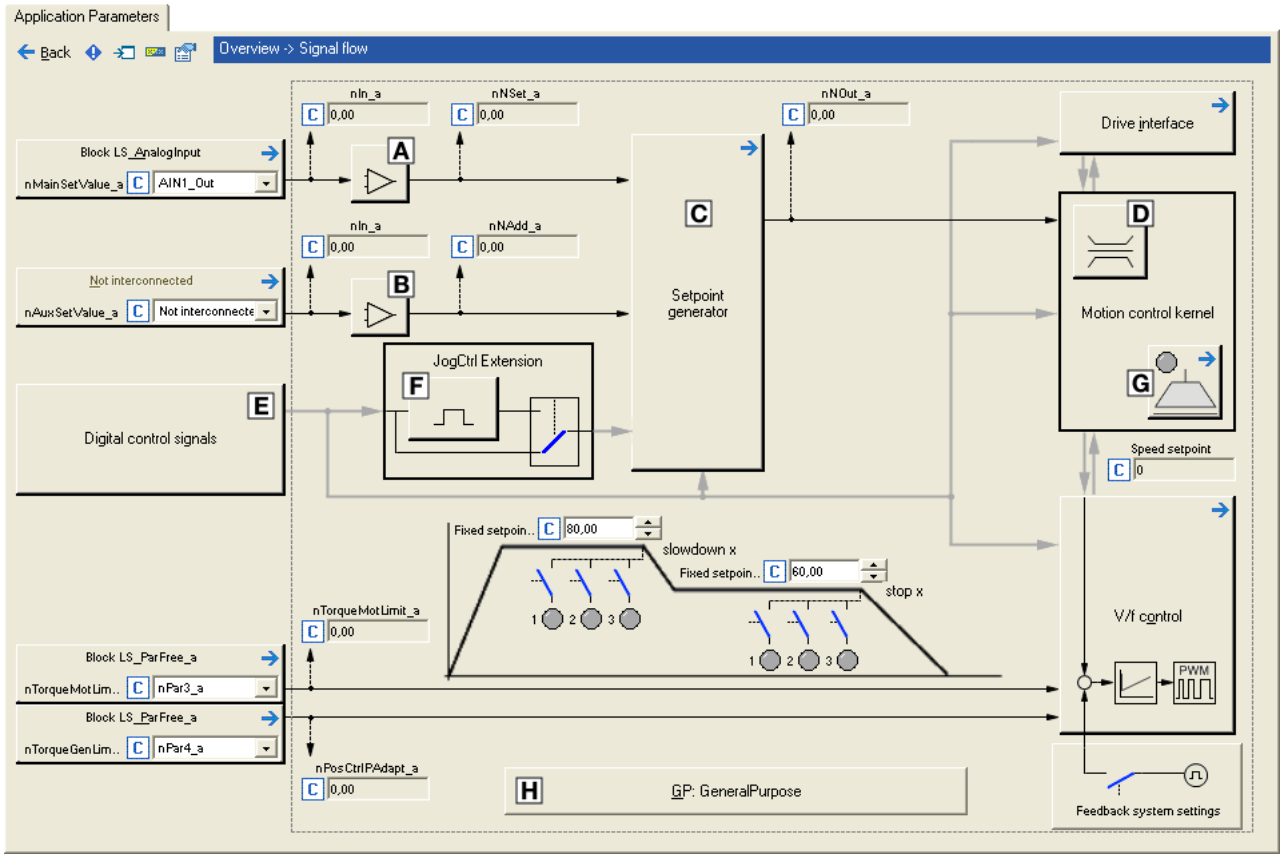
They result from the non-compliance with the decision criteria.

- Compared to systems with speed feedback, the positioning and repeat accuracy is reduced.
- Due to the mechanical hardware limit switches, this concept is only applicable for systems with only a few fixed positions. Changing the target position during the operation or the teaching is not possible.
- If necessary, additional functions like manual jog or homing must be realised externally, e.g. via a control.
- As the 8400 TopLine inverter does not meet safety-related functions except STO (Safe Torque Off), you must observe that all safety-related aspects are realised by the plant instructor.
- Especially in the case of an outdoor use or in wet areas, you must consider the corresponding discharge currents when operated with a fault current circuit breaker.
- A table positioning or sequential positioning control is required for highly dynamic applications and jerk-free traversing profiles which is available with the "HighLine" device version.

**Related topics:**

▸ Commissioning of the "Switch-off positioning" technology application (📖 85)

## 8.5.1 Basic signal flow




[8-18] Signal flow of the switch-off positioning

- A Main speed setpoint offset and gain (L_OffsetGainP_1)
- B Additional speed setpoint offset and gain (L_OffsetGainP_2)
- C Setpoint generator (L_NSet_1)
- D Speed setpoint input limitation
- E Terminal assignment & display of digital control signals
- F Selection of edge/level for tripping the ramp down and stop functions (L_JogCtrlExtension_1)
- G Holding brake control
- H Integrated disposable "GeneralPurpose" functions: Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop

### 8.5.2 Internal interfaces | application block "LA_SwitchPos"

**Note!**

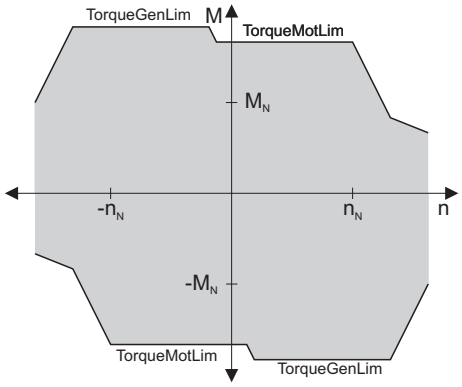
The connectors greyed out in the following table are hidden in the function block editor in the Lenze setting.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wCANDriveControl<br>WORD | Control word via system bus (CAN) for device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. | |
| wMCIDriveControl<br>WORD | Control word via communication module (e.g. PROFIBUS) for device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. | |
| wSMControl<br>WORD | Interface to the optional safety system.<br>• Setting control bit 0 ("SafeStop1") in this control word causes e.g. the automatic deceleration of the drive to standstill within this application (in the **Motion Control Kernel**).<br>• See the subchapter "Interface to safety system" of the chapter on basic drive functions for a detailed description of the individual control bits. | |
| bCInh<br>BOOL | Enable/inhibit inverter | |
| | FALSE | Enable inverter: The inverter switches to the "OperationEnabled" device status if no other source for controller inhibit is active.<br>• C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit. |
| | TRUE | Inhibit inverter (controller inhibit): The inverter switches to the "SwitchedOn" device status. |
| bFailReset<br>BOOL | Reset error message<br>In the Lenze setting this input is connected to the digital input controller enable so that a possibly existing error message is reset together with the controller enable (if the cause for the fault is eliminated). | |
| | TRUE | The current fault is reset, if the cause for the fault is eliminated.<br>• If the fault still exists, the error status remains unchanged. |
| bSetQuickstop<br>BOOL | Activate quick stop (QSP)<br>• Also see device command "Activate/deactivate quick stop". | |
| | TRUE | Activate quick stop<br>• Motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0).<br>• The motor is kept at a standstill during closed-loop operation.<br>• A pulse inhibit is set if the auto-DCB function has been activated via C00019. |
| | FALSE | Deactivate quick stop<br>• The quick stop is deactivated if no other source for the quick stop is active.<br>• C00159 displays a bit code of active sources/causes for the quick stop. |

| Designator<br>Data type | Information/possible settings |
|---|---|
| bSetDCBrake<br>BOOL | Manual DC-injection braking (DCB)<br>• Detailed information on DC-injection braking is provided in the motor control chapter, subchapter "DC-injection braking". |
| | **Note!**<br>Holding braking is not possible when this braking mode is used!<br>Use the basic "Holding brake control" function for controlling the holding brake with a low rate of wear. |
| | FALSE: Deactivate DC-injection braking. |
| | TRUE: Activate DC-injection braking, i.e. the drive is brought to a standstill by means of DC-injection braking.<br>• The braking effect stops when the rotor is at standstill.<br>• After the hold time (C00107) has expired, the controller sets the pulse inhibit. |
| bRFG_Stop<br>BOOL | Ramp function generator: Maintain the current value of the main setpoint integrator<br>• The speed, for instance, of a running ramp process is immediately kept constant when *bRFG_Stop* is activated. At the same time, the acceleration/deceleration jumps to the value "0".<br>• For a detailed functional description see the L_NSet FB. |
| | TRUE: The current value of the main setpoint integrator is held. |
| nVoltageAdd_a<br>INT | Additive voltage impression<br>• An additional setpoint for the motor voltage can be specified via this process input.<br>• If there are, for instance, different loads at the motor output end, it is possible to apply a voltage boost at the starting time.<br>• If the value is negative, the voltage is reduced.<br>• Scaling: 16384 ≡ 1000 V |
| | **Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! |
| nBoost_a<br>INT | Additional setpoint for the motor voltage at speed = 0<br>• The entire voltage-frequency characteristic is provided with an offset.<br>• Scaling: 16384 ≡ 1000 V |
| | **Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! |
| nPWMAngleOffset<br>INT | Additional offset for the electrical angle of rotation<br>• If a torque is connected, e.g. dynamic acceleration processes can be generated.<br>• Scaling: ±32767 ≡ ±180 ° angle of rotation |

| Designator<br>Data type | Information/possible settings |
|---|---|
| nTorqueMotLim_a<br>nTorqueGenLim_a<br>INT | Torque limitation in motor mode and in generator mode<br>• These input signals are directly transferred to the motor control to limit the inverter's maximum torque in motor and generator mode.<br>• The drive cannot output a higher torque in motor/generator mode than set here.<br>• The applied values (any polarity) are internally interpreted as absolute values.<br>• If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.<br>• If sensorless vector control (SLVC) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| | Torque limits in motor and generator mode:<br>  <br> |
| bSetSpeedCcw<br>BOOL | Change of direction of rotation<br>• For instance if a motor or gearbox is fixed laterally reversed to a machine part, but the setpoint selection should still be executed for the positive direction of rotation. |
| | FALSE: Clockwise rotation (Cw) |
| | TRUE: Direction of rotation to the left (Ccw) |
| bRLQCw<br>BOOL | Activate clockwise rotation (fail-safe)<br>• For a detailed functional description see the L_RLQ FB. |
| | FALSE: Quick stop |
| | TRUE: CW rotation |
| bRLQCcw<br>BOOL | Activate counter-clockwise rotation (fail-safe)<br>• For a detailed functional description see the L_RLQ FB. |
| | FALSE: Quick stop |
| | TRUE: Counter-clockwise rotation |
| nMainSetValue_a<br>INT | Main speed setpoint<br>• Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The main setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.<br>• Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.<br>• For a detailed functional description see the L_NSet FB. |
| nAuxSetValue_a<br>INT | Additional speed setpoint<br>• Offset and gain of this input signal can be set in C00697 and C00671 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The additional speed setpoint can be linked arithmetically with the main speed setpoint behind the ramp function generator.<br>• The additional speed setpoint can be shown via ramp times of a second ramp function generator.<br>• For a detailed functional description see the L_NSet FB. |

| Designator<br>Data type | Information/possible settings |
|---|---|
| **Switch-off positioning** | |
| bJogCtrlInputSel1<br>bJogCtrlInputSel2<br>BOOL | Selection inputs for a binary coded selection of the switch-off position 1 ... 3<br>• Activation of the signal pairs *bJogCtrlSlowDown1/bJogCtrlStop1*, *bJogCtrlSlowDown2/bJogCtrlStop2* or *bJogCtrlSlowDown3/bJogCtrlStop3* according to the Truth table for activating the pre-switch off. |
| bJogCtrlRfgIn<br>BOOL | Ramping down of the setpoint generator in the downstream L_NSet FB according to the Truth table for activating the pre-switch off |
| bJogCtrlJog1<br>bJogCtrlJog2<br>BOOL | Inputs for overriding fixed setpoints (JOG setpoints) for the main setpoint<br>• If the pre-switch off is inactive (*bJogCtrlInputSel1* and *bJogCtrlInputSel2* are both set to FALSE), the two control signals are passed through 1:1 to the downstream FB L_NSet.<br>• To achieve the desired behaviour (starting at high speed, pre-switch off at low speed), both inputs must be set to TRUE.<br>• Fixed setpoint 2 must be less than fixed setpoint 3!<br>Otherwise, the drive will start at a low speed and accelerate after the pre-switch off.<br>• If in additional to the inputs *bJogCtrlJog1* and *bJogCtrlJog2* the selection inputs *bJogSpeed4* and *bJogSpeed8* are assigned, different fixed setpoints can result from this and the drive may travel with different speeds than selected via *bJogCtrlJog1* and *bJogCtrlJog2*. |
| bJogCtrlSlowDown1<br>bJogCtrlSlowDown2<br>bJogCtrlSlowDown3<br>BOOL | Activation of fixed setpoint 2 in the downstream L_NSet FB<br>• These inputs only have a function if they have been previously activated via *bJogCtrlInputSel1* and *bJogCtrlInputSel2* (see Truth table for activating the pre-switch off). |
| bJogCtrlStop1<br>bJogCtrlStop2<br>bJogCtrlStop3<br>BOOL | Ramping down of the ramp function generator in the downstream L_NSet FB<br>• These inputs only have a function if they have been previously activated via *bJogCtrlInputSel1* and *bJogCtrlInputSel2* (see Truth table for activating the pre-switch off). |
| bJogSpeed4<br>bJogSpeed8<br>BOOL | Inputs for overriding fixed setpoints (JOG setpoints) for the main setpoint<br>• A fixed setpoint for the setpoint generator can be activated instead of the main setpoint via these selection inputs.<br>• The selection inputs are binary coded.<br>• For a detailed functional description see the L_NSet FB. |
| bJogRamp1 ... bJogRamp8<br>BOOL | Selection inputs for alternative acceleration/deceleration times for the main setpoint<br>• The four selection inputs are binary coded, therefore 15 alternative acceleration/deceleration times can be selected.<br>• For main setpoint *nMainSetValue_a*, the set acceleration time (C00012) and deceleration time (C00013) are active in the case of the binary coded selection "0" (all inputs = FALSE or not assigned).<br>• Alternative acceleration times are selected in C00101/1...15.<br>• The selection of the alternative deceleration times is carried out in C00103/1...15.<br>• For a detailed functional description see the L_NSet FB. |
| **MCK basic functions** | |
| bMBrakeRelease<br>BOOL | Holding brake control: Release/apply brake<br>• In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off"). |
| | FALSE: Apply brake.<br>• During automatic operation, the internal brake logic controls the brake. |
| | TRUE: Release brake manually (forced release).<br>• **Note!**<br>The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If a controller inhibit has been set by the brake control, it will be deactivated.<br>• In semi-automatic operation, the brake is released including feedforward control. |

| Designator<br>Data type | Information/possible settings |
|---|---|
| **GP: GeneralPurpose**<br>The following inputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | |
| nGPAnalogSwitchIn1_a<br>nGPAnalogSwitchIn2_a<br>INT | Analog switch: Input signals<br>• The input signal selected via the selection input *bGPAnalogSwitchSet* is output at output *nGPAnalogSwitchOut_a*. |
| bGPAnalogSwitchSet<br>BOOL | Analog switch: Selection input<br>FALSE: *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn1_a*<br>TRUE: *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn2_a* |
| nGPArithmetikIn1_a<br>nGPArithmetikIn2_a<br>INT | Arithmetic: Input signals<br>• The arithmetic function is selected in C00338.<br>• The result is output at output *nGPArithmetikOut_a*. |
| nGPMulDivIn_a<br>INT | Multiplication/Division: Input signal<br>• The factor for the multiplication can be set in C00699/1 (numerator) and C00699/2 (denominator).<br>• The result is output at output *nGPMulDivOut_a*. |
| bGPDigitalDelayIn<br>BOOL | Binary delay element: Input signal<br>• The on-delay can be set in C00720/1.<br>• The off-delay can be set in C00720/2.<br>• The time-delayed input signal is output at output *bGPDigitalDelayOut*. |
| bGPLogicIn1<br>bGPLogicIn2<br>bGPLogicIn3<br>BOOL | Binary logic: Input signals<br>• The logic operation is selected in C00820.<br>• The result is output at output *bGPLogicOut*. |
| nGPCompareIn1_a<br>nGPCompareIn2_a<br>INT | Analog comparison: Input signals<br>• The comparison operation is selected in C00680.<br>• Hysteresis and window size can be set in C00680 and C00682.<br>• If the comparison statement is true, the output *bGPCompareOut* will be set to TRUE. |
| bGPDFlipFlop_InD<br>bGPDFlipFlop_InClk<br>bGPDFlipFlop_InClr<br>BOOL | D-FlipFlop: Input signals<br>• Data, clock and reset input |
| **Free inputs**<br>The following inputs can freely be interconnected on the application level.<br>The signals can be transferred from the I/O level to the application level via these inputs. | |
| bFreeIn1 ... bFreeIn8<br>BOOL | Free inputs for digital signals |
| wFreeIn1 ... wFreeIn4<br>WORD | Free inputs for 16-bit signals |
| dnFreeIn1_p ... dnFreeIn2_p<br>DINT | Free inputs for 32-bit signals |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wDriveControlStatus<br>WORD | Status word of the inverter<br>• The status word contains information on the currents status of the inverter.<br>• See the "wDeviceStatusWord status word" subchapter of the chapter on device control for a detailed description of the bit assignment. |
| wStateDetermFailNoLow<br>WORD | Display of the status determining error (LOW word) |
| wStateDetermFailNoHigh<br>WORD | Display of the status determining error (HIGH word) |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bDriveFail<br>BOOL | TRUE | Inverter in error status.<br>• "Fault" device status is active. |
| bWarningActive<br>BOOL | TRUE | A monitoring in the inverter, for which the error response "Warning" or "Warning locked" has been parameterised, responded. |
| bSafeTorqueOff<br>BOOL | TRUE | Safe torque off.<br>• "SafeTorqueOff" device status is active. |
| bDriveReady<br>BOOL | TRUE | Inverter is ready for operation.<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the inverter is still inhibited by the user (controller inhibit). |
| bCInhActive<br>BOOL | TRUE | Controller inhibit is active. |
| bImpIsActive<br>BOOL | TRUE | Pulse inhibit is active. |
| bQSPIsActive<br>BOOL | TRUE | Quick stop is active. |
| bSpeedCcw<br>BOOL | Current direction of rotation | |
| | FALSE | Clockwise rotation (Cw) |
| | TRUE | Direction of rotation to the left (Ccw) |
| bSpeedActCompare<br>BOOL | Result of the speed comparison (detection of speed=0) | |
| | TRUE | During open-loop operation:<br>Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>Actual speed value < Comparison value (C00024) |
| bImaxActive<br>BOOL | "Current setpoint inside the limitation" status signal | |
| | TRUE | The current setpoint is internally limited (the inverter operates at the maximum current limit). |
| bSpeedSetReached<br>BOOL | Status signal "setpoint = 0" | |
| | TRUE | Speed setpoint from the ramp function generator = 0 |
| bSpeedActEqSet<br>BOOL | TRUE | Actual speed value = speed setpoint |
| nMotorCurrent_a<br>INT | Current stator current/effective motor current<br>• Scaling: 16384 ≡ 100 % $I_{max_mot}$ (C00022) | |
| nMotorSpeedSet_a<br>INT | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorSpeedAct_a<br>INT | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorTorqueAct_a<br>INT | Actual torque<br>• In the "VFC (+encoder)" operating mode of the motor control, this value is determined from the current motor current and corresponds to the actual torque only by approximation.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| nDCVoltage_a<br>INT | Actual DC-bus voltage<br>• Scaling: 16384 ≡ 1000 V | |
| nMotorVoltage_a<br>INT | Current motor voltage/inverter output voltage<br>• Scaling: 16384 ≡ 1000 V | |

| Designator<br>Data type | Value/meaning |
|---|---|
| **MCK basic functions** | |
| bMBrakeReleaseOut<br>BOOL | Holding brake control: Trigger signal for the holding brake control switching element via a digital output<br>• Use bit 0 in C02582 to activate inverted switching element triggering. |
| | FALSE: Apply brake. |
| | TRUE: Release brake. |
| bMBrakeReleased<br>BOOL | Holding brake control: "Brake released" considering the brake release time<br>• When the holding brake is triggered to close, *bMBrakeReleased* is immediately set to FALSE even if the brake closing time has not yet elapsed! |
| | TRUE: Brake released (after the brake release time has expired). |
| **GP: GeneralPurpose**<br>The following outputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | |
| nGPAnalogSwitchInOut_a<br>INT | Analog switch: Output signal |
| nGPArithmetikOut_a<br>INT | Arithmetic: Output signal |
| nGPMulDivOut_a<br>INT | Multiplication/Division: Output signal |
| bGPDigitalDelayOut<br>BOOL | Binary delay element: Output signal |
| bGPLogicOut<br>BOOL | Binary logic: Output signal |
| bGPCompareOut<br>BOOL | Analog comparison: Output signal |
| bGPSignalOut1<br>...<br>bGPSignalOut4<br>BOOL | Binary signal monitor: Output signals<br>• The signal sources to be output are selected in C00411/1...4.<br>• A bit coded inversion of the output signals can be parameterised in C00412. |
| nGPSignalOut1_a<br>...<br>nGPSignalOut4_a<br>BOOL | Analog signal monitor: Output signals<br>• The signal sources to be output are selected in C00410/1...4.<br>• Gain and offset for each output signal can be parameterised in C00413/1...8. |
| bGPDFlipFlop_Out<br>BOOL | D-FlipFlop: Output signal |
| bGPDFlipFlop_NegOut<br>BOOL | D-FlipFlop: Negated output signal |
| **Free outputs**<br>The following outputs can freely be interconnected on the application level.<br>The signals from the application level can be transferred to the I/O level via these outputs. | |
| bFreeOut1 ... bFreeOut8<br>BOOL | Free outputs for digital signals |
| wFreeOut1 ... wFreeOut4<br>WORD | Free outputs for 16-bit signals |
| dnFreeOut1_p<br>dnFreeOut2_p<br>WORD | Free outputs for 32-bit signals |

### 8.5.2.1 Truth table for activating the pre-switch off

| Input | | Function | Response in the setpoint generator (FB L_NSet) |
|---|---|---|---|
| bJogCtrl InputSel1 | bJogCtrl InputSel2 | | |
| FALSE | FALSE | Pre-switch off inactive | No response<br>• The input signal *bJogCtrlRfgIn* is output directly at output *bRfgOut*.<br>• The input signals *bJogCtrlJog1* and *bJogCtrlJog2* are passed through 1:1 to the downstream FB L_NSet for the selection of fixed setpoints. |
| TRUE | FALSE | The *bJogCtrlSlowDown1* and *bJogCtrlStop1* inputs are evaluated. | **Pre-switch off can be activated**<br>• If the slowdown function is activated via the selected *bJogCtrlSlowDown* input, fixed setpoint 2 is activated in the setpoint generator.<br>• If the stop function is activated via the selected *bJogCtrlStop* input, the setpoint generator is deactivated. |
| FALSE | TRUE | The *bJogCtrlSlowDown2* and *bJogCtrlStop2* inputs are evaluated. | |
| TRUE | TRUE | The inputs *bSJogCtrllowDown3* and *bJogCtrlStop3* are evaluated. | |

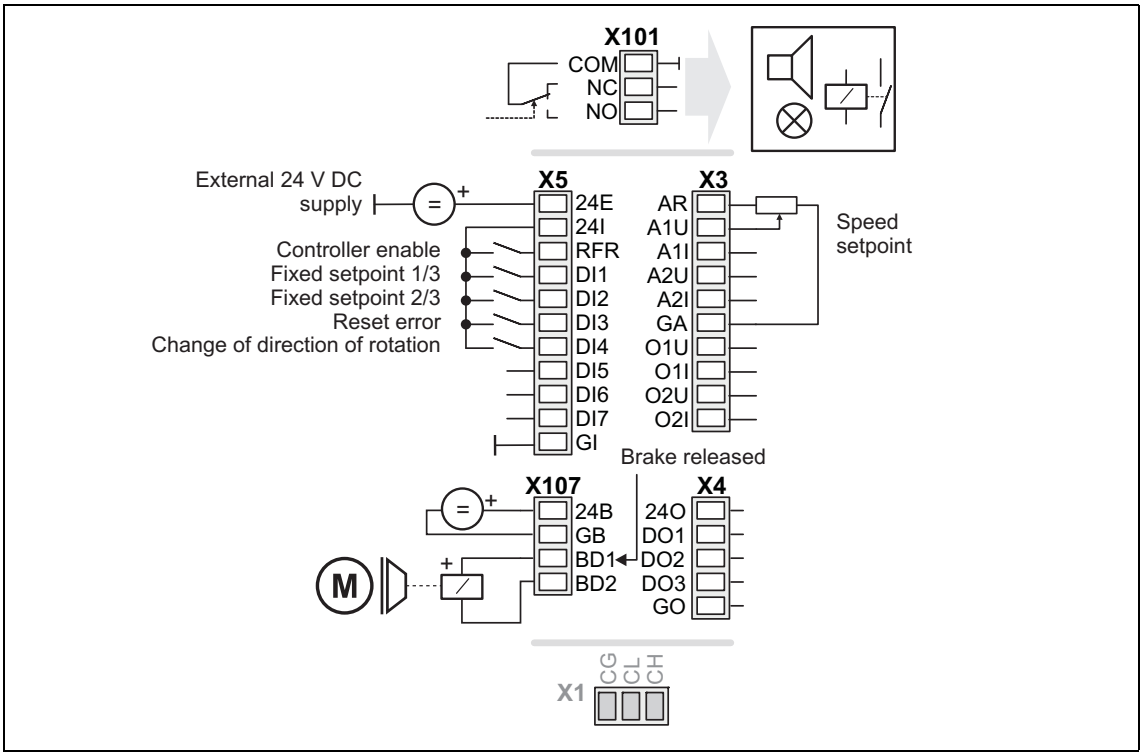
[8-1] Truth table for activating the pre-switch off

### 8.5.3 Terminal assignment of the control modes

The following comparison provides information about which inputs/outputs of the application block **LA_SwitchPos** are interconnected to the digital and analog input/output terminals of the inverter in the different control modes.

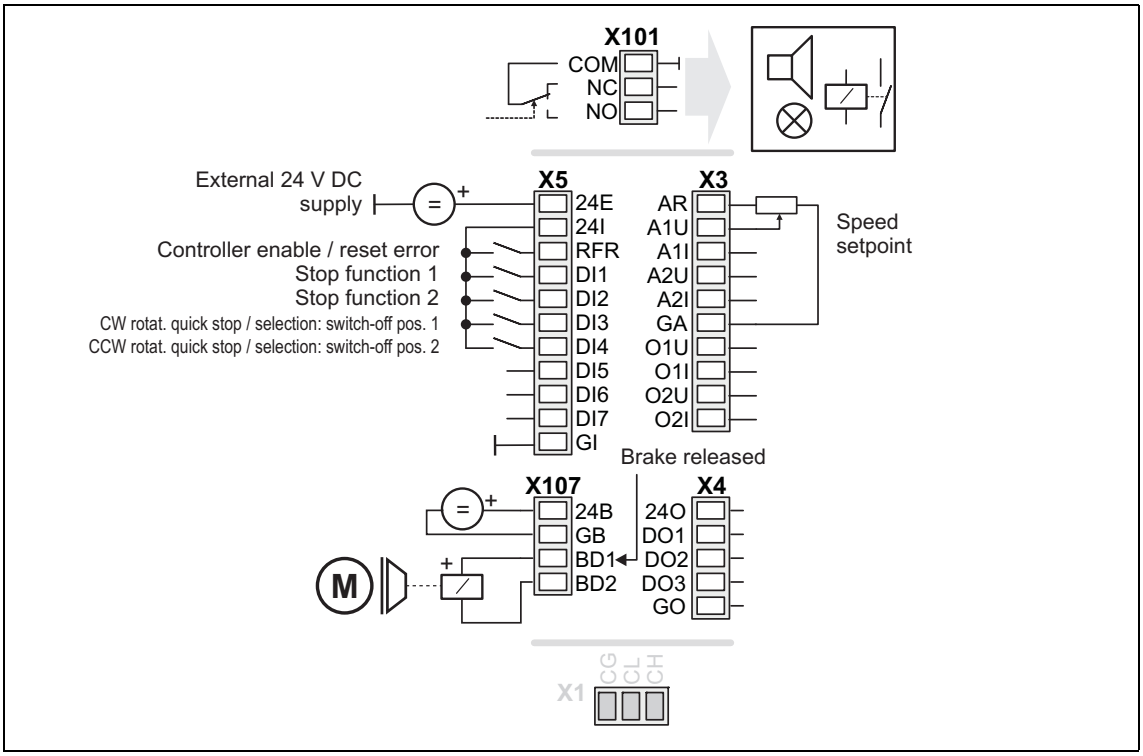
| | Control mode (C00007) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 10: Terminals 0 | 12: Terminals 2 | 14: Terminals 11 | 16: Terminal 16 | 20: Keypad | 21: PC | 30: CAN | 40: MCI |
| **Digital input terminals** | | | | | | | | |
| **X5/RFR** | **Controller enable** | **Controller enable / Reset of error message**<br>bFailReset | | | | | | |
| **X5/DI1** | **Fixed setpoint 1/3**<br>bJogCtrlJog1 | **Stop function 1**<br>bJogCtrlStop1 | | **Fixed setpoint 1/3**<br>bJogCtrlJog1 | - | - | **Stop function 1**<br>bJogCtrlStop1 | |
| **X5/DI2** | **Fixed setpoint 2/3**<br>bJogCtrlJog2 | **Stop function 2**<br>bJogCtrlStop2 | **Selection: Pre-switch off 1**<br>bJogCtrlSlowDown 1 | **Fixed setpoint 2/3**<br>bJogCtrlJog2 | - | - | **Selection: Pre-switch off 1**<br>bJogCtrlSlowDown 1 | |
| **X5/DI3** | **Reset error message**<br>bFailReset | **CW rotation quick stop**<br>bRLQCw<br>Selection: Switch-off position 1<br>bJogCtrlInputSel1 | | **CW rotation quick stop**<br>bRLQCw | - | - | **Stop function 2**<br>bJogCtrlStop2 | |
| **X5/DI4** | **Change of direction of rotation**<br>bSetSpeedCcw | **CCW rotation quick stop**<br>bRLQCcw<br>Selection: Switch-off position 2<br>bJogCtrlInputSel2 | | **CCW rotation quick stop**<br>bRLQCcw | - | - | **Selection: Pre-switch off 2**<br>bJogCtrlSlowDown 2 | |
| **X5/DI5** | - | - | **Stop function 2**<br>bJogCtrlStop2 | - | - | - | **Stop function 3**<br>bJogCtrlStop3 | |
| **X5/DI6** | - | - | **Selection: Pre-switch off 2**<br>bJogCtrlSlowDown 2 | - | - | - | **Selection: Pre-switch off 3**<br>bJogCtrlSlowDown 3 | |
| **X5/DI7** | - | - | - | - | - | - | - | - |
| **Analog input terminals** | | | | | | | | |
| **X3/A1U, A1I** | **Main speed setpoint**<br>nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) | | | | - | - | **Additional speed setpoint**<br>nAuxSetValue_a<br>10 V ≡ 100 % reference speed (C00011) | |
| **X3/A2U, A2I** | - | - | - | - | - | - | - | - |
| **Digital output terminals** | | | | | | | | |
| **X4/DO1 ... DO3** | - | - | - | - | - | - | - | - |
| **X107/BD1, BD2** | **Control of the holding brake**<br>bMBrakeReleaseOut | | | | | | | |
| **X101/COM, NO** | - | - | - | - | - | - | - | - |
| **Analog output terminals** | | | | | | | | |
| **X3/O1U, O1I** | - | - | - | - | - | - | - | - |
| **X3/O2U, O2I** | | | | | | | | |

### 8.5.3.1 Terminals 0



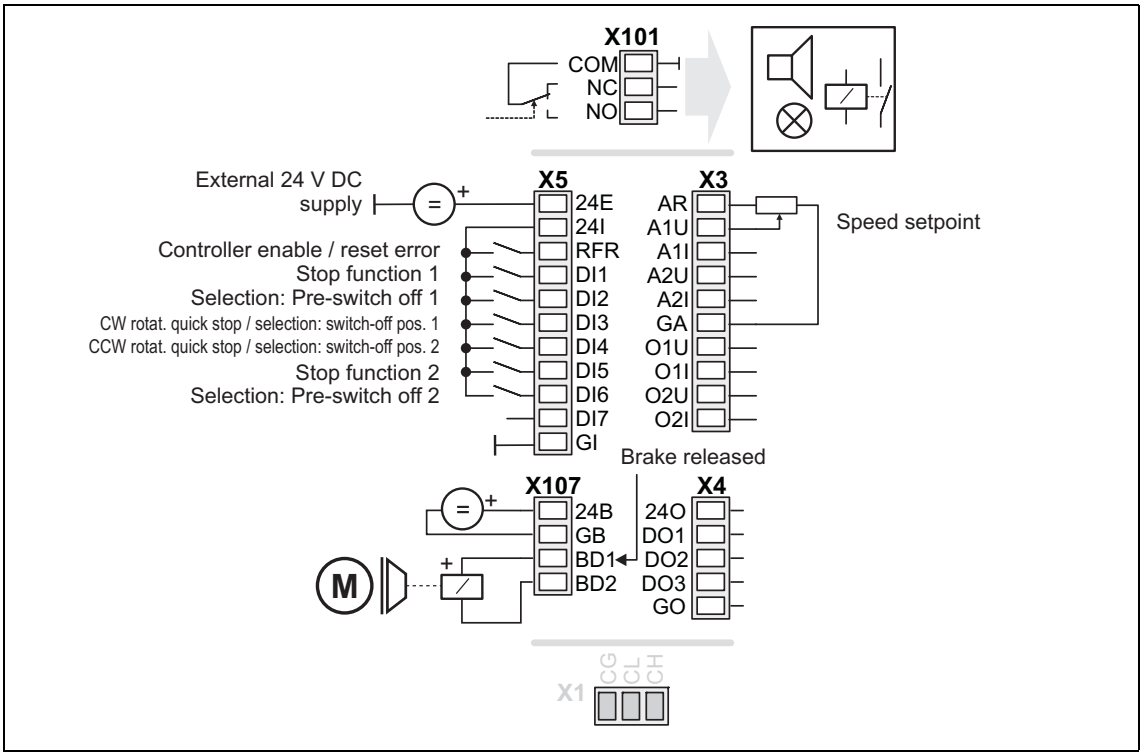

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | - | | | |
| | | | | |
| **X5/RFR** | - | | **X3/A1U** | LA_SwitchPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | LA_SwitchPos.bJogCtrlJog1 | | **X3/A1I** | |
| **X5/DI2** | LA_SwitchPos.bJogCtrlJog2 | | **X3/A2U** | - |
| **X5/DI3** | LA_SwitchPos.bFailReset | | **X3/A2I** | - |
| **X5/DI4** | LA_SwitchPos.bSetSpeedCcw | | **X3/O1U** | - |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | |
| **X107/BD1** | LA_SwitchPos.bMBrakeReleaseOut | | **X4/DO1** | - |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.5.3.2 Terminals 2



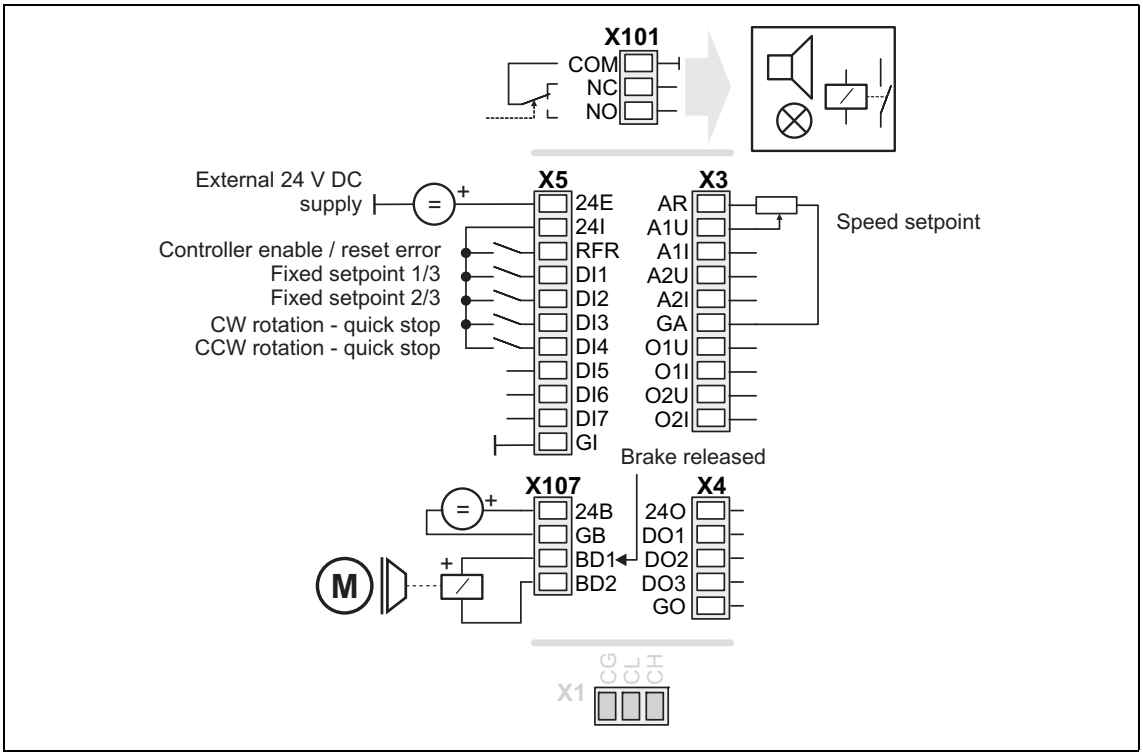

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | - | | | |
| | | | | |
| **X5/RFR** | LA_SwitchPos: bFailReset | | **X3/A1U** | LA_SwitchPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | LA_SwitchPos: bJogCtrlStop1 | | **X3/A1I** | |
| **X5/DI2** | LA_SwitchPos: bJogCtrlStop2 | | **X3/A2U** | - |
| **X5/DI3** | LA_SwitchPos: bRLQCw | | **X3/A2I** | - |
| | LA_SwitchPos: bJogCtrlInputSel1 | | **X3/O1U** | - |
| **X5/DI4** | LA_SwitchPos: bRLQCcw | | **X3/O1I** | - |
| | LA_SwitchPos: bJogCtrlInputSel2 | | **X3/O2U** | - |
| **X5/DI5** | - | | | |
| **X5/DI6** | - | | **X4/DO1** | - |
| **X5/DI7** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |
| **X107/BD1** | LA_SwitchPos.bMBrakeReleaseOut | | | |
| **X107/BD2** | - | | | |

### 8.5.3.3 Terminals 11



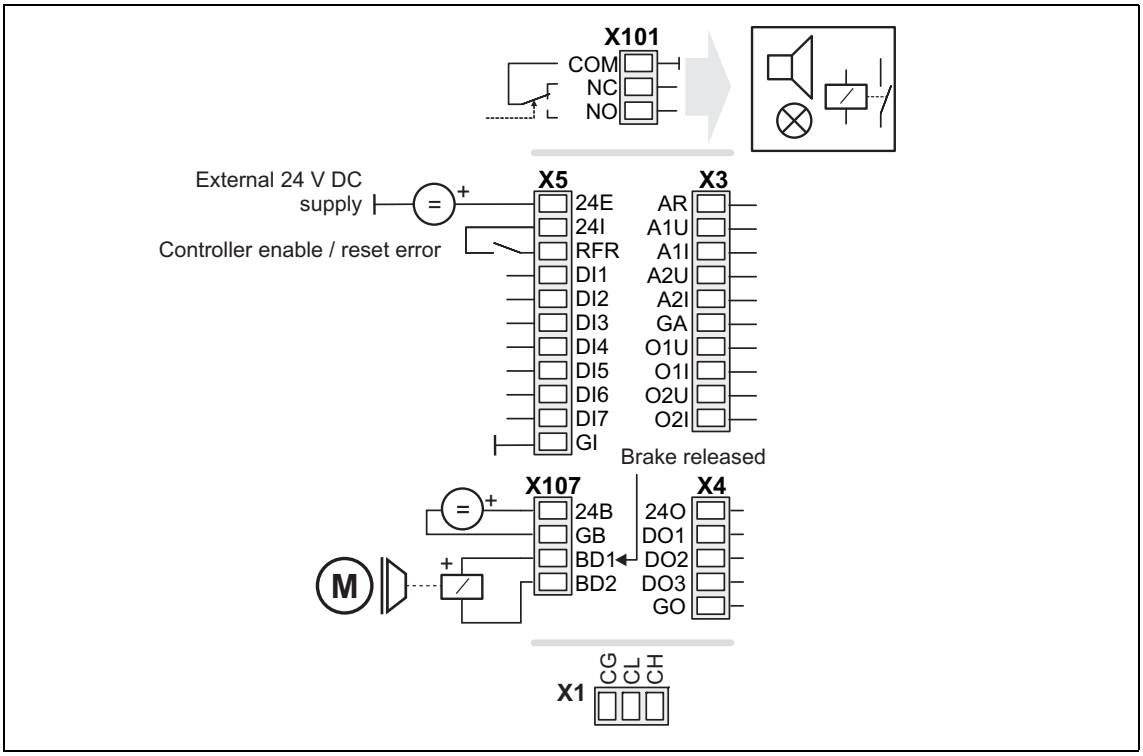

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | - | | | |
| | | | | |
| **X5/RFR** | LA_SwitchPos: bFailReset | | **X3/A1U** | LA_SwitchPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | LA_SwitchPos: bJogCtrlStop1 | | **X3/A1I** | |
| **X5/DI2** | LA_SwitchPos: bJogCtrlSlowDown1 | | **X3/A2U** | - |
| **X5/DI3** | LA_SwitchPos: bRLQCw | | **X3/A2I** | - |
| | LA_SwitchPos: bJogCtrlInputSel1 | | **X3/O1U** | - |
| **X5/DI4** | LA_SwitchPos: bRLQCcw | | **X3/O1I** | - |
| | LA_SwitchPos: bJogCtrlInputSel2 | | **X3/O2U** | - |
| **X5/DI5** | LA_SwitchPos: bJogCtrlStop2 | | | |
| **X5/DI6** | LA_SwitchPos: bJogCtrlSlowDown2 | | **X4/DO1** | - |
| **X5/DI7** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |
| **X107/BD1** | LA_SwitchPos.bMBrakeReleaseOut | | | |
| **X107/BD2** | - | | | |

### 8.5.3.4 Terminal 16



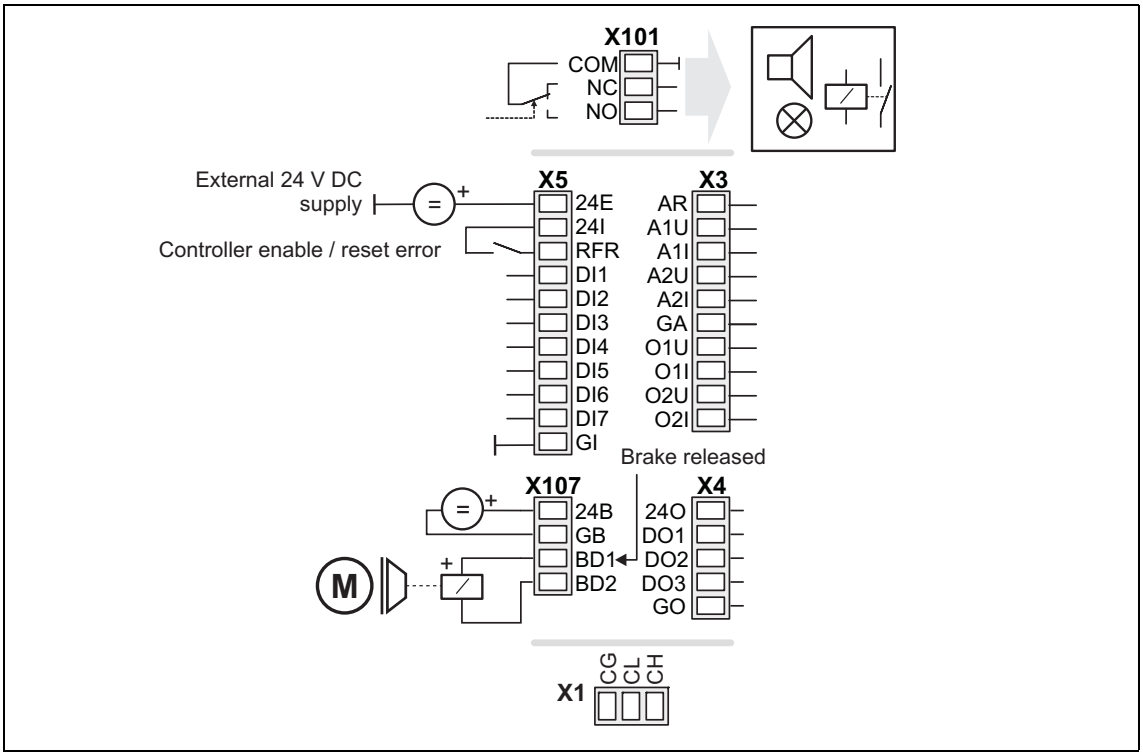

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | - | | | |
| | | | | |
| **X5/RFR** | LA_SwitchPos: bFailReset | | **X3/A1U** | LA_SwitchPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | LA_SwitchPos: bJogCtrlJog1 | | **X3/A1I** | |
| **X5/DI2** | LA_SwitchPos: bJogCtrlJog2 | | **X3/A2U** | - |
| **X5/DI3** | LA_SwitchPos: bRLQCw | | **X3/A2I** | - |
| **X5/DI4** | LA_SwitchPos: bRLQCcw | | **X3/O1U** | - |
| **X5/DI5** | - | | **X3/O1I** | - |
| **X5/DI6** | - | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | |
| **X107/BD1** | LA_SwitchPos.bMBrakeReleaseOut | | **X4/DO1** | - |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

### 8.5.3.5 Keypad



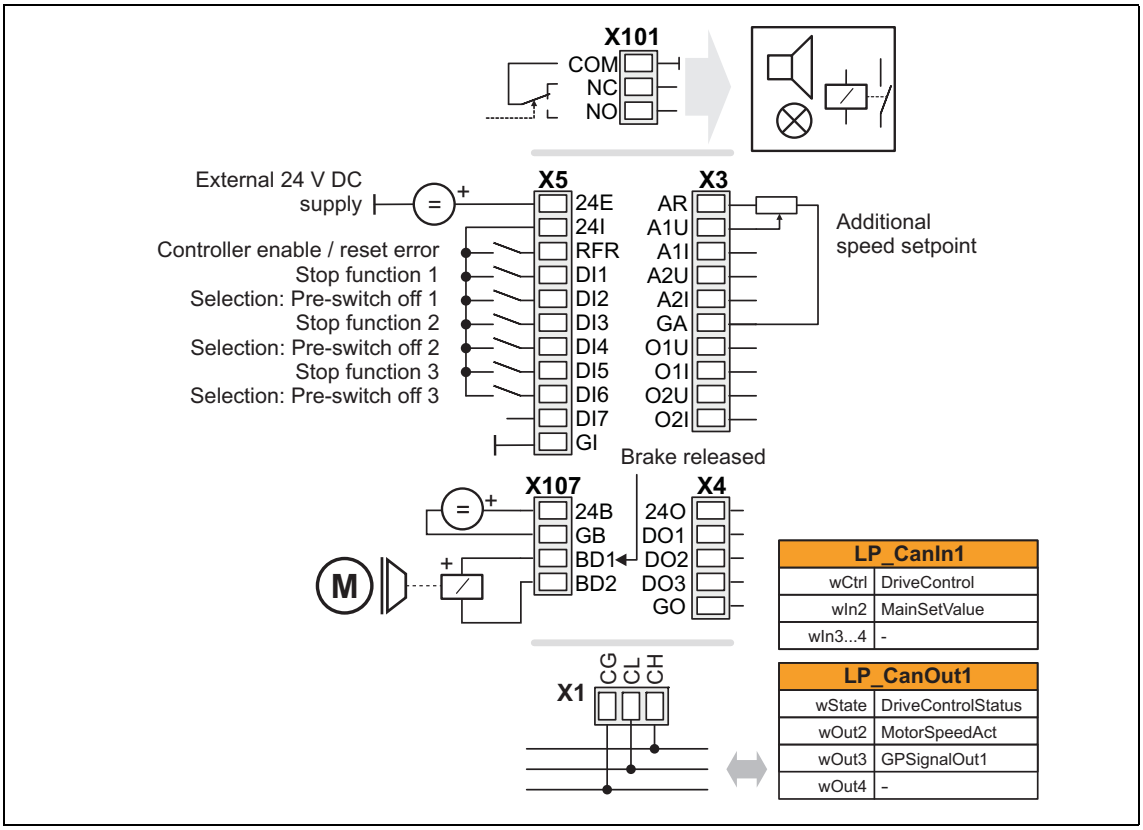

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| X101/NC-NO | - | | | |
| | | | | |
| X5/RFR | - | | X3/A1U | - |
| X5/DI1 | - | | X3/A1I | - |
| X5/DI2 | - | | X3/A2U | - |
| X5/DI3 | - | | X3/A2I | - |
| X5/DI4 | - | | X3/O1U | - |
| X5/DI5 | - | | X3/O1I | - |
| X5/DI6 | - | | X3/O2U | - |
| X5/DI7 | - | | X3/O2I | - |
| | | | | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut | | X4/DO1 | - |
| X107/BD2 | - | | X4/DO2 | - |
| | | | X4/DO3 | - |

### 8.5.3.6 PC




| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| X101/NC-NO | - | | | |
| | | | | |
| X5/RFR | - | | X3/A1U | - |
| X5/DI1 | - | | X3/A1I | - |
| X5/DI2 | - | | X3/A2U | - |
| X5/DI3 | - | | X3/A2I | - |
| X5/DI4 | - | | X3/O1U | - |
| X5/DI5 | - | | X3/O1I | - |
| X5/DI6 | - | | X3/O2U | - |
| X5/DI7 | - | | X3/O2I | - |
| | | | | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut | | X4/DO1 | - |
| X107/BD2 | - | | X4/DO2 | - |
| | | | X4/DO3 | - |

### 8.5.3.7 CAN



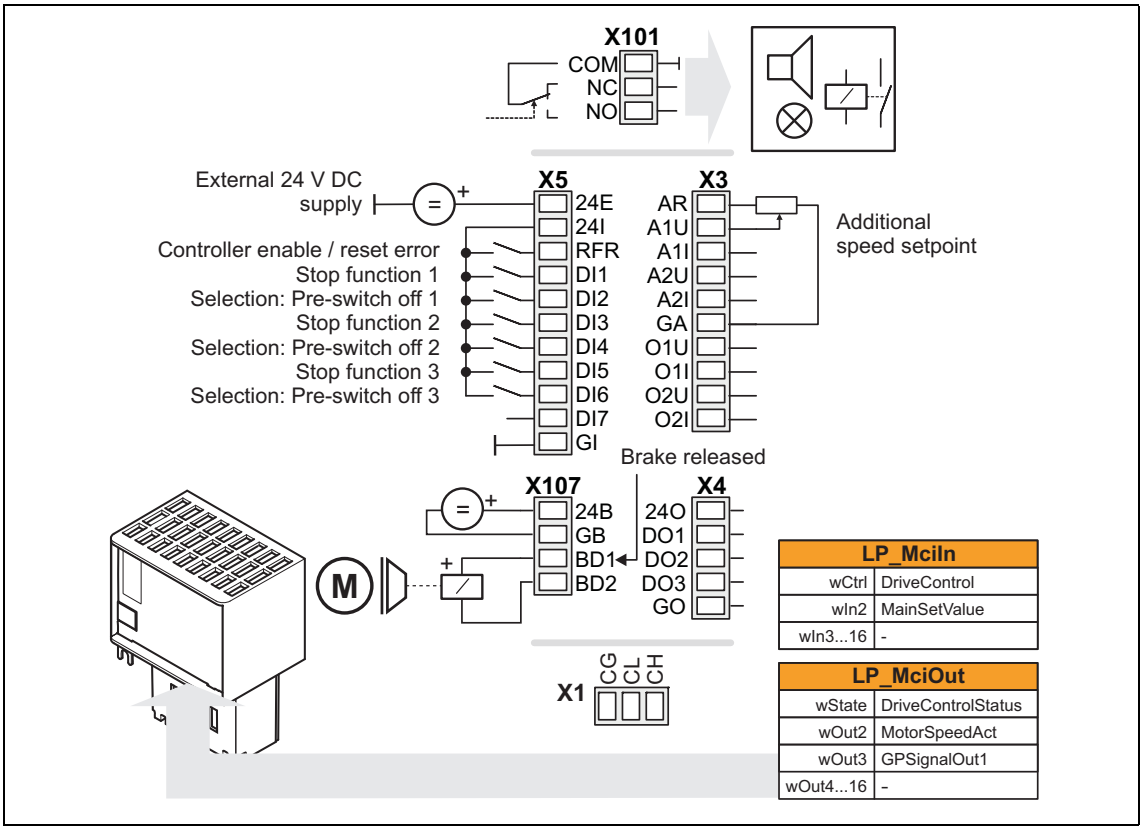

| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| **X101/NC-NO** | - | | | |
| | | | | |
| **X5/RFR** | LA_SwitchPos: bFailReset | | **X3/A1U** | LA_SwitchPos.nAuxSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X5/DI1** | LA_SwitchPos: bJogCtrlStop1 | | **X3/A1I** | |
| **X5/DI2** | LA_SwitchPos: bJogCtrlSlowDown1 | | **X3/A2U** | - |
| **X5/DI3** | LA_SwitchPos: bJogCtrlStop2 | | **X3/A2I** | - |
| **X5/DI4** | LA_SwitchPos: bJogCtrlSlowDown2 | | **X3/O1U** | - |
| **X5/DI5** | LA_SwitchPos: bJogCtrlStop3 | | **X3/O1I** | - |
| **X5/DI6** | LA_SwitchPos: bJogCtrlSlowDown3 | | **X3/O2U** | - |
| **X5/DI7** | - | | **X3/O2I** | - |
| | | | | |
| **X107/BD1** | LA_SwitchPos.bMBrakeReleaseOut | | **X4/DO1** | - |
| **X107/BD2** | - | | **X4/DO2** | - |
| | | | **X4/DO3** | - |

▸ Process data assignment for fieldbus communication (📖 547)

**Note!**

You must set the setpoint arithmetic in C00190 to "1: NOut = NSet + NAdd" so that the additional speed setpoint selected via the analog input A1U has an additive effect.

### 8.5.3.8 MCI




| Connection | Assignment | | Connection | Assignment |
|---|---|---|---|---|
| X101/NC-NO | - | | | |
| | | | | |
| X5/RFR | LA_SwitchPos: bFailReset | | X3/A1U | LA_SwitchPos.nAuxSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X5/DI1 | LA_SwitchPos: bJogCtrlStop1 | | X3/A1I | |
| X5/DI2 | LA_SwitchPos: bJogCtrlSlowDown1 | | X3/A2U | - |
| X5/DI3 | LA_SwitchPos: bJogCtrlStop2 | | X3/A2I | - |
| X5/DI4 | LA_SwitchPos: bJogCtrlSlowDown2 | | X3/O1U | - |
| X5/DI5 | LA_SwitchPos: bJogCtrlStop3 | | X3/O1I | - |
| X5/DI6 | LA_SwitchPos: bJogCtrlSlowDown3 | | X3/O2U | - |
| X5/DI7 | - | | X3/O2I | - |
| | | | | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut | | X4/DO1 | - |
| X107/BD2 | - | | X4/DO2 | - |
| | | | X4/DO3 | - |

▸ Process data assignment for fieldbus communication (📖 547)

**Note!**

You must set the setpoint arithmetic in C00190 to "1: NOut = NSet + NAdd" so that the additional speed setpoint selected via the analog input A1U has an additive effect.

### 8.5.4 Process data assignment for fieldbus communication

The fieldbus communication is connected (preconfigured) to the previously selected technology application by selecting the corresponding control mode in C00007:

- "30: CAN" for the connection to the system bus (CAN)
- "40: MCI" for the connection to a plugged-on communication module (e.g. PROFIBUS)

The assignment of the process data words depends only on the application, not on the bus system used:

| Input words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControl | Control word<br>• For bit assignment see the table below. |
| Word 2 | MainSetValue | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| Word 3 | - | Not preconfigured |
| Word 4 | - | Not preconfigured |
| Words 5 ... 16 | - | Not preconfigured<br>• Only available in control mode "40: MCI". |

| Control word | Name | Function |
|---|---|---|
| Bit 0 | SwitchOn | 1 ≡ Change to the "SwitchedOn" device status<br>• This bit must be set in the CAN/MCI control word to ensure that the device changes to the "SwitchedOn" device status after mains connection without the need for a master control specifying this bit via fieldbus.<br>• If control via a bus system is not wanted (e.g. in the case of control via terminals), the *wDriveCtrl* output signal of the LS_ParFix system block can be connected to the control word inputs. |
| Bit 1 | DisableVoltage | 1 ≡ Inhibit inverter control (pulse inhibit) |
| Bit 2 | SetQuickStop | 1 ≡ Activate quick stop (QSP).<br>▸ Activate/deactivate quick stop (📖 111) |
| Bit 3 | EnableOperation | 1 ≡ Enable inverter (RFR)<br>• If control via terminals is performed, this bit must be set both in the CAN control word and in the MCI control word. Otherwise, the controller is inhibited.<br>▸ Enable/inhibit inverter (📖 110) |
| Bit 4 | ModeSpecific_1 | Reserved (currently not assigned) |
| Bit 5 | JogCtrlInputSel1 | Binary coded selection of the switch-off position 1 ... 3<br>• Activation of the signal pairs *bJogCtrlSlowDown1/bJogCtrlStop1*, *bJogCtrlSlowDown2/bJogCtrlStop2* or *bJogCtrlSlowDown3/bJogCtrlStop3* according to the Truth table for activating the pre-switch off. |
| Bit 6 | JogCtrlInputSel2 | |
| Bit 7 | ResetFault | 1 ≡ Reset fault (trip reset)<br>• Acknowledge error message (if the error cause has been eliminated).<br>▸ Reset error (📖 114) |
| Bit 8 | bJogCtrlRfgIn | Ramping down of the setpoint generator in the downstream L_NSet FB according to the Truth table for activating the pre-switch off |
| Bit 9 | reserved_1 | Reserved (currently not assigned) |
| Bit 10 | reserved_2 | |

| Control word | Name | Function |
|---|---|---|
| Bit 11 | MBrkRelease | Holding brake control:<br>0 ≡ Apply brake<br>1 ≡ Release brake<br>• In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off"). |
| Bit 12 | JogCtrlJog1 | Binary coded selection of the fixed setpoints (JOG setpoints) |
| Bit 13 | JogCtrlJog2 | |
| Bit 14 | SetFail | 1 ≡ Set error (trip set) |
| Bit 15 | SetSpeedCcw | 0 ≡ Direction of rotation to the right (Cw)<br>1 ≡ Direction of rotation to the left (Ccw) |

| Output words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControlStatus | Status word<br>• For bit assignment see the table below. |
| Word 2 | MotorSpeedAct | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| Word 3 | GPSignalOut1 | Analog signal monitor: Output signal 1<br>• The selection of the signal source to output is executed in C00410/1.<br>• Gain and offset for the output signal can be parameterised in C00413/1 and C00413/2.<br>• For a detailed functional description see the L_SignalMonitor_a FB. |
| Word 4 | - | Not preconfigured |
| Words 5 ... 16 | - | Not preconfigured<br>• Only available in control mode "40: MCI". |

| Status word | Name | Status |
|---|---|---|
| Bit 0 | DriveFail | 1 ≡ Inverter in the error status<br>• "Fault" device status is active. |
| Bit 1 | PowerDisabled | 1 ≡ Inverter control inhibited (pulse inhibit is active) |
| Bit 2 | DriveReady | 1 ≡ Inverter is ready for operation<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the inverter is still inhibited by the user (controller inhibit). |
| Bit 3 | SpeedCcw | 0 ≡ Direction of rotation to the right (Cw)<br>1 ≡ Direction of rotation to the left (Ccw) |
| Bit 4 | QSPIsActive | 1 ≡ Quick stop is active |
| Bit 5 | BrakeReleased | 1 ≡ Brake released (after the brake opening time has elapsed) |
| Bit 6 | ActSpeedIsZero | During open-loop operation:<br>1 ≡ Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>1 ≡ Actual speed value < Comparison value (C00024) |
| Bit 7 | ControllerInhibit | 1 ≡ Inverter is inhibited (controller inhibit is active) |
| Bit 8 | StatusCodeBit0 | Bit coded display of the active device status<br>▸ Device state machine and device states (see table [4-1]) |
| Bit 9 | StatusCodeBit1 | |
| Bit 10 | StatusCodeBit2 | |
| Bit 11 | StatusCodeBit3 | |
| Bit 12 | Warning | 1 ≡ A warning is indicated |

| Status word | Name | Status |
|---|---|---|
| Bit 13 | Trouble | 1 ≡ Inverter is in the "Trouble" device status<br>• E.g. if an overvoltage has occurred. |
| Bit 14 | JogCtrlInputSel1 | Binary coded selection of the switch-off position 1 ... 3<br>• Bit 5 and bit 6 of the control word. |
| Bit 15 | JogCtrlInputSel2 | |

### 8.5.5 Setting parameters (short overview)

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00011 | Appl.: Reference speed | 1500 | rpm |
| C00012 | Accel. time - main setpoint | 2.000 | s |
| C00013 | Decel. time - main setpoint | 2.000 | s |
| C00105 | Decel. time - quick stop | 2.000 | s |
| C00039/1 | Fixed setpoint 1 | 40.00 | % |
| C00039/2 | Fixed setpoint 2 | 60.00 | % |
| C00039/3 | Fixed setpoint 3 | 80.00 | % |
| C00039/4...15 | Fixed setpoint 4 ... 15 | 0.00 | % |
| C00101/1...15 | Add. accel. time 1 ... 15 | 0.000 | s |
| C00103/1...15 | Add. decel. time 1 ... 15 | 0.000 | s |
| C00105 | Decel. time - quick stop | 2.000 | s |
| C00106 | Auto-DCB: Hold time | 0.500 | s |
| C00107 | DCB braking: Hold time | 999.000 | s |
| C00134 | L_NSet_1: Ramp smoothing | 0: Off | |
| C00182 | L_NSet_1: S-ramp time PT1 | 20.00 | s |
| C00190 | L_NSet_1: Setpoint arithmetic | 0: Out = Set | |
| C00220 | L_NSet_1: Acceleration time - add. setpoint | 0.000 | s |
| C00221 | L_NSet_1: Deceleration time - add. setpoint | 0.000 | s |
| C00241 | L_NSet_1: Hyst. NSet reached | 0.50 | % |
| C00488/1 | InputSens.SlowDown1 | 0: Level | |
| C00488/2 | InputSens.Stop1 | 0: Level | |
| C00488/3 | InputSens.SlowDown2 | 0: Level | |
| C00488/4 | InputSens.Stop2 | 0: Level | |
| C00488/5 | InputSens.SlowDown3 | 0: Level | |
| C00488/6 | InputSens.Stop3 | 0: Level | |
| C00632/1 | L_NSet_1: Blocking speed 1 max | 0.00 | % |
| C00632/2 | L_NSet_1: Blocking speed 2 max | 0.00 | % |
| C00632/3 | L_NSet_1: Blocking speed 3 max | 0.00 | % |
| C00633/1 | L_NSet_1: Blocking speed 1 min | 0.00 | % |
| C00633/2 | L_NSet_1: Blocking speed 2 min | 0.00 | % |
| C00633/3 | L_NSet_1: Blocking speed 3 min | 0.00 | % |
| C00635 | L_NSet_1: nMaxLimit | 199.99 | % |
| C00636 | L_NSet_1: nMinLimit | -199.99 | % |
| C00670 | L_OffsetGainP_1: Gain | 1.0000 | |
| C00671 | L_OffsetGainP_2: Gain | 1.0000 | |

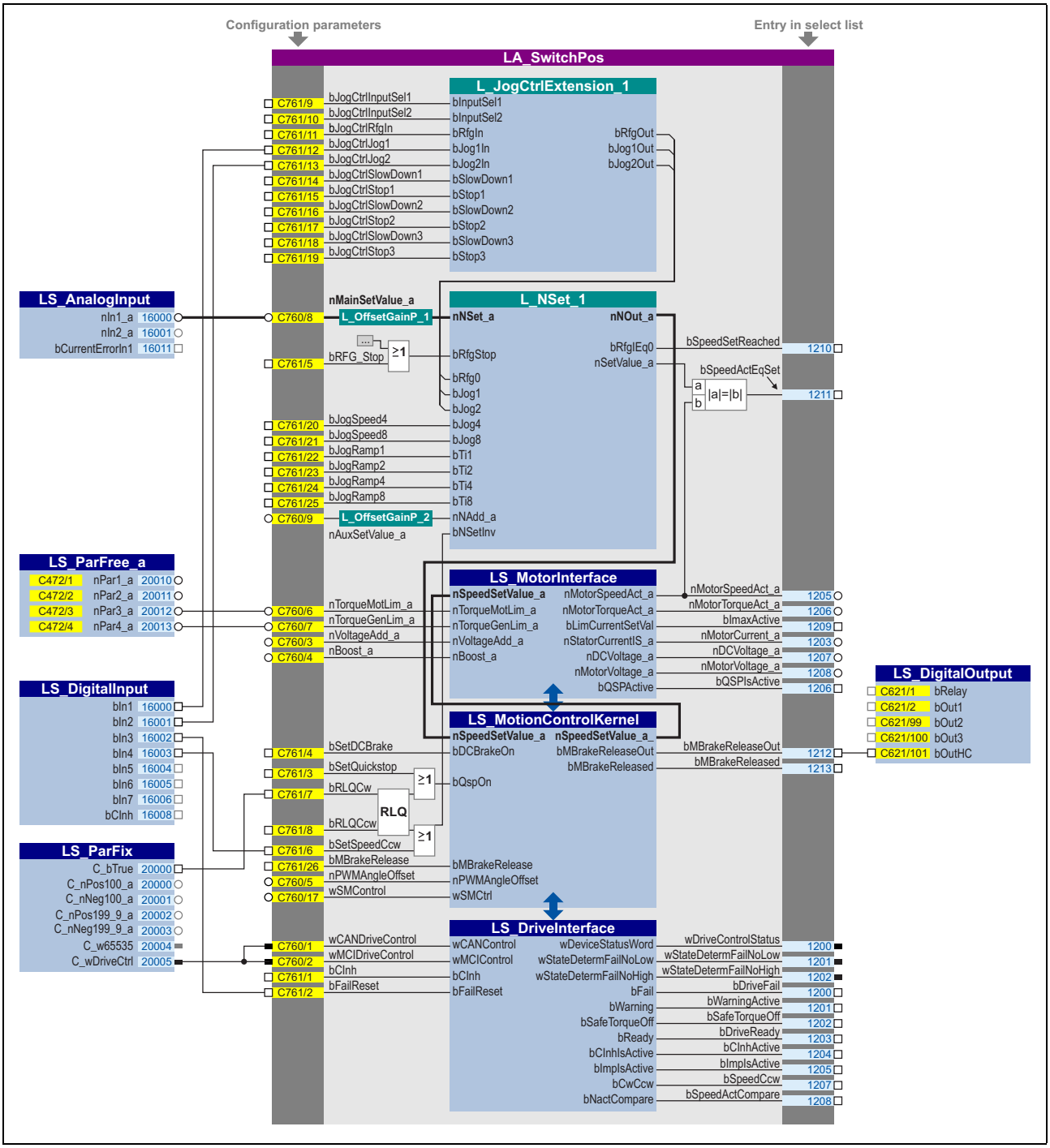
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00672 | L_OffsetGainP_3: Gain | 1.0000 | |
| C00696 | L_OffsetGainP_1: Offset | 0.00 | % |
| C00697 | L_OffsetGainP_2: Offset | 0.00 | % |
| C00698 | L_OffsetGainP_3: Offset | 0.00 | % |
| C00800 | L_MPot_1: Upper limit | 100.00 | % |
| C00801 | L_MPot_1: Lower limit | -100.00 | % |
| C00802 | L_MPot_1: Acceleration time | 10.0 | s |
| C00803 | L_MPot_1: Deceleration time | 10.0 | s |
| C00804 | L_MPot_1: Inactive fct. | 0: Retain value | |
| C00805 | L_MPot_1: Init fct. | 0: Load last value | |
| C00806 | Use of motor potentiometer | 0: No | |
| C02610/2 | MCK: Ramp time synchr. setpoint | 2.000 | s |
| C02611/1 | MCK: Pos. max. speed | 199.99 | % |
| C02611/2 | MCK: Pos. min. speed | 0.00 | % |
| C02611/3 | MCK: Neg. min. speed | 0.00 | % |
| C02611/4 | MCK: Neg. max. speed | 199.99 | % |

**Related topics:**

▸ "GeneralPurpose" functions (📖 554)

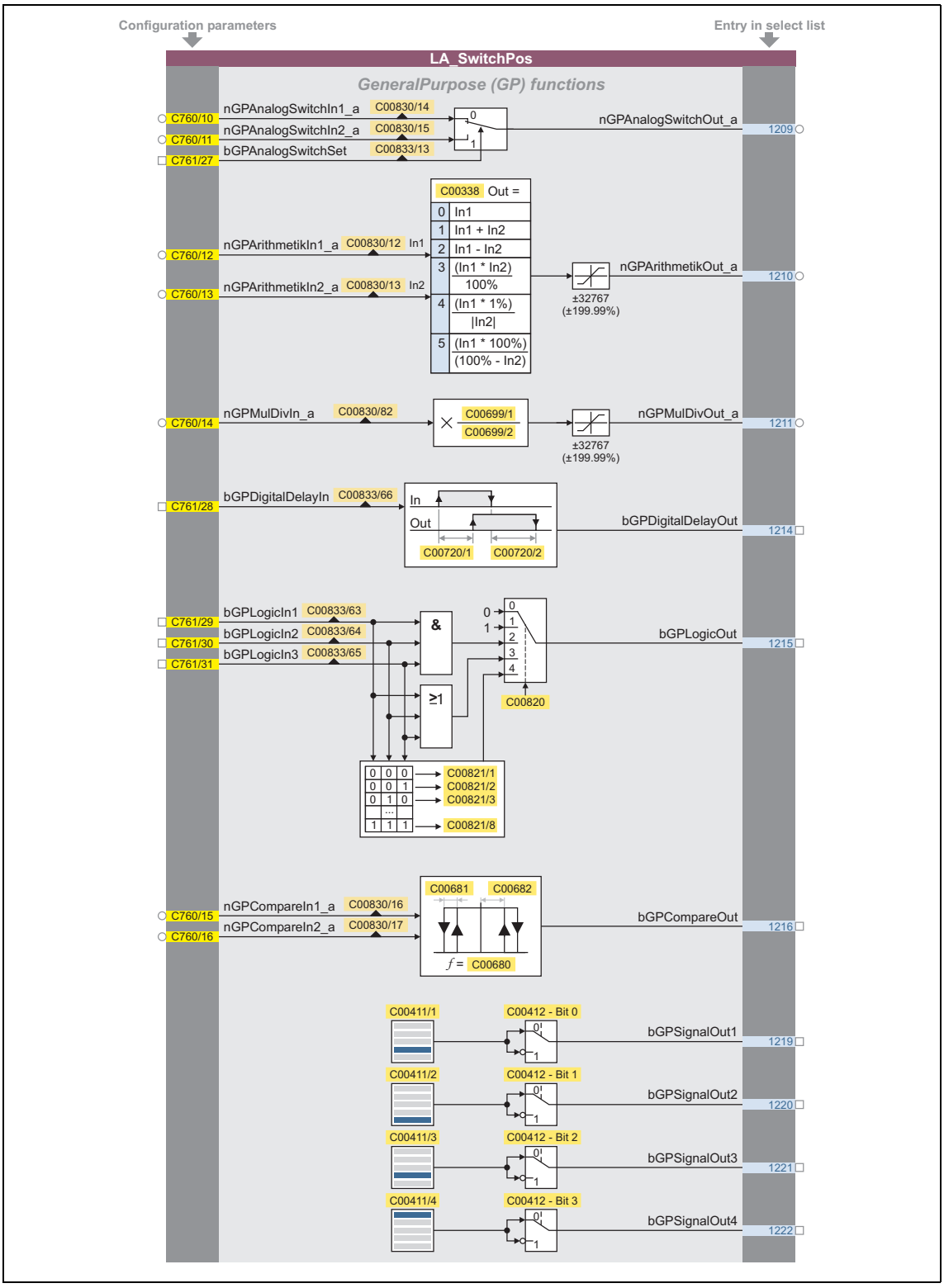
### 8.5.6 Configuration parameters

If required, the subcodes of C00760 and C00761 serve to change the pre-configured assignment of the application inputs:

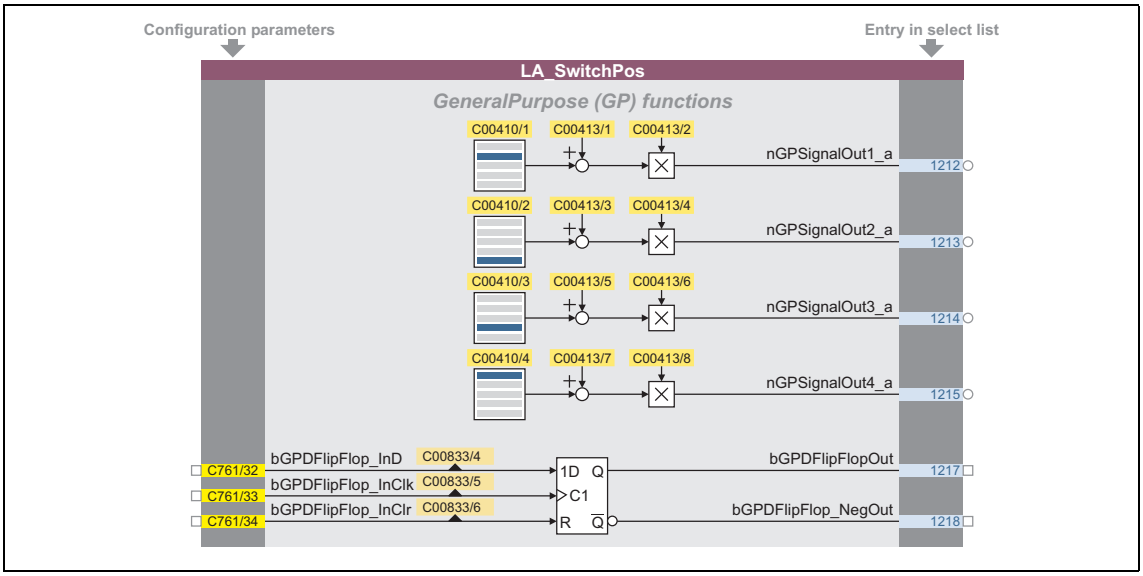



[8-19] Pre-assignment of the "Switch-off positioning" application in the "Terminals 0" control mode

### Configuration parameters for "GeneralPurpose" functions



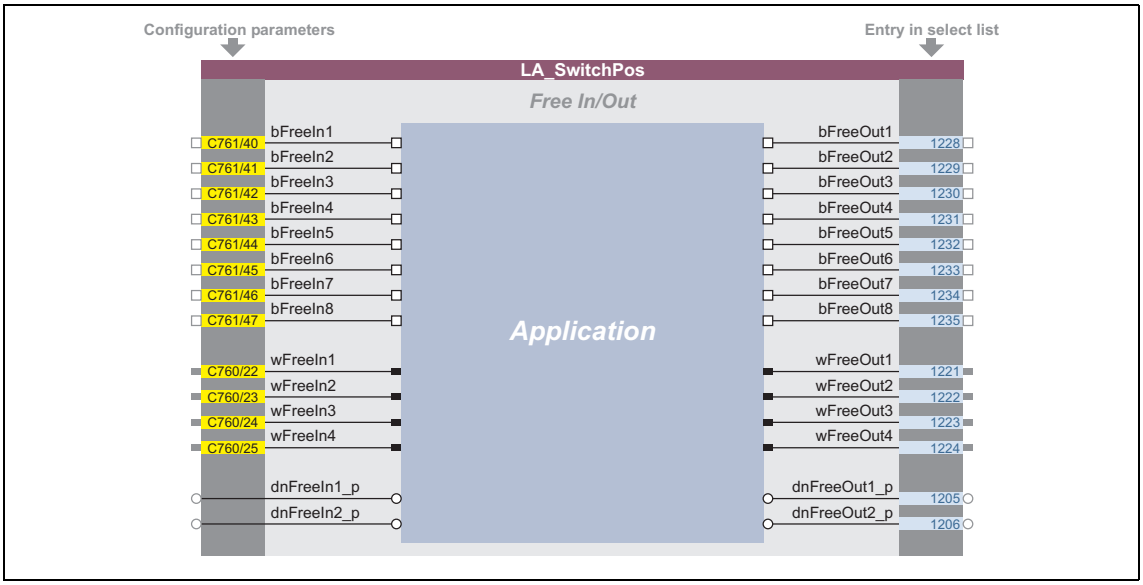

[8-20] "GeneralPurpose" functions




[8-21] "GeneralPurpose" functions (continuation)

### Free inputs and outputs

These inputs can be freely interconnected in the application level. They can be used to transfer signals from the I/O level to the application level and vice versa.




[8-22] Free inputs/outputs

**Related topics:**

- User-defined terminal assignment (📖 429)
- "GeneralPurpose" functions (📖 554)

## 8.6 "GeneralPurpose" functions

Each technology application provides different free logic and arithmetic functions, so-called "GeneralPurpose" functions.

For the interconnection of these functions, the application block features inputs and outputs on the I/O level, which are linked to the logic/arithmetic function.

**Note!**

In the Lenze setting, the connectors for the "GeneralPurpose" functions are hidden in the function block editor.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

**Tip!**

The inputs of the "GeneralPurpose" functions can also be linked to other output signals via the configuration parameters of the technology application.

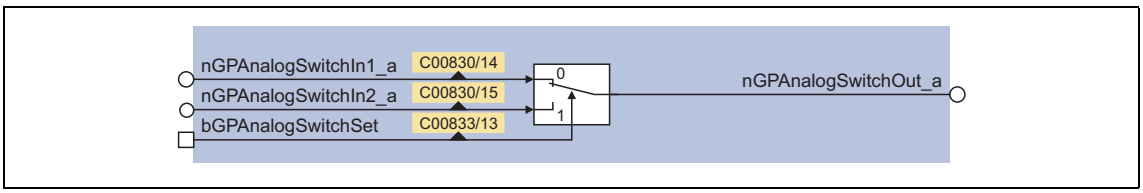
On the other hand, the outputs of the "GeneralPurpose" functions can be selected in the configuration parameters of other inputs.

**Related topics:**

- User-defined terminal assignment (📖 429)
- TA "Actuating drive speed": Configuration parameters (📖 462)
- TA "Table positioning": Configuration parameters (📖 524)
- TA "Switch-off positioning": Configuration parameters (📖 551)

### 8.6.1 Analog switch

This function switches between two analog input signals. The switch-over is controlled by a boolean input signal.

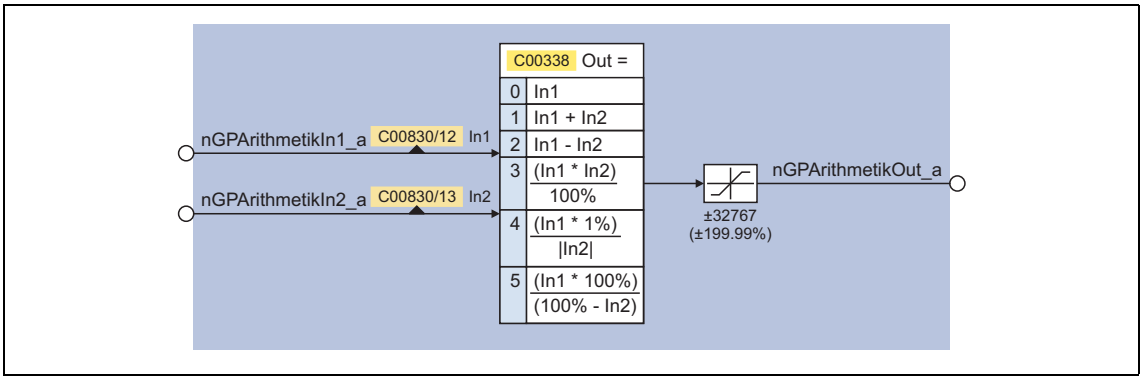



[8-23] GeneralPurpose function "Analog switch"

- For a detailed functional description see FB L_AnalogSwitch.

## 8.6.2 Arithmetic

This function links two analog signals arithmetically. The arithmetic function can be parameterised.



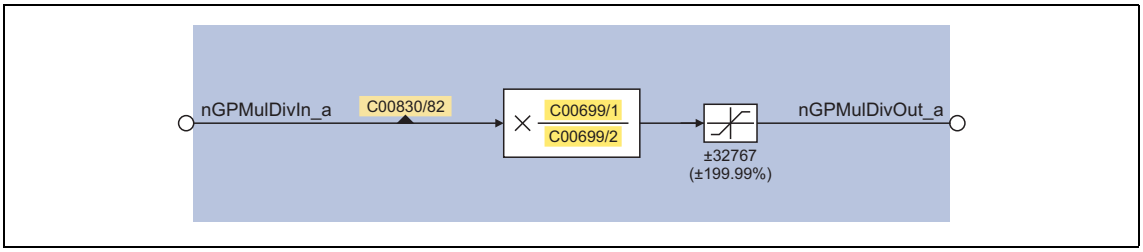

[8-24] GeneralPurpose function "Arithmetic"

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00338 | L_Arithmetik_1: Function | 0: nOut_a = nIn1_a | |

- For a detailed functional description see the L_Arithmetik FB.

## 8.6.3 Multiplication/Division

This function multiplies an analog input signal with a parameterisable factor. The factor must be selected in the form of a quotient (numerator and denominator).



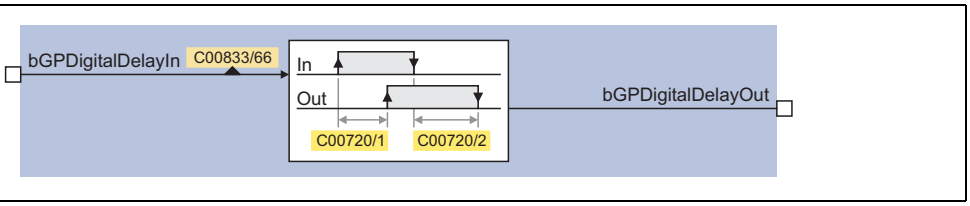

[8-25] GeneralPurpose function "Multiplication/division"

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00699/1 | L_MulDiv_1: Numerator | 0 | |
| C00699/2 | L_MulDiv_1: Denominator | 10000 | |

- For a detailed functional description see FB L_MulDiv.

### 8.6.4 Binary delay element

This function timely delays binary signals. On-delay and off-delay can be parameterised separately.



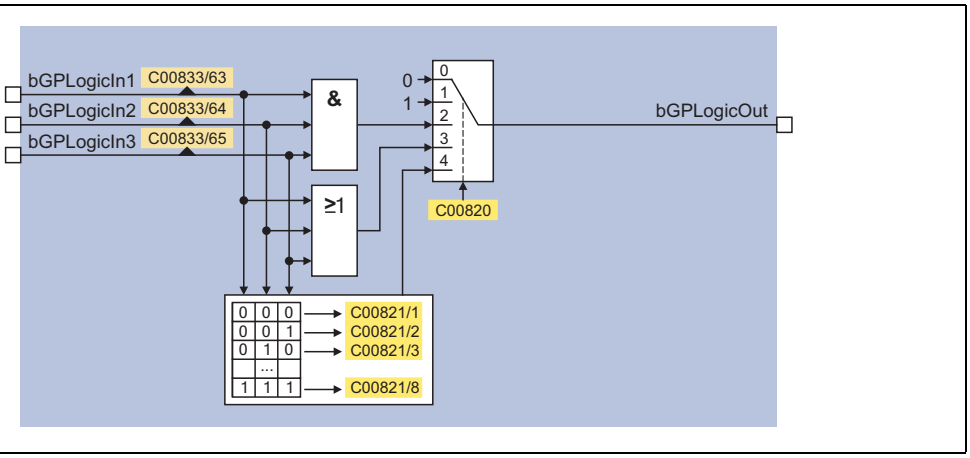

[8-26] GeneralPurpose function "Binary delay element"

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00720/1 | L_DigitalDelay_1: On delay | 0.000 | s |
| C00720/2 | L_DigitalDelay_1: Off delay | 0.000 | s |

- For a detailed functional description see FB L_DigitalDelay.

### 8.6.5 Binary logic

This function provides a binary output signal which is formed by a logic operation of the input signals. Alternatively, you can also select a fixed binary value which is independent of the input signals.



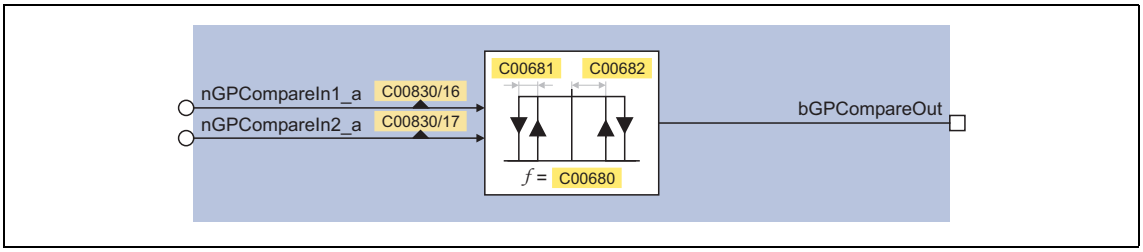

[8-27] GeneralPurpose function "Binary logic"

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00820 | L_DigitalLogic_1: Function | 0: bOut = 0 | |
| C00821/1 | bIn1=0/bIn2=0/bIn3=0 | 0: FALSE | |
| C00821/... | ... | ... | |
| C00821/8 | bIn1=1/bIn2=1/bIn3=1 | 0: FALSE | |

- For a detailed functional description see FB L_DigitalLogic.

### 8.6.6 Analog comparison

This function compares two analog signals and can be used e.g. to realise a trigger. The comparison operation, hysteresis and window size can be parameterised.



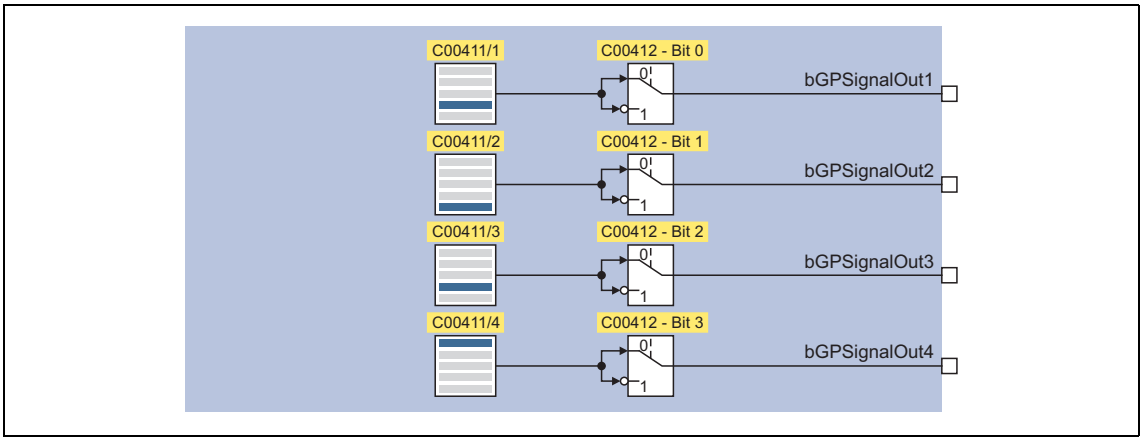

[8-28] GeneralPurpose function "Analog comparison"

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00680 | L_Compare_1: Fct. | 6: \|In1\| < \|In2\| | |
| C00681 | L_Compare_1: Hysteresis | 0.50 | % |
| C00682 | L_Compare_1: Window | 2.00 | % |

- For a detailed functional description see FB L_Compare.

### 8.6.7 Binary signal monitor

This function serves to output four binary signals selected from a list of all binary output signals available in the inverter. You can set an inversion of the output signals.



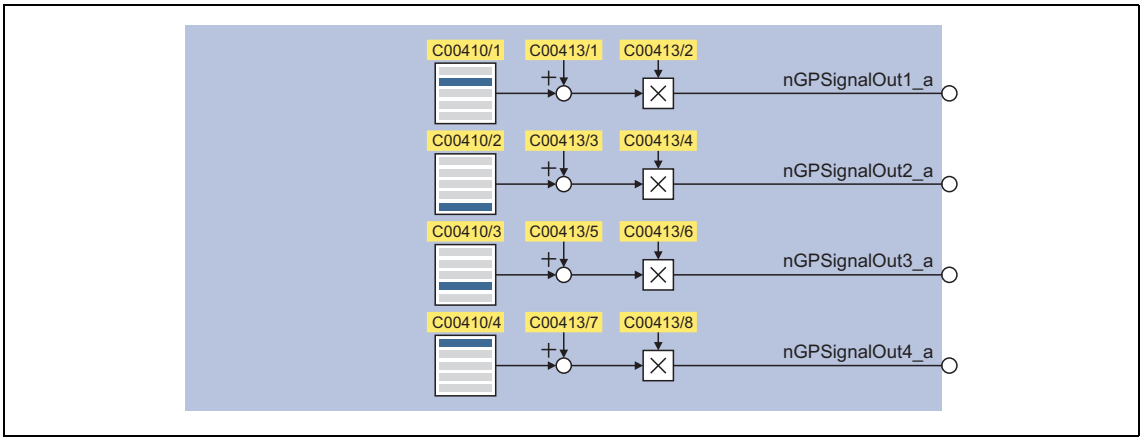

[8-29] GeneralPurpose function "Binary signal monitor"

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00411/1...4 | L_SignalMonitor_b: Signal 1 ... 4 | 0: Not connected | |
| C00412 | L_SignalMonitor_b: Inversion | Bit coded | |

- For a detailed functional description see FB L_SignalMonitor_b.

### 8.6.8 Analog signal monitor

This function serves to output four analog signals selected from a list of all analog output signals available in the inverter. Offset and gain of the source signals can be adjusted.



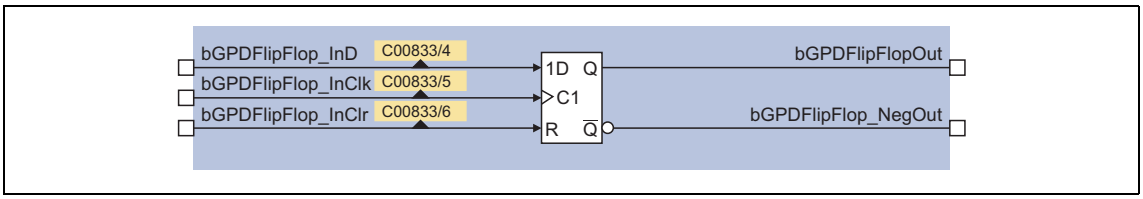

[8-30] GeneralPurpose function "Analog signal monitor"

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00410/1...4 | L_SignalMonitor_a: Signal 1 ... 4 | 0: Not connected | |
| C00413/1 | L_SignalMonitor_a: Signal 1 offset | 0.00 | % |
| C00413/2 | L_SignalMonitor_a: Signal 1 gain | 100.00 | % |
| C00413/3 | L_SignalMonitor_a: Signal 2 offset | 0.00 | % |
| C00413/4 | L_SignalMonitor_a: Signal 2 gain | 100.00 | % |
| C00413/5 | L_SignalMonitor_a: Signal 3 offset | 0.00 | % |
| C00413/6 | L_SignalMonitor_a: Signal 3 gain | 100.00 | % |
| C00413/7 | L_SignalMonitor_a: Signal 4 offset | 0.00 | % |
| C00413/8 | L_SignalMonitor_a: Signal 4 gain | 100.00 | % |

- For a detailed functional description see the L_SignalMonitor_a FB.

### 8.6.9 D-FlipFlop

This function saves the logic status of the data input (1D) in case of an active clock edge at the clock input (C1) and puts out its value in sequence at the output Q. If there is no active clock edge, the input value is not accepted.

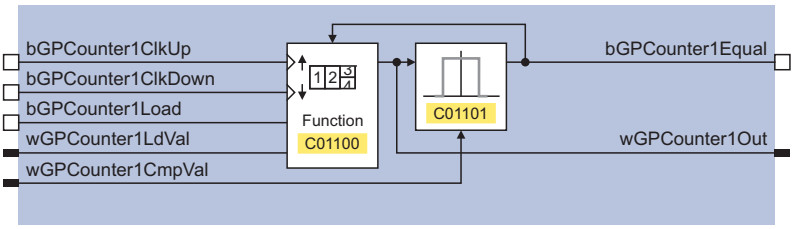



[8-31] GeneralPurpose function "D-FlipFlop" (clock-edge controlled)

- For a detailed functional description see FB L_DFlipFlop.

### 8.6.10 Counter

This function is a digital upcounter and downcounter with a comparison operation.




[8-32] GeneralPurpose function "Counter"

| Parameters | Info | Lenze setting<br>Value \| Unit |
|---|---|---|
| C01100/1 | L_Counter_1: Function | 0: Normal counting |
| C01101/1 | L_Counter_1: Comparison | 0: Greater than or equal to |

- Only available with TA "Table positioning".
- For a detailed functional description see FB L_Counter.

# 9 Basic drive functions (MCK)

In this chapter, the standard and basic drive functions integrated in the **Motion Control Kernel** (MCK) of the 8400 TopLine are described to which the active technology application can gain access via defined internal interfaces. As a result, the time-consuming creation of individual FB interconnections is avoided and the amount of work and complexity involved in the implementation of standard functions is minimised.

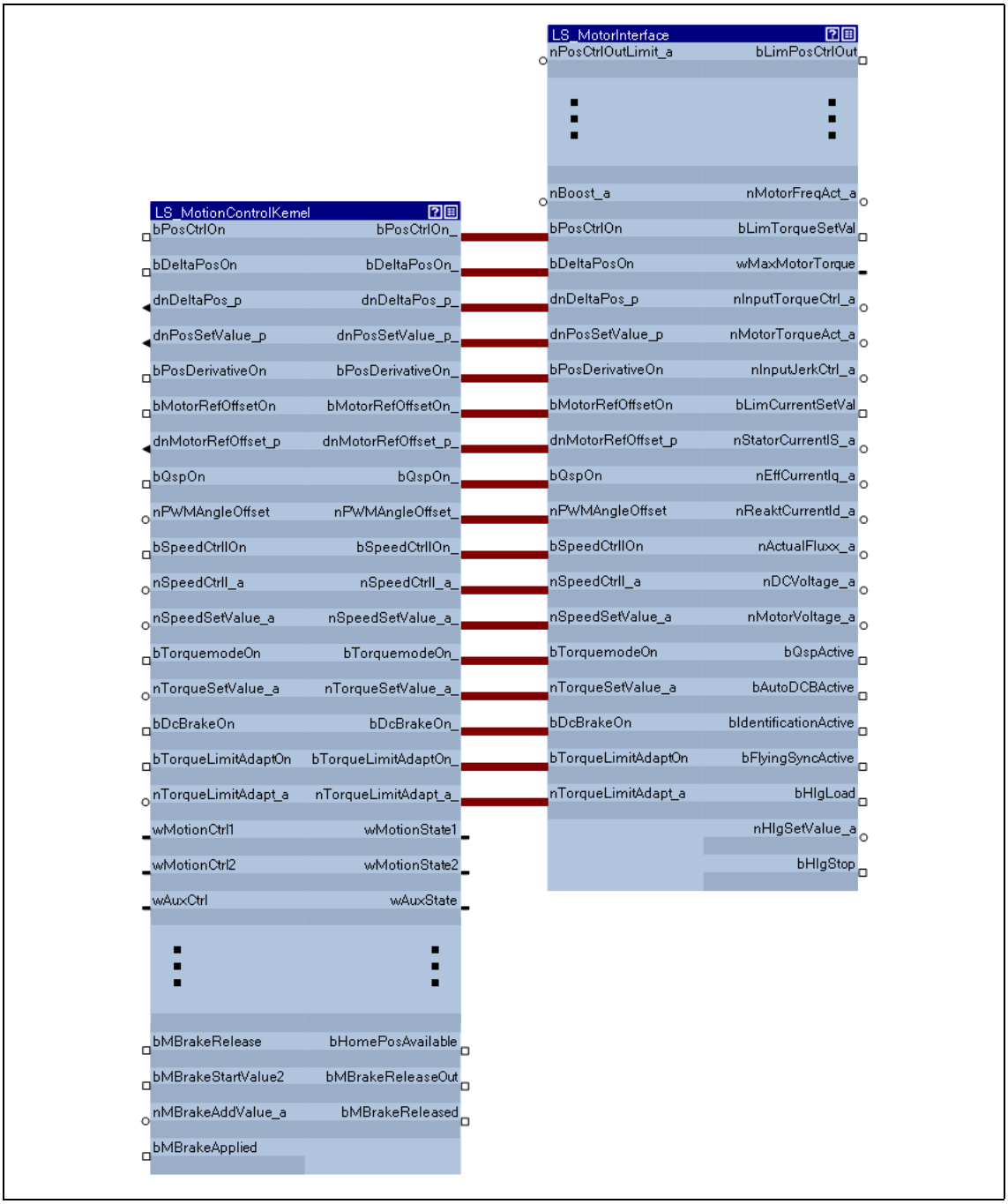
In the **Motion Control Kernel**, for example, an automatic holding brake control function is integrated which controls the holding brake in relation to the speed setpoint and various other internal control signals. Due to integrated automatic brake operation, the user is relieved of the task of managing these control signals.

Other standard functions integrated in the **Motion Control Kernel** for the 8400 TopLine inverter are, for example:

- Manual jog, e.g. for manual setting-up operation
- Homing for positioning functions of the drive
- Position profile generator for positioning functions of the drive

## 9.1 Basic signal flow

The **Motion Control Kernel** is connected between setpoint generator (e.g. ramp generator, PID process controller, etc.) and the motor control function in the case of the available technology applications. For problem-free interaction of the **Motion Control Kernel** and motor control function, the two associated system blocks LS_MotionControlKernel and LS_MotorInterface have interfaces with corresponding inputs/outputs. These are visible in the FB Editor for monitoring purposes and must be connected to each other:




[9-1] Interconnection of Motion Control Kernel and motor control function

In the interconnection previously shown, the **Motion Control Kernel** monitors every interface. Some of the signals such as a quick stop request or a DC-injection braking request are directly passed through to the motor control. However, other signals are passed through or modified depending on the operating mode (e.g. synchronising a setpoint selection via ramp function).

## 9.2 Internal interfaces | System block "LS_MotionControlKernel"

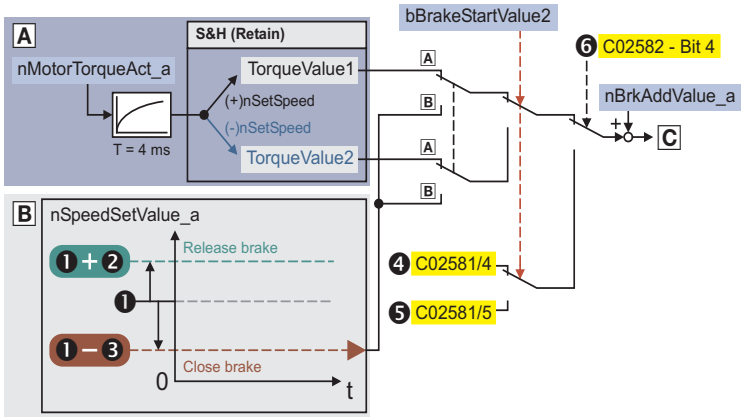
In the Function Block editor, the system block **LS_MotionControlKernel** provides the interfaces to the **Motion Control Kernel**.

### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| **Control and setpoint signals for motor control**<br>The purpose of the following inputs is to transfer control signals and setpoints to the internal motor control function (LS_MotorInterface). | | |
| bPosCtrlOn<br>BOOL | Activation of position/angle control | |
| | FALSE | Position/angle control deactivated. |
| | TRUE | Position/angle control activated. |
| bDeltaPosOn<br>BOOL | Activate position difference as setpoint selection<br>• In order to position the motor shaft, the position control function can work within the motor control function with the absolute position setpoint *dnPosSetValue_p* or alternatively with the speed setpoint *nSpeedSetValue_a* and the position difference *dnDeltaPos_p*. | |
| | FALSE | Positioning with position setpoint *dnPosSetValue_p*. |
| | TRUE | Positioning with speed setpoint *nSpeedSetValue_a* and position difference *dnDeltaPos_p*.<br>**Note:** In this case, no position encoder should be set to ensure a correct function (C00490 = "No encoder: nSpeedSetValue_a"). |
| dnDeltaPos_p<br>DINT | Position difference (following error input)<br>• Difference between setpoint position and actual position in [increments]<br>• Is used for position control if *bDeltaPosOn* = TRUE.<br>• Scaling: A revolution is displayed with 65536 increments or steps. | |
| dnPosSetValue_p<br>DINT | Absolute position setpoint in [increments]<br>• Is used for position control if *bDeltaPosOn* = FALSE.<br>• Scaling: A revolution is displayed with 65536 increments or steps. | |
| bPosDerivativeOn<br>BOOL | Create a setpoint for the speed controller from the position setpoint<br>• For highly dynamic control systems, the setpoint for the speed controller can be created from the absolute position setpoint *dnPosSetValue_p* instead of the speed setpoint *nSpeedSetValue_a* ▸ Position control/additive speed specification | |
| | TRUE | Create a speed setpoint from the position setpoint.<br>• The absolute position setpoint *dnPosSetValue_p* is differentiated and a speed value is created which is the setpoint for the speed controller.<br>• Internal limitation of 65536 increments/ms. |
| bMotorRefOffsetOn<br>BOOL | In preparation - Input has no function at present time!<br>Adopt home position ("Homing on the fly") | |
| | FALSE↗TRUE | Adopt home position *dnMotorRefOffset_p*. |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnMotorRefOffset_p<br>DINT | Home position in [increments]<br>• Value is adopted when control bit 9 ("HomeSetPos") is set from "0" to "1" in the MCK control word and is sent to the motor control function via the *dnMotorRefOffset_p_* output.<br>• Transfer is possible in any operating mode ("Homing on the fly").<br>• For this function, the two outputs *bMotorRefOffsetOn_* and *dnMotorRefOffset_p_* must be connected to the inputs of the same name of the LS_MotorInterface system block. | |
| bQspOn<br>BOOL | Trigger quick stop (QSP) via the MCK<br>• Also see device command "Activate/deactivate quick stop". | |
| | TRUE | Activate quick stop<br>• Motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0).<br>• A pulse inhibit is set if the auto-DCB function has been activated via C00019.<br>• The motor is kept at a standstill during closed-loop operation (function in preparation). |
| | FALSE | Deactivate quick stop<br>• The quick stop is deactivated if no other source for the quick stop is active.<br>• C00159 displays a bit code of active sources/causes for the quick stop. |
| nPWMAngleOffset<br>INT | Angular offset input<br>• Scaling: 16384 ≡ 100 %<br>• Setting range: 0 ... 199.99 % | |
| bSpeedCtrlIOn<br>BOOL | Directly set the I-component of speed controller<br>• In order to statically specify a minimum torque, e.g. when a load is being lifted. | |
| | TRUE | Se the I-component of the speed controller to the value *nSpeedCtrlI_a*. |
| nSpeedCtrlI_a<br>INT | I-component of the speed controller<br>• Value is adopted in the case of a FALSE-TRUE edge at the input *bSpeedCtrlIOn*. | |
| nSpeedSetValue_a<br>INT | Rotation speed/velocity setpoint | |
| bTorquemodeOn<br>BOOL | TRUE | Switch on torque-controlled operation |
| nTorqueSetValue_a<br>INT | Torque setpoint | |
| bDcBrakeOn<br>BOOL | Manual DC-injection braking (DCB)<br>• For this function, the *bDcBrakeOn_* output signal must be connected to the *bDcBrakeOn* input of the same name of the LS_MotorInterface system block.<br>• Detailed information on DC-injection braking is provided in the motor control chapter, subchapter "DC-injection braking". | |
| | **Note!**<br>Holding braking is not possible when this braking mode is used!<br>Use the basic "Holding brake control" function for controlling the holding brake with a low rate of wear. | |
| | FALSE | Deactivate DC-injection braking. |
| | TRUE | Activate DC-injection braking, i.e. the drive is brought to a standstill by means of DC-injection braking.<br>• The braking effect stops when the rotor is at standstill.<br>• After the hold time (C00107) has expired, the controller sets the pulse inhibit. |

| Designator<br>Data type | Information/possible settings |
|---|---|
| bTorqueLimitAdaptOn<br>BOOL | Adaptation of torque limitation<br>TRUE \| Activate adaptation of torque limitation. |
| nTorqueLimitAdapt_a<br>INT | Value for adaptation of torque limitation<br>• Scaling: 16384 ≡ 100 % of the torque limit specified at the LS_MotorInterface system block via the *nTorqueMotLimit_a* and *nTorqueGenLimit_a* inputs. |
| **Control words** | |
| MCK: wMotionCtrl1<br>wMotionCtrl2<br>WORD | MCK control word 1 & 2<br>• The two control words together form a 32-bit double control word with which the entire Motion Control Kernel is controlled.<br>• All motion profiles in the different operating modes can be operated via this interface.<br>• See the "MCK control word" subchapter for a detailed description of the individual control bits.<br>• Display parameter: C01240 |
| wAuxCtrl<br>WORD | For future extensions - Input has no function at present time!<br>Additional control word |
| wSMCtrl<br>WORD | Interface to the optional safety system.<br>• Setting control bit 0 ("SafeStop1") in this control word causes e.g. the automatic deceleration of the drive to standstill within this application (in the **Motion Control Kernel**).<br>• See the "Interface to safety system" subchapter for a detailed description of the individual control bits. |
| **Control and setpoint signals for Motion Control Kernel function** | |
| dnProfilePosition_p<br>DINT | Profile position in [increments]<br>• Position which is to be entered into a profile data set selected via the MCK control word.<br>• In the "absolute" mode, this position is a target position. However, in the "relative" mode it is a relative traverse path.<br>▸ Consideration of residual value in case of external profile calculation |
| nSpeedAddValue_v<br>INT | Additive speed setpoint in [inc/ms] |
| nSpeedOverride_a<br>INT | Value for Speed override<br>• Percentage multiplier (0 ... 199.99 %) for the currently active speed.<br>• 16384 ≡ 100 % of the maximum traversing speed (display in C01211/1).<br>• If the override value is 0 %, the drive is brought to a standstill. |
| nAccOverride_a<br>INT | Value for Acceleration override<br>• Percentage multiplier (0 ... 199.99 %) for the currently active acceleration.<br>• 16384 ≡ 100 % of the parameterised acceleration of the corresponding operating mode.<br>• If the override value is 0 %, acceleration ceases. |
| nSRampOverride_a<br>INT | Value for S-ramp smoothing override<br>• Percentage multiplier (0 ... 100 %) for the currently active acceleration.<br>• 16384 ≡ 100 % of the parameterised S-ramp time (C01306/1...15).<br>• Values > 16384 are ignored. |
| | **Note!**<br>If the *nSRampOverride_a* input remains unconnected or if an override value of "0 %" is selected, activation of the S-ramp override causes deactivation of the S-ramp time.<br>• Deactivation of the S-ramp time before the start of a profile with S-ramp time causes linear ramp generation.<br>• Deactivation of the S-ramp time during a traversing process, however, is not accepted immediately in the profile generator, but the profile generator checks automatically when an online change of the ramp form can be carried out and then initiates it automatically. |

| Designator<br>Data type | Information/possible settings |
|---|---|
| bLimitSwitchPos<br>BOOL | Input for Hardware limit switches (positive) |
| bLimitSwitchNeg<br>BOOL | Input for Hardware limit switches (negative) |
| bHomingMark<br>BOOL | Input for pre-stop mark/pre-stop signal for homing<br>• Relevant for homing modes "4" ... "7". |
| bMBrakeRelease<br>BOOL | Holding brake control:<br>Releasing/applying the brake in connection with the selected operating mode |
| | FALSE: Apply brake.<br>• During automatic operation, the internal brake logic controls the brake. |
| | TRUE: Release brake manually (forced release).<br>• **Note!**<br>The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If a controller inhibit has been set by the brake control, it will be deactivated.<br>• In semi-automatic operation, the brake is released including feedforward control. |
| bMBrakeStartValue2<br>BOOL | Holding brake control:<br>Selection of the torque feedforward control value<br>▸ Feedforward control of the motor before release |
| | FALSE: Starting value 1 is active (see the following figure). |
| | TRUE: Starting value 2 is active (see the following figure). |
| | Creation of the feedforward control value for the release process of the brake:<br>A Signal path for motor control with feedback<br>B Signal path for motor control without feedback<br>C Feedforward control value<br>  <br><br>❶ C02581/1: Switching threshold<br>❷ C02581/2: Hysteresis for release<br>❸ C02581/3: Hysteresis for application<br>❹ C02581/4: Starting value 1 for manual feedforward control<br>❺ C02581/5: Starting value 2 for manual feedforward control<br>❻ C02582/Bit 4: Selection of the feedforward control value (automatic/manual selection) |
| nMBrakeAddValue_a<br>INT | Holding brake control:<br>Additive feedforward control value (speed or torque) in [%] for torque feedforward control when the respective control mode is started<br>• For speed control: 100 % ≡ reference speed (C00011)<br>• For torque control: 100 % ≡ maximum torque (C00057)<br>▸ Feedforward control of the motor before release |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bMBrakeApplied<br>BOOL | Holding brake control:<br>Input for status detection via switching contacts at the brake<br>• Only effective if bit 5 in C02582 is set to "1". | |
| | FALSE | Brake is released. |
| | TRUE | Brake is applied. |
| **Process data interface for entering profile data** | | |
| dnProcessIn1_p<br>..<br>dnProcessIn4_p<br>DINT | Profile position as process data in [increments] | |
| nProcessIn1_a<br>..<br>nProcessIn8_a<br>DINT | Speed, acceleration, deceleration, S-ramp time, final speed as process data in [%]<br>• The 16 bit value has an adaptive effect on the corresponding parameters | |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| **Control and setpoint signals for motor control**<br>The following outputs are used to transfer control signals and setpoints to the internal motor control function (LS_MotorInterface). | | |
| bPosCtrlOn_<br>BOOL | TRUE | Activate position control. |
| bDeltaPosOn_<br>BOOL | TRUE | Control to adjust for following errors. |
| dnDeltaPos_p_<br>DINT | Following error input | |
| dnPosSetValue_p_<br>DINT | Absolute position setpoint | |
| bPosDerivativeOn_<br>BOOL | TRUE | Activate precontrol function of speed controller. |
| bMotorRefOffsetOn_<br>BOOL | FALSE↗TRUE | Accept home position. |
| dnMotorRefOffset_p_<br>DINT | Home position | |
| bQspOn_<br>BOOL | TRUE | Activate quick stop |
| nPWMAngleOffset_a_<br>INT | Angular offset input | |
| bSpeedCtrlIOn_<br>BOOL | TRUE | Set I-component of speed controller. |
| nSpeedCtrlI_a_<br>INT | I-component of the speed controller | |
| nSpeedSetValue_a_<br>INT | Main setpoint of speed | |
| bTorqueModeOn_<br>BOOL | TRUE | Switch on torque-guided operation. |
| nTorqueSetValue_a_<br>INT | Torque setpoint | |
| bDcBrakeOn_<br>BOOL | TRUE | Activate DC-injection braking. |
| bTorqueLimitAdaptOn_<br>BOOL | TRUE | Activate adaptation of torque limitation. |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| nTorqueLimitAdapt_a_<br>INT | Value for adaptation of torque limitation | |
| **Status words** | | |
| wMotionState1<br>wMotionState2<br>WORD | MCK status word 1 & 2<br>• For a detailed description of the individual status bits, see subchapter entitled "MCK status word.<br>• Display parameter: C01241 | |
| wAuxState<br>WORD | For future extensions - Output has no function at present time!<br>Additional status word | |
| **Status signal and actual-value signals from Motion Control Kernel functions** | | |
| nSpeedSet_v<br>INT | Speed setpoint selection in [increments/ms]<br>• 16384 ≡ 15000 rpm | |
| dnPosTarget_p<br>DINT | Target position in [increments]<br>• 65535 ≡ 1 revolution of the motor shaft | |
| dnPosSet_p<br>DINT | Modulo position in [increments]<br>• 65535 ≡ 1 revolution of the motor shaft<br>▸ Activation of the modulo measuring system | |
| dnPosSetRelative_p<br>DINT<br>(from version 02.00.00) | Relative feed in positioning processes in [increments]<br>• 65535 ≡ 1 revolution of the motor shaft | |
| wActProfileNo<br>WORD | Number of current profile | |
| wFollowProfileNo<br>WORD | Number of sequence profile | |
| bReadyToOperate<br>BOOL<br>(from version 16.00.00) | TRUE | Readiness to process setpoint signals and setpoint commands is active |
| bBusy<br>BOOL | TRUE | Setpoint profile generation is active. |
| bDone<br>BOOL | TRUE | Setpoint profile generation is completed<br>(set position = target position). |
| bHomingDone<br>BOOL | TRUE | Homing has been carried out.<br>• The *bHomingDone* output, in contract to the *bHomePosAvailable* output, remains set to TRUE even if a travel command that resulted in a reset of the home position has been carried out. |
| bHomePosAvailable<br>BOOL | TRUE | Home position is known.<br>▸ Status bit "HomPosAvailable" |
| bMBrakeReleaseOut<br>BOOL | Trigger signal for switching element holding brake control via a digital output<br>• Use bit 0 in C02582 to activate inverted switching element triggering.<br>▸ Holding brake control | |
| | FALSE | Apply brake. |
| | TRUE | Release brake. |
| bMBrakeReleased<br>BOOL | "Brake released" status signal considering the brake release time<br>• When the holding brake is triggered to close, *bMBrakeReleased* is immediately set to FALSE even if the brake closing time has not yet elapsed!<br>▸ Holding brake control | |
| | TRUE | Brake released (after the brake release time has expired). |
| bFollowErrLim1<br>BOOL<br>(from version 12.00.00) | TRUE | The current following error has exceeded the limit 1 for the following error set in C01215/1 via the time set in C01244/2.<br>▸ Following error monitoring system |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bFollowErrLim2<br>BOOL<br>(from version 12.00.00) | TRUE | The current following error has exceeded the limit 2 for the following error set in C01244/3 via the time set in C01215/2.<br>▸ Following error monitoring system |
| wGearNum<br>wGearDenom<br>WORD<br>(from version 02.00.00) | Output of the gearbox factor set in C01202/1 and C01202/2<br>• These outputs can be connected to the *GearNum-/GearDenom* inputs of the FBs which process a gearbox factor (FB L_PhiIntegrator_1, FB L_DFSET_1, FB L_CalcDiameter_1).<br>• C01067/1...3 can be used to inform these FBs that the gearbox direction of rotation is inverted is mostly done by selecting "motor mounting position inverted"). | |

### 9.2.1 MCK control word

The motion control function implemented in the LS_MotionControlKernel system block can be controlled via the *wMotionCtrl1* and *wMotionCtrl2* control words. Together, both control words form a 32-bit control double word which serves to control the entire MotionControlKernel. All motion profiles in the various operating modes can be operated via this interface.

For direct control via a fieldbus system, the two control words can be triggered by the field bus via a port block. In addition to a few other signals (e.g. limit switch, speed override) that are connected to the digital inputs of the inverter, all control signals can therefore be activated/deactivated via the fieldbus used (CAN, PROFIBUS, etc.).

As an alternative to this, changes to the individual control bits or bit fields (e.g. for specifying the profile number) can also be carried out via separate process inputs provided by the L_MckCtrlInterface function block. The control words *wOutMckPosCtrl_1* and *wOutMckPosCtrl_2* output by this function block constitute the input information for the LS_MotionControlKernel system block in this case.

#### MCK control word 1 (wMotionCtrl1)

| Bit | Designation | Description | | | | |
|---|---|---|---|---|---|---|
| 0<br>...<br>3 | OpMode_Bit0<br>...<br>OpMode_Bit3 | **Operating mode** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Speed follower | 0 | 0 | 0 | 0 |
| | | Homing | 0 | 0 | 0 | 1 |
| | | Manual jog | 0 | 0 | 1 | 0 |
| | | Positioning | 0 | 0 | 1 | 1 |
| | | Stop | 0 | 1 | 0 | 0 |
| | | Position follower | 0 | 1 | 0 | 1 |
| | | All other possible settings are reserved for future extensions! | | | | |
| 4<br>5 | ManJogPos<br>ManJogNeg | **Manual jog** | | | **Bit 5** | **Bit 4** |
| | | Stop manual jog | | | 0 | 0 |
| | | Manual jog CW | | | 0 | 1 |
| | | Manual jog in counter-clockwise direction | | | 1 | 0 |
| | | No change from previous status | | | 1 | 1 |
| 6 | ManExecute<br>2ndSpeed | "1" ≡ Change over to manual speed 2 | | | | |
| 7 | ReleaseLimitSwitch | "1" ≡ Retract operated hardware limit switch | | | | |
| 8 | HomStartStop | "1" ≡ Start/stop homing process | | | | |
| 9 | HomSetPos | "1" ≡ Set homing position | | | | |
| 10 | HomResetPos | "1" ≡ Reset the "Reference known" status<br>• *bHomePosDone* and *bHomePosAvailable* are reset to FALSE.<br>• The positions remain unaffected. | | | | |
| 11 | EnableSpeedOverride | "1" ≡ Activate speed override | | | | |
| 12 | EnableAccOverride | "1" ≡ Activate acceleration override | | | | |
| 13 | Enable<br>SRampOverride | "1" ≡ Activate S-ramp override | | | | |
| 14 | PosTeachSetPos | "1" ≡ Teach MCK set position into the selected profile | | | | |
| 15 | PosTeachActPos | "1" ≡ Teach current position into the selected profile | | | | |

### MCK control word 2 (wMotionCtrl2)

| Bit | Designation | Description | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 16 | PosExecute | "0↗1" ≡ Start travelling | | | | | | | |
| 17 | PosFinishTarget | "0↗1" ≡ Complete cancelled profile | | | | | | | |
| 18 | PosDisable FollowProfile | "1" ≡ Do not travel sequence profile | | | | | | | |
| 19 | PosStop | "1" ≡ Cancel travelling<br>From version 02.00.00, more travel requests will be inhibited ("PosExecute" will be blocked). | | | | | | | |
| 20 ... 23 | PosModeBit0 ... PosModeBit3 | **Positioning mode** | | | | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Positioning mode = setting in C01300/1...15 | | | | 0 | 0 | 0 | 0 |
| | | Absolute (shortest path) | | | | 0 | 0 | 0 | 1 |
| | | Continuous | | | | 0 | 0 | 1 | 0 |
| | | Relative | | | | 0 | 0 | 1 | 1 |
| | | absolute (Cw) | | | | 0 | 1 | 0 | 0 |
| | | absolute (Ccw) | | | | 0 | 1 | 0 | 1 |
| | | Absolute (shortest path) to TP | | | | 1 | 0 | 0 | 0 |
| | | Continuous to TP | | | | 1 | 0 | 0 | 1 |
| | | Relative to TP | | | | 1 | 0 | 1 | 0 |
| | | Absolute (Cw) on TP | | | | 1 | 0 | 1 | 1 |
| | | Absolute (Ccw) on TP | | | | 1 | 1 | 0 | 0 |
| | | All other possible settings are reserved for future extensions! | | | | | | | |
| 24 ... 31 | ProfileNo_Bit0 ... ProfileNo_Bit7 | **Profile** | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Profile 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Profile 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | Profile 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | | ... | ... | | | | | | | |
| | | Profile 15 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | | All other possible settings are reserved for future extensions! | | | | | | | | |

**Note!**

The profile 0 is no valid profile for the "Positioning" operating mode.

If a driving request with an invalid profile number is started, the response set in C00595/12 occurs (Lenze setting: "WarningLocked").

**Tip!**

Travel requests/profiles can also be started while the drive is running. The drive does not need to be at standstill.

## 9.2.2 MCK status word

### MCK status word 1 (wMotionState1)

| Bit | Designation | Description | | | | |
|---|---|---|---|---|---|---|
| 0<br>...<br>3 | ActOpModeBit00<br>...<br>ActOpModeBit03 | **Active operating mode** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Speed follower | 0 | 0 | 0 | 0 |
| | | Homing | 0 | 0 | 0 | 1 |
| | | Manual jog | 0 | 0 | 1 | 0 |
| | | Positioning | 0 | 0 | 1 | 1 |
| | | Stop or Safe stop 1 (SS1) | 0 | 1 | 0 | 0 |
| | | Position follower | 0 | 1 | 0 | 1 |
| | | StandBy (internal operating mode in the event of quick stop, pulse inhibit and DC-injection braking) | 1 | 1 | 1 | 1 |
| 4 | Busy | "1" ≡ Internal profile generation is active. A speed profile is being generated. | | | | |
| 5 | Done | "1" ≡ Generation of a speed profile with the selected position has been completed. | | | | |
| 6 | AcceleratingActive | "1" ≡ Profile generation phase is in the acceleration process. | | | | |
| 7 | ConstSpeedDuty | "1" ≡ Profile generation phase at constant speed active. | | | | |
| 8 | DeceleratingActive | "1" ≡ Profile generation phase is in the deceleration process. | | | | |
| 9 | S_ShapingActive | From version 12.00.00:<br>"1" ≡ Rounding during acceleration/deceleration active.<br>▸ Status bit "S_ShapingActive" | | | | |
| 10 | Pos. HW-Limit Detected | "1" ≡ Positive limit switch has triggered.<br>• Reset only possible via"Manual jog" mode!<br>▸ Hardware limit switches | | | | |
| 11 | Neg. HW-Limit Detected | "1" ≡ Negative limit switch has triggered.<br>• Reset only possible via"Manual jog" mode!<br>▸ Hardware limit switches | | | | |
| 12 | HomPosDone | "1" ≡ Homing has been completed. | | | | |
| 13 | HomPosAvailable | "1" ≡ The home position has been detected and is known in the drive.<br>▸ Status bit "HomPosAvailable" | | | | |
| 14 | Pos. SW limit detected | "1" ≡ Positive software limit position overtravelled.<br>▸ Software limit positions | | | | |
| 15 | Neg. SW limit detected | "1" ≡ Negative software limit position overtravelled.<br>▸ Software limit positions | | | | |

**Note!**

The internal "StandBy" operating mode is assumed if controller inhibit, pulse inhibit, quick stop and/or DC-injection braking are activated.

- In this operating mode, no setpoint generation takes place through the **Motion Control Kernel**.
- If the holding brake control sets a controller inhibit when the holding brake is closed, the internal "StandBy" operating mode is not assumed.

### MCK status word 2 (wMotionState2)

| Bit | Designation | Description |
|---|---|---|
| 16 | DwellTime | "1" ≡ Dwell time after reaching the setpoint position is active.<br>▸ Target position monitoring (status "drive in target") |
| 17 | InTarget | "1" ≡ Dwell time has expired and current actual position is in the set target window.<br>▸ Target position monitoring (status "drive in target") |
| 18 | PosDone | "1" ≡ Positioning profile has been completed in the "Positioning" or "Homing" mode. Setpoint position of a profile data set is in target. |
| 19 | ReadyToOperate | "1" ≡ Readiness to process setpoint signals and setpoint commands.<br>▸ Status bit "bReadyToOperate" |
| 20 ... 23 | ActPosMode_Bit00 ... ActPosMode_Bit03 | |

| Active positioning mode | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|
| Absolute (shortest path) | 0 | 0 | 0 | 1 |
| Continuous | 0 | 0 | 1 | 0 |
| Relative | 0 | 0 | 1 | 1 |
| absolute (Cw) | 0 | 1 | 0 | 0 |
| absolute (Ccw) | 0 | 1 | 0 | 1 |
| Absolute (shortest path) to TP | 1 | 0 | 0 | 0 |
| Continuous to TP | 1 | 0 | 0 | 1 |
| Relative to TP | 1 | 0 | 1 | 0 |
| Absolute (Cw) on TP | 1 | 0 | 1 | 1 |
| Absolute (Ccw) on TP | 1 | 1 | 0 | 0 |

**Note:** The display of the active positioning mode depends on further factors. See subchapter "Override of the parameterised positioning mode".

| 24 ... 31 | ActProfileNo_Bit00 ... ActProfileNo_Bit07 | |
|---|---|---|

| Active profile | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Profile 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Profile 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Profile 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| ... | ... | | | | | | | |
| Profile 15 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

## Note!

The effective positioning mode is generated via the MCK control word (bit 20 ... 23) and/or the parameters C01300/1...15 for defining the positioning mode in the profile data. Here, the setting in the MCK control word superimposes the mode selection via the corresponding parameter. This means that the mode selection and the touch probe enable are possible via process data.

For generating the effective positioning mode, the following applies:

- Valid PosMode in the MCK control word:
  → Use positioning mode of the MCK control word
- PosMode in the MCK control word = 0:
  → Use positioning mode set in C01300/x
- Invalid PosMode in the MCK control word:
  → Error message "Ck09: Positioning mode invalid"

#### 9.2.2.1 Status bit "HomPosAvailable"

Bit 13 ("HomPosAvailable") in the MCK status word 1 shows that the reference position has been detected and is known in the drive.

An available reference information (bit 13 = "1") and thus a valid measuring system is required for the following functions:

- Positioning in the "Positioning" operating mode in the following (C01300/1...15) positioning modes:
  - 1: absolute (beeline)
  - 4: absolute (Cw)
  - 5: absolute (Ccw)
  - 8: absolute (shortest path) to TP
  - 11: absolute (Cw) to TP
  - 12: absolute (Ccw) to TP
- Response to set software limit positions
- Stopping on breakpoints in the "Manual jog" operating mode
- Traversing of a sequence profile with absolute measuring reference in the "Homing" mode after homing is completed

**Resetting the reference information**

The following events reset bit 13 ("HomPosAvailable") in the MCK status word 1:

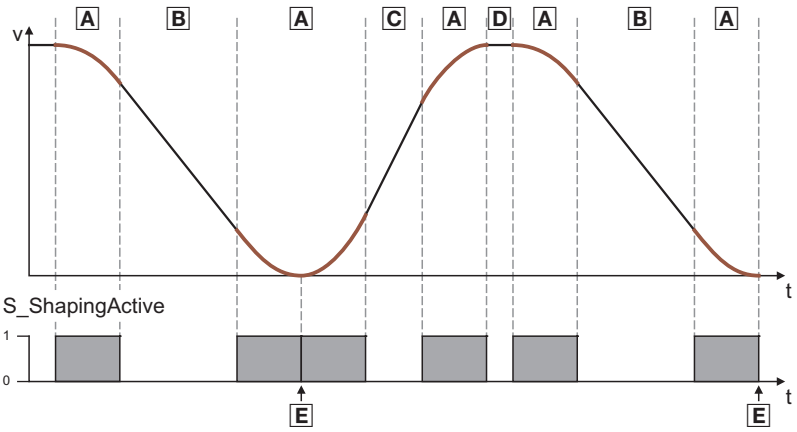
- Setting bit 10 ("HomResetPos") in the MCK control word 1
- Overtravelling the 32-bit position display area at the limit ±2147483647 increments when the cycle (C1201/1) = "0 units".
- Change of the position encoder selection in C00490
- Change of the cycle in C1201/1
- Writing on the machine parameters if bit 1 ("delete reference when machine data is changed") is set in C2652.
  - Machine parameters are cycle (C1201/1), gearbox factors (C1202/1...2, C1203/1...2) and feed constant (C1204).
  - The mere writing of these parameters (even when the previous value is used) causes the reference to be deleted!
- Encoder error when a position encoder is used at the multi-encoder interface
- Resolver error when a resolver is used as position encoder
- Mains recovery at a cycle set unlike "0 units" in C1201/1 and if the position difference at reference recovery is higher than 1000 cycles.

### 9.2.2.2 Status bit "S_ShapingActive"

This function extension is available from version 12.00.00!

Bit 9 ("S_ShapingActive") in the MCK status word 1 indicates in the operating modes "Homing", "Manual jog", "Positioning", "Stop" and "Position follower" when rounding is effective while S-shaped profiles are being carried out.

- Bit 9 is usually reset in case of profile restarts and after profiles have been completed.
- The following illustration shows this by means of a real temporal characteristic:




A Rounding active
B Constant deceleration
C Constant acceleration
D Constant speed
E Profile completed (bit 9 is reset in case of profile restart and profile completion)

[9-2] Example: Display of the rounding generated via the "S_ShapeActive" status bit

### 9.2.2.3 Status bit "bReadyToOperate"

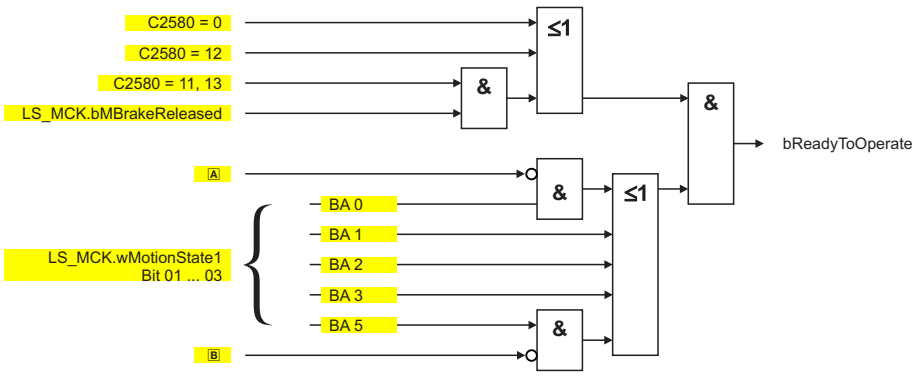
This function extension is available from version 16.00.00!

The **LS_MotionControlKernel** with status bit 19 in the status word 2 (wMotionState2) provides the signalling for the readiness to process the setpoint signals or setpoint commands. The state of bit *bReadyToOperate* provides TRUE if, for instance, a defined speed setpoint or a command (e.g. "manual jog positive") can be directly processed by the **LS_MotionControlKernel**.

The status of this bit depends on three factors:

- Operating mode of the MotionControlKernel
  - *bReadyToOperate* = FALSE if operating mode **Standby** or **Stop**
- Synchronisation status
  - *bReadyToOperate* = FALSE if **LS_MotionControlKernel** executes a synchronisation process in the operating modes **Speed follower** or **Position follower**.

The figure shows the logical relationship:




A Speed follower: Synchronisation active
B Position follower: Synchronisation active
BA 0 ... 5: Operating mode 0 ... 5

[9-3] Creation of the status bit *bReadyToOperate*

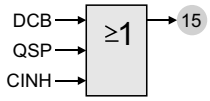
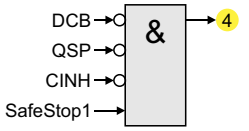
### 9.2.3 MCK diagnostic word

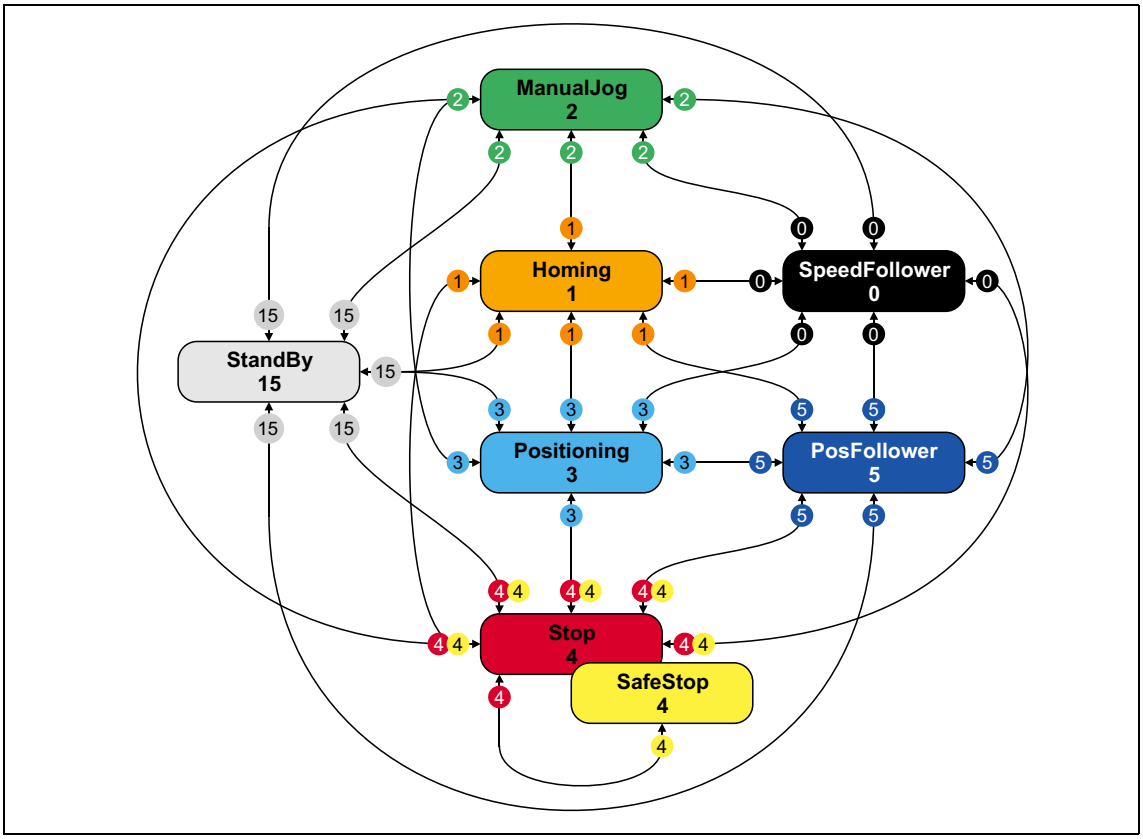
The MCK diagnostic word provides information derived by parameter settings (e.g. effective limitations etc.).

**MCK: Diagnostic word**

| Bit | Designation | Description |
|---|---|---|
| Bit 0 | Traversing range: Limited | 1 ≡ The traversing range is limited:<br>• The software limit positions are set.<br>• The reference is known.<br>• There is no modulo measuring system available (C01201/1 = 0).<br>**Note:**<br>This status does not consider whether the software limit positions are activated or deactivated by the settings in the prevailing operating mode. |
| Bit 1 | Traversing range: Modulo | 1 ≡ A modulo measuring system is available as a cycle length has been set in C01201/1. |
| Bit 2 | Pos. HW limit switch active | Evaluation of travel range limit switch active in positive direction |
| Bit 3 | Neg. HW limit switch active | Evaluation of travel range limit switch active in negative direction |
| Bit 4 | Pos. SW-Limit active | Evaluation of software limit positions active in positive direction |
| Bit 5 | Neg. SW-Limit active | Evaluation of software limit positions active in negative direction |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | SpeedNormConstant : Low-Limit | 1 ≡ Due to the set machine parameters, an internal calculation constant is limited to a possible minimum.<br>Remedy: Increase reference speed in C00011. |
| Bit 9 | SpeedNormConstant : Up-Limit | 1 ≡ Due to the set machine parameters, an internal calculation constant is limited to a possible maximum.<br>Remedy: Reduce reference speed in C00011. |
| Bit 10 | Max. speed limit | 1 ≡ Due to the set machine parameters, the maximum speed for specifications in [units/s] are limited to the factor 150 %. |
| Bit 11 | Quick stop is active | 1 ≡ Quick stop function is active. |
| Bit 12 | Quick stop standstill is active | 1 ≡ Standstill during active quick stop function (no ramp generation). |
| Bit 13 | Max. speed is pending | 1 ≡ The speed setpoint has reached or exceeded the limit of ± 199.99 %. |
| Bit 14 | Position resolution limit | 1 ≡ The position resolution calculated by machine data entries has been limited to the maximum display area of 214748.3647 [Incr./Unit] with reference to the display in C01205 . The real value is higher! |
| Bit 15 .... Bit 31 | Reserved | |

### 9.2.4 MCK state machine

| Prio | Condition | | | | | |
|---|---|---|---|---|---|---|
| **Global conditions:** | | | | | | |
| 1 | 15 | Setpoint generation through Motor control (MCTRL):<br>• DCB = DC-injection braking<br>• QSP = quick stop<br>• CINH = controller inhibit |  | | | |
| 2 | 4 | "Safe stop 1" (SS1) requested<br>▸ Interface to safety system |  | | | |
| **Conditions requested by MCK control word:** | | | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| 3 | 2 | Manual jog requested | 0 | 0 | 1 | 0 |
| 4 | 1 | Homing requested | 0 | 0 | 0 | 1 |
| 5 | 3 | Positioning requested | 0 | 0 | 1 | 1 |
| 6 | 0 | Speed follower requested | 0 | 0 | 0 | 0 |
| 7 | 5 | Position follower requested | 0 | 1 | 0 | 1 |
| 8 | 4 | Stop requested | 0 | 1 | 0 | 0 |




[9-4] MCK state machine

### 9.2.4.1 "StandBy" operating mode

The internal "StandBy" operating mode is implicitly assumed if controller inhibit, pulse inhibit, quick stop and/or DC-injection braking are activated. Thus, there is no setpoint generation by the **Motion Control Kernel** in the "StandBy" operating mode.

- If the holding brake control sets a controller inhibit when the holding brake is closed, the internal "StandBy" operating mode is not assumed.
- The "StandBy" operating mode cannot be activated via the MCK control word.
- When the "StandBy" operating mode is active, bits 0 ... 3 are set in the MCK status word.

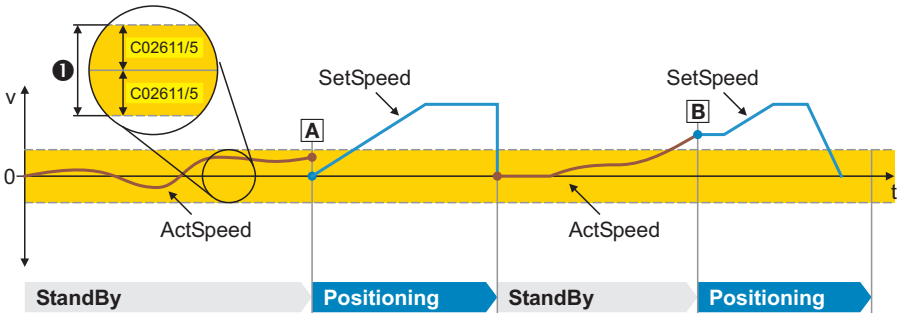
#### Speed acceptance when the "StandBy" operating mode is quit

In addition to cancelling the controller inhibit, pulse inhibit, quick stop and/or DC injection braking, a change to another operating mode from "StandBy" also requires a termination of the magnetisation of the motor (not in case of synchronous motors). If there is a change to a setpoint-controlled operating mode of the **Motion Control Kernel** the current speed is transferred to the speed setpoint.

**Note!**

Due to a motor magnetisation of a motor normally standing still, the current speed can also be non-zero rpm when "StandBy" is quit. If the operating mode now changes to "Positioning", the transfer of this speed to the speed setpoint may cause a permanent drifting of the motor shaft, depending on the activated option in C01216.

From version V12.00.00 onwards, a sensitivity threshold can be set for preventing a drifting of the motor shaft in C2611/5. If the absolute value of the current speed is below the sensitivity threshold, the value "0" is transferred to the speed setpoint instead of the current speed. The following illustration shows the issue:




A Current speed below sensitivity threshold → speed setpoint = 0
B Current speed above sensitivity threshold → speed setpoint = current speed
❶ C02611/5: Sensitivity threshold (in [%] with regard to reference speed)

[9-5] Example: Sensitivity threshold for speed transfer from the "StandBy" operating mode

In the Lenze setting "0.5 %", the sensitivity threshold corresponds to approx. 7.5 rpm at a reference speed of 1500 rpm set in C00011.

### 9.2.5 Interface to safety system

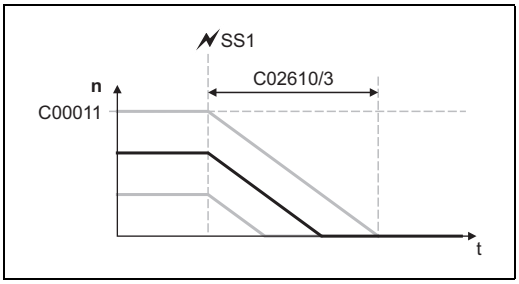
For operation with optional safety system, the LS_MotionControlKernel system block has the *wSMCtrl* input, This interface is used to transfer a control word by means of which the **Motion Control Kernel** can be supplied with information on requested or active safety functions. The **Motion Control Kernel** then initiates the necessary motion sequence (e.g. braking).

At the moment, only bit 0 in the *wSMCtrl* control word has a function. Additional functions are in preparation:

| Bit | Designation | Description |
|---|---|---|
| 0 | SafeStop1 | "1" ≡ Request for "Safe Stop 1" (SS1). |
| 1 ... 15 | Reserved | In preparation - Still without function! |

**Behaviour in case of request for "Safe Stop 1" (SS1)**

The drive is brought to a standstill with the stopping ramp set in C02610/3.




[9-6] Ramping down to standstill

- The time set in C02610/3 refers to the down-ramping of the reference speed set in C00011.
- If the current speed is lower, the time to standstill is accordingly lower as well.

If the request is reset during the down-ramping process (bit 0 = "1↘0"), the behaviour depends on the active operating mode:

- In the "Speed follower" operating mode, direct synchronisation with the target speed takes place with the ramp time set in C02610/2.
- In the "Homing" operating mode, the ramp-down is continued with the deceleration for stop set in C01251 if the bit 8 ("HomStartStop") in the MCK control word = "0". If bit 8 is set, the homing process starts immediately in the selected homing mode.
- In the "Manual jog" operating mode, the behaviour depends on whether the manual jog initiator (*bManJogPos* or *bManJogNeg*) is still set:

| Bit 0 (SafeStop1) | bManJogPos bManJogNeg | Behaviour |
|---|---|---|
| 1 | TRUE | Ramping down to standstill |
| 0 | TRUE | Accelerating to manual speed |
| 0 | FALSE | Ramping down to standstill |

- In the "Positioning" operating mode, the behaviour depends on the setting in C01216.
- In the "Position follower" operating mode, (forward) positioning to the signalled absolute position always takes place if this position differs from the internal position.

### 9.2.6 Consideration of residual value in case of external profile calculation

The *dnProfilePosition_p* input serves to transfer a profile position in [increments] to the SB LS_MotionControlKernel. This will then be entered into the active profile data set selected in the MCK control word via the bits 24 ... 31.

For (external) calculations to be carried out before, the following blocks can be used in the FB Editor.

- From version 12.00.00 onwards, these blocks automatically consider residual values and modulo positions in case of a cycle length definition and correctly forward these internally:

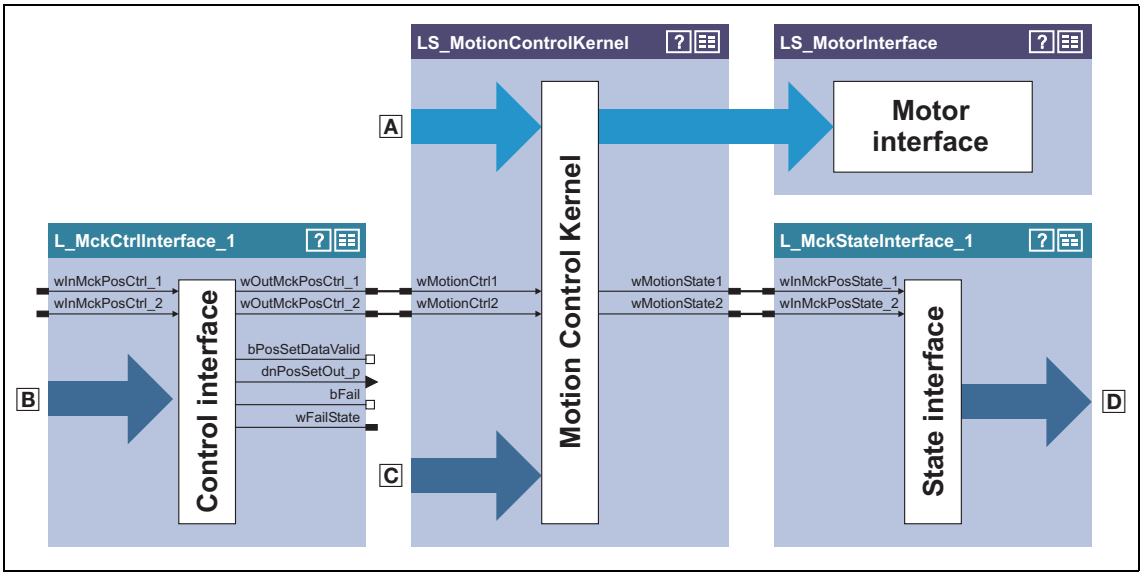
| Block | Function |
|---|---|
| L_ConvUnitsToIncr_1<br>L_ConvUnitsToIncr_2<br>L_ConvUnitsToIncr_3 | ... converts a position value provided in the real unit of the machine into an internal 32-bit position value.<br>• These FBs are available from version 12.00.00. |
| L_MckCtrlInterface_1 | ... provides the application with process inputs for controlling various basic functions of the Motion Control Kernel. |
| LS_ParFreeUnit<br>LS_ParFreeUnit_2 | Output of 16 parameterisable position signals with internal conversion of [unit] in [increments]<br>• These SBs are available from version 02.00.00. |

The following blocks also automatically consider residual values and module positions and can be used for forwarding the increment results of the previously mentioned blocks:

| Block | Function |
|---|---|
| L_Mux_1 | ... switches one of eight selectable input signals to the output. |
| L_SignalSwitch32_1<br>L_SignalSwitch32_2<br>L_SignalSwitch32_3 | ... .switches between two input signals of "DINT" data type.<br>• These FBs are available from version 02.00.00. |
| LA_SwitchPos / LA_TabPos:<br>• dnFreeIn1_p<br>• dnFreeIn2_p | These two inputs serve to transfer 32-bit signals from the I/O level to the application level. The signals are available at the correspondent outputs of the application block LA_SwitchPosIn or LA_TabPosIn. |
| LA_TabPos:<br>• dnPosProfilePosition | Selection of the target position in [increments] for the "table positioning" application. The signal is available in the application level at the correspondent output of the LA_TabPosIn application block. |

## 9.3 MCKInterface

The so-called "MCK interface" described in this chapter consists of the two function blocks L_MckCtrlInterface and L_MckStateInterface, which are connected upstream respectively downstream of the LS_MotionControlKernel system block:




[9-7] Detail of the interconnection architecture for the "table positioning" technology application

A Control and setpoint signals for motor control

B Additional process inputs for controlling the **Motion Control Kernel**, e.g.:

- Selection of the operating mode
- Selection of the profile number
- Override of the positioning mode
- Control inputs for manual jogging, homing, positioning
- Control inputs for speed/acceleration/S-ramp override

C Control and setpoint signals for the **Motion Control Kernel** such as

- Selection of speed setpoint for speed follower
- Selection of positioning setpoint for position follower
- Selection of override values
- Connection for limit switch & pre-stop mark for homing
- Holding brake control

D Output of status signals of the **Motion Control Kernel**

### Control of the Motion Control Kernel

Control of the basic drive functions implemented in the Motion Control Kernel is carried out by means of

- direct specification of the control words, for example via a master control unit also connected to the fieldbus.
  - For this purpose, the control word inputs can be directly connected to the field bus interface **LP_McIIn** respectively **LP_CanIn**.
  - See the "MCK control word" chapter for a detailed description of the individual control bits.
- the specification of individual process signals at the L_MckCtrlInterface FB which are then ORed with the control word.

### Plausibility check

The process signals applied to the L_MckCtrlInterface FB are logically linked by means of an OR logic operation to the specified control word via the two control word inputs *wInMckPosCtrl_1* and *wInMckPosCtrl_2* and, after a plausibility check, are output via the two control word outputs *wOutMckPosCtrl_1* and *wOutMckPosCtrl_2* .

- The profile number, the operating mode, and the positioning mode are checked for plausibility.
- If implausibility is detected, only control bit information is output and the *bFail* output is set to TRUE.
- The result of the plausibility check is provided as the *wFailState* output word and displayed in C01299.

The control words that are output, namely *wOutMckPosCtrl_1* and *wOutMckPosCtrl_2*, constitute the input information for the LS_MotionControlKernel system block.

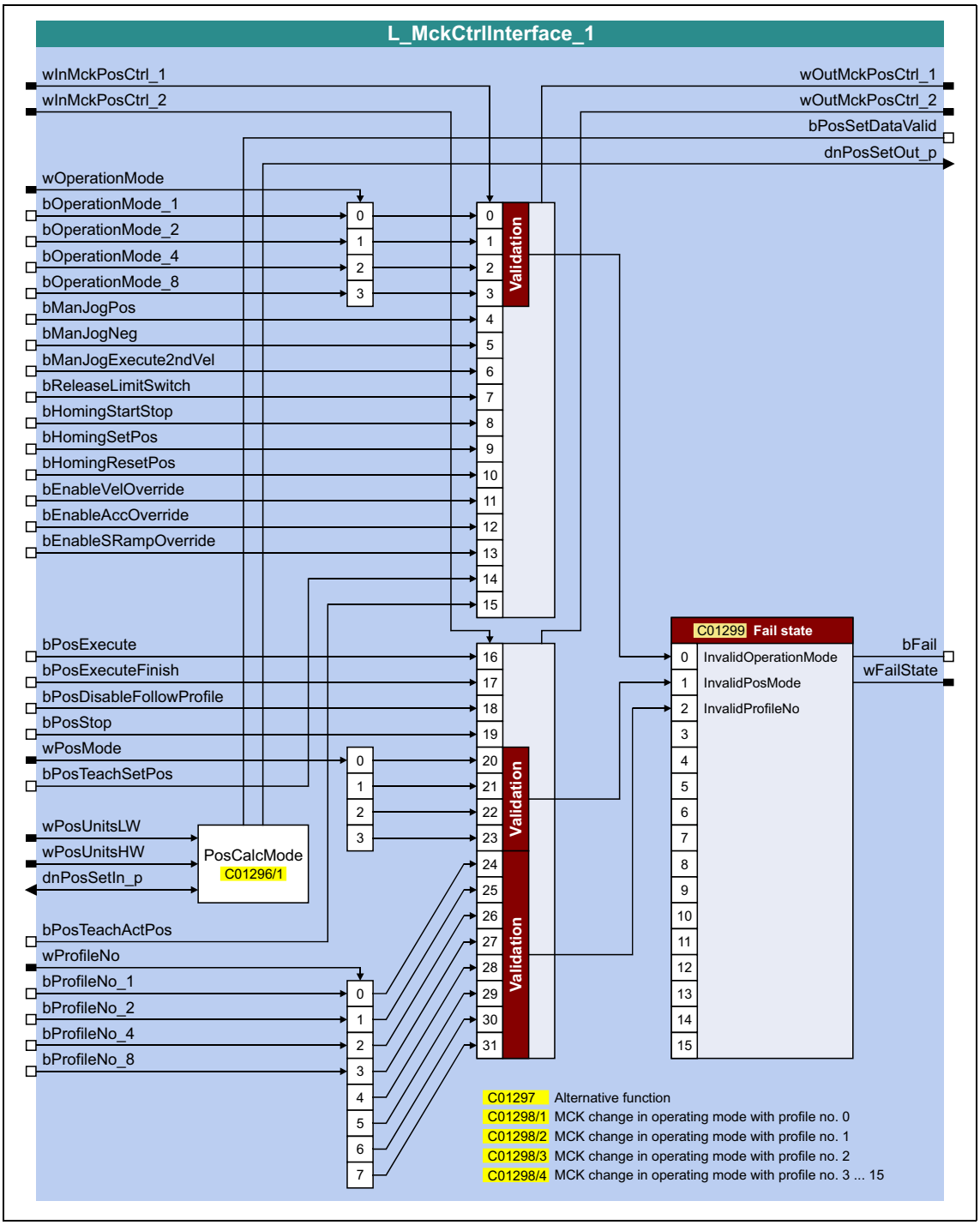
### Status information of the Motion Control Kernel

Output by the LS_MotionControlKernel, the status words *wMotionState1* and *wMotionState2* also constitute input information for the downstream L_MckStateInterface FB, which provides this information to the application in the form of process signals.

### 9.3.1 Control inputs | "L_MckCtrlInterface" function block

This FB provides process inputs for controlling different basic functions of the **Motion Control Kernel**.

In addition to the ORing of discrete input signals with the control word inputs, the FB has further functions that are described in the following subchapters.



### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wInMckPosCtrl_1<br>wInMckPosCtrl_2<br>WORD | Direct specification of the MCK control word 1 & 2<br>• For example, via a master control unit that is also connected to the fieldbus. For this purpose, the control word inputs can be directly connected to the **LP_McIIn** respectively **LP_CanIn1** fieldbus interface.<br>• The two control words together form a 32-bit double control word with which the entire Motion Control Kernel is controlled.<br>• All motion profiles in the different operating modes can be operated via this interface.<br>• See the "MCK control word" chapter for a detailed description of the individual control bits. | |
| wOperationMode<br>WORD | Selection of the operating mode of the **Motion Control Kernel**<br>• Only bit 0 ... bit 3 of *wOperationMode* are evaluated.<br>• If an invalid operating mode is selected, the response set in C00595/11 is activated (Lenze setting: "Warning").<br>• The current operating mode is displayed in C01243. | |
| | 0 | Speed follower |
| | 1 | Homing |
| | 2 | Manual jog |
| | 3 | Positioning |
| | 4 | Stop |
| | 5 | Position follower |
| | 6 ... 15 | Reserved for future extensions |
| bOperationMode_1<br>...<br>bOperationMode_8<br>BOOL | Binary-coded selection of the operating mode of the **Motion Control Kernel**<br>• See the "MCK control word" chapter for a detailed description of the individual control bits.<br>• If an invalid operating mode is selected, the response set in C00595/11 is activated (Lenze setting: "Warning").<br>• The current operating mode is displayed in C01243. | |
| bManJogPos<br>bManJogNeg<br>BOOL | Manual jog:<br>*bManJogPos* = TRUE: Manual jog right<br>*bManJogNeg* = TRUE: Manual jog left<br>Both inputs = TRUE: No change compared to previous state<br>Both inputs = FALSE: Stop manual jog | |
| bManJogExecute2ndVel<br>BOOL | Manual jog: Changeover to speed 2 | |
| | FALSE | Speed 1 (C01231/1) active |
| | TRUE | Speed 2 (C01231/2) active |
| bReleaseLimitSwitch<br>BOOL | Manual jog: Retract operated limit switch | |
| | TRUE | Retract operated limit switch (in opposite direction) |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bHomingStartStop<br>BOOL | Homing: Start/stop homing<br>• Only possible in the "referencing" operating mode. | |
| | TRUE | If one of homing modes "4" ... "15" in C01221 is selected:<br>**Start reference search**<br>• The current status of the reference search is indicated via the status outputs *bHomingDone* and *bHomePosAvailable*. |
| | | If homing mode "100: SetRef" is selected:<br>**Setting the home position manually**<br>• The home position is set manually with the drive at a standstill. The current actual position now corresponds to the reference position set in C01227/2 in the machine measuring system. |
| | TRUE↘FALSE | Stop homing.<br>• If the *bHomingStartStop* input is reset to FALSE during active referencing, homing is cancelled and the drive is brought to a standstill. |
| bHomingSetPos<br>BOOL | Homing: Set home position ("on the fly" homing)<br>• With referencing "on the fly", the home position of a machine can be set during ongoing movement. Jerking and compensating movements do not occur. | |
| | FALSE↗TRUE | The position at the *dnMotorRefOffset_p* input of the LS_MotionControlKernel SB at the instant of activation is the set home position. |
| bHomingResetPos<br>BOOL | Homing: Delete home position<br>**Note:** With this function, positions are not deleted but only the status signals *bHomePosAvailable* and *bHomingDone* are reset. Setpoints and actual positions remain untouched until a renewed reference setting or homing. | |
| | FALSE↗TRUE | The internal status "reference known" is reset.<br>• The inverter is no longer referenced.<br>• The process outputs *bHomePosAvailable* and *bHomingDone* are reset to FALSE. |
| bEnableVelOverride<br>BOOL | Speed override | |
| | TRUE | Activate speed override |
| bEnableAccOverride<br>BOOL | Acceleration override | |
| | TRUE | Activate acceleration override |
| bEnableSRampOverride<br>BOOL | S-ramp smoothing override | |
| | **Note!**<br>If the *nSRampOverride_a* input on the LS MotionControlKernel remains unconnected or if "0 %" is specified as the override value, activation of the S-ramp override results in deactivation of the S-ramp time.<br>• Deactivation of the S-ramp time before the start of a profile with S-ramp time causes linear ramp generation.<br>• Deactivation of the S-ramp time during a traversing process, however, is not accepted immediately in the profile generator, but the profile generator checks automatically when an online change of the ramp form can be carried out and then initiates it automatically. | |
| | TRUE | Activate S-ramp smoothing override |
| bPosExecute<br>BOOL | Positioning: Start travelling | |
| | FALSE↗TRUE | Execute selected profile |
| bPosExecuteFinish<br>BOOL | Positioning: Complete cancelled profile | |
| | FALSE↗TRUE | A positioning process previously cancelled, e.g. by *bPosStop* or due to a device error, is resumed by travelling to the original target. |
| bPosDisableFollowProfile<br>BOOL | Positioning: Do not execute sequence profile (switch-off profile linkage) | |
| | TRUE | Evaluation of the sequence profile number parameterised in C01307/1...15 for the selected profile is suppressed. |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bPosStop<br>BOOL | Positioning: Cancel travelling | |
| | TRUE | Stop positioning<br>From version 02.00.00, more travel requests will be inhibited ("PosExecute" will be blocked). |
| wPosMode<br>WORD | Override of the positioning mode set in the profile data<br>• Via this input, an override of the positioning mode parameterised in C01300/1...15 for the selected profile is possible.<br>• The value set in C01300/1...15 is not overwritten in this case.<br>• Only bit 0 ... bit 3 of *wPosMode* are evaluated.<br>• If *wPosMode* = 0 is selected, the positioning mode set in C01300/1...15 is used. | |
| | 0 | Positioning mode = setting in C01300/1...15 |
| | 1 | Absolute (shortest path) |
| | 2 | Continuous |
| | 3 | Relative |
| | 4 | absolute (Cw) |
| | 5 | absolute (Ccw) |
| | 8 | Absolute (shortest path) to TP |
| | 9 | Continuous to TP |
| | 10 | Relative to TP |
| | 11 | Absolute (Cw) on TP |
| | 12 | Absolute (Ccw) on TP |
| | All other possible settings are reserved for future extensions! | |
| bPosTeachSetPos<br>BOOL | Position teaching: MCK setpoint position | |
| | FALSE↗TRUE | Teach MCK setpoint position into the selected profile. |
| wPosUnitsLW<br>wPosUnitsHW<br>WORD | Selection of the target position in [units]<br>• *wPosUnitsLW* = LOW word, *wPosUnitsHW* = HIGH word<br>• The mode for calculating the position is selected in C01296/1. | |
| dnPosSetIn_p<br>DINT | Selection of the target position in [increments]<br>• The mode for calculating the position is selected in C01296/1. | |
| bPosTeachActPos<br>BOOL | Position teaching: Current position | |
| | FALSE↗TRUE | Teach current position into the selected profile. |
| wProfileNo<br>WORD | Stipulation of the profile to be executed<br>• Optionally as a data word or binary coded.<br>• When the profile is stipulated, this FB carries out a mode change in the Lenze setting at the same time:<br>  • If profile 0 is selected: Activation of "Speed follower" operating mode<br>  • If profile 1 is selected: Activation of "Homing" operating mode<br>  • If profile 2 is selected: Activation of "Manual jog" operating mode<br>  • If profile 3 ... 15 is selected: Activation of "Positioning" operating mode | |
| bProfileNo_1<br>...<br>bProfileNo_8<br>BOOL | | |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| wOutMckPosCtrl_1<br>wOutMckPosCtrl_2<br>WORD | Output of the MCK control word 1 & 2<br>• For transfer to the LS_MotionControlKernel system block.<br>• For a detailed description of the individual control bits, see chapter "MCK control word". (📖 569) | |
| bPosSetDataValid<br>BOOL | Status signal "Position conversion completed, data consistent"<br>• This output is permanently set to TRUE if the g"0: dnPosOut_p=dnPosIn_p" mode has been set for the position calculation in C01296/1 and no modulo measuring system exists.<br>• If a cycle length has been set in C01201/1, a calculation is made under the following conditions without an immediate setting of *bPosSetDataValid*:<br>- Position specification is higher than or equal to the cycle length.<br>- Position specification is negative.<br>For a correct positioning process, the *bPosSetDataValid* = TRUE status is required. | |
| | TRUE | Conversion of the target position from [units] into [increments] has been completed.<br>• The travel profile data are valid and the profile is ready to start. |
| dnPosSetOut_p<br>DINT | Output of the target position in [increments]<br>• Observe the *bPosSetDataValid* status output! | |
| wFailState<br>WORD | Result of the plausibility check<br>• Display parameter: C01299<br>• Result is bit coded: | |
| | Bit 0 | Invalid operating mode selection<br>• "1" ≡ The selected operating mode is not defined/invalid. |
| | Bit 1 | Invalid positioning mode selection<br>• "1" ≡ The selected positioning mode is not defined/invalid. |
| | Bit 2 | Invalid profile number selection<br>• "1" ≡ The selected profile number refers to a profile data set that does not exist. |
| | Bit 3 | Reserved |
| | ... | |
| | Bit 15 | |
| bFail<br>BOOL | FALSE | Okay, no error |
| | TRUE | • Plausibility check error or<br>• control information error (in this case after ORing the individual signals with the control words) |

#### 9.3.1.1 Alternative functions for control bit "PosExecute"

In C01297, alternative functions for bit 16 (PosExecute) in the MCK control word can be selected with bit-coding.

**PosStop with PosExecute = FALSE**

If bit 0 has been set in C01297, positioning can only be started/cancelled with the "PosExecute" control bit.

- If the "Positioning" operating mode has been set on the MCK, the "Pos-Execute" control bit has the following effects when being activated:

| Signals at the input | Signals in the control word to the MCK |
|---|---|
| bPosExecute = TRUE | bPosExecute = TRUE<br>bPosStop = FALSE<br>bHomingStartStop remains unchanged |
| bPosExecute = FALSE | bPosExecute = FALSE<br>bPosStop = TRUE<br>bHomingStartStop remains unchanged |

**HomingStartStop with PosExecute**

If bit 1 has been set in C01297, homing can only be started/stopped with the "PosExecute" control bit.

- If the "Homing" operating mode has been set on the MCK, the "PosExecute" control bit has the following effect when activated:

| Signals at the input | Signals in the control word to the MCK |
|---|---|
| bPosExecute = TRUE | bPosExecute = TRUE<br>bPosStop = FALSE<br>bHomingStartStop = TRUE |
| bPosExecute = FALSE | bPosExecute = FALSE<br>bPosStop remains unchanged<br>bHomingStartStop = FALSE |

**SetProfilPosition with PosExecute**

If bit 2 has been set in C01297 and control bit ("PosExecute") is being set, the setpoint position being applied is incorporated in the currently selected profile and then the profile is started immediately.

- If the "Positioning" operating mode has been set on the MCK, the "PosExecute" control bit has the following effect when activated:

| Signals at the input | Signals in the control word to the MCK |
|---|---|
| bPosExecute = TRUE | bPosExecute = TRUE<br>bPosStop = FALSE<br>bHomingStartStop remains unchanged<br>bPosTeachSetPos = TRUE (edge) |
| bPosExecute = FALSE | bPosExecute = FALSE<br>bPosStop remains unchanged<br>bHomingStartStop remains unchanged<br>bPosTeachSetPos = FALSE |

**SetProfilPosition at position change**

If bit 3 is set in C01297, the setpoint positions at the MCKInterface are automatically accepted into the profile with the applied profile number if a change of data is detected at the corresponding input for the setpoint position.

- If the "0: dnPosOut_p=dnPosIn_p" mode has been set in C01296/1 for converting the position, automatic acceptance is executed if the incremental position selection at the *dnPosIn_p* input was changed.
- If another mode (>0) has been set in C01296/1, automatic acceptance is executed if the incremental position selection at the *wPosUnitsLW* and *wPosUnitsHW* inputs was changed.
- From version 02.00.00 a hysteresis can be set in C01245/3 for the position change.

**PosExecute at position change**

(from version 02.00.00)

If bit 4 is set in C01297, an automatic "PosExecute" takes place if the incremental position selection changes and this change is higher than set in the hysteresis for position change (C01245/3).

If the selection is made in units, so that in a first step it has to be converted into increments, and if this automatic function is then activated, the "PosExecute" will only be created automatically when the internal conversion is completed (*bPosSetDataValid*).

**Tip!**

This option serves to start travel requests very easily by simply defining the new target position.

### 9.3.1.2 Operating mode change with profile number

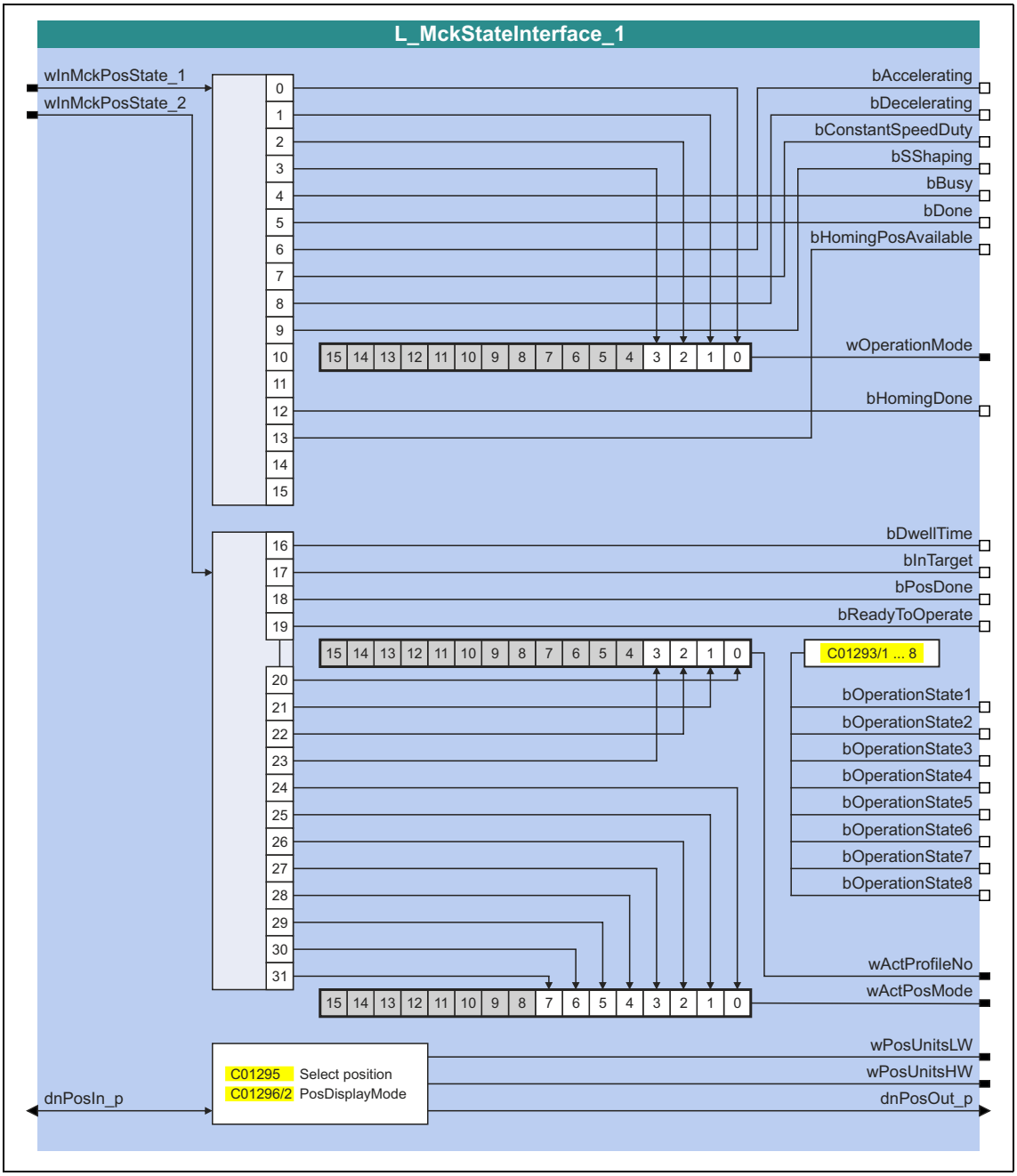
One of the operating modes of the **LS_MotionControlKernel** system block can be assigned to the positioning profiles via the four subcodes of C01298:

| Parameters | Info | Lenze setting |
|---|---|---|
| C01298/1 | Operating mode if profile 0 is selected | Speed follower |
| C01298/2 | Operating mode if profile 1 is selected | Homing |
| C01298/3 | Operating mode if profile 2 is selected | Manual jog |
| C01298/4 | Operating mode if profile 3 ... 15 is selected | Positioning |

- When a changeover to the corresponding profile is carried out, the set operating module is requested at the same time.
- With the setting "0: Op.Mode inputs", no operating modes are changed when the profile is changed. Instead, the operating mode requested "externally" applies (optionally via the inputs *wInMckPosCtrl_1*, *wOperationMode* or *bOperationMode_1...8* of the FB L_MckCtrlInterface).
  **Note:** These inputs are internally connected via OR-logic. Only connect one of these inputs to ensure non-ambiguous operating mode changeovers!

### 9.3.2 Status outputs | FB "L_MckStateInterface"

This FB provides the application with different status information of the **Motion Control Kernel** via process outputs.




**Note!**

When C01296/2 > 0, the position selected in C01295 is calculated in [units]. In this case, the output values *wPosUnitsLW*, *wPosUnitsHW* and *dnPosOut_p* are not updated in a 1-ms cycle (not in real time). However, they are applied consistently to each other. For this reason, we recommend the use of these outputs only for diagnostic purposes where an update in real time is not important.

### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| wInMckPosState_1<br>wInMckPosState_2<br>WORD | Inputs for accepting the status words from the LS_MotionControlKernel system block. |
| dnPosIn_p<br>DINT | Position in [increments]<br>• Is converted into [units] if C01295 = "0: dnPosIn_p" and output at the *wPosUnitsLW* and *wPosUnitsHW* outputs.<br>• The mode for calculating the position is selected in C01296/2. |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bAccelerating<br>BOOL | TRUE | Acceleration phase active. |
| bDecelerating<br>BOOL | TRUE | Braking phase active. |
| bConstantSpeedDuty<br>BOOL | TRUE | Constant phase active. |
| bSShaping<br>BOOL | From version 12.00.00: | |
| | TRUE | Profile rounding active. |
| bBusy<br>BOOL | TRUE | Setpoint generation active. |
| bDone<br>BOOL | TRUE | Target position (setpoint) has been approached. |
| bHomingPosAvailable<br>BOOL | TRUE | Home position is known. |
| wOperationMode<br>WORD | Active setpoint-generating state of the **Motion Control Kernel**.<br>• Bit B0 ... B3 contain the information of the MCK status word.<br>• Bits B4 ... B15 are fixed at "0". | |
| bHomingDone<br>BOOL | TRUE | Homing has been executed. |
| bDwellTime<br>BOOL | TRUE | Settling in target position is active |
| bInTarget<br>BOOL | TRUE | Target position (actual value) is in the target window. |
| bPosDone<br>BOOL | TRUE | Target position from the profile has been approached. |
| bReadyToOperate<br>BOOL | TRUE | Readiness to process setpoint signals and setpoint commands. |
| bOperationState1<br>...<br>bOperationState8<br>BOOL | TRUE | Configurable status display from the MCK status word 1<br><br>▸ Configurable status display bOperationState |
| wActProfileNo<br>WORD | Current traversing profile number<br>• Bit B0 ... B7 contain the information of the MCK status word.<br>• Bits B8 ... B15 are fixed at "0". | |
| wActPosMode<br>WORD | Current positioning mode<br>• Bit B0 ... B3 contain the information of the MCK status word.<br>• Bits B4 ... B15 are fixed at "0". | |

| Designator<br>Data type | Value/meaning |
|---|---|
| wPosUnitsLW<br>wPosUnitsHW<br>WORD | Output of the position selected in C01295 in [units]<br>• *wPosUnitsLW* = LOW word, *wPosUnitsHW* = HIGH word<br>• The mode for calculating the position is selected in C01296/2.<br>• These values are not updated in a 1-ms cycle (not in real time)! |
| dnPosOut_p<br>DINT | Output of the position selected in C01295 in [increments]<br>• The mode for calculating the position is selected in C01296/2.<br>• When C01296/2 > 0, *dnPosOut_p* is updated isochronously with the outputs *wPosUnitsLW* and *wPosUnitsHW* and hence not in a 1-ms cycle (not in real time)! |

### 9.3.2.1 Configurable status display *bOperationState*

This function extension is available from version 16.00.00!

The status display from the MCK status word 1 can be configured with C01293/1 ... C01293/8 and applies to the following states:

| Status bit | Status |
|---|---|
| Bit00 | ActOpModeBit00 |
| Bit01 | ActOpModeBit01 |
| Bit02 | ActOpModeBit02 |
| Bit03 | ActOpModeBit03 |
| Bit04 | Busy |
| Bit05 | Done |
| Bit06 | HomPosAvailable |
| Bit07 | ReadyToOperate |

**Example**

The *bOperationState5* output is to provide the TRUE status if the drive

- is in the current **positioning** mode
- has been referenced
- and is ready for start signals.

The code C01293/5 has to be parameterised as follows:

| Status bit | Status |
|---|---|
| Bit00 | ActOpModeBit00 = TRUE |
| Bit01 | ActOpModeBit01 = TRUE |
| Bit02 | ActOpModeBit02 = FALSE |
| Bit03 | ActOpModeBit03 = FALSE |
| Bit04 | Busy = FALSE |
| Bit05 | Done = FALSE |
| Bit06 | HomPosAvailable = TRUE |
| Bit07 | ReadyToOperate = TRUE |

## 9.4 Basic settings

### 9.4.1 Machine parameters

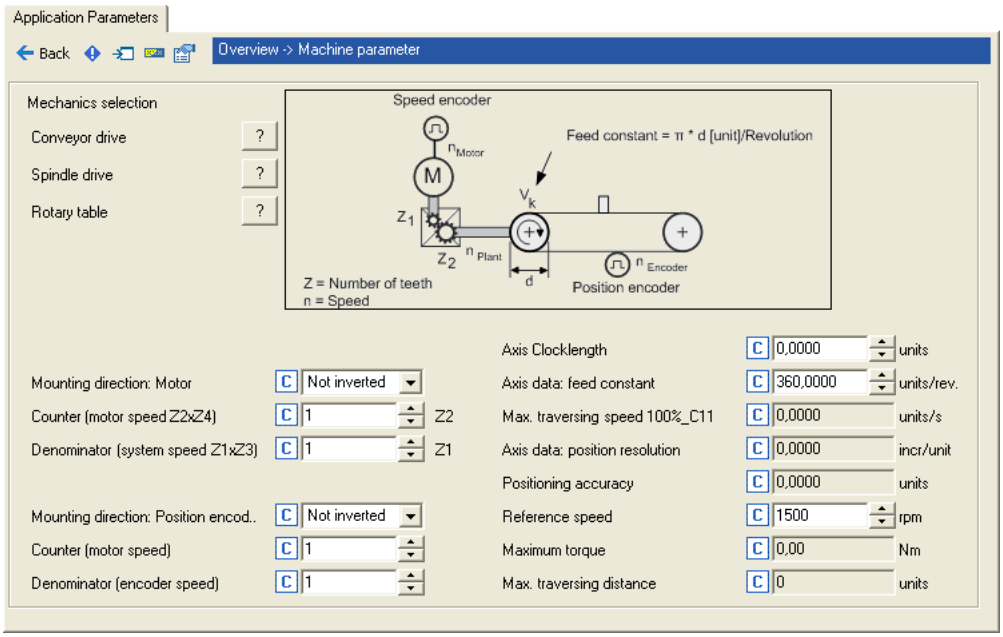
The motor end, among other things, is described by the machine parameters indicated below in respect of the mechanics used.

**Note!**

Setting the machine parameters is a basic prerequisite for the operating modes "Homing", "Manual jog" and "Positioning".

The more precisely the machine parameters are set, the better the results of positioning!

For TA "Table positioning", you can set the machine parameters in the »Engineer» on the tab headed **Application Parameters** on the dialog level *Overview* → *Machine parameters*:




**Tip!**

You are provided with more detailed information on the machine parameters "Gearbox ratio" and "Feed constant" in the following subchapters.

**Short overview of machine parameters:**

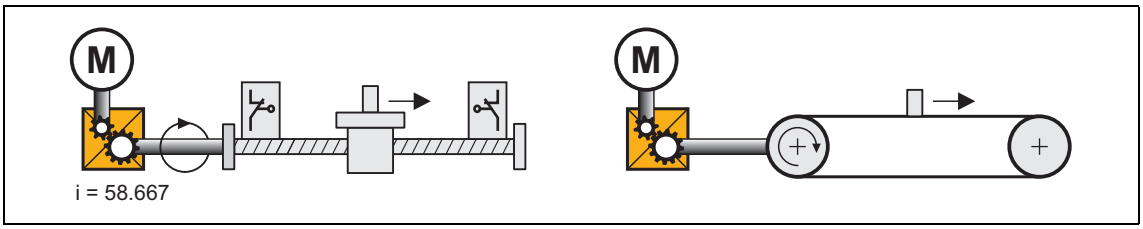
| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01206/1 | Mounting direction: Motor | 0: Not inverted | |
| C01202/1 | iM: Nominator of gearbox factor Z2 | 1 | |
| C01202/2 | iM: Denominator of gearbox factor Z1 | 1 | |
| C01206/2 | Mounting direction: Position encoder | 0: Not inverted | |
| C01203/1 | iG: Numerator (motor speed) | 1 | |
| C01203/2 | iG: Denominator (encoder speed) | 1 | |
| C01201/1 | Axis data: Axis cycle<br>▸ Activation of the modulo measuring system | 0.0000 | units |
| C01204 | Axis data: Feed constant | 360.0000 | units/rev. |
| C00011 | Appl.: Reference speed | 1500 | rpm |

After the machine parameters have been entered, the inverter sends back application-relevant data via the following display parameters:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01211/1 | Max. traversing speed 100%_C11<br>• Is used for orientation when the velocity is being set in the profile sets. | - | units/s |
| C01205 | Axis data: Position resolution<br>• Is needed for the incremental specification of positions. | - | incr/unit |
| C01210/5 | MCK: Positioning accuracy<br>• Theoretical accuracy during positioning with account being taken of the machine data and encoder data. | - | units |
| C00057 | Maximum torque | - | Nm |
| C01213/1 | MCK: Max. traversing distance | - | units |
| Greyed out = display parameter | | | |

### 9.4.1.1 Gearbox ratio

The gearbox ratio indicates the number of revolutions of the motor axis it takes for exactly one revolution of the load axis (e.g. spindle or drive roll) to take place.

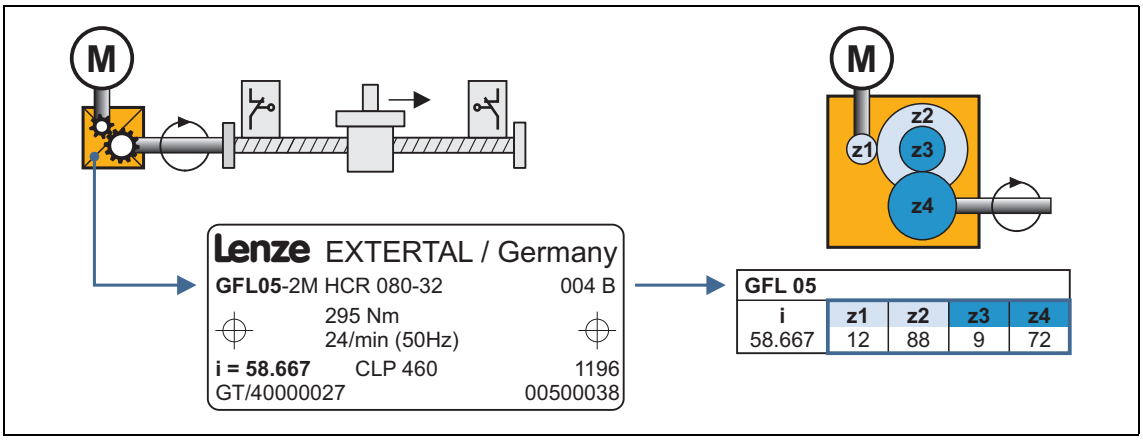



[9-8] Schematic diagram of gearbox ratio

- In the example shown in illustration [9-8] one revolution of the spindle is carried out at exactly 58,667 revolutions of the motor axis.

#### Specification of the gearbox ratio

- The gearbox ratio is to be defined in the form of a quotient (numerator/denominator); the data required can be found in the technical data for the gearbox:




[9-9] Example: Technical data relating to the gearbox (from gearbox catalogue)

**Tip!**

In order to specify the gearbox ratio exactly, use the number of teeth indicated on the data sheet or in the catalogue, if possible, instead of the information on the nameplate (see following calculation).

#### Example calculation on the basis of the technical gearbox data:

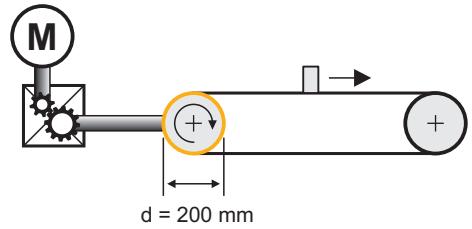
| | | | | |
|---|---|---|---|---|
| Gearbox factor numerator (C01202/1) | = | $z2 \times z4$ = $88 \times 72$ | = | 6336 |
| Gearbox factor denominator (C01202/2) | = | $z1 \times z3$ = $12 \times 9$ | = | 108 |

[9-10] Calculation example (for 2-stage gearbox)

### 9.4.1.2 Feed constant

The feed constant corresponds to the movement of the machine during one revolution of the gearbox output shaft.

- The entry in the **Feed constant** (C01204) field is made in the user unit [units] in respect of the revolution.
- In the case of a conveyor drive, the feed constant is obtained from the drive roll's circumference, which, in the following example, is calculated on the basis of the indicated diameter:

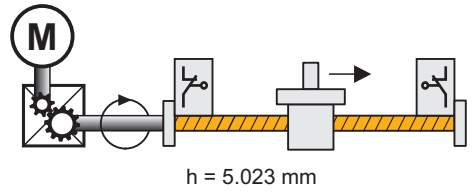



d = diameter

$$\text{Feed constant} = \pi \cdot d \frac{[\text{unit}]}{\text{Revolution}} = \pi \cdot 200 \frac{\text{mm}}{\text{Revolution}} = 628.3185 \frac{\text{mm}}{\text{Revolution}}$$

[9-11] Schematic diagram: Feed constant for a conveyor driver

- In the case of a spindle drive (linear axis), the feed constant is derived from the leadscrew pitch. The feed constant indicates the distance the slide travels during one revolution of the spindle (in the following example: 5.023 mm).



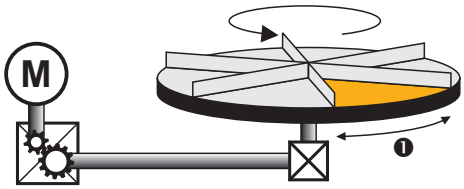

h = leadscrew pitch (can be obtained from the technical data of the linear axis)

[9-12] Schematic diagram: Feed constant for a spindle drive

- In the case of a rotary table and its specification as an angle, the feed constant is = 360°/revolution.

### 9.4.1.3 Activation of the modulo measuring system

The modulo measuring system is also called "rotary table application".




❶ C01201/1: Cycle

[9-13] Example: Rotary table application

- The measuring system is repeated.
- If the set cycle is exceeded, a defined overflow takes place.
  - In a rotary system, the cycle length typically corresponds to one rotation or one tool distance.
- For positioning, the home position must be known.
  - Exception: Positioning modes relative (TP) and continuous (TP)
- Software limit positions are not effective.
- Absolute targets can be approached by exceeding the measuring system limit, e.g. from 10° via 0° to 350°.

#### Activating the modulo measuring system

The Modulo system is activated by setting a cycle (C01201/1) > 0 units.

- The setting of the cycle is possible when the inverter is enabled.
- When the cycle (C01201/1) is set to 0 units (Lenze setting), the traversing range is unlimited (classical measuring system).

#### Creation of the modulo measuring system

When the Modulo measuring system is active, it is displayed internally via an integrator. The Modulo position is provided at the *dnPosSet_p* process output of the SB LS_MotionControlKernel and displayed in C01210/7. When the Modulo measuring system is not active, the continuous (*dnPosSetValue_p*) setpoint position is output instead.

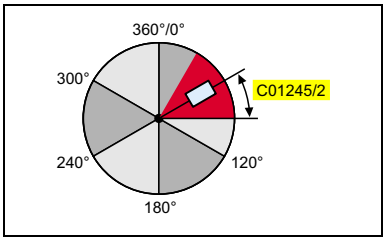
### Blocking zone for "absolute (Cw)" and "absolute (Ccw)" positioning modes

Set a blocking zone in C01245/2 for the "absolute (Cw)" and "absolute (Ccw)" positioning modes in which no target positions are approached. This blocking zone around the current setpoint position serves to consider drift motions of the shaft in case of controller inhibit/enable cycles in order to prevent e.g. an unwanted motion by one cycle.

**Stop!**

With the "absolute (Cw)" and "absolute (Ccw)" positioning modes, target positions that are located in the set blocking zone are always approached by the shortest possible path!

If the blocking zone is set too large, the drive may travel in the opposite direction!




[9-14] Definition of the blocking zone

- Target positions outside the blocking zone are approached with the selected positioning mode.
- The blocking zone is internally limited to half the cycle.

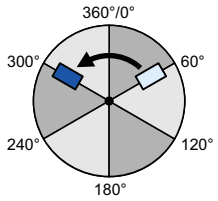
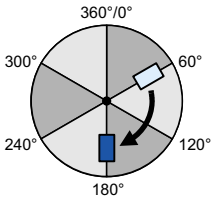
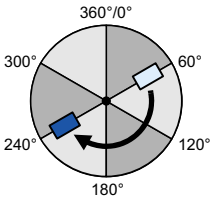
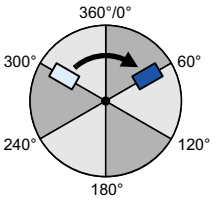
### Display of the target position in the "positioning" mode

| Positioning mode | Target position displayed in C01210/6 in [units] |
|---|---|
| Absolute (shortest path) | Defined position |
| absolute (Cw) | Defined position |
| absolute (Ccw) | Defined position |
| Continuous | 214748.3647 |
| Relative | Modulo position + defined position |

### Case studies for modulo positioning

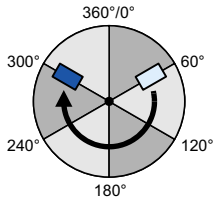
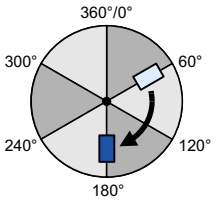
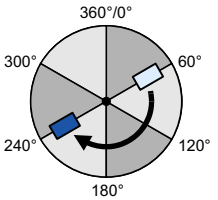
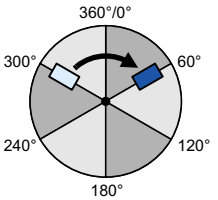
In the following, some case studies with the different positioning modes are represented. The target positions are defined by the user.

**Positioning mode "absolute (shortest path)"**

| Case 1 | Case 2 | Case 3 | Case 4 |
|---|---|---|---|
|  |  |  |  |

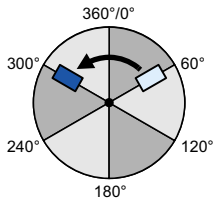
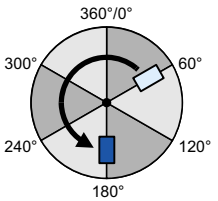
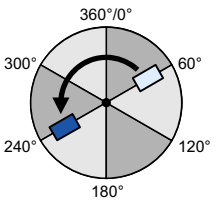
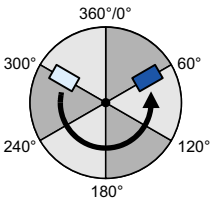


**Positioning mode "absolute (Cw)"**

| Case 1 | Case 2 | Case 3 | Case 4 |
|---|---|---|---|
|  |  |  |  |



**Positioning mode "absolute (Ccw)"**

| Case 1 | Case 2 | Case 3 | Case 4 |
|---|---|---|---|
|  |  |  |  |

### 9.4.2 Min/Max speed

For the "Speed follower" operating mode, yo can initially limit the speed setpoint by means of the following parameters:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02610/2 | MCK: Ramp time synchr. setpoint | 2.000 | s |
| C02611/1 | MCK: Pos. max. speed | 199.99 | % |
| C02611/2 | MCK: Pos. min. speed | 0.00 | % |
| C02611/3 | MCK: Neg. min. speed | 0.00 | % |
| C02611/4 | MCK: Neg. max. speed | 199.99 | % |

In the »Engineer», you can set the initial limit by means of the dialog box *Min/Max speed*.

- Open the *Min/max speed* dialog box by opening the **Application Parameters** tab and clicking on the following button on the dialog level *Overview* ➞ *Signal flow*:

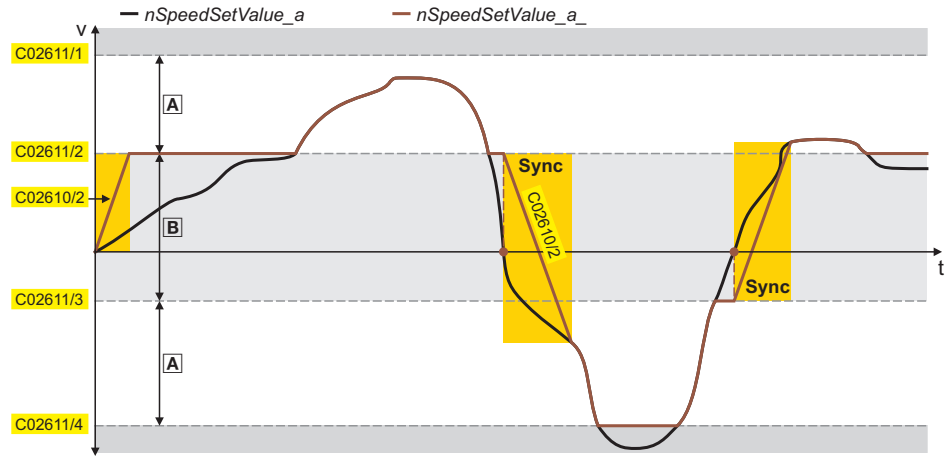


#### Speed setpoint generation

When the speed limit values are set, the **Motion Control Kernel** influences the setpoint generation with a synchronisation mode The synchronisation mode serves to travel the synchronisation range dynamically with the synchronisation ramp set in C02610/2.

For the synchronisation process it is irrelevant whether you are already in the "Speed follower" operating mode or whether you branch to this operating mode. If you branch to the "Speed follower" operating mode, the current speed setpoint is accepted and processed with regard to the input limitation parameterised. The transition to the "Speed follower" operating mode may cause a speed jump if the synchronisation period is set too short. This speed jump is avoided by extending the ramp time with C02610/2.

Synchronisation starts in the zero crossing of the selected speed in each case:




A Setpoint range
B Synchronisation range

[9-15] Example: Speed setpoint generation in the "Speed follower" operating mode (with nSpeedAddValue_v = 0)

### 9.4.3 Limit position monitoring

#### 9.4.3.1 Software limit positions

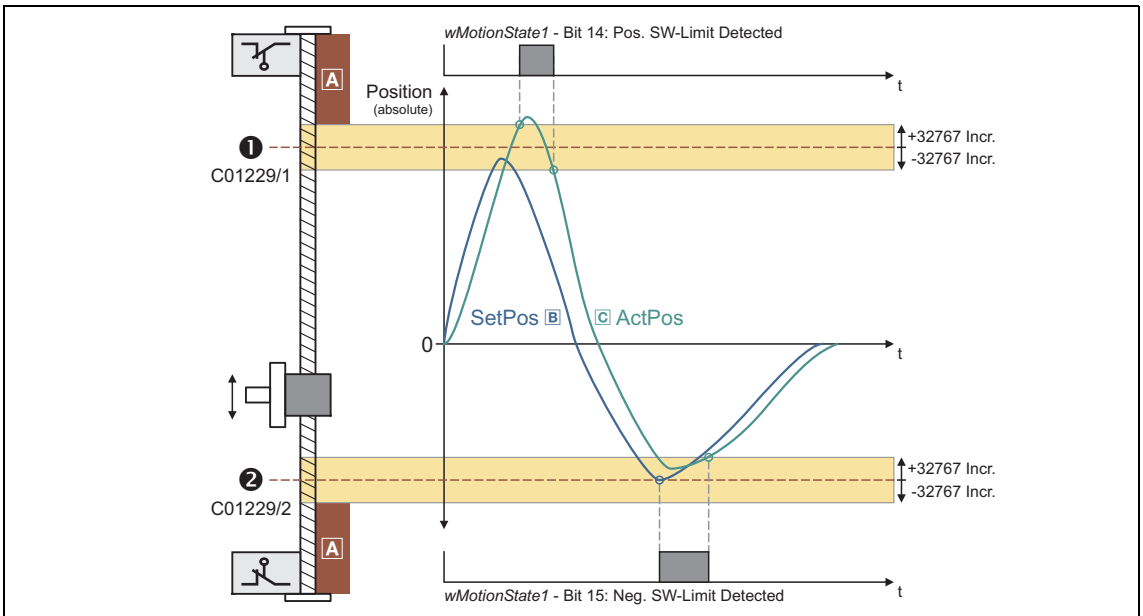
The parameterisable software limit positions serve to limit the traversing range by the software.

The positive software limit position is set in C01229/1 and the negative software limit position is set in C01229/2.

Evaluation and monitoring of the software limit positions are only carried out if

- C01229/1 - C01229/2 > 32767
- no modulo measuring system is set (C01201/1 = 0.0000 units)
- the drive knows the home position (*bHomePosAvailable* = TRUE) and the software limit positions for the respective operating mode have been activated (see the following table)!

| Operating mode | Software limit positions active (if home position is known) |
|---|---|
| Speed follower | Yes (adjustable in C01219 - bit 3) |
| Homing | Yes |
| Manual jog | No (adjustable in C01230 - bit 3) |
| Positioning | Yes |
| Position follower | Yes (adjustable in C01218 - bit 3) |




A Error range (actual position) B Set position C Actual position
❶ C01229/1: Positive software limit position
❷ C01229/2: Negative software limit position

[9-16] Definition of the software limit position trigger limits

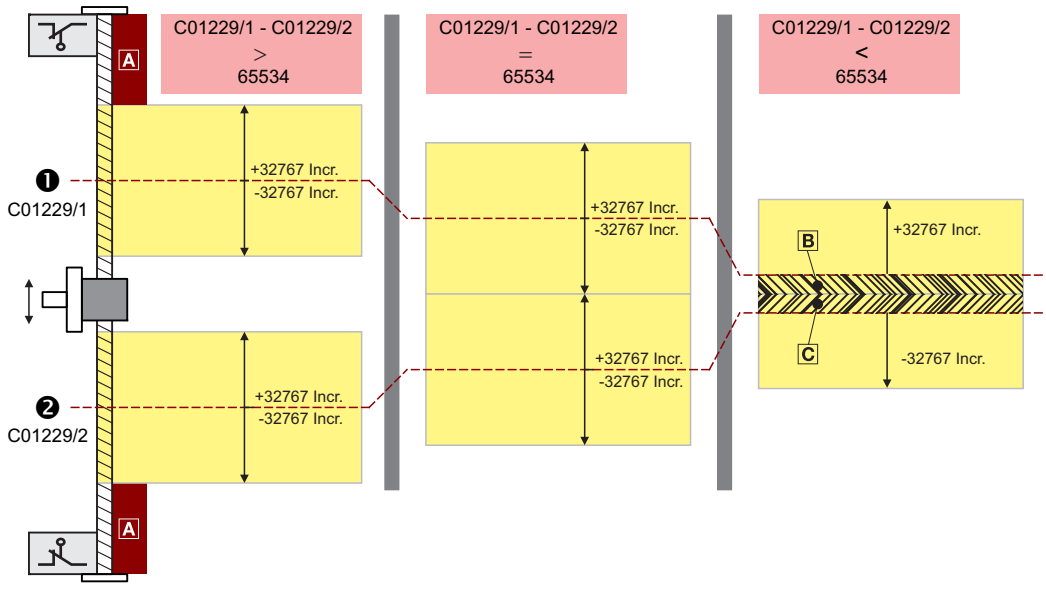
**Exception: hysteresis of both software limit positions meets or overlaps**

In the description of the previous section, the hysteresis areas of both software limit positions are separated from each other. The following applies to this standard case (also see the left-hand diagram below):

- C01229/1 - C01229/2 > 65534 incr

The following applies to applications in which the hysteresis areas meet or overlap (also see the diagram in the middle below/right-hand diagram below):

- C01229/1 - C01229/2 ≤ 65534 incr




A Error area (actual position)
B Reduced hysteresis area in positive direction
C Reduced hysteresis area in negative direction
❶ C01229/1: Positive software limit position
❷ C01229/2: Negative software limit position

Up to version 15.xx.xx, the maximum travel range will be set if hysteresis areas are overlapping (C01229/1 - C01229/2 ≤ 65534 incr.).

**Danger!**

The drive traverses without taking the software limit positions set into consideration!

The software limit positions are therefore deactivated. In order to prevent the system from being damaged, the software limit positions have to be corrected so that the following conditions apply:

1. SW_Limit_POS[incr] > SW_Limit_NEG[incr]
2. SW_Limit_POS[incr] - SW_Limit_NEG[incr] > 32767

From version 16.00.00 onwards, overlapping of the inner hysteresis areas of the software limit positions is avoided by scaling down the hysteresis width, see hatched area:

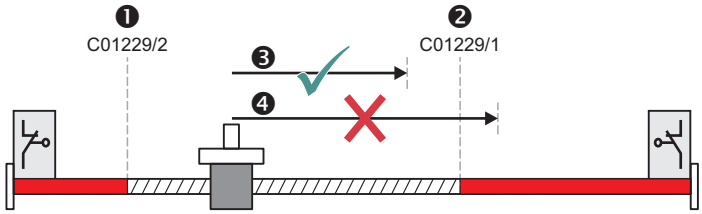
| Hysteresis width [incr] in pos. direction B | Hysteresis width in neg. direction C |
|---|---|
| $C01229/1 - \left(\frac{C01229/1 - C01229/2}{2}\right)$ | $C01229/2 + \left(\frac{C01229/1 - C01229/2}{2}\right)$ |

**Behaviour in the case of active software limit positions**

**Note!**

The"travel commands" mentioned in the following description are no speed setpoint selections. In the operating modes "Speed follower" and "Position follower", an acknowledged software limit position error ensures that traversing into the impermissible travel range remains possible afterwards. This is because in these two operating modes, there is no preview whether a software limit position is approached or not with a setpoint selection.

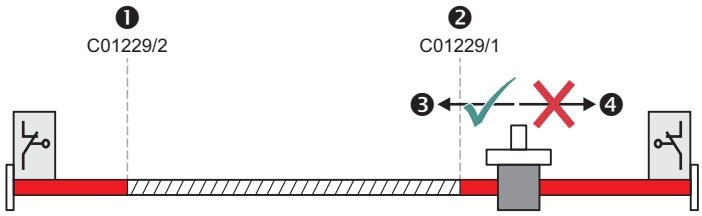
If the software limit positions are active, travelling commands that would result in exiting from the permissible travel range can no longer be executed:




❶ C01229/2: Negative software limit position
❷ C01229/1: Positive software limit position
❸ Travel command possible
❹ Travel command not possible since the target position is outside the permissible traversing range.

[9-17] Example: Traversing range limitation by means of software limit positions

If the drive is already outside the permissible travel range and the software limit positions have been activated, only travel commands that result in the drive moving back into the permissible travel range can be executed:




❶ C01229/2: Negative software limit position
❷ C01229/1: Positive software limit position
❸ Traversing in "retracting direction" possible
❹ Traversing opposite to "retracting direction" not possible

[9-18] Example: Permissible traversing direction if software limit positions active

If the software limit positions are active and a software limit position is passed ("overtravel"):

- The error response "TroubleQuickStop" takes place in the Lenze setting, i.e. the drive is brought to a standstill in the deceleration time set for the quick stop function and does so irrespective of the setpoint selection. The error response can be parameterised in C00595/3 and C00595/4.
- The error message "Ck03: Pos. SW limit position" or "Ck04: Neg. SW limit position" is entered in the logbook of the inverter.
- Depending on the parameterised fault response, the drive cannot traverse until the error has been acknowledged.

- C01247/Bit4 and C01247/Bit5 show if a software limit position is active.

### 9.4.3.2 Hardware limit switches

The travel range limits are monitored by means of limit switches via the inputs *bLimitSwitchPos* and *bLimitSwitchNeg* of the LS_MotionControlKernel system block.

- The two inputs react to the TRUE state.
- In TA "Table positioning", the two inputs are connected to the digital inputs provided for connection of the limit switches.

**Stop!**

The limit switches are only evaluated if the limit switches for the respective operating mode have been activated (see the following table)!

| Operating mode | Hardware limit switch effective |
|---|---|
| Speed follower | Yes (adjustable in C01219 - bit 2) |
| Homing | Depending on the selected homing mode (see description of the homing modes) |
| Manual jog | No (adjustable in C01230 - bit 2) |
| Positioning | Yes |
| Stop | Yes |
| Position follower | Yes (adjustable in C01218 - bit 2) |

**Note!**

If the digital inputs used for the connection of the limit switches are to be designed in a fail-safe manner (activation at LOW level), you simply change the terminal polarity of the corresponding digital inputs in C00114.

**Behaviour when hardware limit switches are active**

If one of the two monitoring inputs is set to TRUE:

- The error response "TroubleQuickStop" takes place in the Lenze setting, i.e. the drive is brought to a standstill in the deceleration time set for the quick stop function and does so irrespective of the setpoint selection. The error response can be parameterised in C00595/1 and C00595/2.
- The error message "Ck01: Pos. HW limit switch" or. "Ck02: Neg. HW limit switch" is entered in the logbook of the inverter.
- Bit 10 ("Pos. HW-Limit Detected") or bit 11 ("Neg. HW-Limit Detected") is set in the MCK status word.
- Depending on the parameterised fault response, the drive cannot traverse until the error has been acknowledged.

Reading out code C01247/0, bit 2 /bit 3 serves to check whether the travel range limit switches are active in positive or negative direction.

**Note!**

An activated limit switch can be retracted again by manual jog in the opposed direction or with the "Retract limit switch" function. ▸ Retracting of an operated limit switch (📖 645)

Only in the "Manual jog" operating mode, retracting of the limit switch resets bit 10 ("Pos. HW-Limit Detected") or bit 11 ("Neg. HW-Limit Detected") in the MCK status word.

#### Re-activation after acknowledging the error

When the error has been acknowledged and the limit switch is still active, the following action is required for a renewed activation of the monitoring depending on the operating mode:

| Operating mode | Action for (re) activation |
|---|---|
| Speed follower | Limit switch is activated and setpoint in direction of the activated limit switch is pending<br>**Note:** If the setpoint e.g. is specified via an analog potentiometer, potentiometer noise can mean an unintentional setpoint causing the error to be set again. |
| Homing | Limit switch is activated and *bHomStartStop* = TRUE |
| Manual jog | Limit switch is activated and manual jog in direction of the activated limit switch |
| Positioning | Limit switch is activated and setpoint command has been transmitted |
| Stop | Setting of the operating mode |
| Position follower | Limit switch is activated and setpoint in direction of the activated limit switch is pending |

### 9.4.4 Target position monitoring (status "drive in target")

The target position monitoring detects whether the drive has reached the target.

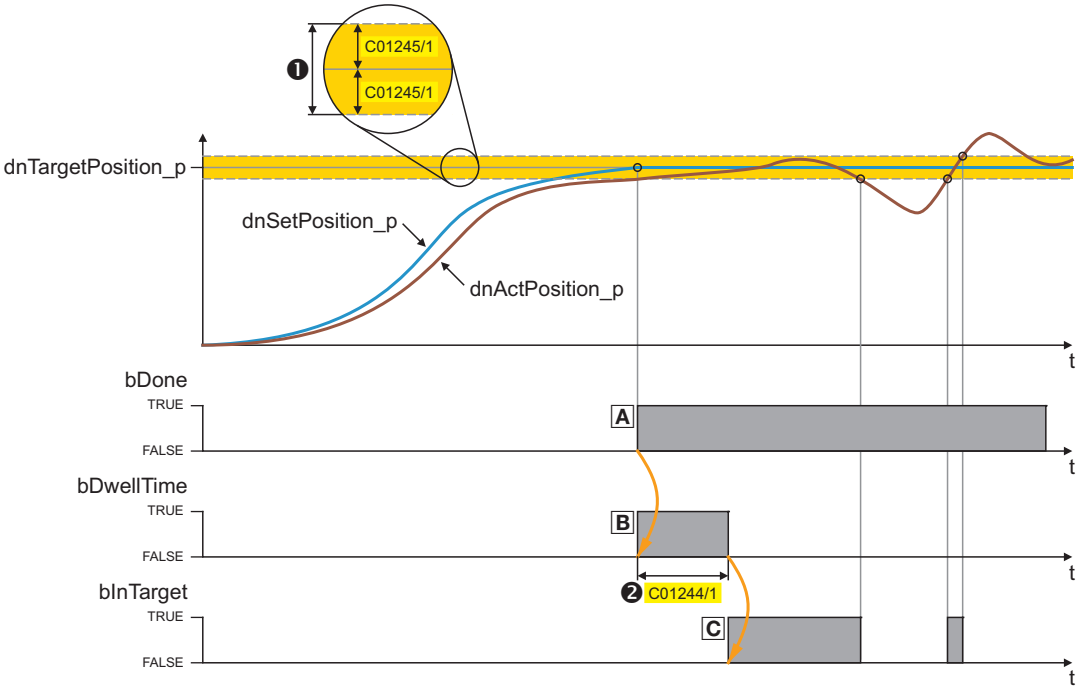
**Note!**

- The target position detection is active with the following operating modes/functions:
  - Operating mode "Positioning": Absolute and relative positioning mode
  - Operating mode "Homing": After approaching the starting position via start profile (if start profile has been selected)
  - "Manual jog" operating mode: when approaching the breakpoints and the software limit positions, respectively
- The target position monitoring is active until the operating mode changes or a new travel command influences the setpoint.

**Short overview of the parameters for target position monitoring:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01244/1 | MCK: Dwell time - target position<br>• For considering mechanical transient phenomenons of the tool when the target position is reached. | 100 | ms |
| C01245/1 | MCK: Window target position<br>• Symmetrical window around the target position. | 1.0000 | units |

**Principal procedure of the position detection**




❶ target position window (symmetrically around the target position; 2 x C01245/1)
❷ dwell time

[9-19] Signal characteristics

**Status "Done" (A)**

- The position setpoint has reached the target position.

**Status "DwellTime" (B)**

- Bit 16 in the MCK status word or the *bDwellTime* output of the L_MckStateInterface FB is set when the setpoint position has been reached after the dwell time set in C01244/1 has expired.
- The status detects the time after the *bDone* signal when settling to target position is executed and actual position detection is not active yet.

**Status "InTarget" (C)**

- If the actual position is within the symmetrical target position window after the dwell time has expired, bit 17 is set to "1" in the MCK status word or the *bInTarget* output of the L_MckStateInterface FB is set to TRUE.

**Note!**

- The target position including the target position window has to be within the maximum traversing range.
- Please observe the following if you use *bInTarget* as stepping condition in a step sequence:
  The *bInTarget* signal needs the set position to be located in the target. A profile restart immediately deletes the setpoint position in the first cycle and thus also resets *bInTarget*.

- The "InTarget" status is reset if:
  - the motor shaft leaves the target position window,
  - a new profile or a new motion process is started,
  - *bInTarget* has been set via manual jog (intermediate stops) and the "Manual jog" mode is left,
  - the target position window has been left in the "Speed follower" mode and a speed setpoint not equal to "0" is injected into the process,
  - a new homing process starts,
  - the reference is set,
  - the device is switched off/on.

### 9.4.5 Monitoring of the maximum travel distance

Continuous travel requests in the "Positioning" operating mode (or relative positioning with feed in the same direction) cause an overflow of the position integrators when the max. display area of the position is reached and the reference is set. The same behaviour takes place in the "Speed follower" and "Manual jog" operating modes. For this reason, a monitoring mode of the maximum travel distance is implemented in the **Motion Control Kernel**.

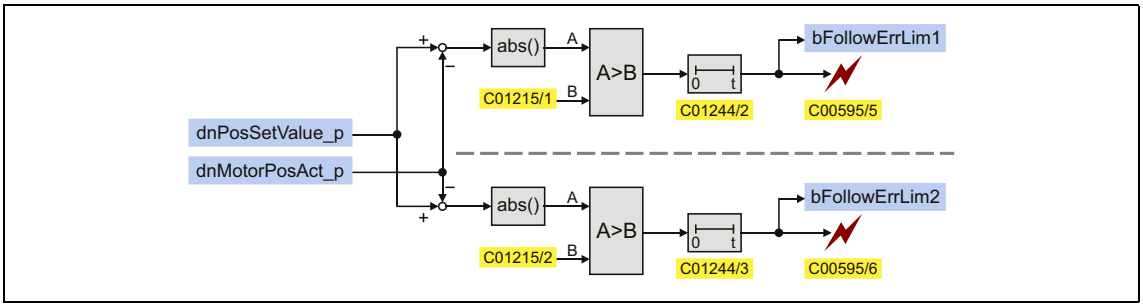
- If the maximum travel distance (C01213/1) is exceeded:
  - The error response set in C00595/7 will be carried out (Lenze setting: "TroubleQuickStop").
  - The "Ck07: Travel range limit exceeded" error message is entered into the logbook.
  - The internal status "reference known" is reset (the inverter is no longer referenced).
- A setting of C00595/7 = "0: No Reaction" deactivates the monitoring.

### 9.4.6 Following error monitoring system

The difference between set position and actual position is called the following error. Ideally, the following error should be "0". The set position is created by the internal definition of the traversing profiles of the **Motion Control Kernel**. The actual position is created by the integration of the speed supplied by the position encoder. If the position control is adjusted optimally, only a minimum following error arises which is always compensated dynamically and not increases continuously.

Certain processes, however, require that a defined limit as a difference between set position and actual position is not exceeded. If it is exceeded, it may have been caused by mechanical blocking in the machine and the system part is not situated at the position defined at that time. In such a case, it makes sense to activate the "Fault" error response to make the motor torqueless.

In the 8400 TopLine inverter, two independent following error monitoring systems can be parameterised:




[9-20] Two-channel following error monitoring system

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01215/1 | MCK: Following error limit 1 | 0.0000 | units |
| C01215/2 | MCK: Following error limit 2 | 0.0000 | units |
| C01244/2 | MCK: Following error deceleration 1 | 0.000 | s |
| C01244/3 | MCK: Following error deceleration 2 | 0.000 | s |
| C00595/5 | MCK: Resp. to following error 1 | Warning | |
| C00595/6 | MCK: Resp. to following error 2 | Warning | |

**Note!**

If the limit for the following error is set to "0.0000 units" (Lenze setting, the following error monitoring system is not active.

**Tip!**

In certain situations (e.g. dynamic acceleration of the load), higher system-dependent following errors occur than while approaching the target position.

In order that no error is triggered during acceleration and a close tolerance limit can be monitored all the same at standstill in the target, the addressing of the following error monitoring system can be decelerated. Thus, dynamic processes or torque impulses occurring for short periods can be "masked out".

**Operating mode**

If the limit for the following error in C01215/1..2 is set higher than "0.0000 units" and if the current following error exceeds this limit over the time set in C01244/x:

- The error response set in C00595/5 or C00595/6 will be carried out (Lenze setting: "Warning").
- The error message "Ck05: Error, following error 1" or "Ck06: Error, following error 2" will be entered into the logbook.
- The status output *bFollowErrLim1* or *bFollowErrLim2* at the SB LS_MotionControlKernel is set to TRUE.

### 9.4.6.1 Monitoring of the following error in case of controller inhibit

Up to and including version 13.xx.xx, the setpoint position is always set = actual position in case of controller inhibit.

From version 14.00.00 onwards, the setpoint position is only set = actual position if the current following error reaches the limit value set in C01215/3.

- In the Lenze setting C01215/3 = "0.0000 units" the behaviour is as before, i.e. the setpoint position is immediately set = actual position in case of controller inhibit.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01215/3 | MCK: Following error SetPos=ActPos | 0.0000 | units |

**Note!**

The limit value set in C01215/3 should not be set higher than the drift of the actual position at standstill of the drive plus a possible safety option. Otherwise, the position controller carries out an intermittent compensation after the controller inhibit is deactivated due to the pending system deviation!

**Application example**

In case ofa continuous relative positioning by 100 units and a setting of the controller inhibit after each positioning process (e.g. by means of the Holding brake control in the "automatically controlled" mode), it is expected that after 10 positioning processes, an absolute target of 10 x 100 units = 1000 units has been reached.

Without setting a limit value (C01215/3 =0), the continuous setting of the setpoint position to the actual position in case of controller inhibit causes a drifting of the setpoint position due to the drifting of the actual position at standstill. After ten positioning processes, only e.g. 999.5 units have been reached instead of 1000 units. If, however, the setpoint position is not set to the actual position (C01215/3 <> 0) in case of controller inhibit, the setpoint position is exactly in the target after 10 positioning processes and 9 x controller inhibits (in case of 10 x relative traverse paths).

### 9.4.7 Setpoint holdback for bus runtime compensation

This function extension is available from version 12.00.00!

**Basics of the "setpoint holdback" function**

In case of a master/slave application in which a slave drive must follow a master drive with a precise angle, the data coupling from axis to axis is mostly effected via real-time capable fieldbuses (e.g. CAN).

- Here, the data is always transferred in a time-controlled way. The axes are synchronised, i.e. the reading of the process data and its internal processing are effected at the same time.
- For such applications, the 8400 TopLine inverter provides the Axis bus which is very easy to configure.

During a synchronous run between a master and several slaves, mostly a position and speed setpoint generated by the master are transferred to the slaves.

- The setpoints are generated in the master, usually by the LS_MotionControlKernel system block.
- The setpoints are transferred to the slaves via PDOs.
- The output to the respective fieldbus is effected via the corresponding port blocks (e.g. LP_CanOut1 or LP_MciOut) or in case of the axis bus via the LS_AxisBusOut system block.

Due to the bus cycle time (e.g. 1 ms or 2 ... 4 ms for buses with a higher load) and the telegram runtimes (approx. 260 µs per PDO @500 kbits), the setpoints generated by the master arrive in the slave axes with a delay. At this time, the master has already transferred the setpoints to the control loops of the motor control. This offset causes an error in the synchronism of the axes.

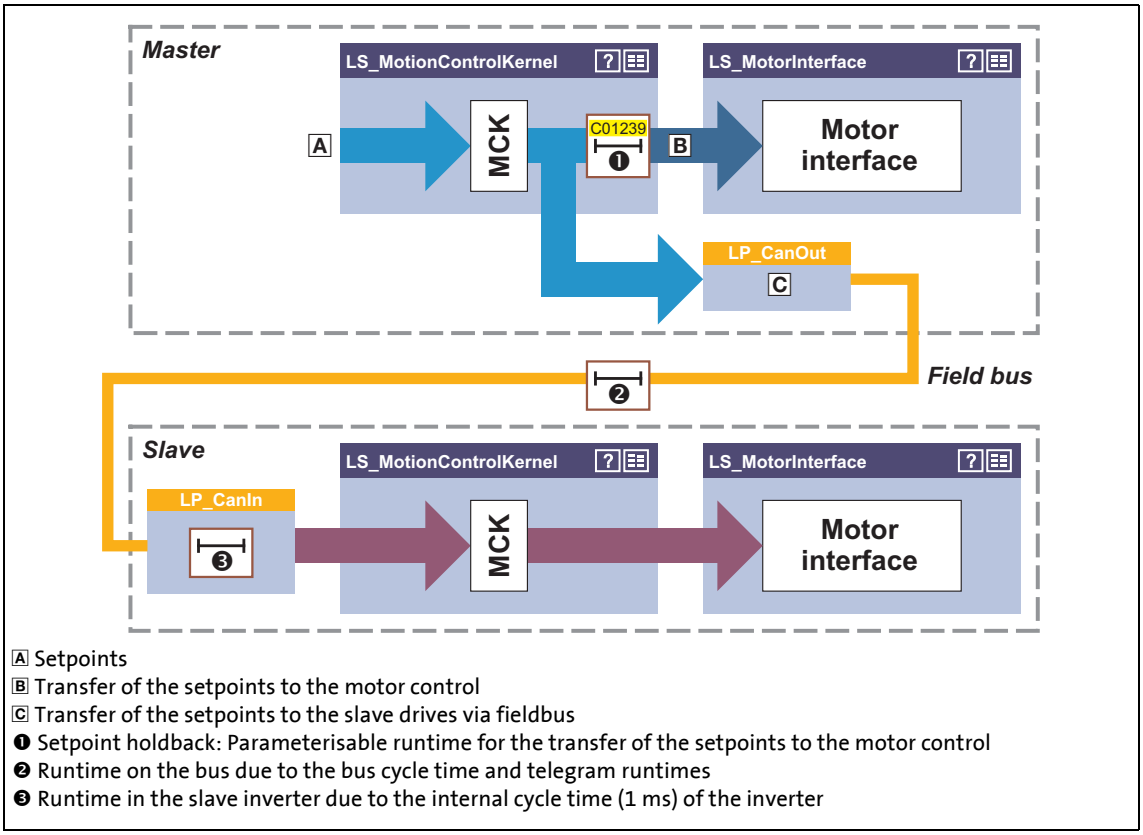
- In case of a mechanical coupling, e.g. a portal feed with two drives, an inclination of the two feed drives is caused.
- This behaviour is more or less distinctive for all serial bus systems depending on the transmission speed and cycle time.

A further delay between the reading of the received data and the forwarding to the control loops is caused by the internal cycle time of the inverters (1 ms for the 8400 device series).

### Parameterising the "setpoint holdback" function

For compensating the runtimes described before, a setpoint holdback can be parameterised in C01239. The function directly affects the setpoint forwarding from the **Motion Control Kernel** to the motor control.

- Internally, the setpoints are transmitted with a delay by the time set in C01239 from the LS_MotionControlKernel system block to the motor control.
- Setting rule: C01239 = bus cycle time + 1 ms
- The Lenze setting "0 ms" corresponds to the previous behaviour (no setpoint holdback).




A Setpoints
B Transfer of the setpoints to the motor control
C Transfer of the setpoints to the slave drives via fieldbus
❶ Setpoint holdback: Parameterisable runtime for the transfer of the setpoints to the motor control
❷ Runtime on the bus due to the bus cycle time and telegram runtimes
❸ Runtime in the slave inverter due to the internal cycle time (1 ms) of the inverter

[9-21] Principle of the setpoint holdback (here only in the master for "setpoint line" topology)

**Note!**

In which drives the function is to be parameterised, depends on the bus topology:

- Setpoint line (1 master, many slaves):
  Only in the master, the setpoint holdback (C01239) has to be set.
- Setpoint cascade (setpoints from one to drive to the other):
  In all drives, the setpoint holdback (C01239) has to be set.

As the setpoint holdback is implemented in the **Motion Control Kernel**, the LS_MotionControlKernel system block is required for this function independent of the setpoint source.

**Behaviour when the "setpoint holdback" function is parameterised online**

The setpoint holdback can also be set "online", i.e. without inhibiting the inverter before. This provides for an adjustment during commissioning while the setpoint generation is running, similar to the setting of the gain of the speed or position controller.

**Stop!**

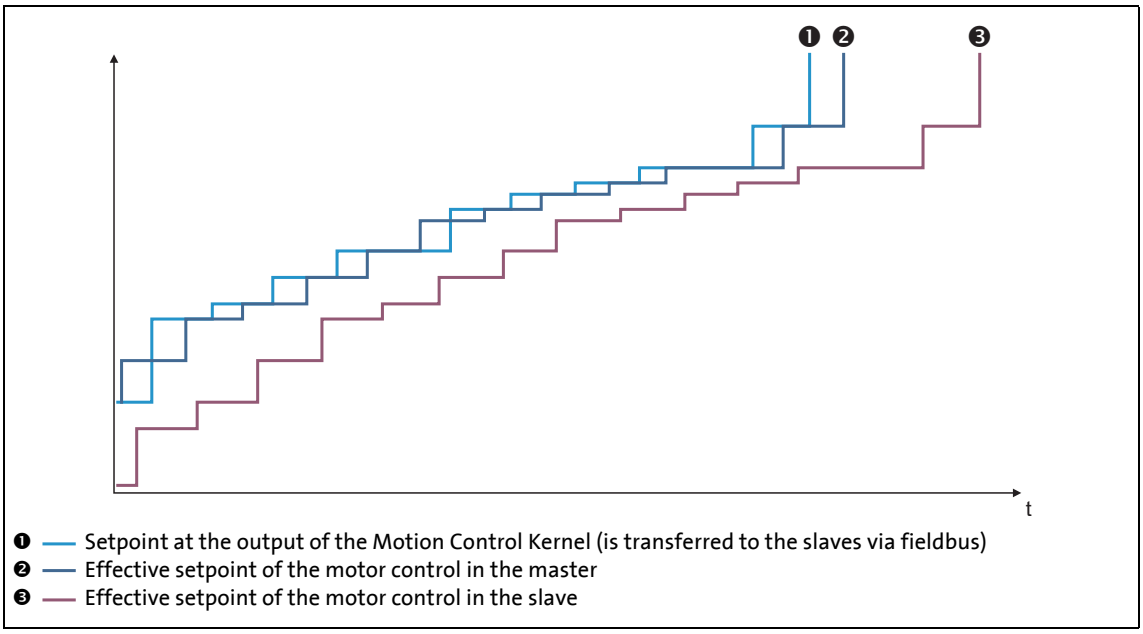
The setting while the setpoint generation is running causes a system-dependent time change of the setpoint processing so that e.g. considerable step-changes or setpoint holding processes occur.

In order to prevent setpoint step-changes which have a negative effect on the mechanics:

- Always change the setting of the setpoint holdback in 1-ms steps only.
- Make the online parameterisation at low speeds only.

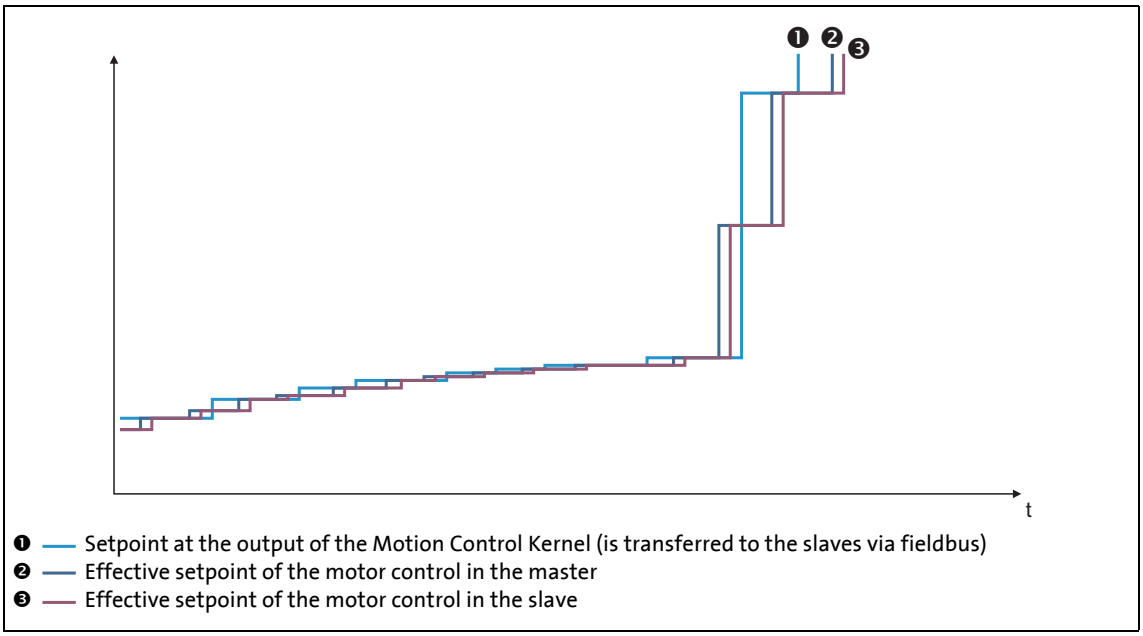
**Example: Runtime behaviour without and with setpoint holdback**

Example 1 shows the behaviour without setpoint holdback. Due to the runtimes, there is an offset between master and slave of 2 ms.




[9-22] Example 1: Runtime behaviour without setpoint holdback

In example 2, the setpoint holdback in the master is set to 2 ms. This removes the offset between master and slave. Only time differences are visible due to the telegram runtimes on the bus.




[9-23] Example 2: Runtime behaviour with setpoint holdback = 2 ms

## 9.4.8 Setting or activation of maximum jerk for traversing profiles

This function extension is available from version 14.00.00!

### Basics of the "maximum jerk" function

A change over of a traversing profile within an acceleration or deceleration ramp to a sequence profile which has a lower jerk than the traversing profile from the calculation of the new acceleration and S-ramp time causes an increase of the speed which may be not wanted for process technology reasons. The reason is that the new jerk reduces the acceleration in a too long time period which generates a too high speed.

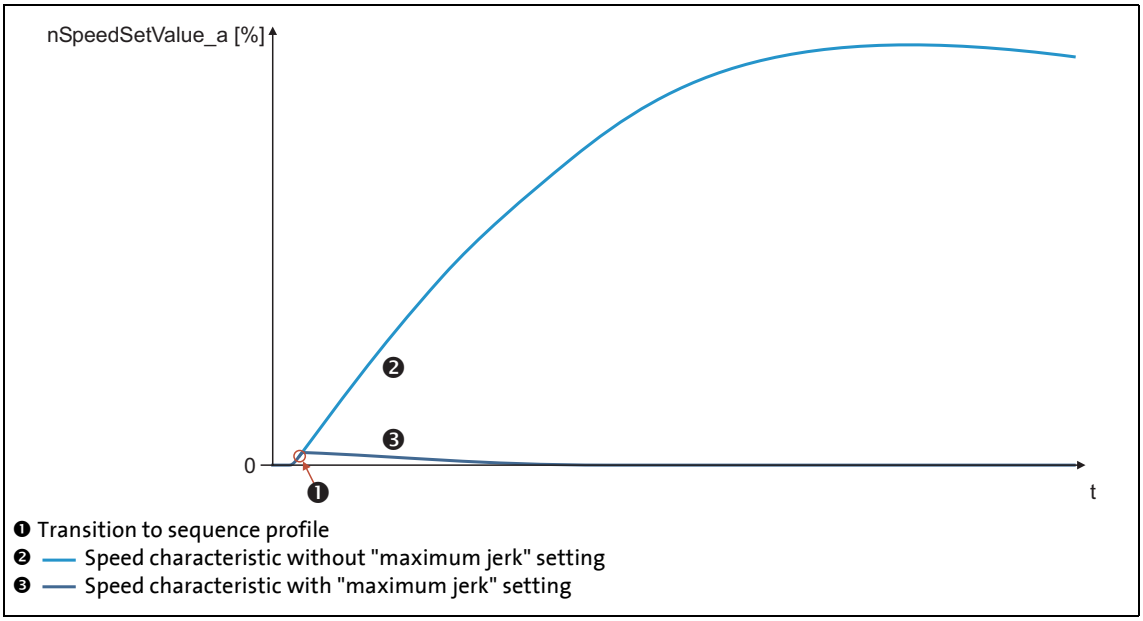
As a remedy, the "maximum jerk" function can be activated which integrates the currently pending acceleration into the acceleration of the new traversing profile or to zero.

### Parameterising the "maximum jerk" function

The "maximum jerk" function can be individually activated for the operating modes "Positioning", "Homing" and "Manual jog" via the following setting parameters:

| Parameters | Info | Lenze setting |
|---|---|---|
| C01216 - bit 5 | Maximum jerk for positioning on/off | Off |
| C01220/1 - bit 5 | Maximum jerk for homing on/off | Off |
| C01230 - bit 5 | Maximum jerk for manual jog on/off | Off |

When the function is activated, the maximum jerk is calculated from the higher value of both acceleration values and the lower value of both S-ramp time values based on the acceleration and S-ramp time of the current and sequence profile. This maximum jerk is then used for an accelerated drive in order to reduce the acceleration to 0 or to the acceleration of the sequence profile.




[9-24] Example: Behaviour during transition to sequence profile with accelerated drive

If the sequence profile is started with a non-accelerated drive, the parameters of the sequence profile are continued to be used for the jerk calculation without maximum comparison to the current profile.

## 9.5 Speed follower

In the "speed follower" operating mode, the drive follows a speed setpoint.

### 9.5.1 Parameter setting

**Short overview of parameters for the "speed follower" operating mode:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01219 | MCK: Speed follower setting | Bit coded | |
| C02610/2 | MCK: Ramp time synchr. setpoint | 2.000 | s |
| C02611/1 | MCK: Pos. max. speed | 199.99 | % |
| C02611/2 | MCK: Pos. min. speed | 0.00 | % |
| C02611/3 | MCK: Neg. min. speed | 0.00 | % |
| C02611/4 | MCK: Neg. max. speed | 199.99 | % |

#### 9.5.1.1 Functional settings

In C01219, various functional settings for the speed follower can be made in bit-coded form.

| Function | | Lenze setting |
|---|---|---|
| Bit 0 | Reserved | Off |
| Bit 1 | Reserved | Off |
| Bit 2 | **HW limit switch on**<br>In the "speed follower" operating mode, a travel range monitoring mode via hardware limit switch is active.<br>▸ Limit position monitoring (📖 601) | On |
| Bit 3 | **SW limit switch on**<br>In the "speed follower" operating mode, a travel range monitoring mode via parameterised software limit positions.<br>▸ Limit position monitoring (📖 601) | On |
| Bit 4 | Reserved | Off |
| Bit 5 | Reserved | Off |
| Bit 6 | Reserved | Off |
| Bit 7 | **Position controller off**<br>In the "Speed follower" operating mode, the position controller is deactivated. Thus, the compensation of the following error is switched off. | Off |

### 9.5.2 Requesting the operating mode

Request for "speed follower" operating mode by means of the MCK control word:

| MCK control word | | | | | | |
|---|---|---|---|---|---|---|
| **Bit 31** | **...** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | 0 | 0 | 0 | 0 |
| X = Status not significant | | | | | | |

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

### 9.5.3 Setpoint selection

The speed setpoint is selected via the *nSpeedSetValue_a* process input and additively via the *nSpeedAddValue_v* process input.

- Usually, the ramp generator L_NSet and, optionally, the process controller L_PCTRL are upstream of the *nSpeedSetValue_a* process input.
- The speed setpoint is limited internally to the speed limits set in C02611/1...4.

**Note!**

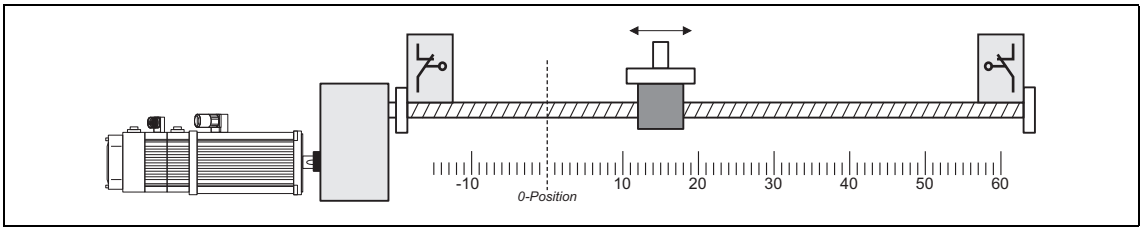
In the "Speed follower operating mode without position control, for an "Electrical shaft" interconnection the actual position value *LS_MotorInterface.dnMotorPosAct_p* must be used as master signal if a drift is to be ruled out.

When the speed limit values are set, the **Motion Control Kernel** influences the setpoint generation with a synchronisation mode The synchronisation mode serves to travel the synchronisation range dynamically with the synchronisation ramp set in C02610/2.

▸ Min/Max speed (📖 600)

## 9.6 Homing

The measuring system in the machine is selected by means of homing and the 0 position is set within the possible physical travel range:




[9-25] Homing (selection of 0 position)

The zero position (home) can be defined by a reference run or by setting a home position:

- If a reference run is carried out, the drive follows a previously selected path to find the home position.
- When the reference is set, it is selected manually when the drive stands still.

**Danger!**

During homing operations, specially assigned profile parameters are effective. If these parameters are not set correctly, the drive may execute unexpected movements!

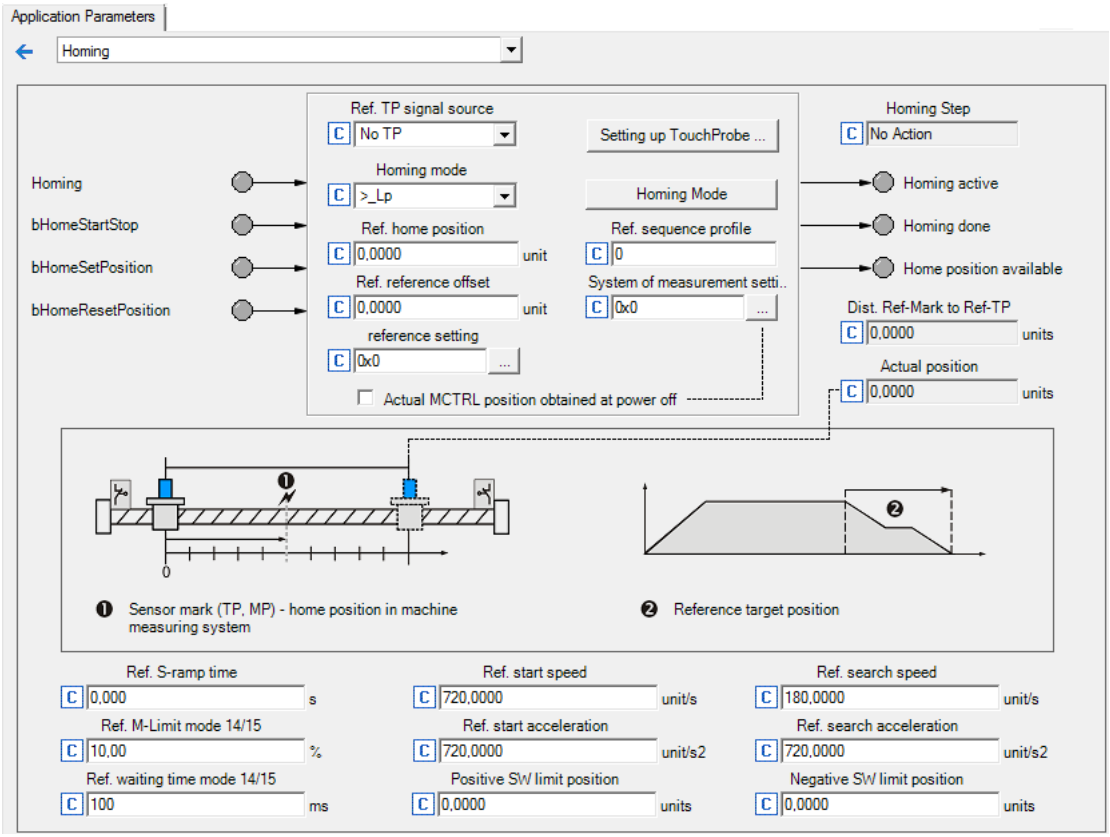
**Tip!**

A reference run is mainly used for systems with feedback via encoders, resolvers, or single-turn absolute value encoders, as in the case of these systems the home position is lost when the supply voltage is switched off.

Usually the reference is set only once during commissioning or in the event of service (e. g. if drive components are replaced), and it is mostly used for systems with feedback via absolute value encoders.

## 9.6.1 Parameter setting

### Parameterisation dialog in the »Engineer«



**Short overview of parameters for "Referencing" operating mode:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01220/1 | MCK: Ref. setting (from version 14.00.00) | Bit coded | |
| C01221 | MCK: Homing mode | 12: >_Lp | |
| C01224/1 | MCK: Ref. initial speed | 720.0000 | unit/s |
| C01225/1 | MCK: Ref. initial acceleration | 720.0000 | unit/s2 |
| C01224/2 | MCK: Ref. search speed | 180.0000 | unit/s |
| C01225/2 | MCK: Ref. search acceleration | 720.0000 | unit/s2 |
| C01226/1 | MCK: Ref. S-ramp time | 0.000 | s |
| C01222 | MCK: Ref. M limit mode 14/15 | 10.00 | % |
| C01223 | MCK: Ref. waiting time mode 14/15 | 100 | ms |
| C01227/1 | MCK: Ref. offset reference degree | 0.0000 | unit |
| C01227/2 | MCK: Ref. home position | 0.0000 | unit |
| C01228 | MCK: Ref. sequence profile | 0 | |
| C01229/1 | MCK: Positive SW limit position | 0.0000 | units |
| C01229/2 | MCK: Negative SW limit position | 0.0000 | units |
| C01246/1 | MCK: Ref. TP signal source | 0: No TP | |
| C01246/2 | MCK: Set.Ref. signal source | 0: No TP | |
| C01248/1 | MCK: Reference step | - | units |
| C01210/9 | MCK: Dist. ref. mark and Ref-TP | - | units |
| C01210/3 | MCK: Actual position | - | - |

### 9.6.1.1 Functional settings

From version 14.00.00, various functional settings for homing can be carried out bit-coded in C01220/1.

| Function | | Lenze setting |
|---|---|---|
| Bit 0 | Reserved | Off |
| Bit 1 | Reserved | Off |
| Bit 2 | Reserved | Off |
| Bit 3 | Reserved | Off |
| Bit 4 | **Start deletes bHomePosAvailable**<br>When homing is started (mode 4 ... 15), the reference information is deleted. | Off |
| Bit 5 | **Maximum jerk**<br>▸ Setting or activation of maximum jerk for traversing profiles (📖 615) | Off |
| Bit 6 | Reserved | Off |
| ... | | |
| Bit 15 | | |

### 9.6.1.2 Homing mode

Specify the referencing mode in C01221 , i.e. the way in which referencing is to take place.

- For reference setting, the referencing mode "100" is to be selected in C01221.
- For a reference search, C01221 contains referencing modes "4"..."15" which can be selected from.

| Referencing mode C01221 | Evaluated signals/sensors | | | |
|---|---|---|---|---|
| | Touch probe sensor (Sensor reference signal) | Travel range limit switch | | Pre-stop mark at *bHomingMark* |
| | | Negative limit switch | Positive limit switch | |
| 4 * | ☑ | | | ☑ |
| 5 * | ☑ | | | ☑ |
| 6 | ☑ | | | ☑ |
| 7 | ☑ | | | ☑ |
| 8 | ☑ | | | |
| 9 | ☑ | | | |
| 10 | ☑ | | ☑ | |
| 11 | ☑ | ☑ | | |
| 12 | | | ☑ | |
| 13 | | ☑ | | |
| 14 | Positive direction of rotation to torque limit. | | | |
| 15 | Negative direction of rotation to torque limit. | | | |
| 100 | Set reference directly. | | | |

* From version 14.00.00

#### Internal interfaces

The switches/sensors are evaluated via the following internal interfaces:

| Switch/sensor | Internal interface for digital input signal |
|---|---|
| Touch probe sensor<br>(Sensor reference signal) | The touch probe signal source can be selected in C01246/1.<br>• If the reference signal is to follow a real touch probe, configure the touch probe signal accordingly. ▸ Touch probe detection (📖 419)<br>**Note:**<br>In case of setting "0: No TP" in C01246/1, the digital input DI3 is used as signal source. An inversion of DI3 via C00114 is considered. |
| Positive travel range limit switch | *bLimitSwitchPos* |
| Negative travel range limit switch | *bLimitSwitchNeg* |
| Pre-stop mark/pre-stop signal | *bHomingMark*<br>• This input has to be connected to the corresponding digital input to which the pre-switch off sensor is connected.<br>• The edge sensitivity of this input and the response to the pre-switch off signal depend on the selected homing mode (see the following description of the homing modes). |

**Note!**

For a reference search with touch probe detection:

- The touch probe signal source can be selected in C01246/1.
- Go to C02810/x and select the edge, the digital input used for the connection of the touch-probe sensor is to respond. In the Lenze setting of C02810/x, no touch probe is detected!
- Moreover, no "acceptance window" must be set in C2813/x and C02814/x for accepting the touch probe signal in order that a touch probe will always be detected independent of the position.

▸ Touch probe detection (📖 419)

### Start and search profile data set

Certain referencing modes use two different profile data sets for homing in order to shorten the homing time and, at the same time, increase accuracy.

- At first, a quick approach of the limit switch/pre-stop mark (depending on the selected mode) is carried out using the start profile data set.
- After reversing at the limit switch/pre-stop mark, the search profile data set results in slower – but more accurate – approaching of the touch probe sensor.

| Start profile data set | | Search profile data set | |
|---|---|---|---|
| C01224/1 | Start speed | C01224/2 | Search speed |
| C01225/1 | Start acceleration (deceleration as well) | C01225/2 | Search acceleration (deceleration as well) |
| C01226/1 | S-ramp time (identical in the two profile data sets) | C01226/1 | S-ramp time (identical in the two profile data sets) |

**Note!**

A changeover to the search profile data set is only carried out of the search speed has been set to (C01224/2) > "0"!

The exact time of the changeover to search profile data set occurs in the respective homing mode is indicated in the process descriptions of the homing modes.

**Tip!**

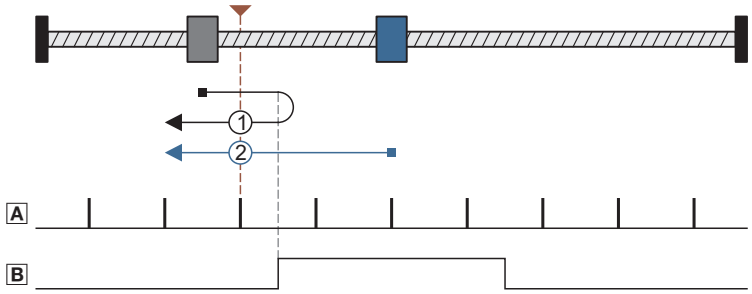
The result of setting a lower search speed in C01224/2 and a high search acceleration in C01225/2 is that deceleration to search speed takes place quickly and position detection is exact (at slower search speed).

**Abbreviations used for the referencing modes:**

| Abbreviation | Meaning |
|---|---|
| > | Movement in positive direction |
| < | Movement in negative direction |
| Lp | Positive travel range limit switch |
| Ln | Negative travel range limit switch |
| Rp | Positive edge pre-stop mark/pre-stop signal |
| Rn | Negative edge - pre-stop mark/pre-stop signal |
| TP | Touch probe or sensor reference signal/reference switch detected |
| Mlim | Torque limit value reached |

### Mode 4: >_Rp_<_TP

(from version 14.00.00)




A Touch probe
B Pre-stop mark

| Mode 4 | > | Rp | < | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|---|---|
| | Start speed | | Search speed | | | Profile speed |
| **Lp** | Active | Active | Active | Active | Active | Active |
| **Ln** | Inactive | Active | Active | Active | Active | Active |

Process of case 1 ➞ Axis has not yet reached the pre-stop mark (*bHomingMark* = FALSE):

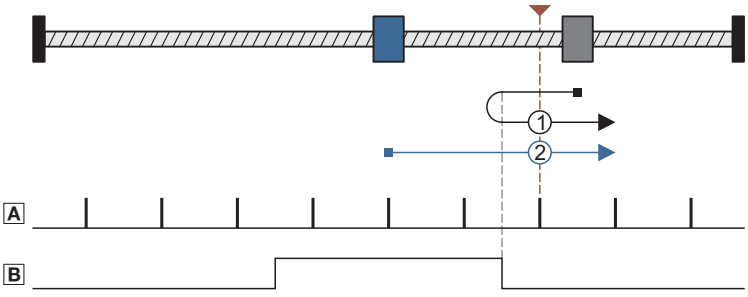
1. Movement in positive direction with start profile data set.
2. Reversing in case of positive edge at *bHomingMark* and, at the same time, activating the search profile data set for continued reference searching.
3. Negative edge at *bHomingMark* enables home position detection.
4. Following edge of the touch probe sensor sets home position.

Process of case 2 ➞ Axis is already on the pre-stop mark (*bHomingMark* = TRUE):

1. Movement in negative direction with search profile data set.
2. Negative edge at *bHomingMark* enables home position detection.
3. Following edge of the touch probe sensor sets home position.

**Mode 5: <_Rp_>_TP**

(from version 14.00.00)




A Touch probe
B Pre-stop mark

| Mode 5 | < | Rp | > | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|---|---|
| | Start speed | | Search speed | | | Profile speed |
| Lp | Inactive | Active | Active | Active | Active | Active |
| Ln | Active | Active | Active | Active | Active | Active |

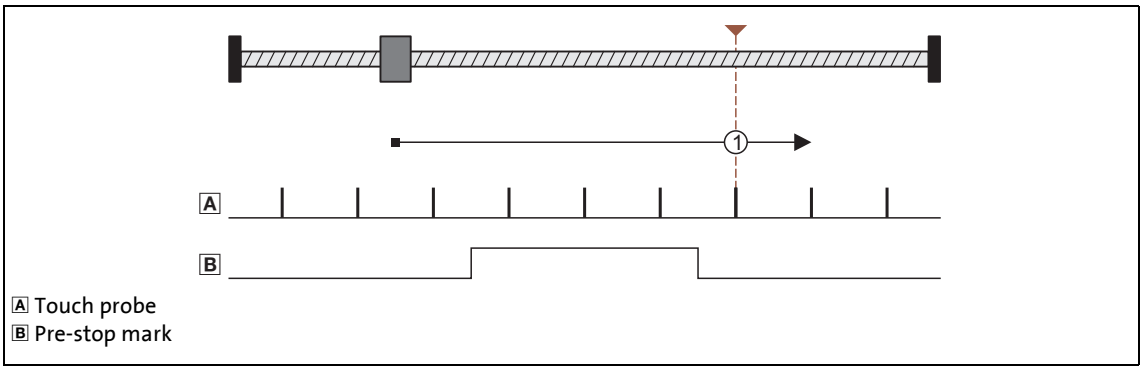
Process of case 1 ➞ Axis has not yet reached the pre-stop mark (*bHomingMark* = FALSE):

1. Movement in negative direction with start profile data set.
2. Reversing in case of positive edge at *bHomingMark* and, at the same time, activating the search profile data set for continued reference searching.
3. Negative edge at *bHomingMark* enables home position detection.
4. Following edge of the touch probe sensor sets home position.

Process of case 2 ➞ Axis is already on the pre-stop mark (*bHomingMark* = TRUE):

1. Movement in positive direction with search profile data set.
2. Negative edge at *bHomingMark* enables home position detection.
3. Following edge of the touch probe sensor sets home position.

### Mode 6: >_Rn_>_TP



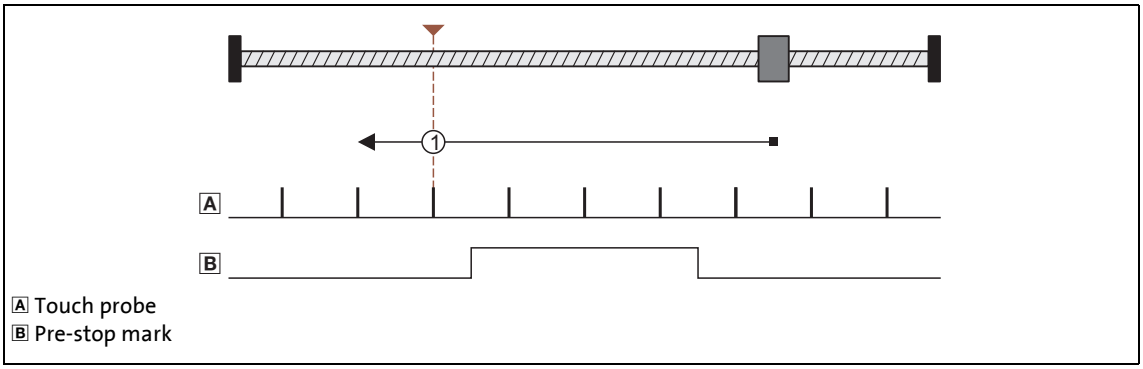

A Touch probe
B Pre-stop mark

| Mode 6 | > | Rn | > | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|---|---|
| | Start speed | | Search speed | | | Profile speed |
| Lp | Active | Active | Active | Active | Active | Active |
| Ln | Inactive | Inactive | Inactive | Inactive | Active | Active |

Procedure:

1. Movement in positive direction with start profile data set.
2. Positive edge at *bHomingMark* activates search profile data set for further reference search.
3. Negative edge at *bHomingMark* enables home position detection.
4. Following edge of the touch probe sensor sets home position.

### Mode 7: <_Rn_<_TP



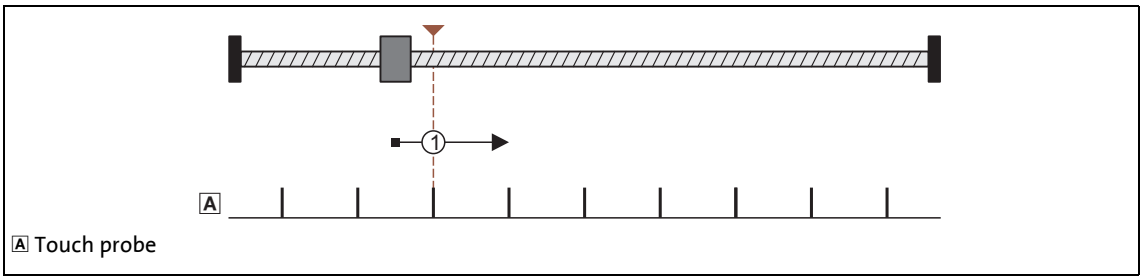

A Touch probe
B Pre-stop mark

| Mode 7 | < | Rn | < | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|---|---|
| | Start speed | | Search speed | | | Profile speed |
| Lp | Inactive | Inactive | Inactive | Inactive | Active | Active |
| Ln | Active | Active | Active | Active | Active | Active |

Procedure:

1. Movement in negative direction with start profile data set.
2. Positive edge at *bHomingMark* activates search profile data set for further reference search.
3. Negative edge at *bHomingMark* enables home position detection.
4. Following edge of the touch probe sensor sets home position.

### Mode 8: >_TP



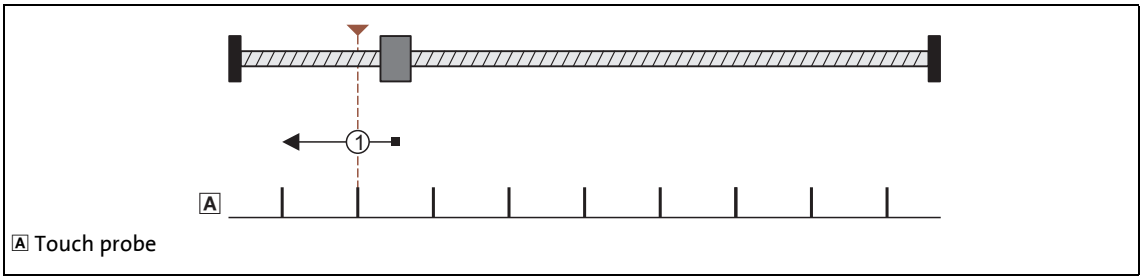

A Touch probe

| Mode 8 | > | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| Lp | Active | Active | Active | Active |
| Ln | Inactive | Inactive | Active | Active |

Procedure:

1. Movement in positive direction with start profile data set.
2. Following edge of the touch probe sensor sets home position.

### Mode 9: <_TP



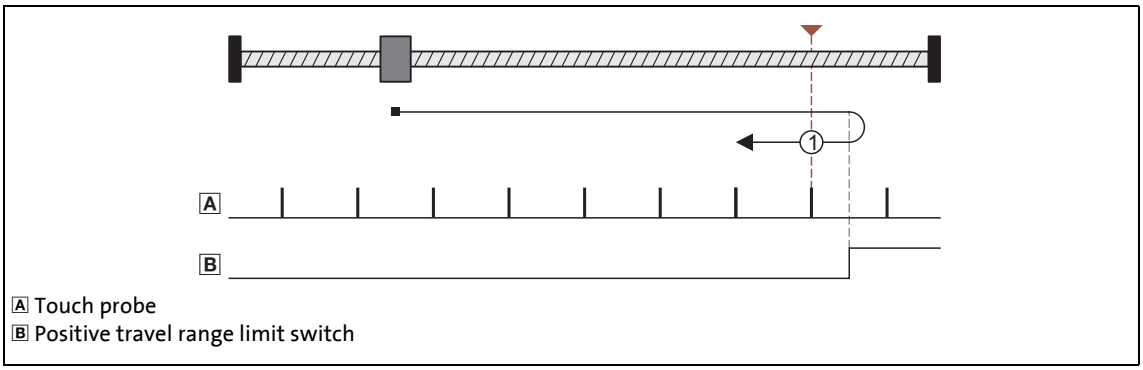

A Touch probe

| Mode 9 | < | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| Lp | Inactive | Inactive | Active | Active |
| Ln | Active | Active | Active | Active |

Procedure:

1. Movement in negative direction with start profile data set.
2. Following edge of the touch probe sensor sets home position.

### Mode 10: >_Lp_<_TP



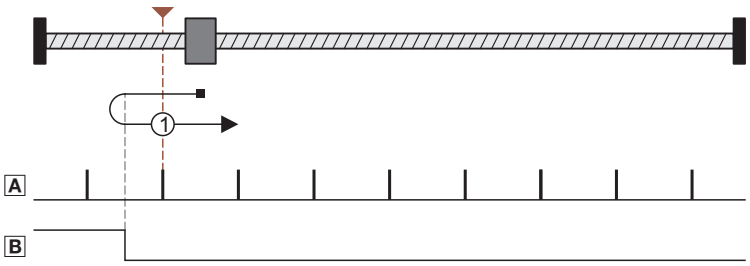

A Touch probe
B Positive travel range limit switch

| Mode 10 | > | Lp | < | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|---|---|
| | Start speed | | Search speed | | | Profile speed |
| **Lp** | Inactive | Inactive | Inactive | Inactive | Inactive | Active |
| **Ln** | Inactive | Inactive | Active | Active | Active | Active |

Procedure:

1. Movement in positive direction with start profile data set.
2. Reversing when the edge of the positive travel range limit switch is positive and, at the same time, activation of search profile data set for continued reference searching.
3. Negative edge of the travel range limit switch enables home position detection.
4. Following edge of the touch probe sensor sets home position.

### Mode 11: <_Ln_>_TP



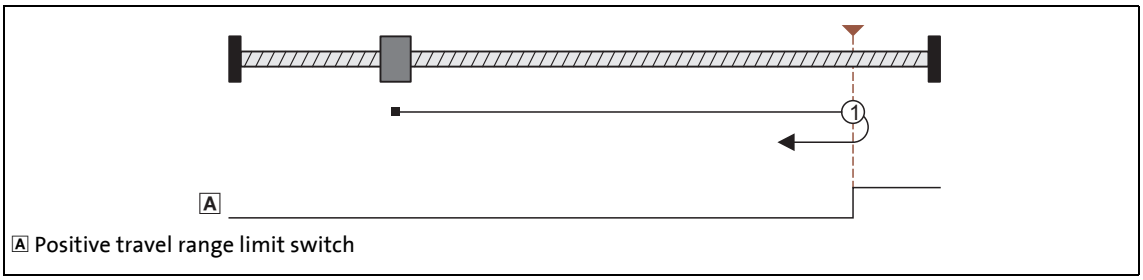

A Touch probe
B Negative travel range limit switch

| Mode 11 | < | Ln | > | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|---|---|
| | Start speed | | Search speed | | | Profile speed |
| **Lp** | Inactive | Inactive | Active | Active | Active | Active |
| **Ln** | Inactive | Inactive | Inactive | Inactive | Inactive | Active |

Procedure:

1. Movement in negative direction with start profile data set.
2. Reversing when the edge of the negative travel range limit switch is positive and, at the same time, activation of search profile data set for continued reference searching.
3. Negative edge of the travel range limit switch enables home position detection.
4. Following edge of the touch probe sensor sets home position.

### Mode 12: >_Lp




A Positive travel range limit switch

| Mode 12 | > | Lp | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| **Lp** | Inactive | Inactive | Inactive | Active |
| **Ln** | Inactive | Inactive | Active | Active |

Procedure:

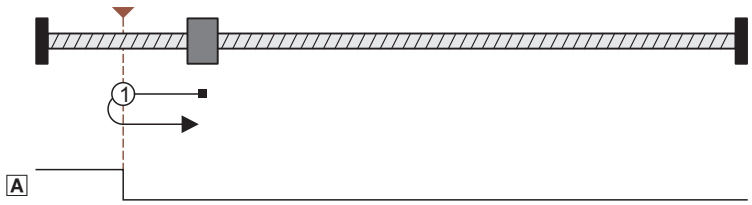
1. Movement in positive direction with start profile data set.
2. Positive edge of the travel range limit switch sets reference.

**Note!**

The load machine can also leave the travel range limit switch. There follows a return to the home position that was set with the positive edge of the travel range limit switch.

- It is possible that, as a result, the machine will remain on an operated limit switch.
- It is therefore recommended that a home value offset be set in C01227/1 in order to release the operated limit switch.

### Mode 13: <_Ln




A Negative travel range limit switch

| Mode 13 | < | Ln | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| **Lp** | Inactive | Inactive | Active | Active |
| **Ln** | Inactive | Inactive | Inactive | Active |

Procedure:

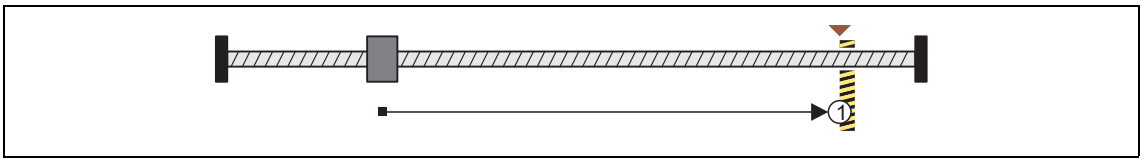
1. Movement in negative direction with start profile data set.
2. Positive edge of the travel range limit switch sets reference.

**Note!**

The load machine can also leave the travel range limit switch. There follows a return to the home position that was set with the positive edge of the travel range limit switch.

- It is possible that, as a result, the machine will remain on an operated limit switch.
- It is therefore recommended that a home value offset be set in C01227/1 in order to release the operated limit switch.

### Mode 14: >_Mlim



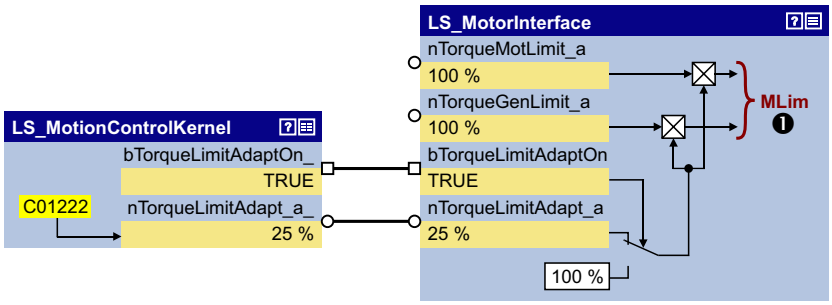

| Mode 14 | > | Mlim | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| Lp | Active | Active | Active | Active |
| Ln | Inactive | Inactive | Active | Active |

Procedure:

1. Movement in a positive direction with reduced torque and start profile data set.
2. The reference is set if it is detected that the torque limit set in C01222 is exceeded beyond the time set in C01223 ("Homing to positive stop").
   - If a home value offset has not been set, this position is the home position.
   - If a home value offset has been set, correctly signed traversing by this offset takes place and the home position is set at the end of this travel distance.

**Tip!**

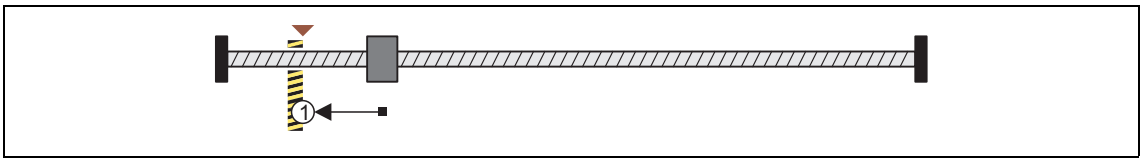
Set a home value offset in C01227/1 to avoid stopping at the positive stop.




❶ Effective torque limit

[9-26] Wiring for torque limit when homing to "Positive stop" (homing modes 14/15)

### Mode 15: <_Mlim




| Mode 15 | < | Mlim | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| Lp | Inactive | Inactive | Active | Active |
| Ln | Active | Active | Active | Active |

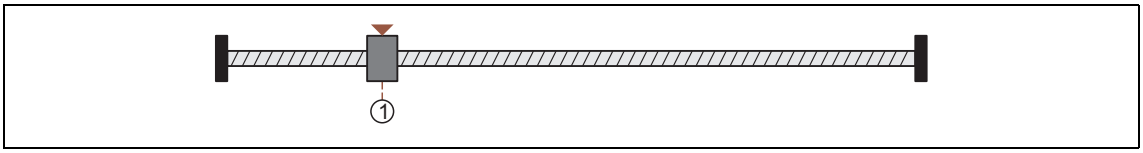
Procedure:

1. Movement in a negative direction with reduced torque and start profile data set.
2. The reference is set if it is detected that the torque limit set in C01222 is exceeded beyond the time set in C01223 ("Homing to positive stop").
   - If a home value offset has not been set, this position is the home position.
   - If a home value offset has been set, correctly signed traversing by this offset takes place and the home position is set at the end of this travel distance.

**Tip!**

Set a home value offset in C01227/1 to avoid stopping at the positive stop.

### Mode 100: SetRef




| Mode 100 | SetRef | Offset path | Sequence profile (optional) |
|---|---|---|---|
| | - | Start speed | Profile speed |
| Lp | Active | Active | Active |
| Ln | Active | Active | Active |

When the drive is at standstill, the measuring system is set via the bit 8 ("HomStartStop") in MCK control word. The current actual position now corresponds to the home position set in C01227/2 in the machine measuring system.

### 9.6.1.3 Home position & home value offset

If the home position is set in the course of the reference search, this determined position in the machine measuring system now corresponds to the home position set in C01227/2.

If a home value offset has been set in C01227/1, the drive continues correctly signed traversing by this offset at search speed after the home position has been found. The home position is not set until after this movement.

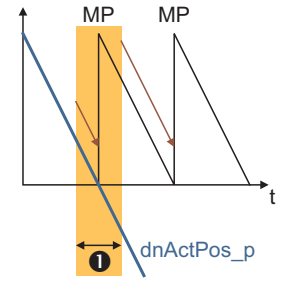
### 9.6.1.4 Traversing a sequence profile after completion of homing

If necessary, a sequence (following or subsequent) profile can be entered in C01228 in order to arrange for a positioning movement to be carried out immediately after completion of homing.

**Note!**

The positioning movement for the sequence profile set in C01228 is also carried out in the "referencing" mode. If a second sequence (following or subsequent) profile has been defined in the sequence profile, it is not automatically started!

### 9.6.1.5 Measurement of the distance between pre-stop mark/limit switch and TP/MP




❶ Jitter of the pre-switch off signal

Depending on how close the pre-stop mark is located to the encoder zero pulse and even jittering around it, it might occur that the drive performs one more or one less revolution after the home position detection has been enabled.

From version 14.00.00, the display parameter C01210/9 is available for this purpose. This parameter shows the distance between the signal which starts the home position detection and the detected touch probe signal or encoder zero pulse (MP) in [units].

The measurement is executed for the homing modes 4, 5, 6, 7, 10 and 11.

| Referencing mode | Display in C01210/9 in [units] |
|---|---|
| 4 / 5 | $dnActPos_p_{(TP/MP)}$ - $dnActPos_p_{(bHomingMark_TRUE_to_FALSE)}$ |
| 6 / 7 | $dnActPos_p_{(TP/MP)}$ - $dnActPos_p_{(bHomingMark_TRUE_to_FALSE)}$ |
| 10 | $dnActPos_p_{(TP/MP)}$ - $dnActPos_p_{(bLimitSwitchPos_TRUE_to_FALSE)}$ |
| 11 | $dnActPos_p_{(TP/MP)}$ - $dnActPos_p_{(bLimitSwitchNeg_TRUE_to_FALSE)}$ |

### 9.6.1.6 Option "Actual MCTRL position received at mains switch-off"

The encoder position can be saved with mains failure protection in the inverter. For this purpose, the bit 0 has to be set in C02652.

- From version 14.00.00, the "Actual MCTRL position received at mains switch-off" option can also be accessed in the »Engineer« via the parameterisation dialog for the basic function **Homing** (depending on the set position encoder system).
- If bit 0 is set in C02652, the actual position of the motor control is saved in the inverter (not in the memory module) and thus remains known to the drive control after mains switching.

The following overview shows the actual position after mains connection depending on the setting C02652/bit 0, C00490, C00422, C00925 (if resolver is selected) and the existence of the reference information:

| C02652 bit 0 | Reference available in case of mains switch-off (Status bit "HomePos Available") | Actual position and status reference in case of mains switch-on with setting | | |
|---|---|---|---|---|
| | | C00490 = 4<br>C00925 = 1 | C00490 = 0, 3, 4, 10<br>C00422 = 0, 1<br>C00925 > 1 (with resolver) | C00490 = 3<br>C00422 = 2, 3, 4 |
| | | • Resolver with number of pole pairs = 1 | • Incremental encoder (TTL)<br>• Sin/cos encoder (Singleturn)<br>• Resolver with number of pole pairs > 1<br>• No sensor | • Absolute value encoder (Multiturn) |
| 0 | 0 | 0 incr. | 0 incr. | *dnActPos*(mains switch-off) ±ϑ |
| | | Reference is unknown | Reference is unknown | Reference is unknown |
| | 1 | 0 incr. | 0 incr. | *dnActPos*(mains switch-off) ±ϑ |
| | | Reference is unknown | Reference is unknown | Reference available |
| 1 | 0 | *dnActPos*(mains switch-off) ±ϑ | *dnActPos*(mains switch-off) | *dnActPos*(mains switch-off) ±ϑ |
| | | Reference is unknown | Reference is unknown | Reference is unknown |
| | 1 | *dnActPos*(mains switch-off) ±ϑ | *dnActPos*(mains switch-off) | *dnActPos*(mains switch-off) ±ϑ |
| | | Reference available | Reference available | Reference available |
| ±ϑ ≡ A rotation within the max. possible angle of rotation is possible at mains switch-off. | | | | |

### 9.6.2 Requesting the operating mode

Request for the "referencing" operating mode by means of the MCK control word:

| MCK control word | | | | | | |
|---|---|---|---|---|---|---|
| Bit 31 | ... | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | 0 | 0 | 0 | 1 |
| X = Status not significant | | | | | | |

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

### 9.6.3 Execute homing

Control is exercised by means of bits 8 ... 10 in the MCK control word:

| Homing | MCK control word | | |
|---|---|---|---|
| | Bit 10 | Bit 9 | Bit 8 |
| | HomResetPos | HomSetPos | HomStartStop |
| Stop homing | 0 | 0 | 0 |
| Start homing | 0 | 0 | 1 |
| Set home position | 0 | 1 | X |
| Delete home position | 1 | 0 | X |
| X = Status not significant | | | |

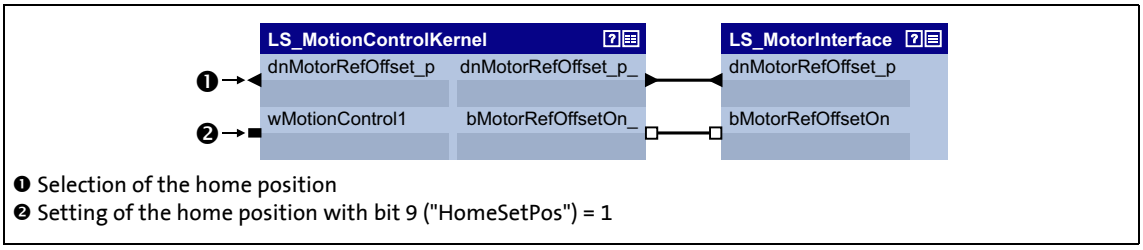
If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process inputs for controlling the operating mode:

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bHomingStartStop<br>BOOL | Start/stop homing<br>• Only possible in the "referencing" operating mode.<br>• Current status of the referencing process, see C01248/1. | |
| | TRUE | If one of homing modes "4" ... "15" in C01221 is selected:<br>**Start reference search**<br>• The current status of the reference search is indicated via the status outputs *bHomingDone* and *bHomePosAvailable*.<br>• If the 100_ Set_Ref_directly" referencing mode has been selected in C01221, the home position can be set manually via the *bHomeStartStop* input with the drive at a standstill. The current actual position is set as the home position (C01227/2) |
| | | If homing mode "100: SetRef" is selected:<br>**Setting the home position manually**<br>• The home position is set manually with the drive at a standstill. The current actual position now corresponds to the reference position set in C01227/2 in the machine measuring system. |
| | TRUE↘FALSE | Stop homing.<br>• If the *bHomingStartStop* input is reset to FALSE during active referencing, homing is cancelled and the drive is brought to a standstill. |
| bHomingSetPos<br>BOOL | Set home position (Homing on the fly)<br>• With referencing "on the fly", the home position of a machine can be set during ongoing movement. Jerking and compensating movements do not occur. | |
| | FALSE↗TRUE | The position applied at the *dnMotorRefOffset_p* input of the SB LS_MotionControlKernel at the time of activation is set as home position if C01246/2 = "0: No TP" is set. |
| bHomingResetPos<br>BOOL | Delete home position<br>**Note:** With this function, positions are not deleted but only the status signals *bHomePosAvailable* and *bHomingDone* are reset. Setpoints and actual positions remain untouched until a renewed reference setting or homing. | |
| | FALSE↗TRUE | The internal status "reference known" is reset.<br>• The inverter is no longer referenced.<br>• The process outputs *bHomePosAvailable* and *bHomingDone* are reset to FALSE. |

### 9.6.3.1 Homing on the fly

With referencing "on the fly", the home position of a machine can be set during ongoing movement. Jerking and compensating movements do not occur.

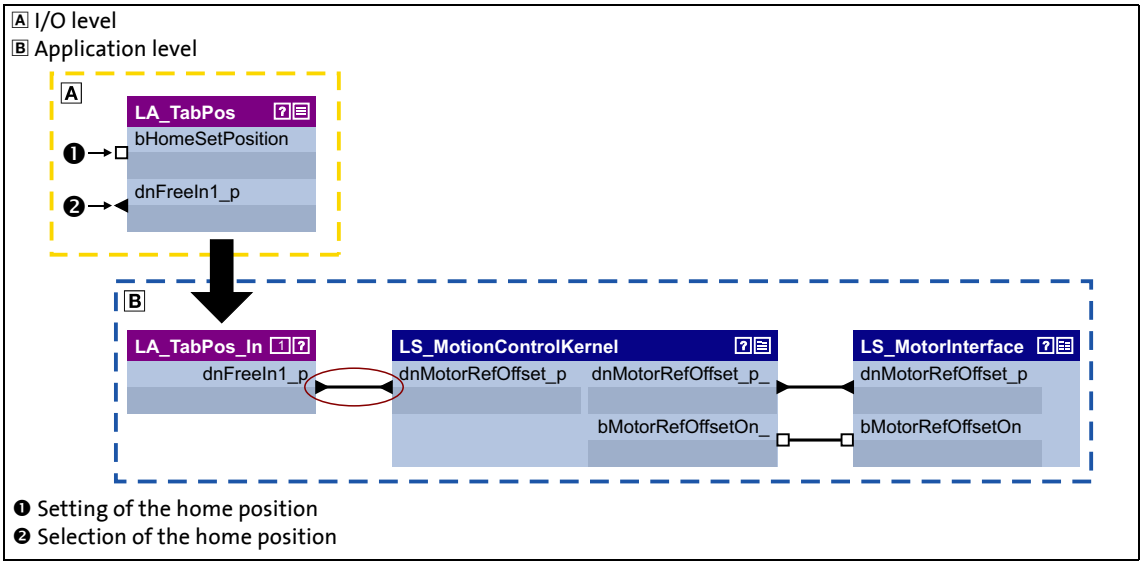
The following illustration shows the relevant interfaces for selecting the home position at the SB LS_MotionControlKernel and SB LS_MotorInterface:




[9-27] Interface for selecting the home position

In order that position selections can be accepted while homing on the fly, an additional connection is required in the application level.

- The following illustration shows the required modification using the example of the "table positioning" technology application.
- For transmitting the home position from the I/O level to the application level, the "free" input *dnFreeIn1_p* is used here:




[9-28] Modified interconnection for selecting the home position

## 9.7 Manual jog

In this operating mode, the drive can be traversed manually in clockwise or anticlockwise direction ("jogging mode").

- As an option, it is possible to change over to a second speed during traversing.
- "Retraction" of operated (travel range) limit switches is also supported. Only traversing in the corresponding retracting direction is then possible.

**Danger!**

During manual jog, specially assigned profile parameters are effective. If these parameters are not set correctly, the drive may execute unexpected movements!

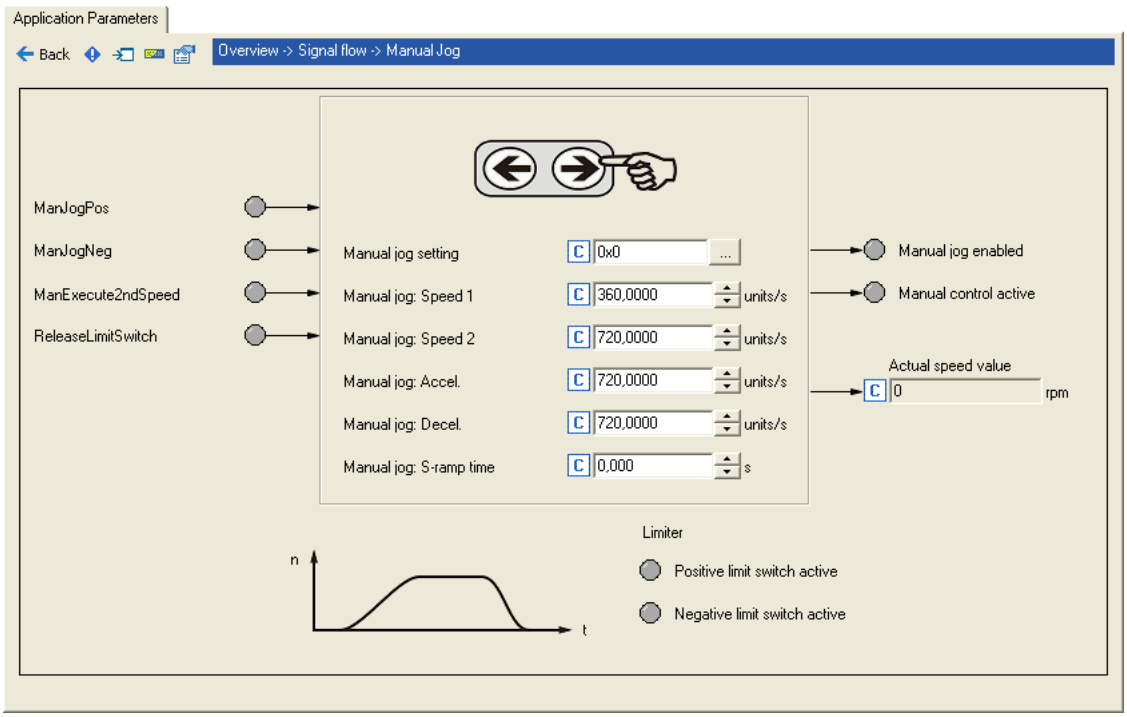
**Stop!**

In the Lenze setting, travel range monitoring is switched-off in C01230 by means of hardware limit switches and software limit positions for the "Manual jog" operating mode!

If travel range monitoring has been deactivated, the drive can travel into a mechanical limit during manual jogging and machine parts can be destroyed or damaged!

## 9.7.1 Parameter setting

### Parameterisation dialog in the »Engineer«




### Short overview of parameters for "manual jogging" mode":

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01230 | MCK: Manual jog setting | Bit coded | |
| C01231/1 | Manual jog: speed 1 | 360.0000 | units/s |
| C01231/2 | Manual jog: Speed 2 | 720.0000 | units/s |
| C01232/1 | Manual jog: Acceleration | 720.0000 | units/s2 |
| C01232/2 | Manual jog: Deceleration | 720.0000 | units/s2 |
| C01233/1 | Manual jog: S-ramp time | 0.000 | s |
| C01235/1 | Waiting time - 2nd speed | 5.000 | s |
| C01234/1 | Manual jog: Breakpoint 1 | 0.0000 | unit |
| C01234/2 | Manual jog: Breakpoint 2 | 0.0000 | unit |
| C01234/3 | Manual jog: Breakpoint 3 | 0.0000 | unit |
| C01234/4 | Manual jog: Breakpoint 4 | 0.0000 | unit |

**Note!**

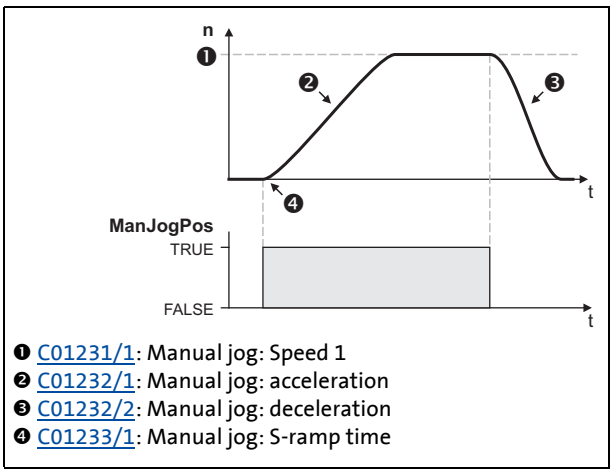
For trouble-free operation, the Machine parameters (at least gearbox factor and feed constant) must also be set correctly!

### 9.7.1.1 Functional settings

In C01230, various functional settings for manual jogging can be carried out in bit-coded form.

| Function | | Lenze setting |
|---|---|---|
| Bit 0 | **Manual jog with breakpoint**<br>During manual jog, a stop is automatically made at the breakpoint positions set in C01234/1...4.<br>• The home position must be known for this function.<br>• Travelling is continued when the pushbutton is "released" and pressed again. This button has been assigned the *ManJogPos* and/or *ManJogNeg* control function. | Off |
| Bit 1 | **Time-based start of second speed**<br>In manual jog mode, an automatic changeover to manual speed 2 takes place after a parameterisable waiting time.<br>▸ Time-based start of second speed (📖 642) | Off |
| Bit 2 | **HW limit switch on**<br>In the "manual jog" operating mode, a travel range monitoring mode via hardware limit switch is active.<br>▸ Limit position monitoring (📖 601) | Off |
| Bit 3 | **SW limit switch on**<br>In the "manual jog" operating mode, a travel range monitoring mode via parameterised software limit positions.<br>▸ Limit position monitoring (📖 601) | Off |
| Bit 4 | Reserved | Off |
| Bit 5 | **Maximum jerk**<br>▸ Setting or activation of maximum jerk for traversing profiles (📖 615) | Off |
| Bit 6 | Reserved | Off |
| Bit 7 | **Position controller off**<br>In the "manual jog" operating mode, the position controller is deactivated. Thus, the compensation of the following error is switched off. | Off |

### 9.7.1.2 Smooth start and quick stop of the drive




❶ C01231/1: Manual jog: Speed 1
❷ C01232/1: Manual jog: acceleration
❸ C01232/2: Manual jog: deceleration
❹ C01233/1: Manual jog: S-ramp time

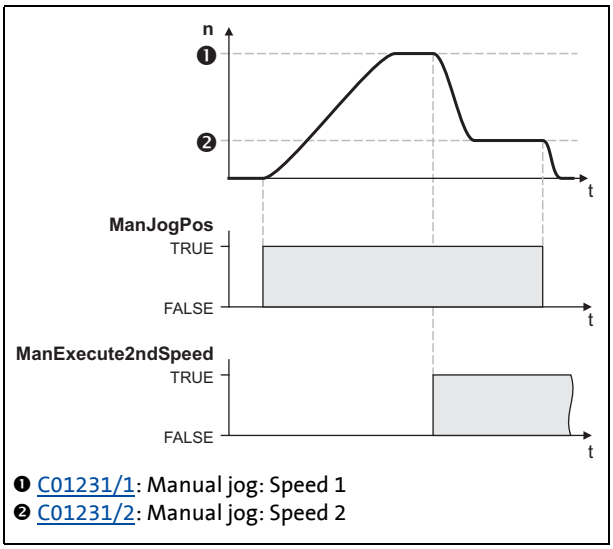
[9-29] Example: Smooth start and quick stop

- For accelerating and decelerating, different values can be set in C01232/1...2 so that smooth starting and quick stopping of the drive can be implemented.
- In order to reduce jerking, the two ramps can be set in such a way that they are s-shaped. This is done by entering a relative S-ramp time in C01233/1.

**Tip!**

Rapid deceleration (C01232/2) reduces the time from releasing of the "jog button" to actual stopping of the drive, with the result that it is easier to position the drive "by sight" and the desired stop position is not passed.

### 9.7.1.3 Second speed




❶ C01231/1: Manual jog: Speed 1
❷ C01231/2: Manual jog: Speed 2

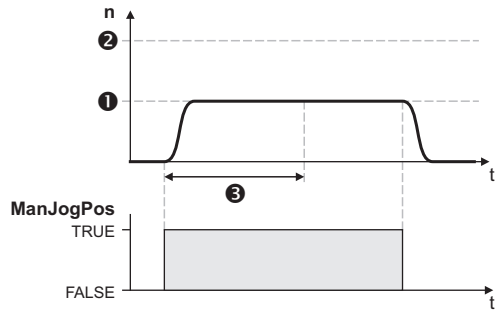
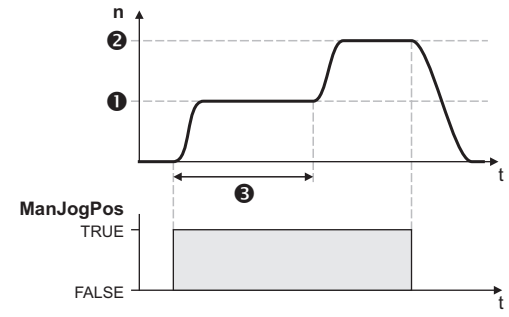
[9-30] Example: Change-over to second speed

- By setting control bit 6 (*ManExecute2ndSpeed*) in theMCK control word, you can changeover to a second speed (C01231/2) during traversing.

### 9.7.1.4 Time-based start of second speed

If this function has been activated by means of bit 1 in C01230 and a waiting time > "0 s" has been set in C01235/1, an automatic change-over to manual jog speed 2 takes place when the manual jog function is activated and after the waiting time has expired.

- When the waiting time is set = 0 s, the automatic change-over is deactivated.

| **Time-based start deactivated** (C01230 - bit 1 = 0; Lenze setting) | **Time-based start activated** (C01230 - bit 1 = 1) |
|---|---|
|  |  |

❶ C01231/1: Manual jog: Speed 1
❷ C01231/2: Manual jog: Speed 2
❸ C01235/1: Waiting time 2nd speed (Lenze setting: 5 s)

[9-31] Time-based start of second speed

**Tip!**

By leaving the corresponding button for manual jog pressed down for longer than the waiting time and parameterising manual jog speed 2 so that it is higher than manual jog speed 1, you can enable longer distances to be travelled by means of manual jogging with this function.

### 9.7.2 Requesting the operating mode

Request for "manual jog" mode via the MCK control word:

| MCK control word | | | | | | |
|---|---|---|---|---|---|---|
| **Bit 31** | **...** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | 0 | 0 | 1 | 0 |
| X = Status not significant | | | | | | |

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

### 9.7.3 Executing manual jogging

Control is exercised by means of bits 4 ... 7 in the MCK control word:

| Manual jog | MCK control word | | | |
|---|---|---|---|---|
| | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** |
| | Release LimitSwitch | ManExecute 2ndSpeed | ManJogNeg | ManJogPos |
| Stop manual jog | 0 | X | 0 | 0 |
| Manual jog CW<br>• With speed 1 (C01231/1) | | 0 | 0 | 1 |
| Manual jog CW<br>• With speed 2 (C01231/2) | | 1 | | |
| Manual jog in counter-clockwise direction<br>• With speed 1 (C01231/1) | | 0 | 1 | 0 |
| Manual jog in counter-clockwise direction<br>• With speed 2 (C01231/2) | | 1 | | |
| No change in relation to previous state | | X | 1 | 1 |
| Retract operated limit switch | 1 | 0 | 0 | 0 |
| X = Status not significant | | | | |

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process inputs for controlling the operating mode:

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bManJogPos<br>bManJogNeg<br>BOOL | bManJogPos = TRUE: Manual jog right<br>bManJogNeg = TRUE: Manual jog left<br>Both inputs = TRUE: No change compared to previous state<br>Both inputs = FALSE: Stop manual jog | |
| bManJogExecute2ndVel<br>BOOL | Change over to speed 2 for manual jog | |
| | FALSE | Speed 1 (C01231/1) active. |
| | TRUE | Speed 2 (C01231/2) active. |
| bReleaseLimitSwitch<br>BOOL | Retract operated limit switch | |
| | TRUE | Retract operated limit switch (in opposite direction) |

**Case 1: Reference known**

If the reference is known and the software limit positions have been set, i.e. at least one software limit position is > 0, manual jogging is carried out until a position at the corresponding software limit position is reached unless manual jogging is aborted first. Overtravelling the set software limit positions is not possible.

**Case 2: Reference not known**

If the reference is not known, the travel range limits are only monitored via the limit switches (if connected).

If you end manual jogging manually by resetting *bManJogNeg* or *bManJogPos*, the drive is brought to a standstill at the rate of deceleration set for manual jogging.

### 9.7.3.1 Manual jog to limit position

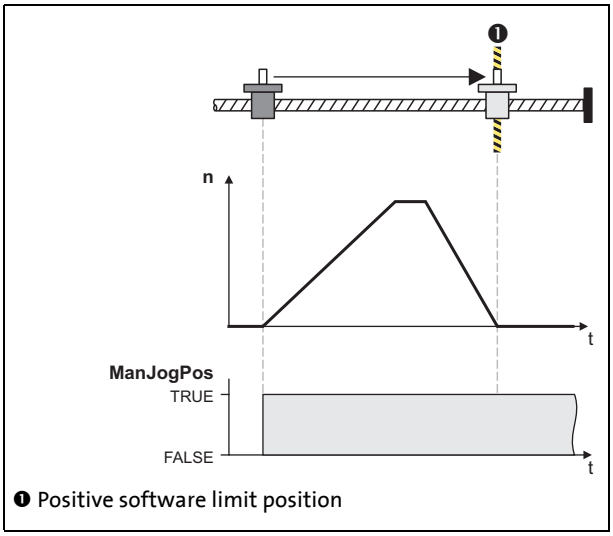
**Stop!**

In the Lenze setting, travel range monitoring is switched-off in C01230 by means of hardware limit switches and software limit positions for the "Manual jog" operating mode!

If travel range monitoring has been deactivated, the drive can travel into a mechanical limit during manual jogging and machine parts can be destroyed or damaged!

▸ Limit position monitoring (📖 601)

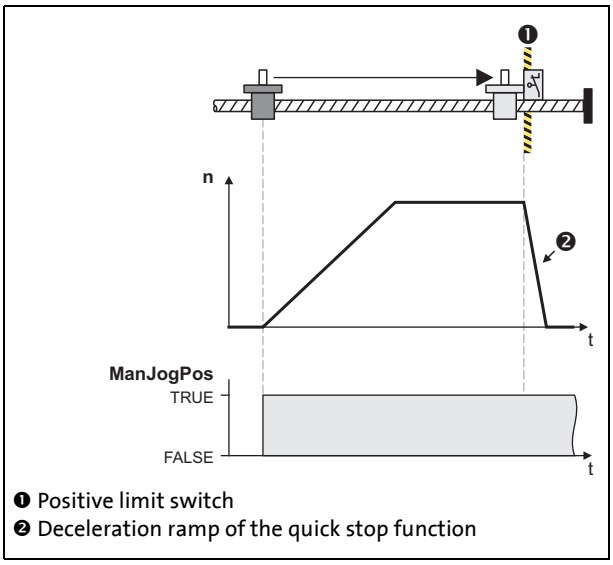
**Manual jog to software limit position**




[9-32] Example: Manual jog to positive software limit position

- If the reference is known and the software limit positions have been set and are active, positioning to the corresponding software limit position is carried out unless you manually stop manual jogging first by resetting control bit 4/5 (*ManJogPos*/*ManJogNeg*) in theMCK control word.
- The drive decelerates at the set rate of deceleration (C01232/2) to the position of the corresponding software limit position.

**Manual jog to hardware limit position (limit switch)**




❶ Positive limit switch
❷ Deceleration ramp of the quick stop function

[9-33] Example: Manual jog to positive limit switch

- When monitoring of the limit switches is switched on and a limit switch is approached during manual jog, the drive is braked within the deceleration time set for the quick stop function if the "TroubleQuickStop" error response is set in C00595/1 or C00595/2.

### 9.7.3.2 Retracting of an operated limit switch

If control bit 7 (*ReleaseLimitSwitch*) in the MCK control word is set, retracting from an operated limit switch is possible. Traversing is carried out in the corresponding retracting direction until the limit switch is not longer operated.

- If a direction preselection is made for retracting by means of control bit 4/5 (*ManJogPos*/*ManJogNeg*) in the retracting direction, travelling is continued, even after the limit switch has been left, until the corresponding control bit is reset.
- If, instead, a direction preselection is made against the retracting direction, the drive remains at a standstill.

**Note!**

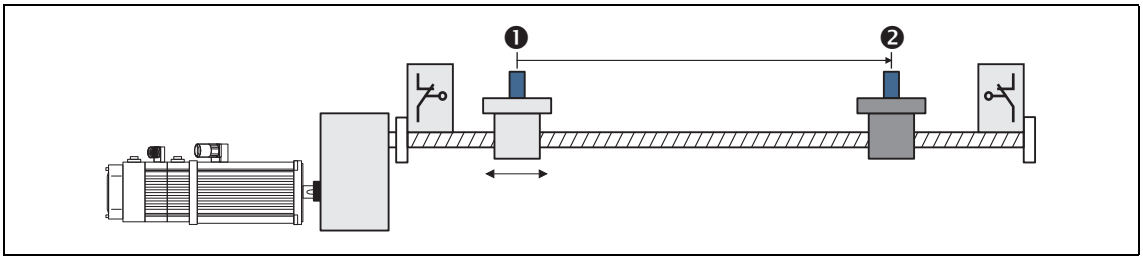
Retracting from a limit switch is only possible if this switch is still in the operated state, i.e. the corresponding limit switch input is still active. You must therefore make sure that, when travelling to a limit switch, its tripping mechanism is not "driven over" due to e.g. an excessively high mass or too much momentum so that the limit switch is no longer in the operated state as a result.

**Tip!**

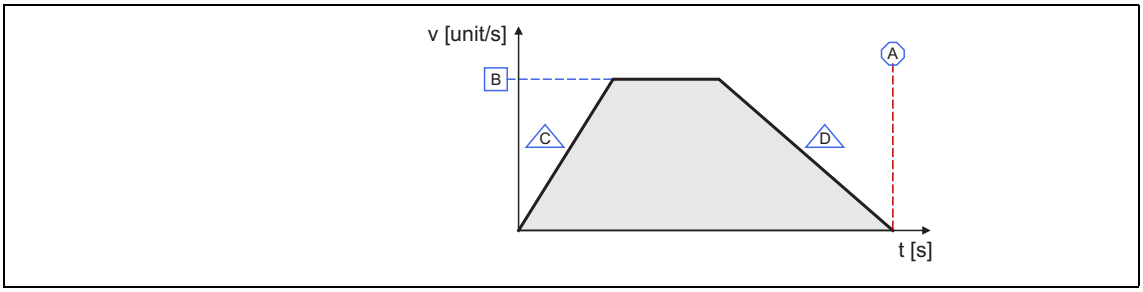
A limit switch that is in the operated state can also be moved away from again through manual jogging in the retracting direction by means of control bits 4/5 (*ManJogPos*/*ManJogNeg*).

## 9.8 Positioning

Positioning means that a workpiece/tool or material is moved from a starting position ❶ to a defined destination ❷:




To carry out positioning, a travel profile has to be stored in the inverter for at least the following profile parameter:

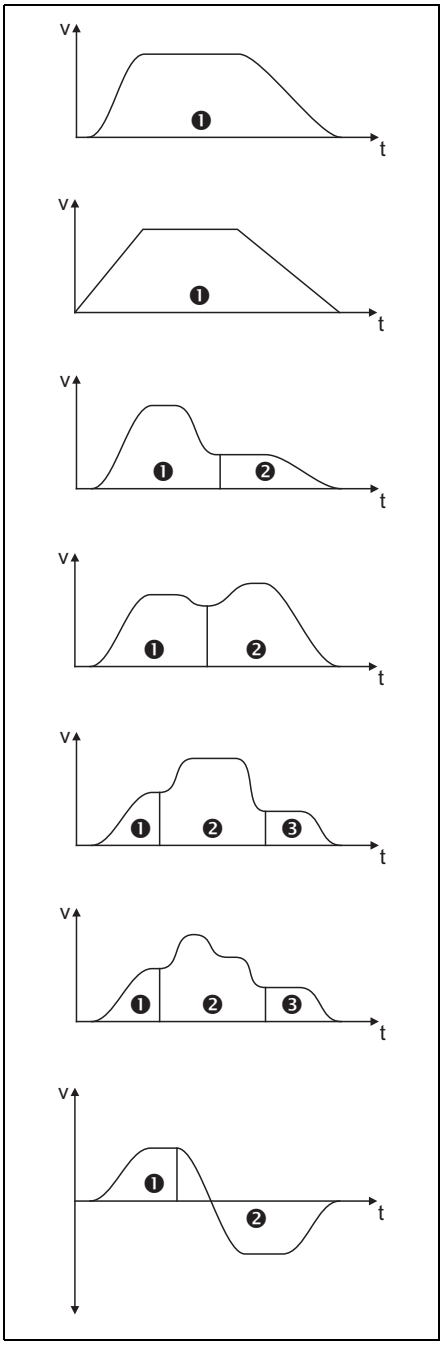



| Symbol | Profile parameters |
|---|---|
| A | **Position**<br>Target position or distance to be traversed. |
| B | **Speed**<br>Maximum velocity during the positioning process. |
| C | **Acceleration**<br>Maximum acceleration during the positioning process. |
| D | **Delay**<br>Maximum deceleration during the positioning process. |

- A profile describes a motion task that can be converted to a rotary motion of the motor shaft by the **Motion Control Kernel** in the "Positioning" operating mode.
- A positioning process can be composed of a large number of profiles that are executed in a fixed manner.
- 15 different profiles can be parameterised for the 8400 TopLine inverter.
- You can find a detailed explanation of all profile parameters in the subchapter entitled "Profile entry". (📖 651)

### 9.8.1 Possible motion profiles

The following illustration shows different movement profiles that can be travelled in the "positioning" mode:




Asymmetrical trapezoidal profile
- with S-shaped ramps

Asymmetrical trapezoidal profile
- with linear ramps

Velocity changeover for profile with S-shaped ramps
- here: $v_{Profile\ 1} > v_{Profile\ 2}$ or $v_{End1} = v_{Profile\ 2}$

Velocity changeover for profile with S-shaped ramps
- here: $v_{Profile\ 1} < v_{Profile\ 2}$

Profile linkage

Profile linkage
- with speed override in profile 2

Profile linkage
- here: forwards/backwards profile

[9-34] Examples of possible motion profiles

### 9.8.2 Parameter setting

**Short overview of parameters for "positioning" mode:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01216 | MCK: Positioning setting | Bit coded | |
| C01300/1...15 | Profile 1 ... 15: Mode | 1: absolute (beeline) | |
| C01301/1...15 | Profiles 1 ... 15: Position | 360.0000 | unit |
| C01302/1...15 | Profile 1 ... 15: Speed | 360.0000 | unit/s |
| C01303/1...15 | Profile 1 ... 15: Accel. | 720.0000 | unit/s2 |
| C01304/1...15 | Profile 1 ... 15: Decel. | 720.0000 | unit/s2 |
| C01305/1...15 | Profile 1 ... 15: Final speed | 0.0000 | unit/s |
| C01306/1...15 | Profiles 1 ... 15: S-ramp time | 0.000 | s |
| C01307/1...15 | Profiles 1 ... 15: Sequence profile | 0 | |
| C01308/1...15 | Profile 1 ... 15: TP profile | 0 | |
| C01309/1...15 | Profile 1 ... 15: TP signal source | 3: TP-DigIn3 | |
| C00595/9 | MCK: Resp. to invalid PosModus | 4: WarningLocked | |
| C00595/10 | MCK: Resp. to invalid profile data | 4: WarningLocked | |
| C00595/12 | MCK: Resp. to invalid profile no. | 4: WarningLocked | |
| C00595/14 | MCK: React target out of travel range | 4: WarningLocked | |
| C01210/1 | MCK: Curr. feed | - | units |
| C01210/2 | MCK: Curr. set position | - | units |
| C01210/3 | MCK: Curr. actual position | - | units |
| C01210/4 | MCK: Curr. following error | - | units |
| C01210/5 | MCK: Positioning accuracy | - | units |
| C01210/6 | MCK: Target position | - | units |
| C01211/1 | Max. traversing speed 100%_C11 | - | units/s |
| C01213/1 | MCK: Max. traversing distance | - | units |
| C01242 | MCK: Current pos profile number | - | |
| Greyed out = display parameter | | | |

**Note!**

For trouble-free operation, the Machine parameters (at least gearbox factor and feed constant) must also be set correctly!

### 9.8.2.1 Functional settings

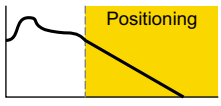
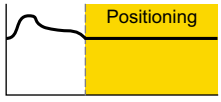
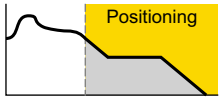
In C01216 , various functional settings regarding behaviour in the case of a changeover to the "positioning" mode can be made in bit-coded form.

- When this is being done, any mode change must be taken into account.
- Positioning with controller inhibit/enable is also possible if the corresponding setting is made.

| Function | | Lenze setting |
|---|---|---|
| Bit 0 | **PosAbort at PosInit**<br>When a changeover to "positioning" mode is made, ramping down at the rate of deceleration set in C01251 for normal stopping is carried out. | On |
| Bit 1 | **PosExecute active at PosInit**<br>When a changeover is made to the "positioning" mode, the specified profile is immediately executed if the "MCK PosExecute" control bit has also been set. If the MCK "PosExecute" control bit has not been set, the setpoint is continued. | Off |
| Bit 2 | Reserved | Off |
| Bit 3 | Reserved | Off |
| Bit 4 | **ProfilStart at PosInit**<br>**When it is changed to the "Positioning" operating mode, the defined profile is executed immediately without the need to set the MCK control bit "PosExecute".**<br>**Note:** If the FB L_MckCtrlInterface writes to the MCK control word due to the setting in C01297 , a profile start may be prevented. Remedy: Set bit 0 in C01297 to "0".<br>▸ Alternative functions for control bit "PosExecute" (📖 588) | Off |
| Bit 5 | **Maximum jerk**<br>▸ Setting or activation of maximum jerk for traversing profiles (📖 615) | Off |
| Bit 6 | Reserved | Off |
| Bit 7 | Reserved | Off |

**Note!**

In the case of multiple selection, the "PosAbort at PosInit" function, which can be activated by means of bit 0, has priority over the other functions (see the following truth table).

| Bit 4<br>ProfilStart | Bit 1<br>PosExecute active | Bit 0<br>PosAbort | MCK control bit "PosExecute" | Behaviour when it is changed to "Positioning" operating mode | |
|---|---|---|---|---|---|
| x | x | 1 | x |  | Ramping down the setpoint |
| **0** | **0** | **0** | x |  | Continuing the setpoint |
| 0 | 1 | 0 | 0 | | |
| 0 | 1 | 0 | 1 |  | Positioning from the setpoint |
| 1 | x | 0 | x* | | |
| printed in bold = Lenze setting; x = any state; * see note on bit 4 in table above | | | | | |

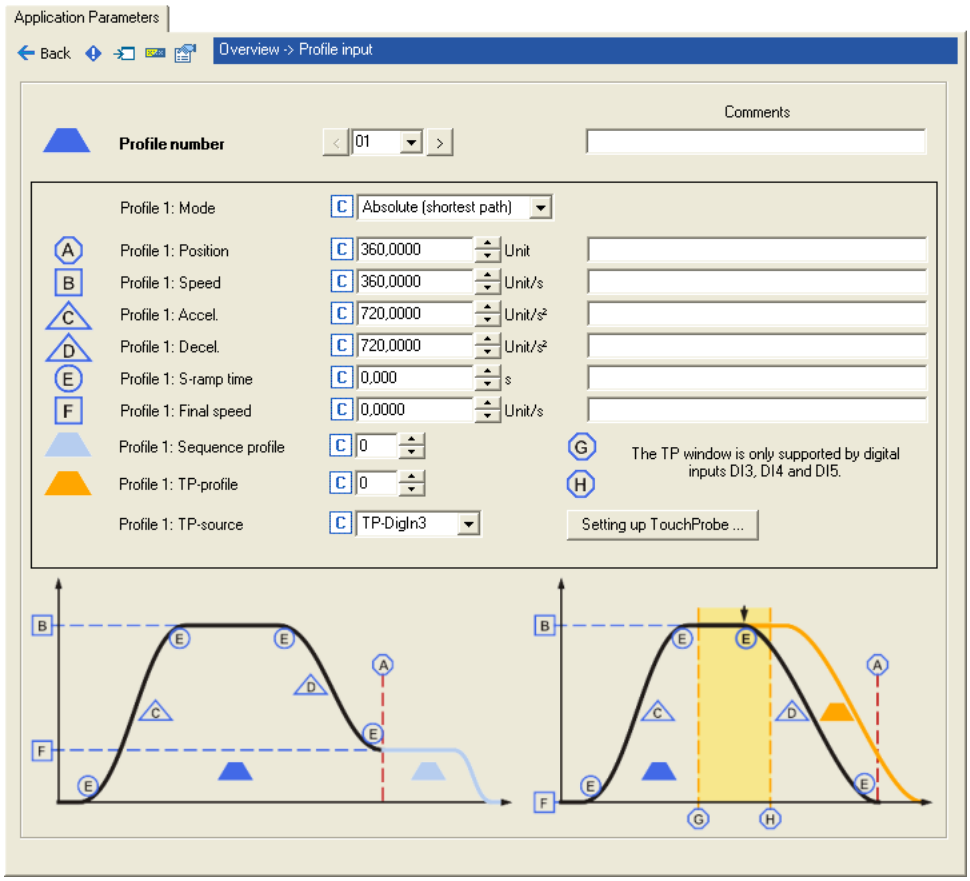
**Related topics:**

- Stipulation of the profile to be executed (📖 661)
- Starting/cancelling a traversing task (📖 662)

### 9.8.2.2 Profile entry

For setting the profile parameters, the »Engineer« provides a parameterisation dialog with easy profile entry via parameters and an extended profile entry (see marking).

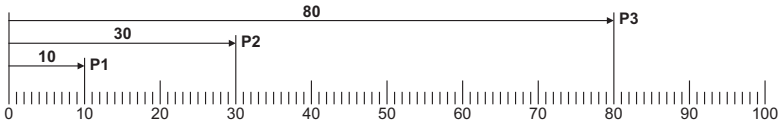
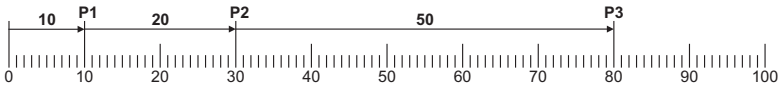
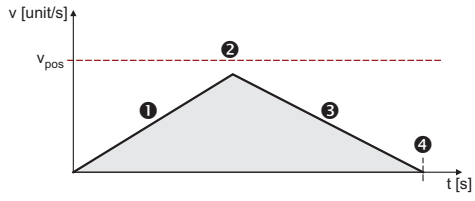
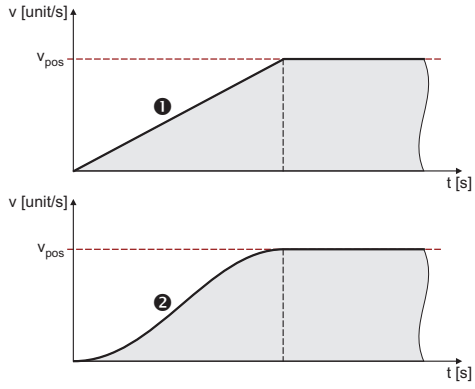
▸ Extended profile entry (📖 654)

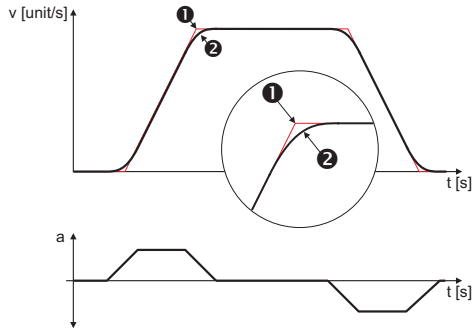
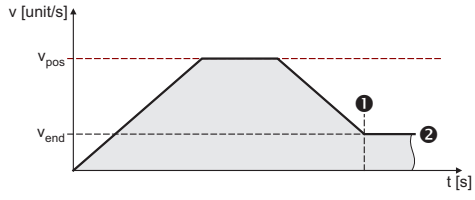



#### Profile entry via parameters

A profile is described by the following profile parameters:

| Symbol | Profile parameters |
|---|---|
| | **(Standard) profile**<br>Profile data set (profile numbers 1 ... 15), in which the profile data are stored. |
| | **Mode** (C01300/1...15)<br>Selection of the way in which positioning is to be carried out. ▸ Positioning modes (📖 657) |

| Symbol | Profile parameters |
|---|---|
| A | **Position** (C01301/1...15)<br>Target position or distance to be traversed.<br>When the position is indicated, a distinction is made between absolute position and relative position:<br>• An absolute position always indicates the distance to the defined zero position:<br>Absolute position = Target position<br>  <br>• A relative position indicates the distance to the starting position (current position):<br>Relative position = Target position - Starting position<br>  |
| B | **Speed** (C01302/1...15)<br>Maximum velocity during the positioning process.<br>• Depending on the profile parameters of position, acceleration and deceleration , it is possible that the drive will not even reach the maximum speed. In this case, the graphic representation will be a trapezium instead of a triangle:<br>  <br>❶ Acceleration<br>❷ Travelling speed (is not reached in this case)<br>❸ Deceleration<br>❹ Target position (or traversing distance) |
| C | **Acceleration** (C01303/1...15)<br>Maximum acceleration during the positioning process.<br>• Two types of acceleration are distinguished:<br>• Constant acceleration: the velocity increases linearly.<br>• Linearly increasing acceleration: Speed increases in S-shape.<br>A linearly increasing acceleration (S-profile) results from the setting of an S-ramp time (see more below).<br>  <br>❶ Constant acceleration (L-profile)<br>❷ Linearly increasing acceleration (S-profile) |
| D | **Deceleration** (C01304/1...15)<br>Maximum deceleration during the positioning process. |

| Symbol | Profile parameters |
|---|---|
| Ⓔ | **S-ramp time** (C01305/1...15)<br>Due to stipulation of an S-ramp time for a profile, the profile is executed with S-shaped ramps, i.e. acceleration and braking processes are initiated smoothly in order to reduce jerk and thus the stress on the drive components.<br>• The acceleration/deceleration stipulated in the profile is not achieved until after the specified S-ramp time.<br>• This kind of acceleration/deceleration is needed for sensitive machine parts with a certain amount of play.<br>• The unavoidable consequence of the slower increase in acceleration in the case of the S profile is that the positioning time is longer compared to the L profile, which is more efficient in terms of time.<br>▸ S-ramp time for jerk limitation (📖 659) |
| |  <br>❶ Without jerk limitation (L profile)<br>❷ With jerk limitation (S profile) |
| Ⓕ | **Final speed** (C01305/1...15)<br>This specifies the velocity at which the drive is to start the next profile after reaching the target position. With a final speed not equal to "0", "velocity changeover" or "overchange" is possible, i.e. when the target position is reached, a second positioning process is started immediately without the drive coming to a standstill at the first target position. |
| |  <br>❶ Target position<br>❷ Final speed (in this case, not equal to "0") |
| | **Sequence profile** (C01307/1...15)<br>**for profile linkage/following block control**<br>A special feature is automatic advancing to subsequent profiles with and without velocity changeover. For this purpose, the profile number of the desired subsequent profile (1 to 15) is simply set in the parameter "Sequence profile" (C01307/1...15) of a profile.<br>After execution of the profile (target position reached), the set following (subsequent) profile is started automatically. In this way, profile chains can be stipulated without additional control processes.<br>• If the profile parameter "Final speed" (C01305/1...15) is set to <> "0", there is a velocity changeover leading into the following (subsequent) profile at the set final speed.<br>• If "0" is set for the following (i.e. subsequent) profile, profile linkage does not take place.<br>• This function can be performed in all positioning modes. |
| | **TP profile** (C01308/1...15)<br>Profile number of the profile (1 ... 15) that is to be executed after a touch probe has been detected.<br>• If "0" is set, there will be no profile stepping through touch probe.<br>• Only relevant for positioning modes with touch-probe.<br>▸ Touch probe positioning (📖 658) |
| | **TP signal source** (C01309/1...15)<br>Selection of the signal source for touch probe detection.<br>• Only relevant for positioning modes with touch-probe.<br>▸ Touch probe positioning (📖 658) |

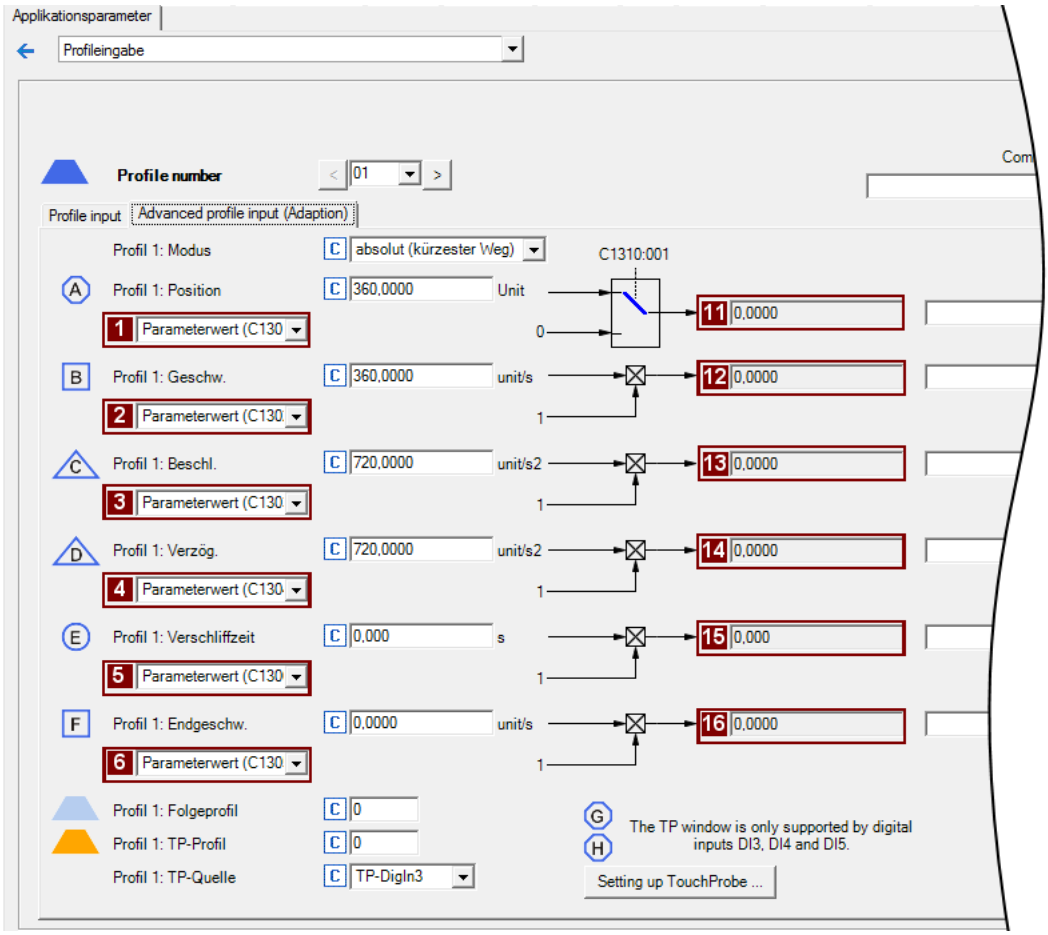
### Extended profile entry

This function extension is available from version 16.00.00!

This function serves to extend the profile entry (1 ... 6) via a process data interface.

The profile data is processed for the profile start in different ways:

- In case of the profile position, the process input value is directly used for the profile start.
- The other profile data (profile speed, acceleration etc.) are evaluated with a percentage factor by means of the selected process input. The resulting value is used for the profile start.

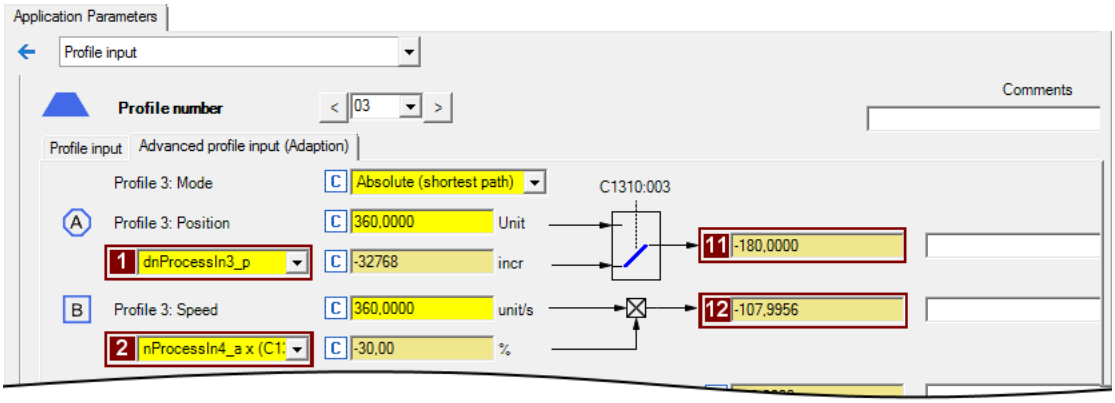


| Parameters x = 1 ... 15 | | Lenze setting Value | Unit | Info |
|---|---|---|---|---|
| 1 | **Profile x: Position** (C01310/x) | 0 | | Selection of the profile position value<br>• 0: C01301/x<br>• 1 ... 4: *dnProcessIn1..4_p* [incr] |
| 2 | **Profile x: Speed** (C01311/x) | 0 | | Selection of the profile speed value<br>• 0: C01302/x<br>• 1 ... 8: *nProcessIn1..8_a* [%] x C01302/x<br>Note:<br>If the adaptation value for a speed value is higher than the speed limitation value C00909/x in the motor control, a following error takes place. |
| 3 | **Profile x: Acceleration** (C01312/x) | 0 | | Selection of the profile acceleration value<br>• 0: C01303/x<br>• 1 ... 8: *nProcessIn1..8_a* [%] x C01303/x |
| 4 | **Profile x: Deceleration** (C01313/x) | 0 | | Selection of the profile deceleration value<br>• 0: C01304/x<br>• 1 ... 8: *nProcessIn1..8_a* [%] x C01304/x |
| 5 | **Profile x: S-ramp time** (C01315/x) | 0 | | Selection of profile S-ramp time value<br>• 0: C01306/x<br>• 1 ... 8: *nProcessIn1..8_a* [%] x C01306/x |
| 6 | **Profile x: Final speed** (C01314/x) | 0 | | Selection of the profile final speed value<br>• 0: C01305/x<br>• 1 ... 8: *nProcessIn1..8_a* [%] x C01305/x<br>Note:<br>If the adaptation value for a speed value is higher than the speed limitation value C00909/x in the motor control, a following error takes place. |
| 11 | **Position** (C01320/x) | | | Display of the effective profile position |
| 12 | **Speed** (C01321/x) | | | Display of the effective profile speed under consideration of the limitation in C01302/x by the maximum value in C01211/1 . |
| 13 | **Acceleration** (C01322/x) | | | Display of the effective profile acceleration |
| 14 | **Delay** (C01323/x) | | | Display of the effective profile deceleration |
| 15 | **S-ramp time** (C01325/x) | | | Display of the effective profile S-ramp time |
| 16 | **Final speed** (C01324/x) | | | Display of the effective final profile speed under consideration of the limitation C01305/x by the maximum value in C01211/1 |

### Example

For the profile data set 3, the profile position and the profile speed are specified. For this purpose, the following values are used for the setting:

| Parameters | | Setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 1 | **Profile 3: Position** (C01310/3) | -32768 | incr | Selection of the profile position at the process input *LS_MotionControlKernel.dnProcessIn3_p* (32 bits) |
| 2 | **Profile 3: Speed** (C01311/3) | -30 | % | Adaptation of the profile speed at the process input *LS_MotionControlKernel.dnProcessIn4_a* (16 bits) |
| 11 | **Position** (C01320/3) | -180 | unit | Display of the resulting profile position **Profile 3** |
| 12 | **Speed** (C01321/3) | -108 | unit/s | Display of the resulting profile speed **Profile 3** |




**Note!**

In order to avoid a following error, the resulting speed ≤ speed limitation value (C00909/x) should be set.

### 9.8.2.3 Positioning modes

For positioning, it is possible to select from different positioning modes in relation to the type of axis/application. These modes are described in the following table.

**Stop!**

In the positioning modes "continuous" and "relative", the "TroubleQuickStop" error response occurs in the Lenze setting when the maximum travel distance is exceeded (C01213/1). ▸ Monitoring of the maximum travel distance (📖 608)

**Note!**

For absolute positioning, the home (reference) position must be known!

- If absolute positioning (positioning modes 1, 4, 5, 8, 11, 12) is started although the home position is not known:
  - The error message set in C00595/8 is displayed (Lenze setting: "WarningLocked").
  - The error message "Ck08: Home position unknown" is entered into the logbook.

| Positioning mode | | Info |
|---|---|---|
| 1 | Absolute (shortest path) | Travelling along the axis takes place until an absolute position is reached via the shortest path.<br>• Reference for the absolute position is the zero position. |
| 8 | Absolute (shortest path) to TP | |
| 2 | Continuous | In these two modes, no particular position is approached but travelling takes place at the traversing speed stipulated by means of the profile.<br>• The values of the profile are used for accelerating and decelerating.<br>• The direction of travel is determined by the sign of the traversing speed. |
| 9 | Continuous to TP | |
| 3 | Relative | The axis is traversed by a distance.<br>• Reference for the distance is the target position of the previously executed profile. |
| 10 | Relative to TP | |
| **Positioning modes for modulo measuring system (rotary table application)**<br>▸ Activation of the modulo measuring system (📖 597) | | |
| 4 | absolute (Cw) | Travelling along the axis takes place clockwise until an absolute position is reached.<br>• Reference for the absolute position is the zero position.<br>• In this direction, the zero position of the axis can be exceeded. |
| 11 | Absolute (Cw) on TP | |
| 5 | absolute (Ccw) | Travelling along the axis takes place counter-clockwise until an absolute position is reached.<br>• Reference for the absolute position is the zero position.<br>• In this direction, the zero position of the axis can be exceeded. |
| 12 | Absolute (Ccw) on TP | |

### 9.8.2.4 Touch probe positioning

In case of a touch probe positioning, first the profile is executed according to the set profile parameters. If a touch probe is detected during the process, it is automatically changed to the profile defined in the "TP profile" profile parameter. This profile mode is also defined via the mode selection like in the following profile start processes.

If no valid TP profile is set, the travel request is aborted ("PosStop").

**Preconditions for touch probe positioning**

- The mode of the current profile data set contains the setting "to TP".
- For absolute positioning, the home (reference) position must be known.
- In the corresponding profile data set, the following profile parameters must be set in addition:
  - TP profile (C1308/x)
  - TP signal source (C1309/x)
- The touch probe interface is configured such that the selected TP signal source is enabled.
  - See chapter entitled "Touch probe detection". (📖 419)



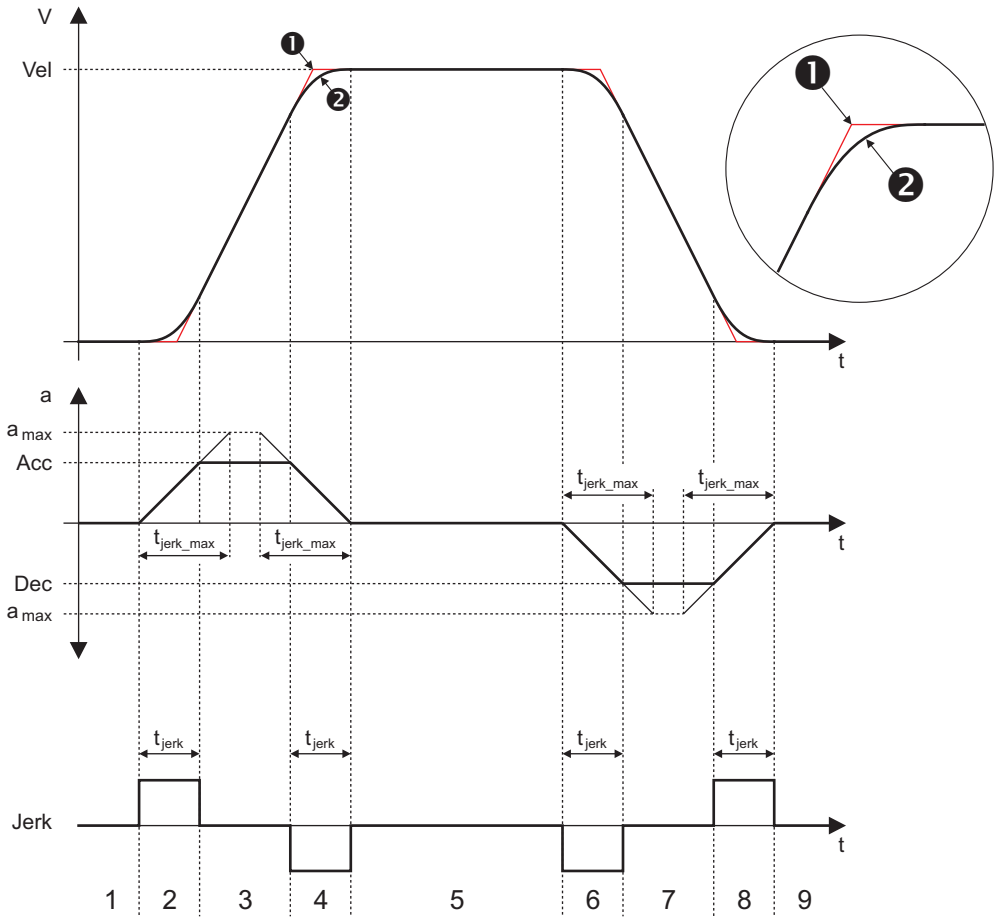
**Note!**

Go to C02810/x and select the edge, the digital input used for the connection of the touch-probe sensor is to respond. In the Lenze setting of C02810/x, no touch probe is detected!

### 9.8.2.5 S-ramp time for jerk limitation

The maximum jerk is defined by the selection of an S-ramp time $t_{jerk_max}$, after which the max. acceleration ($a_{max}$) only is reached.

- The actual jerk time $t_{jerk}$ is reduced according to the actual acceleration Acc:




❶ Without jerk limitation
❷ With jerk limitation

1. Standstill
2nd Acceleration with set jerk limitation
3. Acceleration according to acceleration profile (Acc)
4. Reduce acceleration (jerk limitation)
5th Traversing with Vel according to speed profile
6. Deceleration with set jerk limitation
6. Deceleration according to deceleration profile (Dec)
7. Reduce deceleration (jerk limitation)
8. Standstill (position target reached)

**Note!**

If disproportionately high S-ramp times are set for low acceleration times, this may result in an incorrect profile generation.

Example: v = 100 mm/s, a = 1000 mm/s$^2$

→ $t_{acc}$ = 0.1 s

→ $t_{jerk_max}$ = 1 s

For this reason, you must only set plausible S-ramp times (this should not exceed half the value of $t_{acc}$).

### 9.8.3 Requesting the operating mode

Request for "positioning" mode by means of the MCK control word:

| MCK control word | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 31 | ... | | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | X | 0 | 0 | 1 | 1 |
| X = Status not significant | | | | | | | |

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

### 9.8.4 Carrying out positioning

#### 9.8.4.1 Stipulation of the profile to be executed

The profile to be executed is stipulated by means of the MCK control word in bit-coded form:

| Bit | Designation | Description | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 24 ... 31 | ProfileNo_Bit0 ... ProfileNo_Bit7 | **Profile** | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Profile 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Profile 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | Profile 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | | ... | ... | | | | | | | |
| | | Profile 15 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | | All other possible settings are reserved for future extensions! | | | | | | | | |

If the **Motion Control Kernel** is downstream from the **MCKInterface**, the L_MckCtrlInterface FB provides the following processing inputs for stipulating the profile to be executed:

| Designator<br>Data type | Information/possible settings |
|---|---|
| wProfileNo<br>WORD | Selection of the profile number<br>• Optionally as a data word or binary coded.<br>• In the Lenze setting, an operating mode change is carried out by the L_MckCtrlInterface FB at the same time as the profile is stipulated:<br>  • If profile 0 is selected: Activation of "Speed follower" operating mode<br>  • If profile 1 is selected: Activation of "Homing" operating mode<br>  • If profile 2 is selected: Activation of "Manual jog" operating mode<br>  • If profile 3 ... 15 is selected: Activation of "Positioning" operating mode<br>• The operating mode change with profile number can be set in C01298/1...4. |
| bProfileNo_1<br>...<br>bProfileNo_8<br>BOOL | |

**Note!**

The profile 0 is no valid profile for the "positioning" operating mode.

If a driving request with an invalid profile number is started, the response set in C00595/12 occurs (Lenze setting: "WarningLocked").

### 9.8.4.2 Starting/cancelling a traversing task

Control is exercised by means of bits 16 ... 19 in the MCK control word:

| Homing | MCK control word | | | |
|---|---|---|---|---|
| | Bit 19 | Bit 18 | Bit 17 | Bit 16 |
| | PosStop | Pos DisableFollowProfile | PosFinishTarget | PosExecute |
| Start travelling | 0 | 0 | 0 | 0↗1 |
| Complete interrupted profile | 0 | 0 | 0↗1 | 0 |
| Do not travel sequence profile | 0 | 1 | X | X |
| Cancel travelling* | 1 | X | X | X |
| *From version 02.00.00, more travel requests will be inhibited ("PosExecute" will be blocked). | | | | |
| X = Status not significant | | | | |

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process inputs for controlling the operating mode:

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bPosExecute<br>BOOL | Start travelling | |
| | FALSE↗TRUE | Execute selected profile |
| bPosExecuteFinish<br>BOOL | Complete interrupted profile | |
| | FALSE↗TRUE | A positioning process previously cancelled, e.g. by *bPosStop* or due to a device error, is resumed by travelling to the original target.<br>From version 14.00.00, *bPosExecuteFinish* serves to permanently retravel to the original target even if the profile has been completed once or several times. |
| bPosDisableFollowProfile<br>BOOL | Do not execute following profile (switch off profile linkage) | |
| | TRUE | Evaluation of the sequence profile number parameterised in C01307/1...15 for the selected profile is suppressed. |
| bPosStop<br>BOOL | Cancel travelling | |
| | TRUE | Stop positioning<br>From version 02.00.00, more travel requests will be inhibited ("PosExecute" will be blocked). |

**Tip!**

Travel requests/profiles can also be started while the drive is running. The drive does not need to be at standstill.

**Related topics:**

- Target position monitoring (status "drive in target") (📖 606)
- Monitoring of the maximum travel distance (📖 608)
- Following error monitoring system (📖 609)

### 9.8.4.3 Override of the parameterised positioning mode

The setting of the positioning mode in MCK control word superimposes the mode selection via the "Mode" profile parameter (C1300/1...15). This means that for selected profiles the mode selection and the request of a touch probe positioning via process data are possible. The value set in C01300/1...15 is not overwritten. The following applies to the override:

A. Valid positioning mode in the MCK control word:
The positioning mode set in the MCK control word is used.

B. Positioning mode in the MCK control word = 0:
The positioning mode set in "mode" profile parameter (C1300/1...15) is used.

C. Invalid positioning mode in the MCK control word:
Ck09Error message "Ck09: Invalid positioning mode". The response set in C00595/9 occurs (Lenze setting: "WarningLocked").

| Bit | Designation | Description | | | | |
|---|---|---|---|---|---|---|
| 20 ... 23 | PosModeBit0 ... PosModeBit3 | **Positioning mode** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Positioning mode = setting in C01300/1...15 | 0 | 0 | 0 | 0 |
| | | Absolute (shortest path) | 0 | 0 | 0 | 1 |
| | | Continuous | 0 | 0 | 1 | 0 |
| | | Relative | 0 | 0 | 1 | 1 |
| | | absolute (Cw) | 0 | 1 | 0 | 0 |
| | | absolute (Ccw) | 0 | 1 | 0 | 1 |
| | | Absolute (shortest path) to TP | 1 | 0 | 0 | 0 |
| | | Continuous to TP | 1 | 0 | 0 | 1 |
| | | Relative to TP | 1 | 0 | 1 | 0 |
| | | Absolute (Cw) on TP | 1 | 0 | 1 | 1 |
| | | Absolute (Ccw) on TP | 1 | 1 | 0 | 0 |
| | | All other possible settings are reserved for future extensions! | | | | |

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process input for stipulating an override of the positioning mode:

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wPosMode<br>WORD | Override of the positioning mode set in the profile data<br>• Only bit 0 ... bit 3 of *wPosMode* are evaluated. | |
| | 0 | Positioning mode = setting in C01300/1...15 |
| | 1 | Absolute (shortest path) |
| | 2 | Continuous |
| | 3 | Relative |
| | 4 | absolute (Cw) |
| | 5 | absolute (Ccw) |
| | 8 | Absolute (shortest path) to TP |
| | 9 | Continuous to TP |
| | 10 | Relative to TP |
| | 11 | Absolute (Cw) on TP |
| | 12 | Absolute (Ccw) on TP |
| | All other possible settings are reserved for future extensions! | |

### 9.8.4.4 Position teaching

The MCK control word can be used to initiate "teaching" or "latching" of the MCK setpoint position or the current position into the currently selected profile.

| Bit | Designation | Description |
|---|---|---|
| 14 | PosTeachSetPos | "1" ≡ Teach MCK set position into the selected profile |
| 15 | PosTeachActPos | "1" ≡ Teach current position into the selected profile |

If the **Motion Control Kernel** is downstream from the **MCKInterface**, the L_MckCtrlInterface FB provides the following process inputs for teaching:

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bPosTeachSetPos<br>BOOL | Teach MCK setpoint position | |
| | TRUE | Teach MCK setpoint position into the selected profile. |
| bPosTeachActPos<br>BOOL | Teach current position | |
| | TRUE | Teach current position into the selected profile. |

**Note!**

If request is made at the same time, the current position is taught.

## 9.9 Stop

If the "normal stop" operating mode is active, the drive is brought to a standstill with a parameterisable deceleration ramp.

### 9.9.1 Parameter setting

**Short overview of parameters for "Normal stop" operating mode:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01251/1 | Normal stop: Deceleration | 720.0000 | units/s2 |
| C01252/1 | Stop: S-ramp time | 0.000 | s |

### 9.9.2 Requesting the operating mode

Request for "Normal stop" operating mode by means of the MCK control word:

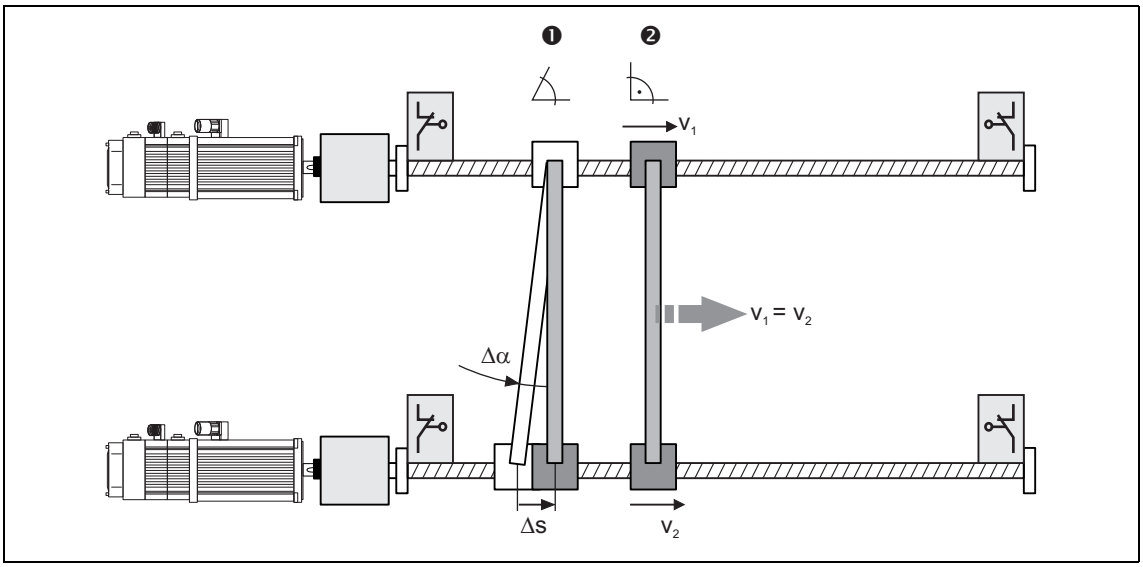
| MCK control word | | | | | | |
|---|---|---|---|---|---|---|
| Bit 31 | ... | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | 0 | 1 | 0 | 0 |
| X = Status not significant | | | | | | |

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

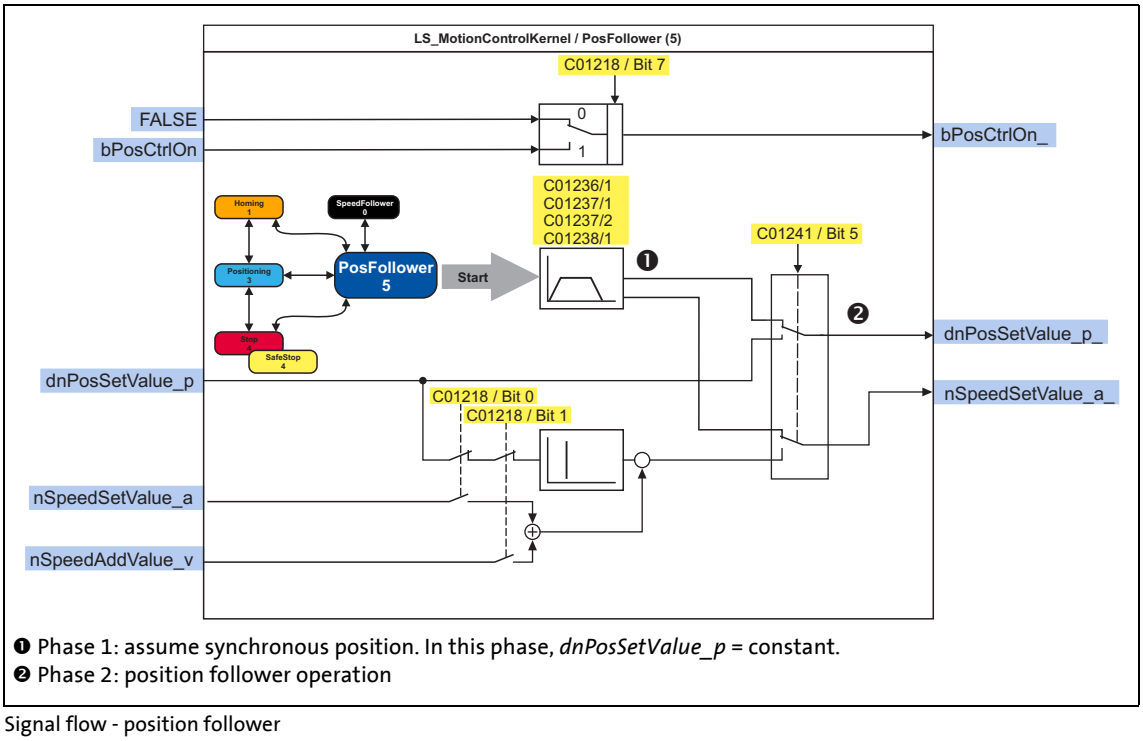
## 9.10 Position follower

In the "Position follower" operating mode, first one of the two drives covers a distance Δs (or an angle Δα) to the synchronous position ❶.

At the time ❷, both drives follow a position setpoint with synchronous speed:




### Signal flow




❶ Phase 1: assume synchronous position. In this phase, *dnPosSetValue_p* = constant.
❷ Phase 2: position follower operation

[9-35] Signal flow - position follower

### 9.10.1 Parameter setting

**Short overview of parameters for "position follower" operating mode:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01218 | MCK: PosFollower setting | Bit coded | |
| C01236/1 | PosFollower: Sync. speed | 360.0000 | units/s |
| C01237/1 | Pos follower: Sync. accel. | 720.0000 | units/s2 |
| C01237/2 | Pos follower: Sync. decel. | 720.0000 | units/s2 |
| C01238/1 | Pos follower: Sync. S-ramp time | 0.000 | s |

#### 9.10.1.1 Functional settings

In C01218, various functional settings for the position follower can be made in bit-coded form.

| Function | | Lenze setting |
|---|---|---|
| Bit 0 | Speed FF control.: nSpeedSetValue_a | Off |
| Bit 1 | Speed FF control: nSpeedAddValue_v | Off |
| Bit 2 | HW limit switch on | On |
| Bit 3 | SW limit switch on | On |
| Bit 4 | Reserved | Off |
| Bit 5 | Reserved | Off |
| Bit 6 | Reserved | Off |
| Bit 7 | Position controller off | Off |

**Speed feedforward control by means of nSpeedSetValue_a**

If this function has been activated by means of bit 0 in C01218, the main setpoint *nSpeedSetValue_a* is used as the speed feedforward control value.

**Speed feedforward control by means of nSpeedAddValue_v**

If this function has been activated by means of bit 1 in C01218, the additive speed value *nSpeedAddValue_v* is used as the speed feedforward control value.

**HW limit switch on**

If this function has been activated by means of bit 2 in C01218, travel range monitoring by means of hardware limit switches is active in this mode.

▸ Limit position monitoring (📖 601)

**SW limit switch on**

If this function has been activated by means of bit 3 in C01218, travel range monitoring by means of parameterised software limit positions is active in this mode.

▸ Limit position monitoring (📖 601)

**Position controller off**

If this function is activated via bit 7 in C01218, the position controller is deactivated in this operating mode. Hence, following error control is switched-off.

### 9.10.2 Requesting the operating mode

Requesting the "position follower" operating mode by means of the MCK control word:

| MCK control word | | | | | | |
|---|---|---|---|---|---|---|
| **Bit 31** | **...** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | 0 | 1 | 0 | 1 |
| X = Status not significant | | | | | | |

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

### 9.10.3 Setpoint selection

The absolute position setpoint is stipulated via the *dnPosSetValue_p* process input.

**Note!**

- In order to prevent a jump between two phases, the position setpoint at the *dnPosSetValue_p* process input (see Signal flow - position follower (📖 666)) must not change while taking the synchronous position.
- The *bPosCtrlOn* process input must be set to TRUE so that position/angle control is active within motor control.

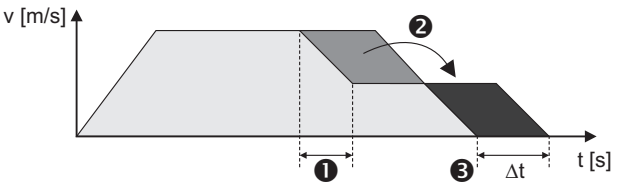
#### Compensating following errors

If the *bDeltaPosOn* process input is set to TRUE, a position difference (following error) is compensated for which is not internally generated from setpoint/actual positions but must be defined via the *dnDeltaPos_p* process input instead.

## 9.11 Override

An "Override" is the change of profile parameters and their acceptance during the positioning process.

- In this case the traversing profile has to be adapted accordingly, so that positioning is carried out exactly to the target position defined, even if for example a change in speed occurs during the positioning process ("Speed override"):




❶ Speed is reduced during the positioning process.
❷ In order to reach the position defined, the missing area has to be "attached" to the profile.
❸ Due to the speed reduction the positioning process takes longer (Δt).

[9-36] Override (here: speed override)

- The override for speed, acceleration, and S-ramp time affects all motion profiles that are controlled by the internal profile generator:
  - Manual jog
  - Retracting from limit switches
  - Homing
  - Point-to-point positioning

### Note!

The online change of speed and acceleration is in effect from the start of the profile until the deceleration phase begins. Changing the deceleration phase by means of an override is therefore not possible!

- In the case of an override value of 0 % for the speed, the drive is brought to a standstill.
- Set override value for the speed ≤ speed limitation value (C00909/x). Otherwise a following error will be generated.
- In the case of an override value of 0 % for the acceleration, acceleration does not take place any longer.

The acceleration override also has the same effect on the deceleration ramp to the same extent specified, but only until the deceleration phase is initiated.

The override has no impact on:

- The "speed follower" operating mode
- Synchronising processes
- Setpoint controls via external setpoints
- Abort via input *bPosStop*
- Speed setpoint selection in case of error (e. g. "Fail-QSP")

### 9.11.1 Speed override

#### Activate override

If control bit 11 (*EnableSpeedOverride*) has been set to "1" in the MCK control word, a speed override is carried out in accordance with the stipulated override value.

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface provides the following process input for activating the speed override:

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bEnableVelOverride<br>BOOL | Speed override | |
| | TRUE | Activate speed override |

#### Stipulate override value

The override value is selected via the *nSpeedOverride_a* input at the SB LS_MotionControlKernel. The override value is a percentage of the speed of the current profile.

| Designator<br>Data type | Information/possible settings |
|---|---|
| nSpeedOverride_a<br>INT | Value for speed override<br>• Percentage multiplier (0 ... 199.99 %) for the currently active speed.<br>• 16384 ≡ 100 % of the maximum traversing speed (display in C01211/1).<br>• If the override value is 0 %, the drive is brought to a standstill. |

#### Deactivate override

If control bit 11 (*EnableSpeedOverride*) is reset to "0", travelling takes place at the speeds that have been specified by means of the profile parameters. Acceleration/deceleration from the speed defined with override to the speed set in the profile takes places immediately in this case.

Deactivating the override within the braking/deceleration ramp does not have any effect.

**Note!**

If the value of a speed override is greater than the speed limitation value C0909/x in the motor control, a following error is generated. C0909/x should always be parameterised greater than or equalling the max. possible speed override value.

### 9.11.2 Acceleration override

#### Activate override

If control bit 12 (*EnableAccOverride*) in the MCK control word has been set to "1", an acceleration override takes place in accordance with the stipulated override value.

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process input for activating the acceleration override:

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bEnableAccOverride<br>BOOL | Acceleration override | |
| | TRUE | Activate acceleration override |

#### Stipulate override value

The override value is stipulated via the *nAccOverride_a* input at the LS_MotionControlKernel SB. The override value is a percentage of the maximum acceleration that has been set for the respective operating mode (referencing, manual jog, positioning etc.) in the corresponding profile parameter.

The acceleration override results in a deceleration override. Both ramps are evaluated by multiplication by the acceleration override until the deceleration process sets in.

| Designator<br>Data type | Information/possible settings |
|---|---|
| nAccOverride_a<br>INT | Value for acceleration override<br>• Percentage multiplier (0 ... 199.99 %) for the currently active acceleration.<br>• 16384 ≡ 100 % of the parameterised acceleration of the corresponding operating mode.<br>• If the override value is 0 %, acceleration ceases. |

#### Deactivate override

If control bit 12 (*EnableAccOverride*) is reset to "0", travelling takes place at the acceleration rates specified by means of the profile parameters. "Ramping up" from the acceleration defined with override to the acceleration set in the profile takes place immediately in this case.

### 9.11.3 S-ramp smoothing override

#### Activate override

If control bit 13 (*EnableSRampOverride*) in the MCK control word is set to "1", S-ramp smoothing override is executed according to the selected override value.

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process input for activating the S-ramp smoothing override:

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bEnableSRampOverride<br>BOOL | S-ramp smoothing override | |
| | TRUE | Activate S-ramp smoothing override |

#### Stipulate override value

Optional stipulation of the override value is carried out via the *nSRampOverride_a* input at the LS_MotionControlKernel SB. The override value is a percentage of the S-ramp time set in the profile data.

**Note!**

If the *nSRampOverride_a* input remains unconnected or if an override value of "0 %" is selected, activation of the S-ramp override causes deactivation of the S-ramp time.

- Deactivation of the S-ramp time before the start of a profile with S-ramp time causes linear ramp generation.
- Deactivation of the S-ramp time during a traversing process, however, is not accepted immediately in the profile generator, but the profile generator checks automatically when an online change of the ramp form can be carried out and then initiates it automatically.

**Tip!**

Thus it is possible to start a travel profile with S-ramp time and then deactivate the S-ramp time, e.g. to traverse with a linear characteristic after reaching the profile speed.

| Designator<br>Data type | Information/possible settings |
|---|---|
| nSRampOverride_a<br>INT | Value for S-ramp smoothing override<br>• Percentage multiplier (0 ... 100 %) for the currently active acceleration.<br>• 16384 ≡ 100 % of the parameterised S-ramp time (C01306/1...15).<br>• Values > 16384 are ignored. |

#### Deactivate override

If control bit 13 (*EnableSRampOverride*) is reset to "0", the S-ramp time specified by means of the profile parameters is used for profile generation.

## 9.12 Holding brake control

This basic function is used for low-wear control of a holding brake.

### Danger!

Please note that the holding brake is an important element of the safety concept of the machine as a whole.

Thus, proceed very carefully when commissioning this system part!

### Stop!

Holding brakes on Lenze motors are not intended for braking during operation. The increased wear caused by braking during operation can destroy the motor holding brake!

### Note!

- **Deactivate automatic DC-injection braking when a holding brake is used!**
  - For this purpose, go to C00019 and set the Auto DCB threshold to "0".
  - Background: Controller inhibit is already activated by the holding brake control.
- If an electrically holding (self-releasing) brake is to be controlled instead of an electrically released (self-holding) brake, the trigger signal must be inverted!
  ▸ Functional settings (📖 679)
- Detailed information on mounting and electrical installation of the motor holding brake can be found in the documentation on the motor holding brake.

#### Intended use

Motor holding brakes are used to lock axes if the controller is inhibited or in case of "mains off" system status. This is not only important for vertical axes but also for e.g. horizontal axes which may cause various problems if the motion is not controlled.

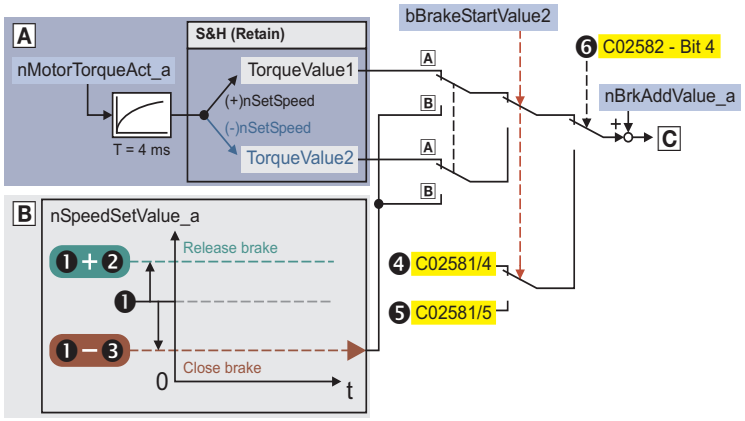
Examples:

- Loss of the reference information after mains OFF and further spinning of the drive.
- Collision with other moving machine parts.

### 9.12.1 Internal interfaces

In the function block editor, the LS_MotionControlKernel system block provides the following internal interfaces for the basic function "holding brake control":

#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bMBrakeRelease<br>BOOL | Releasing/applying the brake in connection with the selected operating mode |
| | FALSE: Apply brake.<br>• During automatic operation, the internal brake logic controls the brake. |
| | TRUE: Release brake manually (forced release).<br>• **Note!** The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If a controller inhibit has been set by the brake control, it will be deactivated.<br>• In semi-automatic operation, the brake is released including feedforward control. |
| bMBrakeStartValue2<br>BOOL | Selection of the torque feedforward control value<br>▸ Feedforward control of the motor before release |
| | FALSE: Starting value 1 is active (see the following figure). |
| | TRUE: Starting value 2 is active (see the following figure). |
| | Creation of the feedforward control value for the release process of the brake:<br>A Signal path for motor control with feedback<br>B Signal path for motor control without feedback<br>C Feedforward control value<br>  <br><br>❶ C02581/1: Switching threshold<br>❷ C02581/2: Hysteresis for release<br>❸ C02581/3: Hysteresis for application<br>❹ C02581/4: Starting value 1 for manual feedforward control<br>❺ C02581/5: Starting value 2 for manual feedforward control<br>❻ C02582/Bit 4: Selection of the feedforward control value (automatic/manual selection) |
| nMBrakeAddValue_a<br>INT | Additive feedforward control value (speed or torque) in [%] for torque feedforward control when the respective control mode is started<br>• For speed control: 100 % ≡ reference speed (C00011)<br>• For torque control: 100 % ≡ maximum torque (C00057)<br>▸ Feedforward control of the motor before release |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bMBrakeApplied<br>BOOL | Input for status detection via switching contacts at the brake<br>• Only effective if bit 5 in C02582 is set to "1". | |
| | FALSE | Brake is released. |
| | TRUE | Brake is applied. |

**Outputs**

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bMBrakeReleaseOut<br>BOOL | Trigger signal for switching element holding brake control via a digital output<br>• Use bit 0 under C02582 to activate inverted switching element triggering.<br>▸ Functional settings | |
| | FALSE | Apply brake. |
| | TRUE | Release brake. |
| bMBrakeReleased<br>BOOL | "Brake released" status signal considering the brake release time<br>• When the holding brake is triggered to close, *bMBrakeReleased* is immediately set to FALSE even if the brake closing time has not yet elapsed! | |
| | TRUE | Brake released (after the brake release time has expired). |




## Stop!

The digital outputs are not suitable for the "direct" control of a holding brake!

- Connect the digital output connected to the trigger signal *bMBrakeReleaseOut* with a relay or power contactor which switches the brake supply.
- When a power contactor is used, the response and release time of the earth contact is added to the response and release time of the brake. Both times must also be considered for parameterising the closing and opening time of the holding brake



**Tip!**

For direct triggering of a holding brake, the high current output of the 8400 TopLine inverter can be used that can switch max. 2.5 A. For this purpose, connect the *bMBrakeReleaseOut* trigger signal to the *bOutHC* input of the SB LS_DigitalOutput.

From version 15.00.00, the *bOutHC_BrakeApplied* status signal of the SB LS_DigitalInput can be used for a simple monitoring of the switching status of a holding brake connected to the high current output. ▸ Switching status of the motor holding brake at the high current output (📖 405)

### 9.12.2 Parameter setting

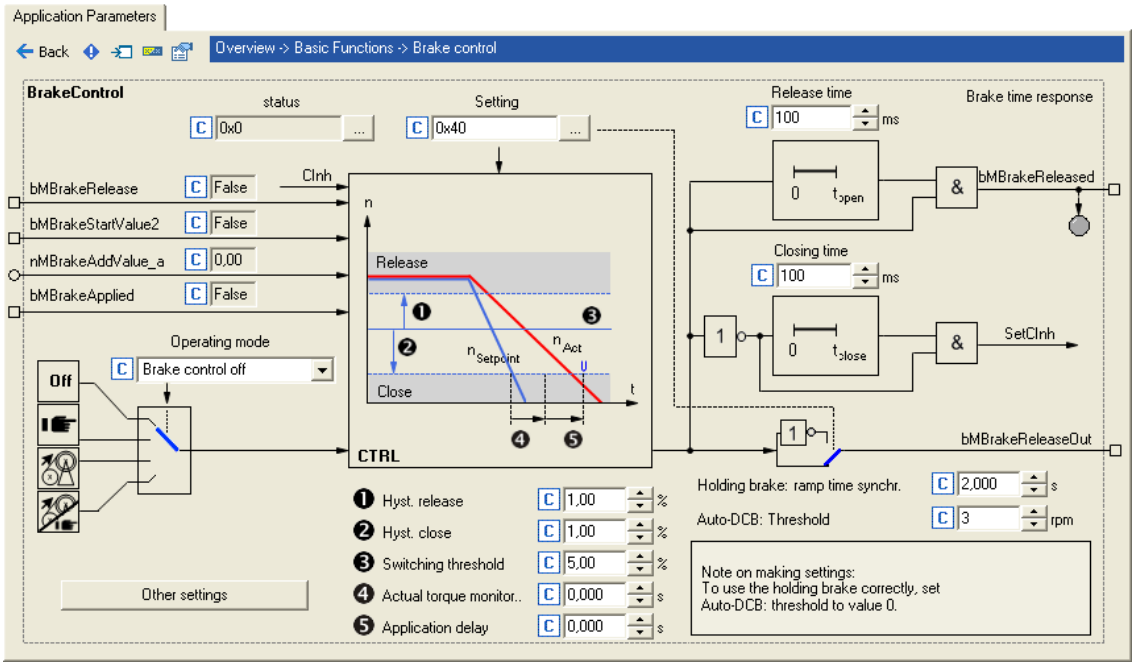
**Danger!**

A faultless holding brake control function requires a correct setting of the different deceleration times in the following parameters!

A wrong setting of the delay times can cause a faulty control of the brake!

**How to go to the parameterisation dialog of the holding brake control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Go to the *Overview* dialog level and click the "**Basic functions**" button.
4. Go to the *Overview* → *Basic functions* dialog box and click the **Holding brake control** button.




**Short overview of parameters for holding brake control:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02580 | Holding brake: Operating mode | 0: Brake control off | |
| C02581/1 | Holding brake: Switching threshold | 5.00 | % |
| C02581/2 | Holding brake: Hyst. release | 1.00 | % |
| C02581/3 | Holding brake: Hyst. close | 1.00 | % |
| C02581/4 | Holding brake: FF control starting value 1 | 0 | % |
| C02581/5 | Holding brake: FF control starting value 2 | 0 | % |
| C02582 | Holding brake: Setting | 0 | |
| Greyed out = display parameter | | | |

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02589/1 | Holding brake: Closing time | 100 | ms |
| C02589/2 | Holding brake: Release time | 100 | ms |
| C02589/3 | Holding brake: Waiting time status | 100 | ms |
| C02589/4 | Holding brake: Ramp FF control | 0 | ms |
| C02593/1 | Holding brake: Actual value monitoring | 0.000 | ms |
| C02593/2 | Holding brake: Application delay | 0.000 | ms |
| C02610/1 | MCK: Holding brake ramp time synchr. | 2.000 | s |
| C02607 | Holding brake: Status | - | |
| C00830/68 | MCK: nMBrakeAddValue_a | - | % |
| C00833/80 | MCK: bMBrakeRelease | - | |
| C00833/81 | MCK: bMBrakeStartValue2 | - | |
| C00833/82 | MCK: bMBrakeApplied | - | |
| Greyed out = display parameter | | | |

### 9.12.2.1 Operating mode

For different applications and tasks, different operating modes are available in C02580. The selected operating mode determines whether the holding brake control is used and how the holding brake will be switched.

#### Mode 0: Brake control off

In this mode, brake control is switched off (not active).

- The *bMBrakeReleaseOut* trigger signal for the holding brake control switching element is set to FALSE.
- The *bMBrakeReleased* status signal is set to FALSE.

**Note!**

In the Lenze setting, the mode 0 is preset to get into a safe state after the mains is switched on.

#### Mode 11: Manual control

In this mode, brake release and brake application can be directly controlled via the *bMBrakeRelease* input without special logic or automatic.

- Setting pulse inhibit or controller inhibit has no influence on the *bMBrakeReleaseOut* trigger signal for the holding brake control switching element.
- After the brake has been activated and the brake application time has expired, the controller is inhibited automatically by the basic "Holding brake control" function.

**Tip!**

You can use mode 11 to easily check if the brake switches correctly.

**Mode 12: Automatic control**

In this mode, the brake is controlled automatically.

**Danger!**

In this mode, the input *bMBrakeRelease* should be permanently set to FALSE unless manual release (supervisor operation) is required.

If the *bMBrakeRelease* input is set to TRUE, the brake is released immediately, even if the controller is inhibited!

- If the requested speed setpoint reaches a parameterisable upper speed threshold that allows traversing of the drive, the brake will be released and operation enabled.
- On the other hand, if speed setpoint and actual speed fall below a parameterisable lower speed threshold, the brake will be applied under consideration of different time parameters.
- For operating modes with setpoint request via control signal (e.g. "PosExecute" in the Positioning" operating mode), the speed thresholds do not apply. Here the control logic opens and closes the holding brake through internal commands in the **Motion Control Kernel**.
- The brake will also be activated automatically if quick stop is activated in the drive, e.g. by a device command or as response to an error, and in the event of controller inhibit or pulse inhibit.
- After automatic brake activation and expiration of the brake application time, the controller is inhibited automatically by the basic "Holding brake control" function.

**Tip!**

The 2/12 mode is the usual mode to control the brake.

**Mode 13: Semi-automatic control**

In this mode, brake release and brake application can be directly controlled via the *bMBrakeRelease* input without special logic or automatic.

In contrast to the manual operation (mode 11)

- the feedforward control is active in this mode, preventing a sagging e.g. in case of a hoist.
- the brake in this mode also closes when the controller is inhibited in order to prevent the axis in a hoist from falling.

**Related topics:**

▸ TroubleQSP (📖 124)
▸ Behaviour in case of pulse inhibit (📖 688)

### 9.12.2.2 Functional settings

The following bit coded functional settings for the holding brake control can be made in C02582:

| Bit | Option | Info |
|---|---|---|
| Bit 0 | bMBrakeReleaseOut invert. | Activation of inverted control<br>• "1" ≡ Inverted logic of the trigger signal for the holding brake control switching element |
| Bit 1 | Horizontal brake protection | Brake response in case of pulse inhibit<br>• "1" ≡ In the case of a pulse inhibit, the actual speed value is monitored which must reach the "Close" threshold value to cause the holding brake to be applied.<br>**Note:**<br>• This function is only active if bit 3 (horizontal/winding technology) is set as well. The function is used in order that, when the controller is inhibited, the holding brake of a drive with horizontal traverse path does not wear out during rotation.<br>• With vertical motion (bit 3 = 0), this function is not active. Especially with hoists and activated pulse inhibit of the inverter, an immediate application of the brake is essential for safety-related reasons! |
| Bit 2 | with hoist inv. feedfwd. control | Direction of feedforward control with vertical/hoist technology:<br>• "0" ≡ Positive direction<br>• "1" ≡ Negative direction<br>**Note:**<br>Reversal (Ccw) is then considered. |
| Bit 3 | Horizontal application | Direction of movement of the axis<br>• "0" ≡ The axis performs vertical movements. Gravitational acceleration causes movements.<br>• "1" ≡ The direction of the axis is horizontal or rotary. The gravitational acceleration does not cause any movement. |
| Bit 4 | Feedforward control C2581 | Selection of the feedforward control value<br>• "0" ≡ Automatic selection.<br>  • The torque saved at the last stop is used.<br>• "1" ≡ Manual selection.<br>  • *bMBrakeStartValue2* = FALSE: The feedforward control value 1 set in C02581/4 is used.<br>  • *bMBrakeStartValue2* = TRUE: The feedforward control value set in C02581/5 is used. |
| Bit 5 | Feedback monitoring | Activation of status monitoring<br>• "1" ≡ The *bMBrakeApplied* input for status detection of the brake (via a switching contact at the brake) is monitored after the waiting time set in C02589/3 has expired. |
| Bit 6 | Sync ramp L_NSet_1<br>(from version 02.00.00) | Selection of the ramp time for the synchronisation process to setpoint speed after the brake opening time has elapsed<br>**Revised behaviour from version 02.00.00:**<br>• "1" ≡ The ramp time of the effective acceleration of the ramp function generator (L_NSet_1) is used (Lenze setting).<br>• "0" ≡ As before, the ramp time set in C02610/1 is used.<br>**Note:**<br>The changeover can be dynamically both via the ramp parameter and via bit 6. |
| Bit 7 | Reserved | |

**Related topics:**

- Behaviour in case of pulse inhibit (📖 688)
- Feedforward control of the motor before release (📖 689)

### 9.12.2.3 Switching thresholds

**Stop!**

Do not set the lower speed threshold for closing the brake too high to prevent excessive wear of the brake!

**Note!**

When comparing speeds, only the absolute value of the motor speed and not the direction of rotation is considered.

**Upper speed threshold for brake release:**

Switching threshold (C02581/1) + hysteresis for release (C02581/2)

**Lower speed threshold for brake application:**

Switching threshold (C02581/1) - hysteresis for application (C02581/3)

**Tip!**

The lower speed threshold for brake application should be set to approximately 5 ... 20 % of the maximum speed to minimise the wear of the brake and provide for an optimum brake reaction by a low grinding of the brake.

**Related topics:**

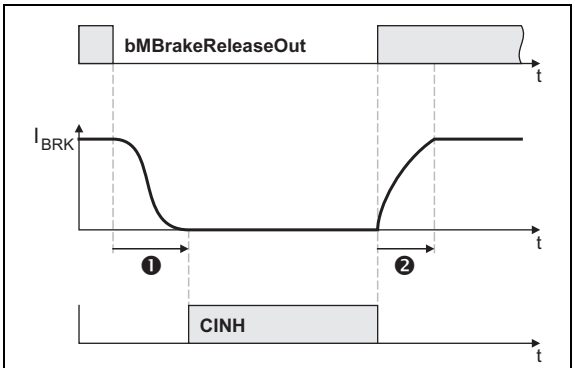
- Process when brake is released (📖 685)
- Process when brake is closed (📖 686)

### 9.12.2.4 Application and release time

**Danger!**

A wrong setting of the application and release time can cause a faulty control of the brake!

- If the application time is set too low, the controller is inhibited and the drive becomes torqueless before the brake is applied completely.




❶ C02589/1: Application time
❷ C02589/2: Release time
CINH = controller inhibit

[9-37] Definition of the application and release time with the example of the PM brake

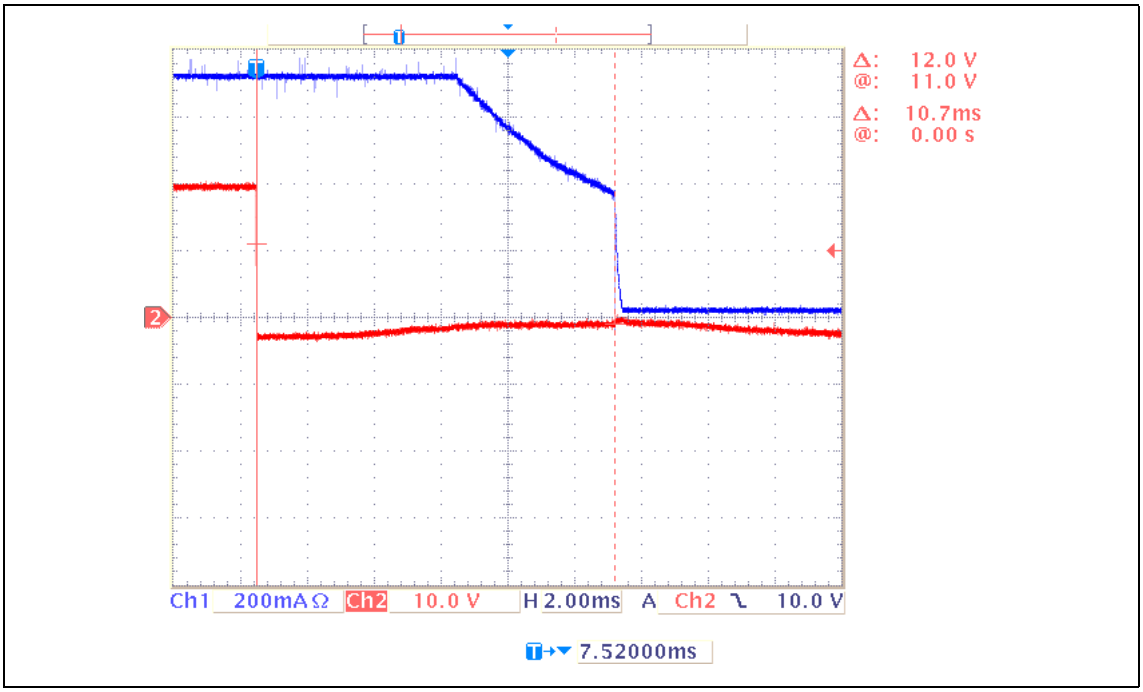
- Every mechanical holding brake comes with a construction-conditioned application and release time which must be considered by the holding brake control and is set in C02589.
- The application and release time of the Lenze holding brake is indicated in the supplied operating instructions in the "Technical data" chapter.
- If the application and release times are too long, this is uncritical in respect of safety but leads to unnecessarily long delays during cyclical braking processes.

**Tip!**

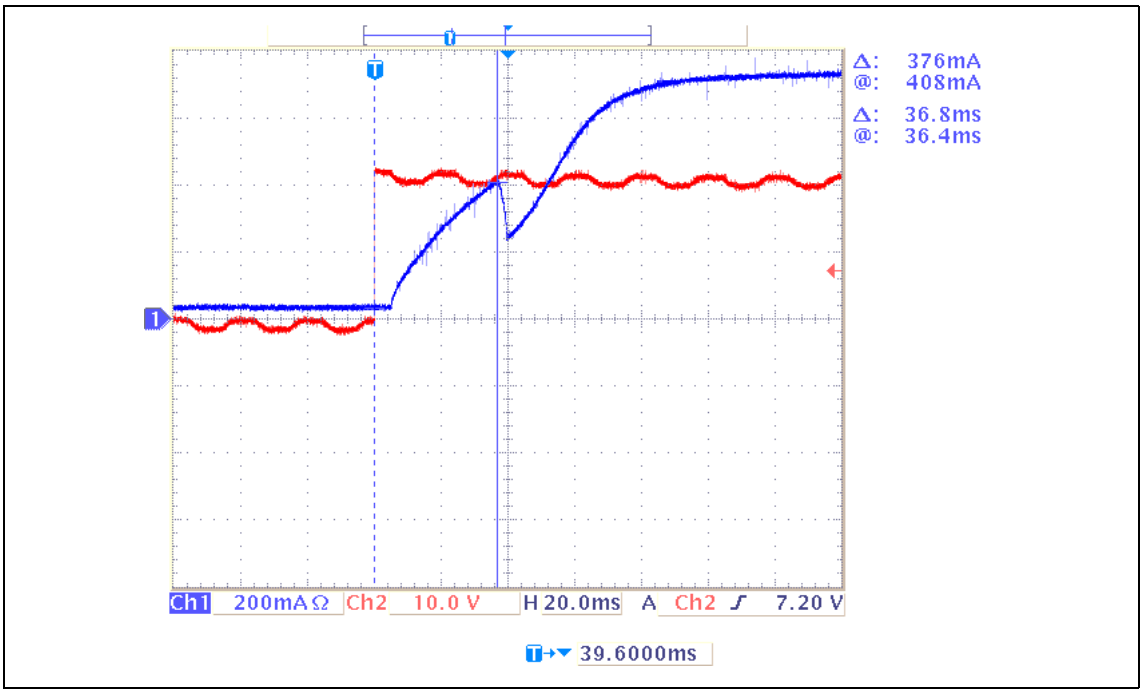
The application and release times do not only vary between the brake types but also depend on the basic conditions in the plant:

- Parameters of the hardware (cable length, temperature, level of supply voltage etc.)
- Contact elements used (brake module or contactor at the digital output)
- Type of overvoltage limitation/suppressor circuit

For optimisation purposes, detect in individual cases the response times by measurement.




[9-38] Oscillogram 1: Current characteristic for the application of a mechanical holding brake (application time: 10.7 ms)




[9-39] Oscillogram 2: Current characteristic for the release of a mechanical holding brake (release time: 36.8 ms)

**Related topics:**

- Process when brake is released (📖 685)
- Process when brake is closed (📖 686)

### 9.12.2.5 Ramp time for approaching the setpoint speed

For the "Speed follower" operating mode, a ramp time can be set in C02610/1 if the setpoint is already out of reach while the holding brake is initiating the feedforward control process.

**Note!**

The "Homing", "Manual jog", and "Positioning" operating modes are based on a different control/release process of the holding brake. In these operating modes, the C02610/1 setting parameter does not have any effect!

**Example :**

A setpoint of 90 % is selected via the ramp function generator while the brake is applied (controller is inhibited).

1. At the set ramp (in most cases C00012), the ramp function generator ramps up to 90 %.
2. The brake identifies the setpoint selection of 5 % (release switching threshold). The feedforward control of the brake provides 3 % of the setpoint and will not report the release of the brake after approx. 1 s has expired.

Conclusion: 90 % of the selected setpoint is already ramped up while the brake is only providing 3 % of the setpoint via the feedforward control.

Since at this point a step change from 3 % to 90 % may cause mechanical jerks, the setpoint is ramped up from 3 % to 90 %, using the ramp time set in C02610/1 (Lenze setting: 2 s).

Our example is based on the V/f characteristic control (VFCplus) operating mode since the servo control (SC) operating mode does not use the switching threshold for the application of the holding brake for speed feedforward control. However, ramping up to a setpoint that is out of reach is carried out for all motor control operating modes because there is always a mechanical/electric delay when controlling a holding brake.

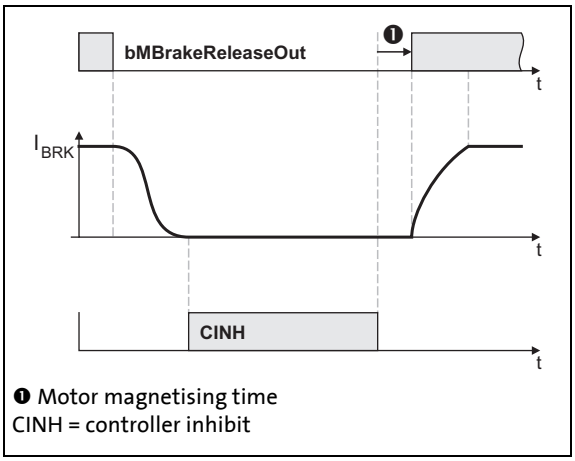
This delay is due to:

- Motor magnetisation (in the case of servo control only)
- Mechanical delay of all switching elements connected upstream to the holding brake
- Mechanical delay of the holding brake itself
- Generation of the holding torque by the motor

**Related topics:**

▸ Process when brake is released (📖 685)

### 9.12.2.6 Motor magnetising time (only with asynchronous motor)




❶ Motor magnetising time
CINH = controller inhibit

[9-40] Considering the motor magnetising time taking the PM brake as an example

- When an asynchronous motor is used, first the magnetic field required for the holding torque is created (which is already available when a synchronous motor is used) after the controller inhibit is deactivated.
- The motor is internally magnetised through internal feedforward control of the lower speed threshold. The release time set in C02589/2 is considered here.

**Related topics:**

- Process when brake is released (📖 685)

### 9.12.2.7 Actual value monitoring

If an actual value monitoring time > 0 s is selected in C02593/1, the actual speed time monitoring is active.

- The monitoring time starts when the speed setpoint has reached the lower switching threshold and the actual speed is still above this threshold. (see illustration [9-43] in chapter "Process when brake is closed".)
- If the actual speed is still above the threshold when the monitoring time has expired, the brake will be automatically applied in the automatic brake control mode (mode 12).

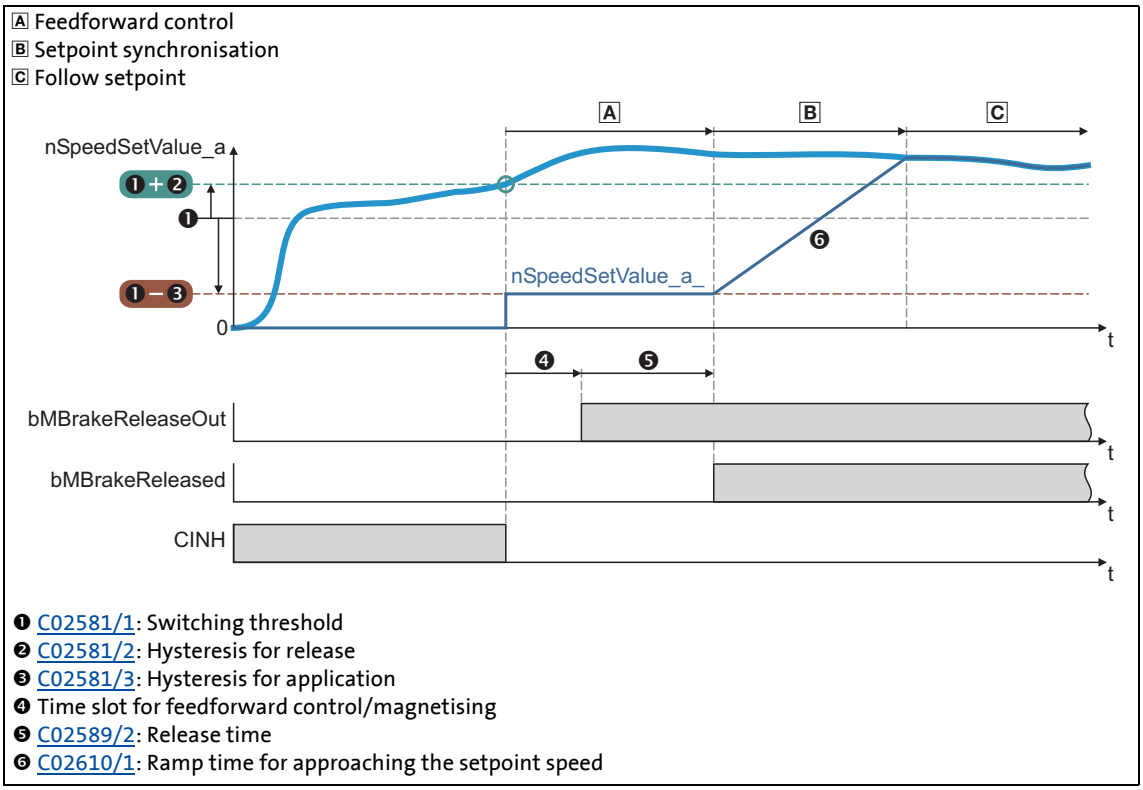
**Note!**

In the Lenze setting, the actual speed time monitoring is deactivated (C02593/1 = "0 s"), i.e. the brake will only be applied when the actual speed has reached the lower switching threshold.

### 9.12.3 Process when brake is released

1. The controller inhibit is deactivated.
2. The magnetic field required for the holding torque is created in the motor (is already available when a synchronous machine is used).
3. The *bMBrakeReleaseOut* trigger signal for holding brake switching element is set to TRUE for releasing the brake.
4. After the brake opening time has elapsed:
   - The *bMBrakeReleased* status signal ("brake released") is set to TRUE.
   - In the "Speed follower" operating mode, the drive synchronises to the already accelerated speed setpoint.
   - In the operating modes with setpoint request via control signal (operating modes "Homing", "Manual jog" and "Positioning"), the ramping process starts after the brake release at 0.
5. After the waiting time set in C02589/3 has additionally expired, the status monitoring starts again (if activated via bit 5 in C02582).

#### Time diagram




[9-41] Release holding brake in automatic mode via speed threshold

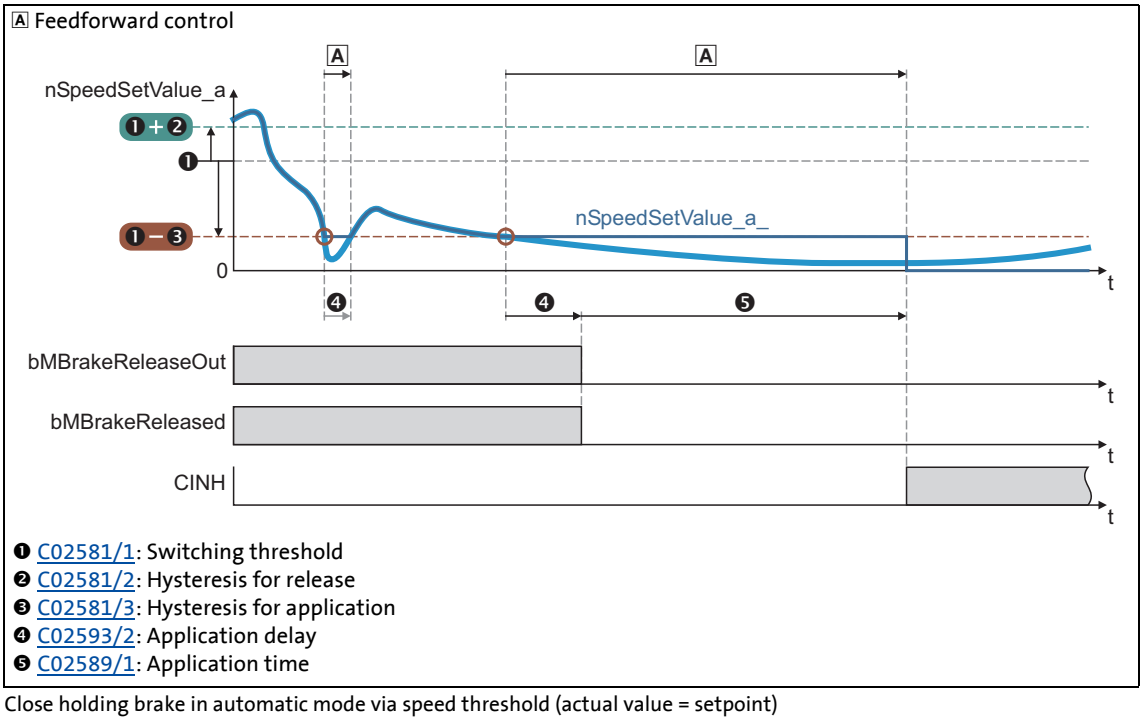
#### Related topics:

- Feedforward control of the motor before release (📖 689)

### 9.12.4 Process when brake is closed

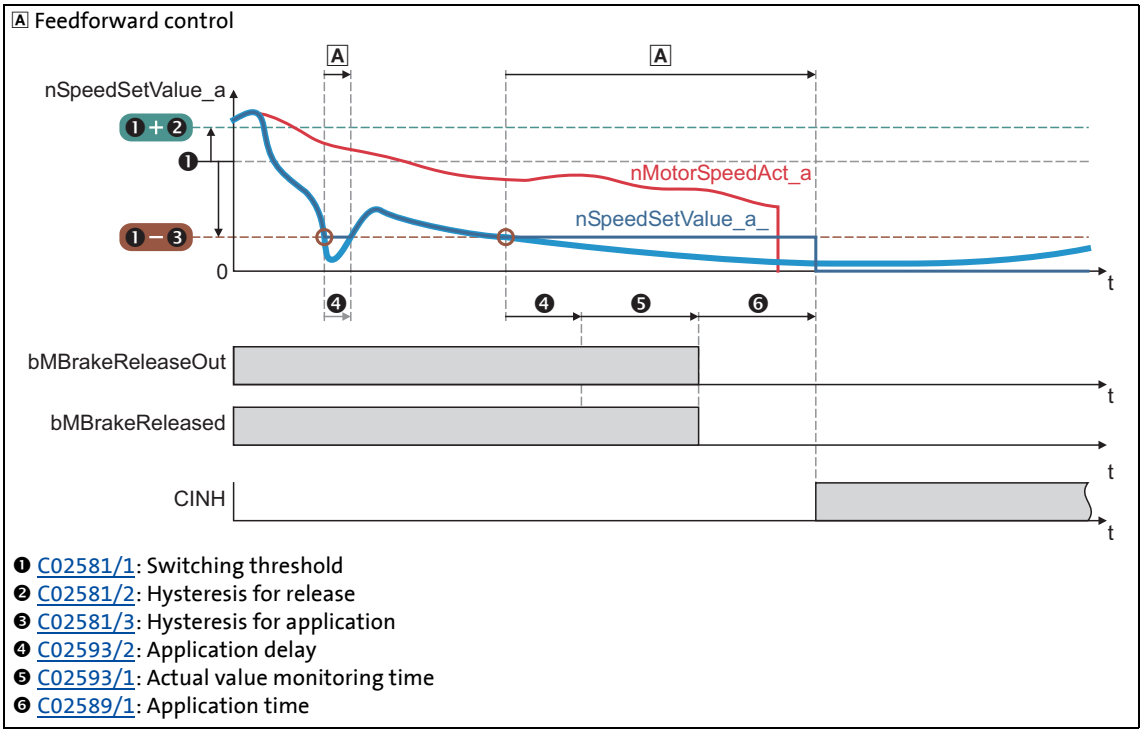
1. The motor is decelerated when the setpoint is reduced by the user (e.g. turn down the potentiometer, setpoint selection via CAN).
   - The motor can also be decelerated by the "Quick stop" function or by "DC-injection braking", either directly requested by the user or as response to an error.
2. If the speed setpoint and the actual speed have fallen below the lower speed threshold or only the speed setpoint has fallen below the lower speed threshold and the actual value monitoring time has expired:
   - The *bMBrakeReleaseOut* trigger signal for the holding brake switching element is set to FALSE for closing the brake.
   - The *bMBrakeReleased* status signal is reset to FALSE.
   - In the operating modes with setpoint request via control signal (operating modes "Homing", "Manual jog" and "Positioning"), the brake closes depending on the internal state *bBusy* (setpoint generation through active profile generator) of the **Motion Control Kernel**.
   - The brake application time starts to expire.
3. After the brake application time has expired, the controller is inhibited.
4. After the waiting time set in C02589/3 has additionally expired, the status monitoring starts again (if activated via bit 5 in C02582).
5. In order to prevent the drive from further rotating/accelerating in the event of an error of the feedback contact, controller inhibit is cancelled again and the drive is held at standstill in a speed-controlled manner.

#### Time diagrams




❶ C02581/1: Switching threshold
❷ C02581/2: Hysteresis for release
❸ C02581/3: Hysteresis for application
❹ C02593/2: Application delay
❺ C02589/1: Application time

[9-42] Close holding brake in automatic mode via speed threshold (actual value = setpoint)




❶ C02581/1: Switching threshold
❷ C02581/2: Hysteresis for release
❸ C02581/3: Hysteresis for application
❹ C02593/2: Application delay
❺ C02593/1: Actual value monitoring time
❻ C02589/1: Application time

[9-43] Close holding brake in automatic mode with actual value monitoring time (C02593/1 > 0 s)

### 9.12.5 Behaviour in case of pulse inhibit

Setting the pulse inhibit causes a load-controlled coasting of the motor until the pulse is enabled again. In the enabled inverter, the pulse can be inhibited e.g. due to a DC overvoltage, DC undervoltage or the "Safe torque off" request.

The brake response to pulse inhibit can be parameterised under C02582.

**Stop!**

For parameterising the response to pulse inhibit in C02582, the energy conditions of the machine should be evaluated first.

The energy stored in the machine can be considerably higher than the permissible switching energy and thus lead to the destruction of the brake if applied directly!

#### Activate brake immediately when pulse is inhibited

If bit 1 is set to "0" in C02582 (Lenze setting), the brake will be immediately applied when the pulse is inhibited to avoid damage to the mechanical components.

Especially in the case of hoist drives, immediate engagement of the brake is absolutely necessary for safety reasons if the pulse inhibit function of the inverter has been activated!

**Danger!**

This behaviour is valid in (semi) automatic operation when the *bMBrakeRelease* input is set to FALSE.

When the *bMBrakeRelease* input is set to TRUE (supervisor operation) in automatic mode, the brake is not applied at pulse inhibit!

#### Only activate brake below threshold for brake activation

If bit 1 and bit 3 are set to "1" in C02582, the brake remains released until the lower speed threshold is reached to avoid an excessive wear of the brake.

- The braking action only takes places due to the friction in the load mechanics.
- The brake will not be applied until the motor speed has reached the threshold for brake activation. Hence, the function depends on the signal of the speed encoder.

During uncritical operation (horizontal loading condition), delayed brake application may be required to protect the brake in case of high centrifugal masses.

In case of vertical motion (bit 3 = 0), this function is not active due to safety-related reasons.

**Related topics:**

▸ Functional settings (📖 679)
▸ Switching thresholds (📖 680)

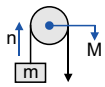
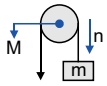
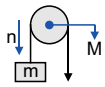
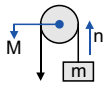
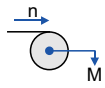
### 9.12.6 Feedforward control of the motor before release

The motor is precontrolled by selecting the lower speed threshold for applying the brake (control modes without feedback). When the upper speed threshold for brake release is reached, the motor is precontrolled with the for the release time set in C02589 with the lower threshold value before the brake switches to the release mode.

Here, the direction of the feedforward control depends on two conditions:

1. On the settings selected under C02582:
   - Bit 2 = feedforward control inverted (for vertical drives/hoists)
   - Bit 3 = direction of the axis
2. On the sign of the setpoint.

#### Truth table for the direction of the feedforward control

| Setpoint | Direction | Feedforward control | Scheme | Direction | |
|---|---|---|---|---|---|
| | | | | Feedforward control value | Start value |
| n ≥ 0 | vertical/hoist (C02582: Bit 3 = 0) | not inverted (C02582: Bit 2 = 0) |  | ⊕ | ⊕ |
| | | inverted (C02582: Bit 2 = 1) |  | ⊖ | ⊕ |
| n < 0 | | not inverted (C02582: Bit 2 = 0) |  | ⊕ | ⊖ |
| | | inverted (C02582: Bit 2 = 1) |  | ⊖ | ⊖ |
| n ≥ 0 | horizontal/winding drive (C02582: Bit 3 = 1) | Inversion via bit 2 with horizontal direction not effective |  | ⊕ | ⊕ |
| n < 0 | | |  | ⊖ | ⊖ |

**Selection of the feedforward control value**

The feedforward control value can be selected via bit 4 in C02582:

- Bit 4 = 0: Automatic selection
  - The torque saved at the last stop is used.
- Bit 4 = 1: Manual selection
  - *bMBrakeStartValue2* = FALSE: The feedforward control value 1 set in C02581/4 is used.
  - *bMBrakeStartValue2* = TRUE: The feedforward control value set in C02581/5 is used.

**Additive torque**

If the servo control (SC) operating mode has been selected, an additive torque value in [%] can be selected via the *nBrkTorqueAdd_a* input.

**Related topics:**

▸ Functional settings (📖 679)
▸ Switching thresholds (📖 680)

# 10 Diagnostics & error management

This chapter provides information on error handling, drive diagnostics, and fault analysis.

## 10.1 Basics on error handling in the inverter

Many of the functions integrated in the inverter can

- detect errors and thus protect the device from damage or overload, e.g. short-circuit detection, Ixt overload detection, overtemperature detection, etc.
- detect an operating error by the user, e.g. a missing memory module, a required or missing communication module, etc.
- output a warning signal if desired, e.g. if the speed is too high or too low, etc.

Depending on the importance, the error detection in the device responds very fast (e.g. short-circuit detection < 1 ms) or in a slower cycle (e.g. temperature monitoring approx. 100 ms).

All functions provided with an error detection (e.g. the motor control) supply information to a so-called error handler. The error handler is processed every 1 ms and evaluates all information.

In this evaluation, the so-called status determining error (display in C00168) and the current error (display in C00170) are generated, and the inverter is caused to take the respective error status (e.g. TroubleQSP).

These two types of error information serve to diagnose errors systematically and contain the following information:

1. The error type (e.g. "Warning")
2. The error subject area (e.g. "CAN generally integrated")
3. The error ID within the error subject area

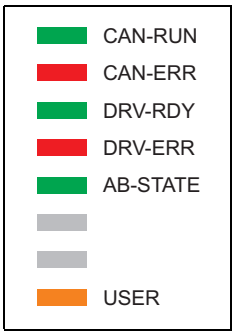
Together all types of information form the real error number which is unique in the whole device system. ▸ Structure of the 32-bit error number (bit coding) (📖 715)

In addition to the control of the device status by the error handler, a logbook function records the errors and their histories. ▸ Logbook (📖 700)

**Tip!**

For many device errors, the error type and hence the response of the inverter to the error can be parameterised. ▸ Setting the error response (📖 710)

## 10.2 LED status displays



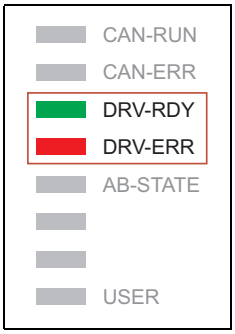

Information on some of the operating states can be quickly obtained via the LED display on the front of the inverter.

| Labelling | Colour | Description | |
|---|---|---|---|
| CAN-RUN | green | CAN bus ok | ▸ LED status displays for the system bus (📖 778) |
| CAN-ERR | red | CAN bus error | |
| DRV-RDY | green | Standard device ready for operation | ▸ LED status displays of the device status (📖 693) |
| DRV-ERR | red | Warning/trouble/fault | |
| AB-STATE | green | Axis bus status | ▸ LED status displays of the axis bus (📖 860) |
| - | - | These LEDs are currently not provided with a function! | |
| - | - | | |
| USER | orange | Starting from version 12.00.00, this LED can be triggered with any digital process signal via the LS_DigitalOutput system block. (📖 408) | |

**Related topics:**

- ▸ Device control (DCTRL) (📖 102)
- ▸ Device state machine and device states (📖 116)
- ▸ System bus "CAN on board" (📖 772)

### 10.2.1 LED status displays of the device status




The control of the two LEDs "DRV-RDY" and "DRV-ERR" on the front of the inverter depends on the device status.

The meaning can be seen from the table below.

| DRV-RDY | DRV-ERR | Description | Device status (Display in C00137) |
|---|---|---|---|
| OFF | OFF | OFF or initialisation active | Init |
| | OFF | Safe torque off is active | SafeTorqueOff |
| | OFF | Device is ready to start | ReadyToSwitchOn |
| | OFF | Device is switched on | SwitchedOn |
| | OFF | Motor data identification/operation | OperationEnabled |
| | | The inverter is ready to switch on, switched on or the operation is enabled and a warning is indicated. | |
| | | Fault active, quick stop | TroubleQSP |
| OFF | | Trouble active | Trouble |
| OFF | | Error active | Fault |
| OFF | | System fault active | SystemFault |

**Legend**
Meaning of the symbols used to describe the LED states:

| | |
|---|---|
| | LED is flashing once approx. every 3 seconds (*slow flash*) |
| | LED is flashing once approx. every 1.25 seconds (*flash*) |
| | LED is flashing twice approx. every 1.25 seconds (*double flash*) |
| | LED is blinking every second |
| | LED is permanently on |

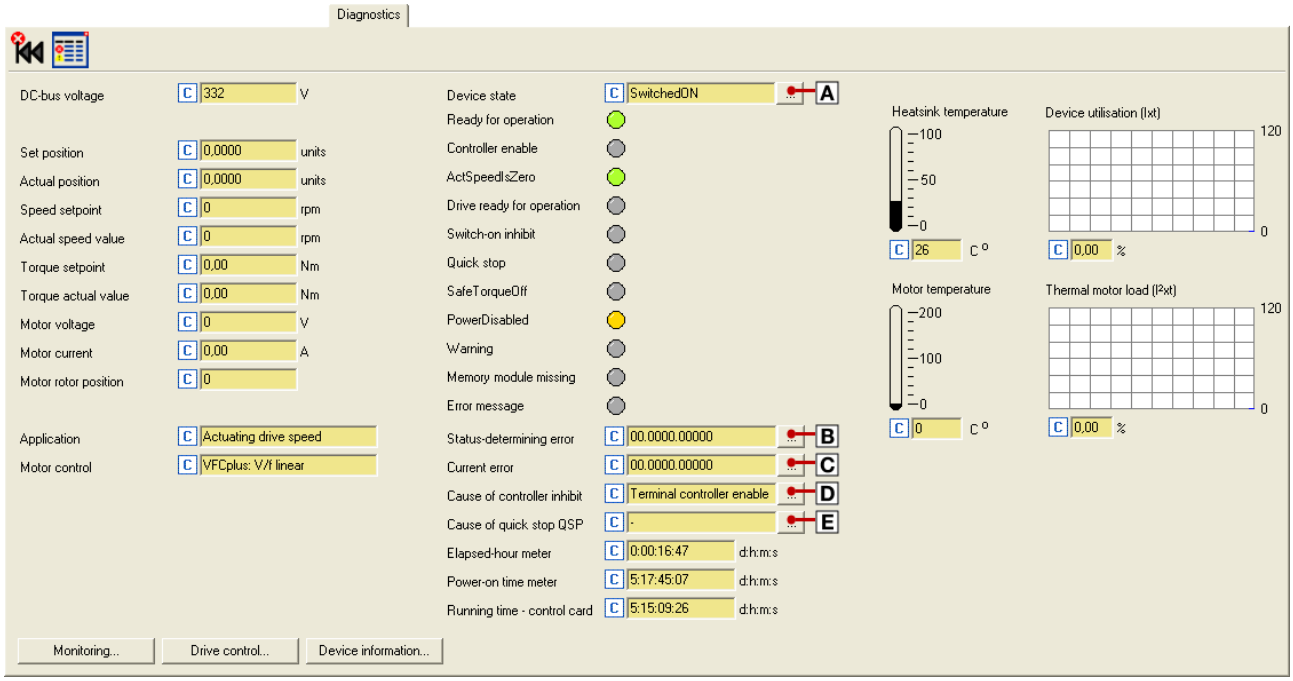
**Related topics:**

- Device state machine and device states (📖 116)

## 10.3 Drive diagnostics with the »Engineer«

When an online connection to the inverter has been established, the connected inverter can be diagnosed and relevant actual inverter states can be displayed in a clearly arranged visualisation using the »Engineer«.

**Example: Representation of the diagnostic information in the »Engineer« V2.19**




| Button | | Function |
|---|---|---|
|  | | Acknowledge error message (if the error cause has been eliminated). |
|  | | Display the Logbook of the inverter. (📖 700) |
|  | A | Display the internal state machine including the current device status. |
| | B | Display details of the status determining error. |
| | C | Display details of the current error. |
| | D | Display all active sources of a controller inhibit. |
| | E | Display all active sources of a quick stop. |
| **Monitoring...** | | Configure the Monitoring. (📖 708) |
| **Drive control...** | | Display the bit assignment of the following control-related words:<br>• MCI control word (C00136/1)<br>• CAN control word (C00136/2)<br>• Cause of controller inhibit (C00158)<br>• Cause of quick stop (C00159)<br>• Status word (C00150)<br>• Status word 2 (C00155) |
| **Device information...** | | Display identification data, e.g. firmware information or serial number of individual inverter components. |

**How to diagnose a drive with the »Engineer«:**

1. Go to the *Project view* and select the 8400 TopLine inverter.
2. Click the icon or select the **Online→Go online** command to build up an online connection with the inverter.
3. Select the **Diagnostics** tab.
   - With an online connection, the **Diagnostics** tab displays current status information about the inverter.

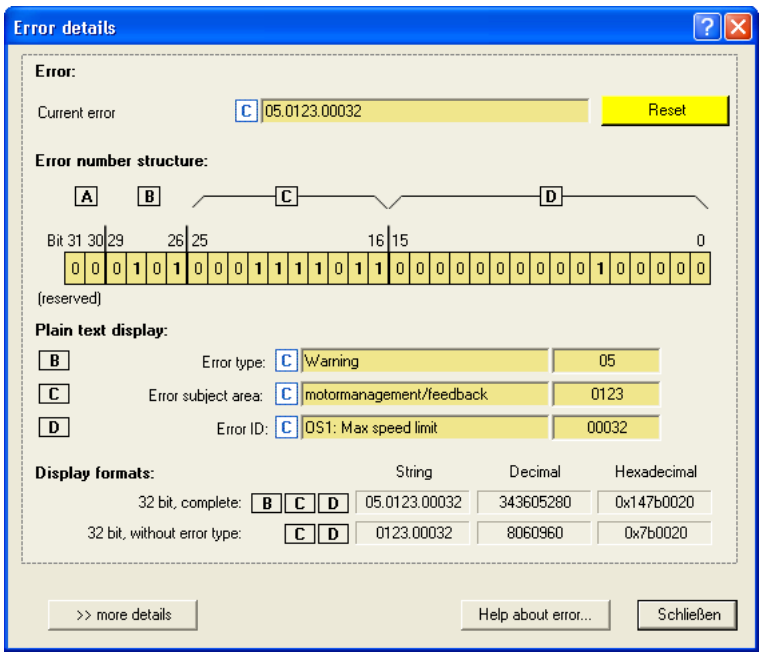
**Tip!**

The online connection to the inverter can be established via the following device interfaces:

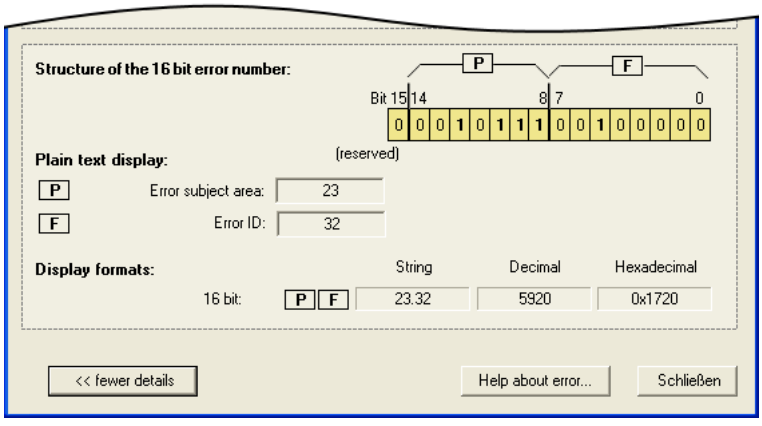
- CAN interface X1
  Diagnostics via the System bus "CAN on board" (📖 772)
- X6 diagnostic interface
  We recommend this diagnostic connection when the X1 CAN interface is used for process communication.

### 10.3.1 Display details of the error

If you go to the **Diagnostics** tab and click the ... button for the status determining or current error, the *Error details* dialog box displays further information on the error:




- Click the **Help about error...** button to open the online help with information on the error cause and possible remedies.
- The **>> more details** button serves to provide more information about the structure of the 16-bit error number:

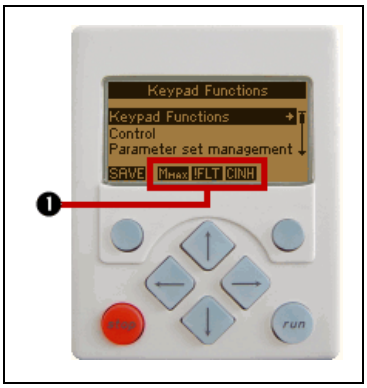



**Related topics:**

- Structure of the 32-bit error number (bit coding) (📖 715)
- Structure of the 16 bit error number (bit coding) (📖 718)

## 10.4 Drive diagnostics via keypad/bus system

### Keypad display of the inverter status




- If the keypad on the front of the inverter is connected to the diagnostic interface X6, the status of the inverter is shown via different icons on the LCD display in the area ❶.

| Symbol | Meaning | Notes |
|---|---|---|
| RDY | Inverter is switched on. | ▸ SwitchedOn (📖 122) |
| RUN | Inverter is enabled. | |
| QSP | Quick stop active | |
| CINH | Inverter is inhibited. | The power outputs are inhibited. |
| Mmax | Speed controller 1 at the limit. | The drive is torque-controlled. |
| Imax | Set current limit has been exceeded in motor or generator mode. | |
| IMP | Pulse inhibit active | The power outputs are inhibited. |
| !SFLT | System fault active | |
| !FLT | Fault | ▸ Fault (📖 126) |
| !TRB | Trouble | ▸ Trouble (📖 125) |
| !TQSP | TroubleQSP | ▸ TroubleQSP (📖 124) |
| WRN | Warning is active | |

### Display parameters

The parameters listed in the following tables serve to query current states and actual values of the inverter for diagnostic purposes, e.g. by using the keypad, a bus system or the »Engineer« (with an online connection to the inverter).

- These parameters are listed in the »Engineer« parameter list and the keypad in the **Diagnostics** category.
- A detailed description of these parameters can be found in the chapter "Parameter reference" (📖 895).

| Parameters | Display |
|---|---|
| C00051 | MCTRL: Actual speed value |
| C00052 | Motor voltage |
| C00053 | DC-bus voltage |
| C00054 | Motor current |
| C00056/1 | Torque demand |
| C00056/2 | Actual torque value |
| C00058 | Output frequency |
| C00061 | Heatsink temperature |
| C00064/1 | Device utilisation (Ixt) |
| C00064/2 | Device utilisation (Ixt) 15s |
| C00064/3 | Device utilisation (Ixt) 3 min |
| C00133 | Brake resistor utilisation |
| C00136/1 | MCI control word |
| C00136/2 | CAN control word |
| C00137 | Device status |
| C00138/1 | SYS control signals |
| C00138/2 | MCK control signals |
| C00138/3 | FWM control signals |
| C00150 | Status word |
| C00158 | Cause of controller inhibit |
| C00159 | Cause of quick stop QSP |
| C00165/1 | Status determining error (displayed as a numeric text) |
| C00165/2 | Current error (displayed as a numeric text) |
| C00168 | Status determining error (display of 32-bit number) |
| C00170 | Current error |
| C00166/1 | Error type, status determining |
| C00166/2 | Error subject area, status determining |
| C00166/3 | Error ID, status determining |
| C00166/4 | Error type, current |
| C00166/5 | Error subject area, current |
| C00166/6 | Error ID, current |
| C00177/1 | Switching cycles mains switching |
| C00177/2 | Switching cycles output relay |
| C00177/3 | Stress counter - short circuit |
| C00177/4 | Stress counter - earth fault |
| C00177/5 | Stress meter clamp |

| Parameters | Display |
|---|---|
| C00177/6 | STO counter after power-on |
| C00177/7 | DigIn CINH counter after power-on |
| C00177/8 | IMP counter after power-on |
| C00178 | Time the inverter was enabled (elapsed-hour meter) |
| C00179 | Power-up time (power-on time meter) |
| C00180/1 | Running time - control card |
| C00180/2 | Running time - heatsink fan |
| C00180/3 | Running time - internal fan |

### Identification data

The parameters listed in the following table belong to the **Identification** category of the »Engineer« parameter list and the keypad and serve to display the identification data of the inverter:

| Parameters | Display |
|---|---|
| C00099 | Firmware version (as a string) |
| C00199/1 | Device name<br>▸ Automatic acceptance of the device name in the »Engineer« |
| C00200 | Firmware product type |
| C00201/1...6 | Firmware of the control card and the power section |
| C00203/1...9 | Product type code of the individual device components |
| C00204/1...9 | Serial numbers of the individual device components |

## 10.5 Logbook

The integrated logbook function of the inverter chronologically logs important events within the system and plays an important role for troubleshooting and inverter diagnostics.

### Events that can be logged

The following events can be logged in the logbook:

- Error messages of the operating system (📖 715)
- Error messages generated by the application (via LS_SetError)
- Loading/saving of parameter sets, loading of the Lenze setting *(in preparation)*
- Transmitting the firmware to the inverter *(in preparation)*
- Switching on/off of the inverter

### Information saved

For each event, the following information is saved in the logbook:

- Type of response to the event (e.g. fault, warning or information)
- Subject area that activated the event (e.g. CAN or USER).
- Event
- Value of power-on time meter
- Selected process values (analog % signals, binary signals)

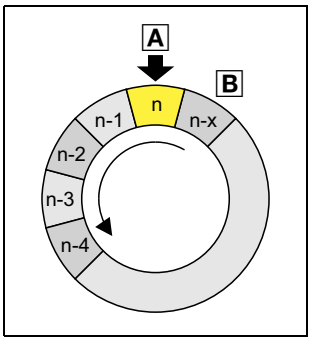
### Memory depth

Maximum number of logbook entries:

- 8400 StateLine: 15 logbook entries
- 8400 HighLine/TopLine: 50 logbook entries

## 10.5.1 Functional description

The structure of the logbook corresponds to a ring buffer:




- As long as free logbook memory is available, the entry is placed in the next free position within the memory (A).
- If all memory units are assigned, the oldest entry (B) is deleted for a new entry.
- The newest entries will always remain available.

**Note!**

In the event of a supply voltage failure, the logbook is saved and reloaded automatically when the inverter is switched on. This ensures that the error history of the device does not get lost. For this reason it is very important to act with caution when deleting the logbook entries.

### 10.5.2 Filtering logbook entries

The logbook adds new entries to the ring buffer after they have been passed through a parameterisable filter. This filter helps you to exclude certain events from being entered into the logbook which would trigger a certain error response (fault, trouble, warning, information, etc.).

C00169 (bit 1 ... bit 6) includes a bit coded specification of the events which are to be entered into the logbook. In the Lenze setting, all events are entered into the logbook.

In the Lenze setting (C00169 = 0x067E), all events are entered into the logbook.

**Note!**

Events with the "No response" setting are not entered into the logbook.

#### Counter for multiple entries

In order to prevent the ring buffer from overflowing with identical errors with frequent occurrence e.g. during commissioning, identical errors will not lead to new line entries in the configuration of the logbook in the Lenze setting. Instead, one counter will be counted up for this error.

The time of the error is always the time of its first occurrence. A new logbook line will only be generated if an error occurs that differs from the previous error.

In C00169, bit 9 and bit 10 determine the logbook update in case of similar errors.

Meaning of bit 9 / bit 10:

- Bit 9: Error counter
- Bit 10: Update of the logbook line

In the Lenze setting (C00169 = 0x067E), bits 9 and 10 are active.

The following cases can be distinguished depending of the activation of bit 9 and bit 10:

1. Bit 9 inactive, bit 10 active/inactive
   - Every time an error occurs, a new line is created in the logbook.
   - The setting of bit 10 has no effect!
2. Bit 9 active, bit 10 inactive
   - Only one line is created for the first occurrence of the error. If the error occurs again, only the error counter in the line is increased.
3. Bit 9 active / bit 10 active (Lenze setting)
   - The time of the last occurrence of similar errors is logged.
   - The time stamp and the optional measured values are updated in the error line of the logbook.
   - When the count value = 255 is reached and an error occurs again, a new line is created.

### 10.5.3 Automatic recording of device-internal signals at the time the error occurs

For purposes of error analysis, two optional device-internal digital signals and an analog signal (16 bit) can be recorded at the time the error occurs.

- The digital signals to be recorded are selected in C00163/1 and C00163/2.
  - Scaling of the value recorded: 0 ≡ FALSE; 1 ≡ TRUE

- The analog signal to be recorded is selected in C00164/1.
  - Scaling of the value recorded: 16384 ≡ 100 %

### 10.5.4 Reading out logbook entries

With an online connection, the existing logbook entries can easily be displayed in the »Engineer«. Alternatively, the logbook entries can also be read out via the corresponding parameters (e.g. using the keypad).

**Note!**

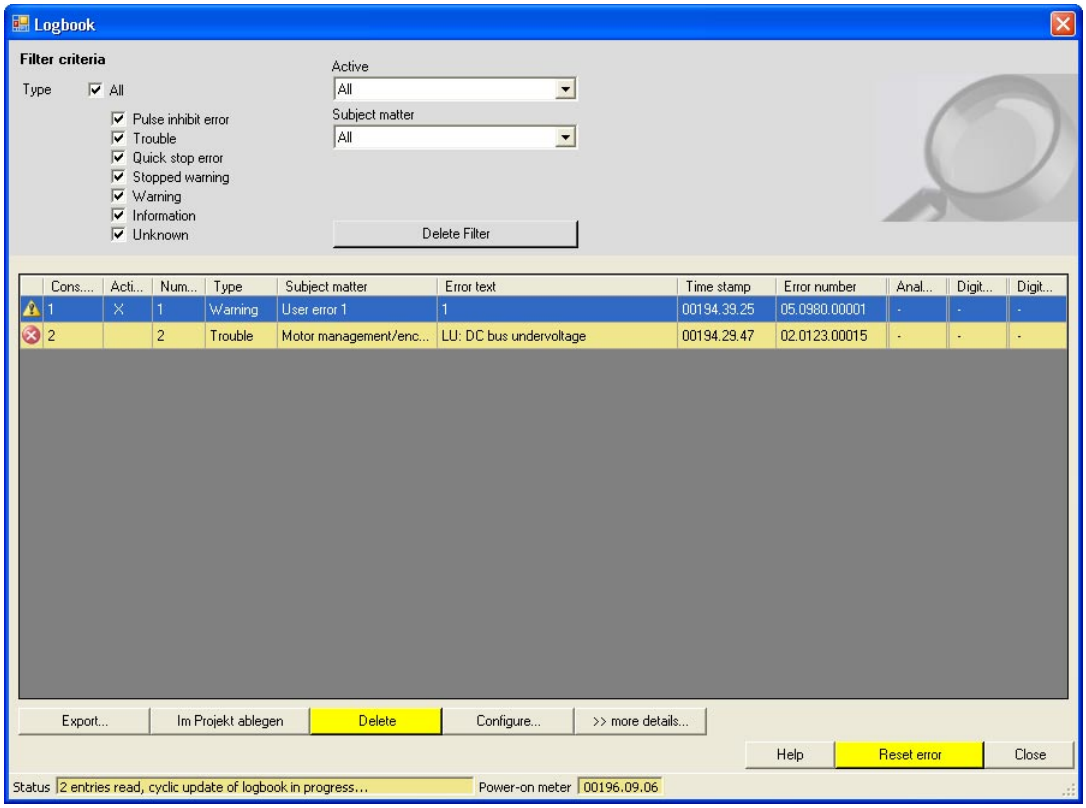
Reading out the logbook entries is only permissible by one engineering tool at a time!

- If the logbook entries are tried to be read out via two engineering tools at the same time, missing lines or a permanent refresh may be caused.
- The parameter interface for reading out the logbook by an external control/visualisation, is available as well. ▸ Reading out the logbook from an external control/visualisation (📖 706)

**How to display logbook entries in the »Engineer«:**

1. Go to the *Project view* and select the 8400 TopLine inverter.
2. Click the icon or select the **Online→Go online** command to build up an online connection with the inverter.
3. Select the **Diagnostics** tab from the *Workspace*.
4. Click **Logbook**.

**Example: Representation of the logbook in the »Engineer« V2.13**



| Button | Function |
|---|---|
| **Delete Filter** | Reset set filter criteria to display all available logbook entries. |
| **Export...** | Export the entries available in the logbook into a *.log file.<br>▸ Exporting logbook entries to a file (📖 704) |
| **Storing in the project** | File the current logbook in the Engineer project to be able to access it offline, too.<br>▸ Storing the logbook in the project (📖 705) |
| **Delete** | Delete all entries available in the logbook. |
| **Configuration of...** | Open parameterisation dialog for configuring the logbook. |
| **>> more details** | Show more details:<br>• Analog value 1, digital values 1 & 2<br>• More output types of the error numbers (32-bit, internal 32-bit and internal 16-bit).<br>• Instead of the **>> more details** button, the **<< less details** is now displayed via which the details can be hidden again. |
| **Help** | Open online help for the logbook. |
| **Resetting an error** | Acknowledge existing error message if the error cause has been eliminated and thus the error is not pending anymore.<br>• After the reset (acknowledgement) of the current error, further errors may be pending which must also be reset. |
| **Close** | Close the *Logbook* dialog box again. |

### 10.5.5 Exporting logbook entries to a file

**How to export the logbook entries to a file:**

1. Click **Export...** in the *Logbook* dialog box.
   - The *Export logbook* dialog box is displayed.
2. Specify the folder, file name, and file type for the file.
3. Click the **Save** button to export the logbook entries into the given file.
   - Hidden logbook entries are not exported, i.e. the filter criteria specified are accounted for during the export.
   - The logbook entries are written to the file in the form of a semicolon separated list.

**Structure of the semicolon separated list**

The list includes the following information:

1. Cons. no.
2. Active
3. Counter
4. Type
5. Subject matter
6. Error text
7. Time stamp
8. Relative time
9. Error number
10. Source - analog value 1
11. Analog value 1
12. Source - digital value 1
13. Digital value 1
14. Source - digital value 2
15. Digital value 2

### 10.5.6 Storing the logbook in the project

If you want to display the currently available logbook entries at a later date in offline mode, i.e. without a connection to the inverter, you can store the current logbook in the project.



**How to store the logbook in the project:**

Go to the *Logbook* dialog box and click the **File in project** button.

- The logbook with all the entries uploaded up to now is stored in the Engineer project independent of the set filter criteria.
- A logbook of the same device already stored before will be overwritten without querying the user.
- The filter settings are not stored in the project.
- When a logbook is filed in the project, the logbook can also be opened in offline mode via the **Logbook** button on the **Diagnostics** tab.



**Note!**

Storing the logbook changes the project.

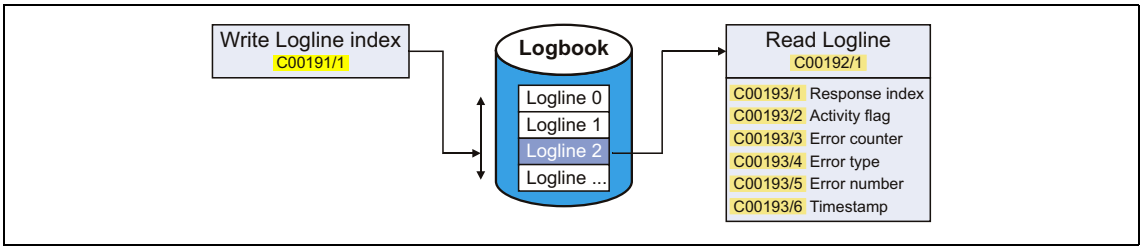
- When the project is closed, you are asked to save the changed project.
- Only if the changed project is saved, the new logbook entries stored in the project remain stored.

### 10.5.7 Reading out the logbook from an external control/visualisation

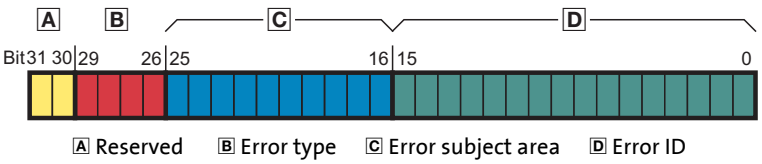
This function extension is available from version 12.00.00!

The parameters described in the following provide a second interface via which an external control or visualisation can exclusively access the logbook and read out logbook contents. Logbook contents can even be read out via this additional interface if the logbook is read out by the »Engineer« or »EASY Starter« at the same time.

#### Structure of the interface




- The line index of the logbook entry to be read out has to be set In C00191/1.
- Then the logbook entry addressed is stored as "OCTET STRING" in C00192/1.
  - Via this display parameter, the logbook entry can be read out by the control.
  - The "OCTET STRING" has a length of 8 double words of 4 successive bytes (= 32 bytes), starting with byte 0 in each case:

| OCTET STRING | Byte 0 | Byte 1 | Byte 2 | Byte 3 |
|---|---|---|---|---|
| **DWORD 0** | Reserved | Reserved | Response index | Reserved |
| **DWORD 1** | Error activity flag | Error counter | Reserved | Reserved |
| **DWORD 2** | 32-bit error number<br>  <br><br>A Reserved B Error type C Error subject area D Error ID | | | |
| **DWORD 3** | Time in [s] during which the power was switched on (power-on time meter). | | | |
| **DWORD 4** | Reserved | Reserved | Reserved | Reserved |
| **DWORD 5** | Source - analog value 1 | Reserved | Analog value 1 (scaling: 16384 ≡ 100 %) | |
| **DWORD 6** | Source - digital value 1 | Reserved | Digital value 1 | Reserved |
| **DWORD 7** | Source - digital value 2 | Reserved | Digital value 2 | Reserved |

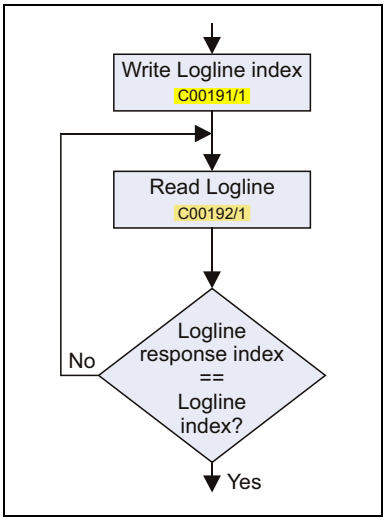
- Additionally, a read access to different individual elements of the logbook entry addressed can be executed via the subcodes of C00193. These parameters have a uniform data format (32 bits) and represent the most important part of the logbook data:

| Parameters | Display |
|---|---|
| C00193/1 | Response index, reference to the index requested (C00191/1) |
| C00193/2 | Error activity flag (0 = error not active; 1 = error active) |
| C00193/3 | Error counter (0 ... 255)<br>• This information depends on the logbook configuration. In the Lenze setting the logbook configured so that identical errors do not produce a new line entry, but a counter is incremented for this error. |
| C00193/4 | Error type (bits 26 ... 29 of the 32-bit error number) |
| C00193/5 | Error subject area + error ID (bits 0 ... 25 of the 32-bit error number) |
| C00193/6 | Time in [s] during which the power was switched on (power-on time meter). |

## Note!

- In the case of a simultaneous read access to the logbook entry (C00192/1) and its individual elements (C00193/x), the line index in C00191/1 must only be reset by the control after the read process is fully completed. Otherwise the data read are inconsistent.
- Depending on the device version, the maximum number of logbook entries can vary:
  - 8400 StateLine: 15 logbook entries
  - 8400 HighLine/TopLine: 50 logbook entries

### Basic workflow




- The logbook can be read out by an external control or visualisation via the procedure shown on the left.
- The "Response index" query ensures that the logbook entry read really corresponds to the logbook entry requested.

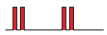
### Related topics:

- Structure of the 32-bit error number (bit coding)

## 10.6 Monitoring

The inverter is provided with various monitoring functions which protect the drive against impermissible operating conditions.

- If a monitoring function responds,
  - an entry will be made into the Logbook of the inverter,
  - the response (TroubleQSP, Warning, Fault, etc.) set for this monitoring function will be triggered,
  - the status of the internal device control changes according to the selected response, controller inhibit is set, and the "DRV- ERR" LED on the front of the inverter goes on:

| Response | Entry in the logbook | Display in C00168 | Pulse inhibit | Disable drive function | Acknowledgement required | LED "DRV-ERR" |
|---|---|---|---|---|---|---|
| None | | | | | | OFF |
| Fault | ☑ | ☑ | ☑ | ☑ | ☑ |  |
| Trouble | ☑ | ☑ | ☑ | ☑ | |  |
| TroubleQSP | ☑ | ☑ | | | ☑ |  |
| WarningLocked | ☑ | ☑ | | | ☑ |  |
| Warning | ☑ | ☑ | | | | |
| Information | ☑ | ☑ | | | | OFF |
| System fault | ☑ | ☑ | ☑ | ☑ | Mains switching is required! |  |

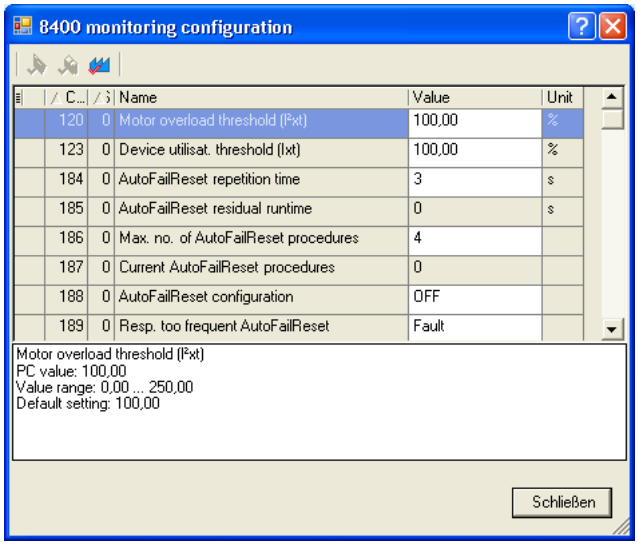
**Related topics:**

- LED status displays of the device status (📖 693)
- Device state machine and device states (📖 116)
- Device overload monitoring (Ixt) (📖 300)
- Motor load monitoring (I2xt) (📖 301)
- Motor temperature monitoring (PTC) (📖 304)
- Brake resistor monitoring (I2xt) (📖 305)
- Motor phase failure monitoring (📖 307)
- Mains phase failure monitoring (📖 310)
- Current monitoring for overload (📖 310)
- Maximum current monitoring (📖 311)
- Maximum torque monitoring (📖 311)
- Resolver/multi-encoder open-circuit monitoring (📖 377)
- Open-circuit monitoring HTL encoder (📖 378)
- Motor temperature monitoring (KTY) (📖 381)

### 10.6.1 Monitoring configuration

**How to configure the monitoring functions using the »Engineer«:**

1. Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Diagnostics** tab from the *Workspace*.
3. Click the **Monitoring...** button.
   - The *8400 monitoring configuration* dialog box is displayed via which the desired settings can be made:



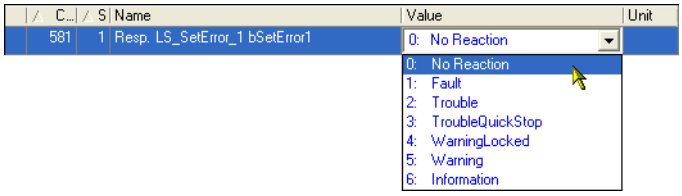

**Related topics:**

- Setting the error response (📖 710)

### 10.6.2 Setting the error response

When a monitoring function responds, the response set for this monitoring function (TroubleQSP, Warning, Fault, etc.) will be triggered.

- For many monitoring functions the response can be individually parameterised via parameters.




**Tip!**

The table in the chapter "Short overview (A-Z)" contains the error messages for which the response can be set. (📖 721)

#### Warning thresholds

Some of the monitoring functions are activated if a defined warning threshold (e.g. temperature) has been exceeded.

- The corresponding preset threshold values can be changed via the following parameters:

| Parameters | Info |
|---|---|
| C00120 | Setting of motor overload (I²xt) |
| C00123 | Device utilisat. threshold (Ixt) |
| C00572 | Brake resistor overload threshold |
| C00599 | Motor phase failure threshold |

### 10.6.3 AutoFailReset function

The AutoFailReset function serves to automatically reset the latching "Fault" and "TroubleQSP" errors as well as the latching "WarningLocked" warning.

The "latching" term means that the effect on the inverter remains active even after the error cause has been removed.

To reset latching errors and warnings, the following options are available:

- Manual reset
  - with device command C00002/19 (activated by Low-High edge)
  - by a Low-High edge at the *bResetFail* input of the LS_DriveInterface (the "FailReset" control bit in the control word must be parameterised with a value of "1").
- Automatic reset
  - using the AutoFailReset function.

#### Overview of the relevant parameters

| Parameters | Info |
|---|---|
| C00184 | Repetition time of the error reset processes |
| C00185 | Time left until the next error reset process |
| C00186 | Max. number of permissible unsuccessful error reset processes<br>• When the number set is reached, the response parameterised in C00189 is executed. |
| C00187 | Current number of unsuccessful error reset processes carried out |
| C00188 | Configuration of the AutoFailReset function<br>• 0: Off<br>• 1: Fault + TroubleQSP<br>• 2: WarningLocked<br>• 3: All locking |
| C00189 | Response after max. number of unsuccessful error reset processes has been reached |
| Greyed out = display parameter | |

## 10.7 Maloperation of the drive

| Maloperation | Cause | Remedy |
|---|---|---|
| Motor does not rotate | DC-bus voltage is too low<br>• Red LED is blinking every 1 s<br>• Display in the keypad: *LU* | Check mains voltage |
| | Inverter is inhibited<br>• Green LED is blinking<br>• Display in the keypad: CINH | Deactivate controller inhibit<br>• Note: Controller inhibit can be set via several sources !<br>• C00158 displays all active sources for controller inhibit. |
| | Automatic start is inhibited<br>(Bit 0 in C00142 = 1) | LOW/HIGH edge at RFR<br>If required, correct starting condition with C00142 |
| | DC-injection braking (DCB) is active | Deactivate DC injection brake |
| | Mechanical motor brake is not released | Release mechanical motor brake manually or electrically |
| | Quick stop (QSP) is active<br>• Display in the keypad: IMP | Deactivate quick stop<br>• Note: Quick stop can be set via several sources!<br>• C00159 displays all active sources for quick stop. |
| | Setpoint = 0 | Select setpoint |
| | JOG frequency = 0 at activated JOG setpoint | Set JOG setpoint in C00039/1...15 |
| | Trouble active | Clear fault |
| | With C00006 = 4, "SLVC: Vector control" has been set, but no motor parameter identification has been carried out. | Execute automatic motor parameter identification with the C00002/23 device command |
| | Assignment of several mutually exclusive functions with a signal source in C00701 | Correct configuration in C00701 |
| Motor rotates irregularly | Motor cable is defective | Check motor cable |
| | Maximum motor current in motor or generator mode is set too low | Adjust settings to the application: C00022: Imax in motor mode<br>C00023: Imax in generator mode |
| | Motor is underexcited or overexcited | Check parameterisation:<br>C00006: Motor control<br>C00015: VFC: V/f base frequency<br>C00016: VFC: Vmin boost |
| | Rated motor data (stator resistance, speed, current, frequency, voltage) and cos φ and/or magnetising inductance is not adapted to the motor data | Execute automatic motor parameter identification with the C00002/23 device command<br>- or -<br>Adjust motor parameters manually:<br>C00084: Motor stator resistance<br>C00087: Rated motor speed<br>C00088: Rated motor current<br>C00089: Rated motor frequency<br>C00090: Rated motor voltage<br>C00091: Motor cosine phi<br>C00092: Motor magnetising inductance |
| | Motor windings are wired incorrectly | Reverse from star connection to delta connection |

| Maloperation | Cause | Remedy |
|---|---|---|
| Motor consumes too much current | $V_{min}$ boost has been selected too high | Correct setting with C00016 |
| | V/f base frequency has been selected too low | Correct setting with C00015 |
| | Rated motor data (stator resistance, speed, current, frequency, voltage) and cos φ and/or magnetising inductance is not adapted to the motor data | Execute automatic motor parameter identification with the C00002/23 device command<br>- or -<br>Adjust motor parameters manually:<br>C00084: Motor stator resistance<br>C00087: Rated motor speed<br>C00088: Rated motor current<br>C00089: Rated motor frequency<br>C00090: Rated motor voltage<br>C00091: Motor cosine phi<br>C00092: Motor magnetising inductance |
| Motor parameter identification is aborted with error LP1 | Motor is too small compared to the rated device power (>1 : 3) | Use device with lower rated power |
| | DC injection brake (DCB) is active via terminal | Deactivate DC injection brake |
| Drive behaviour with vector control is not satisfactory | different | Optimise or manually adapt vector control |
| | | Execute automatic motor parameter identification with the C00002/23 device command |
| Torque dip in field weakening range<br>or<br>motor stalling when being operated in the field weakening range | Motor is overloaded | Check motor load |
| | Motor windings are wired incorrectly | Reverse from star connection to delta connection |
| | V/f reference point is set too high | Correct setting with C00015 |
| | Override point of field weakening is set too low | Correct setting with C00080 |
| An asynchronous motor with feedback rotates without control and with too low speed | Motor phases have been interchanged<br>• Thus the rotating field of the motor is not identical anymore with the rotating field of the feedback system.<br>• Therefore, the drive shows the following behaviour if V/f characteristic control (C00006 = 7) is performed:<br>  • The motor rotates faster than the speed setpoint by the value set in C00074.<br>  • After the controller has been enabled, the inverter will not stop if the speed setpoint = 0 or a quick stop (QSP) occurs.<br>  • Among other things, the final motor current depends on the value set for the $V_{min}$ boost and may rise up to $I_{max}$ which can trigger the "oC5: Ixt overload" fault message. | Check the phase position of the motor cable<br>If possible: Actuate the motor with deactivated feedback (C00006 = 6) and check direction of rotation of the motor |
| Motor phase (LP1) monitoring does not respond if the motor phases are interrupted | Monitoring is not active (C00597 = 0) | Activate monitoring (C00597 = 1) |

## 10.8 Operation without mains supply

### Note!

Observe the following restrictions for operation without mains supply:

**Safety state**

Inverters of the 8400 series can be optionally equipped with the integrated "Safe torque off (STO)" safety system.

- Up to and including version 13.xx.xx, the following applies:
  If merely the external 24-V supply of the inverter is switched on, the "Safe torque off" status in C00137 (bit 10) will not be updated.
- From version 14.00.00 the following applies:
  If merely the external 24-V supply of the inverter is switched on, the "Safe torque off" status in C00137 (bit 10) will be updated.

**Fan monitoring functions**

The fan monitoring functions are only active if the mains supply is switched on.

The following display parameters have a value of "0" if the mains supply is switched off and the external 24 V supply of the inverter is switched on:

| Parameters | Info |
|---|---|
| C00050 | MCTRL: Speed setpoint |
| C00051 | MCTRL: Actual speed value |
| C00052 | Motor voltage |
| C00053 | DC-bus voltage |
| C00054 | Motor current |
| C00058 | Output frequency |
| C00061 | Heatsink temperature |
| C00064/1...3 | Device utilisation (Ixt) |
| C00066 | Thermal motor load (I²xt) |
| C00177 | Switching cycles |
| C00725 | Current switching frequency |

## 10.9 Error messages of the operating system

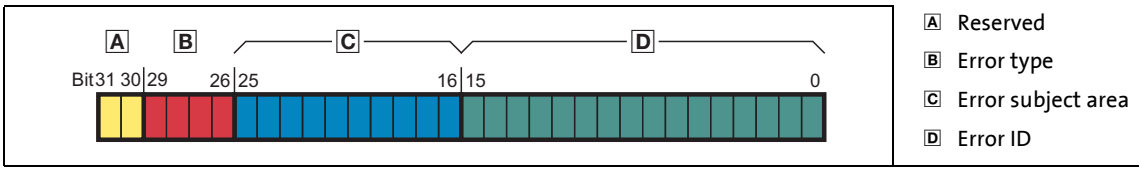
This chapter describes all error messages of the inverter operating system and possible causes & remedies.

**Tip!**

Each error message is also saved to the logbook in chronological order. ▸ Logbook (📖 700)

### 10.9.1 Structure of the 32-bit error number (bit coding)

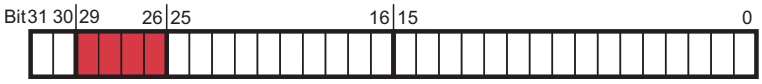
If an error occurs in the inverter, the internal fault memory saves a 32-bit value which contains the following information:




[10-1] Structure of the error number

- Display parameter: C00168
  - From version 13.00.00: C00162/1 additionally displays the 32-bit error number without error type, i.e. this error number only contains the error subject area and the error ID (bit 0 ... bit 25).
- The LS_DriveInterface system block shows the 32-bit error number at the outputs *wStateDetermFailNoLow* (Low Word) and *wStateDetermFailNoHigh* (High Word).
  - If the "Use 16BitFailNo." option is activated in C00148 (bit 15 = "1"), the LS_DriveInterface system block shows the short 16-bit error number at the output *wStateDetermFailNoLow*, and the value "0" at the output *wStateDetermFailNoHigh* (see the following chapter).
- For the sake of legibility, the error number in the logbook and in C00165 is displayed with the following syntax:
  [**Error type**].[**Error subject area no.**].[**Error ID**]

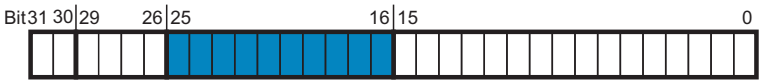
### Error type




The error type gives information about the behaviour/response of the inverter to the error. The error type for some device errors can also be parameterised.

| Bit 29 | Bit 28 | Bit 27 | Bit 26 | Meaning |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0: No Response |
| 0 | 0 | 0 | 1 | 1: Fault |
| 0 | 0 | 1 | 0 | 2: Trouble |
| 0 | 0 | 1 | 1 | 3: TroubleQSP |
| 0 | 1 | 0 | 0 | 4: WarningLocked |
| 0 | 1 | 0 | 1 | 5: Warning |
| 0 | 1 | 1 | 0 | 6: Information |

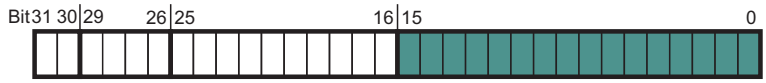
### Error subject area




The error subject area indicates the internal "function unit" of the inverter in which the error has occurred:

| Error subject area | | Assigned errors | Remedy possible by user? |
|---|---|---|---|
| No. | Name | | |
| 111 | Supply voltage | Errors that occur in connection with the supply voltage of the device. | Yes |
| 119 | Temperature | Errors that occur for temperature reasons. | Yes |
| 123 | Motor management / encoder | Errors that occur within the motor control or encoder evaluation. | Yes |
| 125 | Analog I/O integrated | Errors that occur in connection with the analog inputs and outputs. | Yes |
| 126 | Axis bus | Errors that occur in connection with the axis bus. | Yes |
| 127 | Extension module slot 1 | Errors that are reported by the extension module, and communication errors to the plugged-in extension module. | Yes if it is a fieldbus error. |
| 131 | CAN integrated (general) | Errors related to general CAN functions. | Yes |
| 135 | CAN process data object (PDO) | Errors that are explicitly only related to the CAN-PDO (process data objects). | Yes |
| 140 | Device configuration | Errors that occur due to incompatibilities of the plugged-in individual components (fieldbus module, safety module, et al.). | Yes if the error relates to a module plugged-in by the user. |
| 144 | Parameter set | Errors that occur in connection with the parameter set or the parameter set memory (memory module). | Yes if the error relates to a missing or incompatible memory module. |
| 145 | Device firmware (internal error) | Internal error of the device firmware. | No |
| 184 | MotionControlKernel | Errors that occur within the MotionControl basic functions (e.g. profile generation, brake control, positioning). | Yes |
| 400 | Defective device hardware | Errors that occur due to defective device hardware. | No |
| 444 | Fieldbus | Errors that occur in connection with fieldbus communication. | Yes |
| 980<br>...<br>983 | User error 1<br>...<br>User error 4 | Errors generated by the user (by the application) via the LS_SetError_1 system block. | Yes |
| 984<br>...<br>987 | User error 5<br>...<br>User error 8 | Errors generated by the user (by the application) via the LS_SetError_2 system block. | Yes |

### Error ID

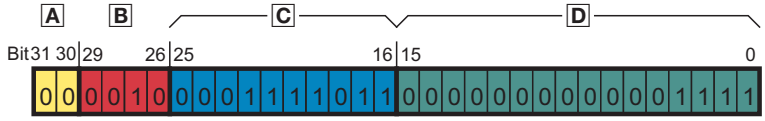



16-bit value (0 ... 65535) for error identification within the error subject area.

### Example for bit coding of the error number

C00168 displays the internal error number "142278671".

- This decimal value corresponds to the following bit sequence:

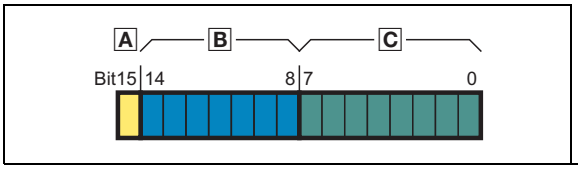



| Assignment | Information | Meaning in the example |
|---|---|---|
| 0 0 | Reserved | - |
| 0 0 1 0 | Error type | 2: Trouble |
| 0 0 0 1 1 1 1 0 1 1 | Error subject area | 123: Motor management / encoder |
| 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 | Error ID | 15: "LU: DC bus undervoltage" |

- Thus, error number "142278671" means:
  A DC bus undervoltage has been detected in the "Motor management / encoder" subject area.
  The error response is a "Fault".

### 10.9.2 Structure of the 16 bit error number (bit coding)

In addition to the 32-bit error number, a 16-bit error number is generated if an error occurs. It consists of the following information:

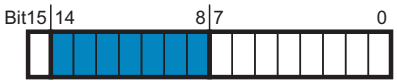



A Reserved
B Error subject area
C Error ID

[10-2] Structure of the error number

- Display parameter: C00160
- The LS_DriveInterface system block shows the 16-bit error number at the output *wStateDetermFailNoShort*.
- If the "Use 16BitFailNo." option is activated in C00148 (bit 15 = "1"), the LS_DriveInterface system block also shows the short 16-bit error number at the output *wStateDetermFailNoLow* (Low Word of the 32-bit error number).
  - The output *wStateDetermFailNoHigh* (High Word of the 32-bit error number) in this case is "0".
  - Advantage: The bus transfer of the error numbers is possible via a data word without changing the interconnection of the technology application.
- For the sake of legibility, the 16-bit error number in the logbook is displayed with the following syntax::
  [**Error subject area no.**].[**Error ID**]

#### Error subject area




The error subject area indicates the internal "function unit" of the inverter in which the error has occurred.

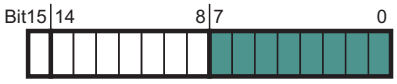
**Note!**

Due to the smaller value range (0 ...127), the number assignment to the error subject area differs from the 32-bit error number.

| Error subject area | | Assigned errors | Remedy possible by user? |
|---|---|---|---|
| No. | Name | | |
| 11 | Supply voltage | Errors that occur in connection with the supply voltage of the device. | Yes |
| 19 | Temperature | Errors that occur for temperature reasons. | Yes |
| 23 | Motor management / encoder | Errors that occur within the motor control or encoder evaluation. | Yes |
| 25 | Analog I/O integrated | Errors that occur in connection with the analog inputs and outputs. | Yes |
| 26 | Defective device hardware | Errors that occur due to defective device hardware. | No |
| 27 | Extension module slot 1 | Errors that are reported by the extension module, and communication errors to the plugged-in extension module. | Yes if it is a fieldbus error. |
| 31 | CAN integrated (general) | Errors related to general CAN functions. | Yes |
| 35 | CAN process data object (PDO) | Errors that are explicitly only related to the CAN-PDO (process data objects). | Yes |

| Error subject area | | Assigned errors | Remedy possible by user? |
|---|---|---|---|
| No. | Name | | |
| 40 | Device configuration | Errors that occur due to incompatibilities of the plugged-in individual components (fieldbus module, safety module, et al.). | Yes if the error relates to a module plugged-in by the user. |
| 44 | Parameter set | Errors that occur in connection with the parameter set or the parameter set memory (memory module). | Yes if the error relates to a missing or incompatible memory module. |
| 45 | Device firmware (internal error) | Internal error of the device firmware. | No |
| 54 | Fieldbus | Errors that occur in connection with fieldbus communication. | Yes |
| 84 | MotionControlKernel | Errors that occur within the MotionControl basic functions (e.g. profile generation, brake control, positioning). | Yes |
| 100<br>...<br>103 | User error 1<br>...<br>User error 4 | Errors generated by the user (by the application) via the LS_SetError_1 system block. | Yes |
| 104<br>...<br>107 | User error 5<br>...<br>User error 8 | Errors generated by the user (by the application) via the LS_SetError_2 system block. | Yes |

**Error ID**




8-bit value (0 ... 255) for error identification within the error subject area.

**Tip!**

All possible 16-bit error numbers are listed in the table entitled "Short overview (A-Z)" in the second column. (📖 721)

## 10.9.3 Reset error message

An error message with the response "Fault", "Trouble", "TroubleQSP" or "Warning locked" must be explicitly reset (acknowledged) after the cause of the error has been eliminated.

To reset (acknowledge) a pending error message, execute device command C00002/19 = "1".

**Tip!**

When an online connection to the inverter has been established, use the **Diagnostics** tab of »Engineer« and click **Reset error** to reset a pending error message.

### 10.9.4 Export error texts

All error texts of the inverter can be exported into a text file (*.txt) for further processing.

- The error text is preceded with the corresponding 32-bit error number (no error type) and the 16-bit error number, both as decimal number.
- If there is no corresponding 16-bit error number for a 32-bit error number, the field remains empty.

**Example**

Output of the German and English error texts:

```
32-BitError   16-BitError   DE-de                       EN-en
0             0             No error                 No error
111           11            Versorgungsspannung         Supply voltage
119 / 19
12323 : Motor management / encoder
125           25            E/A integriert              I/O integrated
...
26214416      6672          dH10: Lüfterausfall         dH10: Fan failure
26214505      6761          dH69: Abgleichdatenfehler   dH69: Adjustment fault
```



**How to export the error texts into a text file:**

1. Go to the *Project view* in the *Context menu* of the 8400 TopLine inverter and execute the **Export error texts...** command.
2. Define the following options in the *Export error texts* dialog box:
   - Output file and memory location
   - Languages to be exported (German/English/French)
   - Device/module to be exported
   - Separator (tabulator or semicolon)
   - Font (UTF8, standard font or ASCII)
3. Click **OK** to start the export.
   - After the export, a message appears indicating whether the export was successful.



**Tip!**

From version 13.00.00, the 32-bit number of the state-determining error is displayed in C00162/1 without error type.

If, for instance, the error texts are stored in a master control or on an operator panel, the error text to be displayed can be detected by reading the C00162/1 code.

### 10.9.5 Short overview (A-Z)

The table below contains all error messages of the inverter operating system in alphabetical order.

**Note!**

For the sake of legibility, the Logbook and C00165 display the 32-bit error number with the following syntax:
**[Error type].[Error subject area no.].[Error ID]**

In this documentation, "xx", a wildcard, stands for the error type since it is configurable for many error messages.

**Tip!**

If you click the cross-reference in the first column, "Error number", you will reach the detailed description of the respective error message in the following chapter "Cause & possible remedies". (📖 724)

| Error number 32 bits | 16 bits$_{hex}$ | 16 bits$_{dec}$ | Display in C00162/1 | Error message | Response (Lenze setting) | Can be set in | CAN Emergency Error Code |
|---|---|---|---|---|---|---|---|
| * Only the lower 8 bits of the adjustable error ID (C161/x) can be used. | | | | | | | |
| xx.0126.00001 | 0x1a01 | 6657 | 8257537 | Ab01: Axis bus timeout | No Reaction | C00591/1 | |
| xx.0126.00002 | 0x1a02 | 6658 | 8257538 | Ab02: Axis bus IO error | No Reaction | C00591/2 | |
| xx.0125.00001 | 0x1901 | 6401 | 8192001 | An01: AIN1_I < 4 mA | TroubleQuickStop | C00598/1 | 0xF000 |
| xx.0125.00002 | 0x1902 | 6402 | 8192002 | An02: AIN2_I < 4 mA | TroubleQuickStop | C00598/2 | 0xF000 |
| xx.0131.00006 | 0x1f06 | 7942 | 8585222 | CA06: CAN CRC error | No Reaction | C00592/1 | 0x8000 |
| xx.0131.00007 | 0x1f07 | 7943 | 8585223 | CA07: CAN bus warn | No Reaction | C00592/3 | 0x8000 |
| xx.0131.00008 | 0x1f08 | 7944 | 8585224 | CA08: CAN Bus Stopped | No Reaction | C00592/4 | 0x8000 |
| xx.0131.00011 | 0x1f0b | 7947 | 8585227 | CA0b: CAN HeartBeatEvent | No Reaction | C00592/5 | 0x8130 |
| xx.0131.00015 | 0x1f0f | 7951 | 8585231 | CA0F: CAN control word | Fault | C00594/1 | 0xF000 |
| xx.0127.00002 | 0x1b02 | 6914 | 8323074 | CE04: MCI communication error | No Reaction | C01501/1 | 0x7000 |
| xx.0127.00015 | 0x1b0f | 6927 | 8323087 | CE0F: MCI control word | Fault | C00594/2 | 0xF000 |
| xx.0135.00001 | 0x2301 | 8961 | 8847361 | CE1: CAN RPDO1 | No Reaction | C00593/1 | 0x8100 |
| xx.0135.00002 | 0x2302 | 8962 | 8847362 | CE2: CAN RPDO2 | No Reaction | C00593/2 | 0x8100 |
| xx.0135.00003 | 0x2303 | 8963 | 8847363 | CE3: CAN RPDO3 | No Reaction | C00593/3 | 0x8100 |
| xx.0131.00000 | 0x1f00 | 7936 | 8585216 | CE4: CAN Bus Off | No Reaction | C00592/2 | 0x8000 |
| xx.0135.00004 | 0x2304 | 8964 | 8847364 | CE5: CAN RPDO4 | No Reaction | C00593/4 | 0x8100 |
| xx.0140.00013 | 0x280d | 10253 | 9175053 | CI01: Module missing/incompatible | No Reaction | C01501/2 | 0x7000 |
| xx.0184.00001 | 0x5401 | 21505 | 12058625 | Ck01: Pos. HW limit switch | TroubleQuickStop | C00595/1 | 0x8600 |
| xx.0184.00002 | 0x5402 | 21506 | 12058626 | Ck02: Neg. HW limit switch | TroubleQuickStop | C00595/2 | 0x8600 |
| xx.0184.00007 | 0x5407 | 21511 | 12058631 | Ck03: Pos. SW limit position | TroubleQuickStop | C00595/3 | 0x8600 |
| xx.0184.00008 | 0x5408 | 21512 | 12058632 | Ck04: Neg. SW limit position | TroubleQuickStop | C00595/4 | 0x8600 |
| xx.0184.00153 | 0x5499 | 21657 | 12058777 | Ck05: Error following error 1 | Warning | C00595/5 | 0x8611 |
| xx.0184.00154 | 0x549a | 21658 | 12058778 | Ck06: Contouring error 2 | Warning | C00595/6 | 0x8611 |
| xx.0184.00155 | 0x549b | 21659 | 12058779 | Ck07: Travel range limit exceeded | TroubleQuickStop | C00595/7 | 0x8612 |
| xx.0184.00156 | 0x549c | 21660 | 12058780 | Ck08: Home position unknown | WarningLocked | C00595/8 | 0x8612 |
| xx.0184.08005 | 0x54cd | 21709 | 12066629 | Ck09: Invalid positioning mode | WarningLocked | C00595/9 | 0x8600 |
| xx.0184.08007 | 0x54cf | 21711 | 12066631 | Ck10: Implausible profile data | WarningLocked | C00595/10 | 0x8600 |
| xx.0184.08009 | 0x54d1 | 21713 | 12066633 | Ck11: Invalid operating mode | Warning | C00595/11 | 0x8600 |
| xx.0184.08014 | 0x54d6 | 21718 | 12066638 | Ck12: Invalid profile no. | WarningLocked | C00595/12 | 0x8600 |
| xx.0184.08015 | 0x54d7 | 21719 | 12066639 | Ck13: Error FB MCKCtrlInterface | Warning | C00595/13 | 0x8600 |

| Error number 32 bits | 16 bits$_{hex}$ | 16 bits$_{dec}$ | Display in C00162/1 | Error message | Response (Lenze setting) | Can be set in | CAN Emergency Error Code |
|---|---|---|---|---|---|---|---|
| xx.0184.00015 | 0x540f | 21519 | 12058639 | Ck14: Target position beyond SW limit position | WarningLocked | C00595/14 | 0x8600 |
| xx.0184.00005 | 0x5405 | 21509 | 12058629 | Ck15: Error status sign. brake | TroubleQuickStop | - | 0x8600 |
| xx.0184.00064 | 0x5440 | 21568 | 12058688 | Ck16: Time overflow manual control | Fault | - | |
| xx.0145.00035 | 0x2d23 | 11555 | 9502755 | dF10: AutoTrip Reset | Fault | C00189 | 0xF000 |
| xx.0145.00014 | 0x2d0e | 11534 | 9502734 | dF14: SW/HW invalid | Fault | - | |
| xx.0145.00015 | 0x2d0f | 11535 | 9502735 | dF15: DCCOM CU2 error | Fault | - | |
| xx.0145.00024 | 0x2d18 | 11544 | 9502744 | dF18: BU RCOM error | Fault | - | 0x6100 |
| xx.0145.00033 | 0x2d21 | 11553 | 9502753 | dF21: BU Watchdog | Fault | - | 0x6100 |
| xx.0145.00034 | 0x2d22 | 11554 | 9502754 | dF22: CU watchdog | Fault | - | 0x6100 |
| xx.0145.00025 | 0x2d19 | 11545 | 9502745 | dF25: CU RCOM error | Fault | - | 0x6100 |
| xx.0145.00026 | 0x2d1a | 11546 | 9502746 | dF26: Appl. watchdog | No Reaction | C00580/1 | 0x6200 |
| xx.0145.00050 | 0x2d32 | 11570 | 9502770 | dF50: Retain error | Fault | - | 0x6100 |
| xx.0145.00051 | 0x2d33 | 11571 | 9502771 | dF51: CuCcr error | Fault | - | 0x6100 |
| xx.0145.00052 | 0x2d34 | 11572 | 9502772 | dF52: BuCcr error | Fault | - | 0x6100 |
| xx.0400.00009 | 0x1a09 | 6665 | 26214409 | dH09: EEPROM power section | Fault | - | 0x5500 |
| xx.0400.00016 | 0x1a10 | 6672 | 26214416 | dH10: Fan failure | Warning | C00566 | 0x5000 |
| xx.0400.00104 | 0x1a68 | 6760 | 26214504 | dH68: Adjustment data error CU | Fault | - | 0x5500 |
| xx.0400.00105 | 0x1a69 | 6761 | 26214505 | dH69: Adjustment data error BU | Fault | - | 0x5500 |
| xx.0400.00106 | 0x1a6a | 6762 | 26214506 | dH70: ControlUnit is unequal to BaseUnit | Fault | - | 0x5500 |
| xx.0123.00094 | 0x175e | 5982 | 8061022 | FC01: Switching frequency reduction | No Reaction | C00590 | 0x2000 |
| xx.0123.00095 | 0x175f | 5983 | 8061023 | FC02: Maximum speed for Fchop | No Reaction | C00588 | 0xF000 |
| xx.0123.00099 | 0x1763 | 5987 | 8061027 | FC03: Field controller limitation | No Reaction | C00570/4 | 0xF000 |
| xx.0123.00057 | 0x1739 | 5945 | 8060985 | Id1: Motor data identification error | Fault | - | 0xF000 |
| xx.0123.00056 | 0x1738 | 5944 | 8060984 | Id2: Motor data identification error | Fault | - | 0xF000 |
| xx.0123.00058 | 0x173a | 5946 | 8060986 | Id3: CINH identification | WarningLocked | - | 0xF000 |
| xx.0123.00059 | 0x173b | 5947 | 8060987 | Id4: Resistance identification error | Warning | - | 0xF000 |
| xx.0123.00074 | 0x174a | 5962 | 8061002 | Id5: Pole position identification error | Fault | C00643/1 | |
| xx.0123.00075 | 0x174b | 5963 | 8061003 | Id6: Resolver ident error | Fault | - | |
| xx.0123.00060 | 0x173c | 5948 | 8060988 | Id7: Motor control does not match motor data | Information | C00571/1 | 0xF000 |
| xx.0123.00061 | 0x173d | 5949 | 8060989 | Id8: Speed sensor has not been set | Fault | C00571/2 | 0x7120 |
| xx.0123.00145 | 0x1791 | 6033 | 8061073 | LP1: Motor phase failure | No Reaction | C00597 | 0x3000 |
| xx.0123.00015 | 0x170f | 5903 | 8060943 | LU: DC bus undervoltage | Trouble | C00600/1 | 0x3100 |
| xx.0123.00016 | 0x1710 | 5904 | 8060944 | oC1: Power section - short circuit | Fault | - | 0x2000 |
| xx.0123.00017 | 0x1711 | 5905 | 8060945 | oC2: Power section - earth fault | Fault | - | 0x2000 |
| xx.0119.00050 | 0x1332 | 4914 | 7798834 | oC5: Ixt overload | Warning | C00604 | 0x2000 |
| xx.0123.00105 | 0x1769 | 5993 | 8061033 | oC6: I2xt motor overload | Warning | C00606 | 0x2000 |
| xx.0123.00007 | 0x1707 | 5895 | 8060935 | oC7: Motor overcurrent | Fault | - | 0x2000 |
| xx.0123.00030 | 0x171e | 5918 | 8060958 | oC10: Maximum current reached | No Reaction | C00609 | 0x2000 |
| xx.0123.00071 | 0x1747 | 5959 | 8060999 | oC11: Clamp operation active | Fault | - | 0xF000 |
| xx.0123.00065 | 0x1741 | 5953 | 8060993 | oC12: I2xt brake resistor overload | Fault | - | 0xF000 |
| xx.0123.00090 | 0x175a | 5978 | 8061018 | oC13: Maximum current for Fch exceeded | Fault | - | 0xF000 |
| xx.0123.00096 | 0x1760 | 5984 | 8061024 | oC14: Direct-axis current controller limitation | No Reaction | C00570/1 | 0xF000 |
| xx.0123.00097 | 0x1761 | 5985 | 8061025 | oC15: Cross current controller limitation | No Reaction | C00570/2 | 0xF000 |
| xx.0123.00098 | 0x1762 | 5986 | 8061026 | oC16: Torque controller limitation | No Reaction | C00570/3 | 0xF000 |
| xx.0123.00031 | 0x171f | 5919 | 8060959 | oC17: Clamp sets pulse inhibit | No Reaction | C00569/1 | 0xF000 |
| xx.0123.00034 | 0x1722 | 5922 | 8060962 | oC18: Current monitoring overload | No Reaction | C00584/1 | 0x2000 |
| xx.0119.00001 | 0x1301 | 4865 | 7798785 | oH1: Heatsink overtemperature | Fault | - | 0x4000 |
| xx.0119.00015 | 0x130f | 4879 | 7798799 | oH3: Motor temperature (X106) triggered | Fault | C00585 | 0x4000 |
| xx.0119.00000 | 0x1300 | 4864 | 7798784 | oH4: Heatsink temp. > shutdown temp. -5°C | No Reaction | C00582 | 0x4000 |
| xx.0119.00020 | 0x1314 | 4884 | 7798804 | oH6: Motor temperature MultiEncoder >= C121 | Warning | C00583/4 | |
| xx.0119.00002 | 0x1302 | 4866 | 7798786 | oH7: Motor temperature resolver >= C121 | Warning | C00583/3 | |

| Error number 32 bits | 16 bits$_{hex}$ | 16 bits$_{dec}$ | Display in C00162/1 | Error message | Response (Lenze setting) | Can be set in | CAN Emergency Error Code |
|---|---|---|---|---|---|---|---|
| xx.0119.00003 | 0x1303 | 4867 | 7798787 | oH9: Motor overtemperature resolver | Fault | C00583/1 | |
| xx.0119.00021 | 0x1315 | 4885 | 7798805 | oH12: Motor overtemperature MultiEncoder | Fault | C00583/2 | |
| xx.0123.00032 | 0x1720 | 5920 | 8060960 | oS1: Maximum speed limit reached | No Reaction | C00579 | 0x8400 |
| xx.0123.00033 | 0x1721 | 5921 | 8060961 | oS2: Max. motor speed | Fault | - | 0x8400 |
| xx.0123.00001 | 0x1701 | 5889 | 8060929 | ot1: Max. torque reached | No Reaction | C00608 | 0x8300 |
| xx.0123.00093 | 0x175d | 5981 | 8061021 | ot2: Speed controller output limited | No Reaction | C00567 | 0xF000 |
| xx.0123.00014 | 0x170e | 5902 | 8060942 | OU: DC bus overvoltage | Trouble | - | 0x3100 |
| xx.0144.00001 | 0x2c01 | 11265 | 9437185 | PS01: No memory module | Warning | - | 0x6300 |
| xx.0144.00002 | 0x2c02 | 11266 | 9437186 | PS02: Invalid par. set | Fault | - | 0x6300 |
| xx.0144.00003 | 0x2c03 | 11267 | 9437187 | PS03: Invalid device par. set | Fault | - | 0x6300 |
| xx.0144.00004 | 0x2c04 | 11268 | 9437188 | PS04: Invalid MCI par. set | Fault | - | 0x6300 |
| xx.0144.00007 | 0x2c07 | 11271 | 9437191 | PS07: Invalid memory module par. set | Fault | - | 0x6300 |
| xx.0144.00008 | 0x2c08 | 11272 | 9437192 | PS08: Invalid device par. | Fault | - | 0x6300 |
| xx.0144.00009 | 0x2c09 | 11273 | 9437193 | PS09: Invalid par. format | Fault | - | 0x6300 |
| xx.0144.00010 | 0x2c0a | 11274 | 9437194 | PS10: Memory module binding invalid | Fault | - | |
| xx.0123.00024 | 0x1718 | 5912 | 8060952 | Sd2: Resolver open circuit | Fault | C00603/2 | |
| xx.0123.00205 | 0x17cd | 6093 | 8061133 | Sd3: Open circuit HTL 2-fold or 4-fold | Fault | C00586 | 0x7300 |
| xx.0123.00027 | 0x171b | 5915 | 8060955 | Sd4: MultiEncoder open circuit | Fault | C00603/1 | |
| xx.0119.00012 | 0x130c | 4876 | 7798796 | Sd6: Error thermal detector resolver | Fault | C00583/5 | |
| xx.0123.00026 | 0x171a | 5914 | 8060954 | Sd7: Error encoder communication | Fault | C00603/4 | |
| xx.0123.00062 | 0x173e | 5950 | 8060990 | Sd8: Encoder angular drift monit. | No Reaction | C00603/3 | |
| xx.0123.00028 | 0x171c | 5916 | 8060956 | Sd9: Status message Hiperface | Information | C00603/5 | 0x7300 |
| xx.0123.00200 | 0x17c8 | 6088 | 8061128 | Sd10: Speed limit for feedback system 12 | Fault | C00607 | 0x7300 |
| xx.0123.00201 | 0x17c9 | 6089 | 8061129 | Sd11: Speed limit for feedback system 67 | Fault | C00607 | 0x7300 |
| xx.0119.00022 | 0x1316 | 4886 | 7798806 | Sd12: Error thermal detector MultiEncoder | Fault | C00583/6 | |
| xx.0123.00206 | 0x17ce | 6094 | 8061134 | Sd13: Inaccuracy SinCos | Information | - | 0x7300 |
| xx.0123.00029 | 0x171d | 5917 | 8060957 | Sd14: Position invalid Hiperface | Information | C00603/6 | 0x7300 |
| xx.0123.00207 | 0x17cf | 6095 | 8061135 | Sd15: Open circuit HTL 4-fold | Fault | C00605/1 | |
| xx.0123.00208 | 0x17d0 | 6096 | 8061136 | Sd16: Hiperface-SinCos deviation | Fault | C00603/8 | |
| xx.0111.00002 | 0x0b02 | 2818 | 7274498 | Su02: One mains phase is missing | Warning | C00565 | 0x3000 |
| xx.0111.00003 | 0x0b03 | 2819 | 7274499 | Su03: Too frequent mains switching | Fault | - | 0x3000 |
| xx.0111.00004 | 0x0b04 | 2820 | 7274500 | Su04: CU Insufficiently Supplied | Warning | - | 0x3000 |
| xx.0111.00006 | 0x0b06 | 2822 | 7274502 | Su06: Mains input overload | Fault | - | 0x3000 |
| Freely configurable user error messages (see LS_SetError_1 and LS_SetError_2) | | | | | | | |
| xx.0980.00001 | 25600$_{dec}$ + C161/1* | | | User error 1 | No Reaction | C00581/1 | 0x6200 |
| xx.0981.00002 | 25856$_{dec}$ + C161/2* | | | User error 2 | No Reaction | C00581/2 | 0x6200 |
| xx.0982.00003 | 26112$_{dec}$ + C161/3* | | | User error 3 | No Reaction | C00581/3 | 0x6200 |
| xx.0983.00004 | 26368$_{dec}$ + C161/4* | | | User error 4 | No Reaction | C00581/4 | 0x6200 |
| xx.0984.00001 | 26624$_{dec}$ + C161/5* | | | User error 5 | No Reaction | C00581/5 | 0x6200 |
| xx.0985.00002 | 26880$_{dec}$ + C161/6* | | | User error 6 | No Reaction | C00581/6 | 0x6200 |
| xx.0986.00003 | 27136$_{dec}$ + C161/7* | | | User error 7 | No Reaction | C00581/7 | 0x6200 |
| xx.0987.00004 | 27392$_{dec}$ + C161/8* | | | User error 8 | No Reaction | C00581/8 | 0x6200 |
| * Only the lower 8 bits of the adjustable error ID (C161/x) can be used. | | | | | | | |

### 10.9.6 Cause & possible remedies

This chapter contains all error messages of the inverter operating system in numerical order of the error numbers. The list provides detailed information on the response to the error message as well as information on the cause & possible remedies.

**Note!**

For the sake of legibility, the Logbook and C00165 display the error number with the following syntax:
**[Error type].[Error subject area no.].[Error ID]**

In this documentation, "xx", a wildcard, stands for the error type since it is configurable for many error messages.

**Tip!**

A list of all error messages of the inverter operating system in alphabetical order can be found in the previous chapter "Short overview (A-Z)" (📖 721).

**User error 1 [xx.0980.00000 ... xx.0980.65535]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| User error 1 has been tripped via the *bSetError1* input of the LS_SetError_1 system block. | User-defined. |

**User error 2 [xx.0981.00000 ... xx.0981.65535]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| User error 2 has been tripped via the *bSetError2* input of the LS_SetError_1 system block. | User-defined. |

**User error 3 [xx.0982.00000 ... xx.0982.65535]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/3 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| User error 3 has been tripped via the *bSetError3* input of the LS_SetError_1 system block. | User-defined. |

**User error 4 [xx.0983.00000 ... xx.0983.65535]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/4 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| User error 4 has been tripped via the *bSetError4* input of the LS_SetError_1 system block. | User-defined. |

**User error 5 [xx.0984.00000 ... xx.0984.65535]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/5 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| User error 5 has been tripped via the *bSetError1* input of the LS_SetError_2 system block. | User-defined. |

**User error 6 [xx.0985.00000 ... xx.0985.65535]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/6 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| User error 6 has been tripped via the *bSetError2* input of the LS_SetError_2 system block. | User-defined. |

**User error 7 [xx.0986.00000 ... xx.0986.65535]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/7 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| User error 7 has been tripped via the *bSetError3* input of the LS_SetError_2 system block. | User-defined. |

**User error 8 [xx.0987.00000 ... xx.0987.65535]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/8 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| User error 8 has been tripped via the *bSetError4* input of the LS_SetError_2 system block. | User-defined. |

**Su02: One mains phase is missing [xx.0111.00002]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00565 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| One mains phase of a three-phase supply has failed. | Check mains connection (terminal X100). |

### Su03: Too frequent mains switching [xx.0111.00003]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Too frequent mains switching of the power section.<br>• The device recognises if the power section is switched on and off too frequently.<br>• To protect internal charging connections from destruction, the device reports this error and prevents the controller inhibit. All other functions are active.<br>Use of a power supply module in the DC-bus connection, the DC terminals of which are connected downstream to the charging connection for the voltage DC bus (e.g. 9400 from 45 kW). | The error must be acknowledged by mains switching.<br>The charging circuit can only cool down when the mains is switched off.<br>• After switching the mains 3 times in one minute, there must be a switching pause of 9 minutes.<br>• Cyclic mains switching every 3 minutes is permissible.<br>From version 12.00.00 onwards, this power supply module can be used in the DC-bus connection by enabling it via C02865 (bit 8).<br>**Note:**<br>For further configuration of devices in the DC-bus connection with 8400, the DC terminals of which are connected downstream to the charging connection for the voltage DC bus (e.g. 9400 from 45 kW with 8400) contact Lenze. |

### Su04: CU insufficiently supplied [xx.0111.00004]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information |

| Cause | Remedy |
|---|---|
| After switching on the device, the 24V supply voltage for the control electronics is too low (100 ms after switch-on U is < 19V).<br>• The current supply voltage is displayed in C00065. | With internal supply voltage via the power electronics, the inverter must be replaced.<br>With external supply voltage, check the correct connection and/or the stability of the supply voltage. |

### Su06: Mains input overload [xx.0111.00006]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| In order to protect the device from overload, the following device outputs have a hardware detection in the mains input: 7.5 kW, 11 kW, 15 kW, 30 kW, 37 kW, 45 kW.<br>In case of the error message "Su06", this hardware detection has responded. | • Check whether all mains phases are connected (a 2-phase supply may be existent).<br>• Provide for sufficient cooling of the device. |

### Su07: 24V supply off [xx.0111.00007]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| | |

**oH4: Heatsink temp. > shutdown temp. -5°C [xx.0119.00000]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00582 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The heatsink temperature now only differs by 5 °C from the shutdown temperature of the motor. | Prevent further heating, i.e. reduce motor load or set controller inhibit so that the heatsink can cool down again. |

**oH1: Heatsink overtemperature [xx.0119.00001]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The heatsink temperature is higher than the fixed limit temperature (90 ° C).<br>Maybe the ambient temperature of the inverter is too high or the fan or its ventilation slots are dirty. | • Check control cabinet temperature.<br>• Clean filter.<br>• Clean the inverter.<br>• If required, clean or replace the fan.<br>• Provide for sufficient cooling of the device. |

**oH7: Motor temperature resolver >= C121 [xx.0119.00002]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/3 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The motor temperature detected via the resolver interface has reached the warning threshold set in C00121/1. | • Check motor temperature monitoring.<br>• Provide for sufficient cooling of the motor.<br>• Check terminals for open circuit or loose contact. |

**oH9: Motor overtemperature resolver [xx.0119.00003]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The motor temperature detected via the resolver interface has reached the fixed limit temperature (150 °C). | • Check motor temperature monitoring.<br>• Provide for sufficient cooling of the motor.<br>• Check terminals for open circuit or loose contact. |

**Sd6: Error thermal detector resolver [xx.0119.00012]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/5 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The signals of the encoder connected to the resolver interface are outside the defined operating range of the motor temperature detection. | • Check contacts of the encoder cable at the motor and inverter.<br>• Check the selected motor temperature sensor in C01190/1 as to whether it complies with the assembly in the motor.<br>• If required, switch off thermal sensor monitoring (C00583/5="0"). |

**oH3: Motor temperature (X106) triggered [xx.0119.00015]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00585 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The motor temperature monitoring function at the plug connector X106, terminal T1 /T2, has tripped.<br>Possible causes:<br>• The motor is overheated so that the thermal contact integrated into the motor has been switched.<br>• An open circuit or a loose contact at the connections mentioned above has occurred. | • Check motor temperature monitoring.<br>• Provide for sufficient cooling of the motor.<br>• Check terminals for open circuit or loose contact. |

**oH6: Motor temperature MultiEncoder >= C121 [xx.0119.00020]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/4 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The motor temperature detected via the encoder interface has reached the warning threshold set in C00121/2. | • Check motor temperature monitoring.<br>• Provide for sufficient cooling of the motor.<br>• Check terminals for open circuit or loose contact. |

**oH12: Motor overtemperature MultiEncoder [xx.0119.00021]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The motor temperature detected via the encoder interface has reached the fixed limit temperature (150 °C). | • Check motor temperature monitoring.<br>• Provide for sufficient cooling of the motor.<br>• Check terminals for open circuit or loose contact. |

**Sd12: Error thermal detector MultiEncoder [xx.0119.00022]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/6 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The signals of the encoder connected to the encoder interface are outside the defined operating range of the motor temperature detection. | • Check contacts of the encoder cable at the motor and inverter.<br>• Check the selected motor temperature sensor in C01190/2 as to whether it complies with the assembly in the motor.<br>• If required, switch off thermal sensor monitoring (C00583/6="0"). |

**oC5: Ixt overload [xx.0119.00050]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00604 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The Ixt overload check has tripped.<br>• Operating threshold = 100 % Ixt (adjustable in C00123)<br>Possible causes:<br>• Wrong dimensioning of the device with regard to its motor load.<br>• Load cycles are not complied with. | • Check and, if required, correct dimensioning of the device and the motor load with regard to technical data.<br>• Reduce motor load cycles (observe load cycles according to documentation). |

**ot1: Maximum torque reached [xx.0123.00001]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00608 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The device indicates that the maximally possible torque at the motor shaft has been reached.<br>• C00057 displays the current torque. | Reduce motor load. |

**oC7: Motor overcurrent [xx.0123.00007]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Maximum current monitoring has tripped.<br>• The instantaneous value of the motor current has exceeded the limit value set in C00939. | Check and, if required, correct dimensioning of the load with regard to the installed device power. |

**oU: DC bus overvoltage [xx.0123.00014]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☑ 1: Fault ☒ **2: Trouble** ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device has detected an overvoltage in the DC bus. To protect the device hardware, the inverter control is switched off.<br>• Depending on the configuration of the auto-start lock function, set C00142 so that, when this error is tripped, the inverter only restarts after the controller inhibit has been switched.<br>• If this error message remains active longer than the time set in C00601, a "Fault" is tripped. Otherwise, the deactivation of the error message causes the inverter control to be enabled again<br>  • In case of the control types VFCplus and SLVC, the motor voltage is approached to the voltage setpoint alongside a ramp.<br>  • From version 15.00.00, this voltage ramp can be set in C00983/2. If the described remedies are not possible or do not have any effect, it may be required to increase this voltage ramp as otherwise an overcurrent interruption may be caused. This only happens in case of high motor power and mass inertia so that the Lenze setting of 1 s should be sufficient in the majority of cases. | • Reduce load in generator mode.<br>• Use a brake resistor.<br>• Use a regenerative power supply unit.<br>• Establish a DC-bus connection.<br>• Select a braking method in C00175 which stops the ramp function generator when reaching the brake chopper threshold ("HlgStop").<br>• In case of servo control (SC), set the speed controller parameters correctly. |

**LU: DC bus undervoltage [xx.0123.00015]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00600/1 (☑ Adjustable response) |
|---|---|
| ☐ 0: No Reaction ☑ 1: Fault ☒ **2: Trouble** ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The device has detected a DC bus undervoltage. The inverter control is switched off because the drive properties of the motor control cannot be provided anymore due to the DC bus undervoltage.<br>• Depending on the configuration of the auto-start lock function, set C00142 so that, when this error is tripped, the inverter only restarts after the controller inhibit has been switched. | • Switch on mains supply or ensure sufficient supply via DC bus.<br>• Adjust setting in C00142 if required. |

**oC1: Power section - short circuit [xx.0123.00016]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device has recognised a short circuit of the motor phases. To protect the device electronics, the inverter control is switched off.<br>• Mostly, incorrectly executed motor connections are the cause.<br>• If the device is inappropriately dimensioned with regard to the motor load and the current limitation in the controller (Imax controller) is set incorrectly, this error message may also occur.<br>▸ Motor control: Defining current limits | • Check motor connections and the corresponding plug connector on the device.<br>• Only use permissible combinations of device power and motor power.<br>• Do not set the dynamics of the current limitation controller too high. |

**oC2: Power section - earth fault [xx.0123.00017]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device has recognised an earth fault at one of the motor phases. To protect the device electronics, the inverter control is switched off.<br>• Mostly, incorrectly executed motor connections are the cause.<br>• If motor filter, motor cable length, and cable type (shielding capacity) are dimensioned incorrectly, this error message may occur due to leakage currents to PE.<br>• If motor filters with additional terminals for +UG and –UG and devices greater or equal 3 kW are used, the earth fault detection may be triggered due to leakage currents to +UG and –UG.<br>• A cause can also be the use of shielded motor cables longer than 50 m. | • Check motor connections and the corresponding plug connector on the device.<br>• Use motor filters, cable lengths, and cable types recommended by Lenze.<br>• If motor filters with additional terminals for +UG and –UG and devices greater or equal 3 kW are used:<br>  • Deactivate earth-fault detection during operation by setting the filter time (C01770) to 250 ms.<br>• If motor cables longer than 50 m are used:<br>  • Increase filter time for earth-fault detection during operation (C01770). |

### Sd2: Open circuit resolver [xx.0123.00024]

| **Response** (Lenze setting printed in bold) | **Setting:** C00603/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| • Encoder signal interferences (EMC).<br>• Resolver cable interrupted.<br>• Resolver defective. | • Check resolver cable.<br>• Check resolver.<br>• Switch off monitoring (C00603/2 = "0: No reaction") when the resolver is not used. |

### Sd7: Encoder communication error [xx.0123.00026]

| **Response** (Lenze setting printed in bold) | **Setting:** C00603/4 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| • Encoder cable interrupted.<br>• Encoder defective. | • Check encoder cable<br>• Check encoder.<br>• Check parameter setting. |

### Sd4: Open circuit MultiEncoder [xx.0123.00027]

| **Response** (Lenze setting printed in bold) | **Setting:** C00603/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| • Encoder cable interrupted.<br>• Encoder defective.<br>• Supply voltage for the encoder not sufficient. | • Check encoder cable.<br>• Check encoder.<br>• Check the set encoder supply voltage in C00421. Adapt setting to the max. permissible encoder supply voltage. If required, additional voltage drops of the encoder cable have to be taken into consideration.<br>• Switch off monitoring (C00603/1 = "0: No reaction") when the encoder is not used. |

### Sd9: Status message Hiperface [xx.0123.00028]

| **Response** (Lenze setting printed in bold) | **Setting:** C00603/5 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☒ **6: Information** | |

| Cause | Remedy |
|---|---|
| Sd9 is output when the inverter has received a Hiperface encoder error code from the encoder during initialisation.<br>• The received error codes are displayed in C00491/1...4. The error codes are explained in the data sheet of the respective encoder. | Eliminating the fault described in the error code. Afterwards, reinitialising the encoder by a renewed writing of the encoder type in C00422. |

### Sd14: Position invalid Hiperface [xx.0123.00029]

| **Response** (Lenze setting printed in bold) | **Setting:** C00603/6 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☒ **6: Information** | |

| Cause | Remedy |
|---|---|
| No standstill of the Hiperface encoder during initialisation. Hence, the absolute Hiperface track and the incremental SinCos track cannot be initiated exactly and the actual position cannot be detected correctly. | Stopping encoder movement. Afterwards reinitialising the encoder by a renewed writing of the encoder type in C00422. |

### oC10: Maximum current reached [xx.0123.00030]

| **Response** (Lenze setting printed in bold) | **Setting:** C00609 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The device displays that the maximum current has been reached. | • Check and, if required, correct dimensioning of the load with regard to the installed device power.<br>• Check the maximum current settings in C00022 (Imax in motor mode) and C00023 (Imax in generator mode). |

### oC17: Clamp sets pulse inhibit [xx.0123.00031]

| **Response** (Lenze setting printed in bold) | **Setting:** C00569/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Due to a short overcurrent, the inverter was switched off for a short time (clamp disconnection). | • Check and, if required, correct dimensioning of the load with regard to the installed device power.<br>• Reduce the dynamics of the setpoint change or speed control. |

### oS1: Maximum speed limit reached [xx.0123.00032]

| **Response** (Lenze setting printed in bold) | **Setting:** C00579 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The device has recognised that the maximum speed has been reached. | • Limit setpoint selection to maximum values.<br>• Adjust set speed limitation (C00909) and frequency limitation (C00910). |

### oS2: Max. motor speed [xx.0123.00033]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device has recognised that the maximally permissible motor speed has been reached. | • Limit setpoint selection to the maximally permissible motor speed.<br>• If required, adapt set maximum motor speed (C00965). |

### oC18: Current monitoring overload [xx.0123.00034]

| **Response** (Lenze setting printed in bold) | **Setting:** C00584/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The current monitoring overload has tripped because the apparent motor current has exceeded the switch-off threshold set in C00124/1 for the delay time set in C00563/1C00563/1. | • Reduce overload.<br>• Increase switch-off threshold (C00124/1). |

**Id2: Motor data identification error [xx.0123.00056]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| During the identification of the motor parameters, an error has occurred that caused an abortion of the identification process.<br>Possible causes:<br>• Interrupted motor cable.<br>• Switched-off power section during the identification.<br>• Implausible start parameter settings. | • Check the motor connections and the corresponding plug connector on the device and, if necessary, the motor terminal box.<br>• Correct start parameters for the motor parameter identification (motor nameplate data).<br>• Stable power supply of the device. |

**Id1: Motor data identification error [xx.0123.00057]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| During the identification of motor parameters, an error has occurred.<br>Possible causes:<br>• Interrupted motor cable.<br>• Switched-off power section during the identification.<br>• Implausible start parameter settings. | • Check the motor connections and the corresponding plug connector on the device and, if necessary, the motor terminal box.<br>• Correct start parameters for the motor parameter identification (motor nameplate data).<br>• Stable power supply of the device. |

**Id3: CINH identification [xx.0123.00058]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☒ **4: WarningLocked** ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device has detected controller inhibit during the motor data identification.<br>• This cancels the identification process. The Lenze setting of the motor data is used. | • Do not set controller inhibit during the motor data identification.<br>• Do not execute any device function which may activate controller inhibit. |

**Id4: Resistor identification error [xx.0123.00059]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device has recognised that an error has occurred in the calculation of the motor cable resistance.<br>• The parameters for cable cross-section and cable length are implausible. | Enter sensible values for cable cross-section and motor cable length. |

### Id7: Motor control does not match motor data [xx.0123.00060]

| **Response** (Lenze setting printed in bold) | **Setting:** C00571/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☒ **6: Information** | |

| Cause | Remedy |
|---|---|
| At controller enable, the device has detected that the motor control type set in C00006 cannot control the motor type set.<br>• Example: Motor nameplate data for an asynchronous motor have been entered; however, a motor control type for a synchronous motor is set in C00006.<br>**Note:**<br>Since the "VFCplus" control types are able to control every motor to a certain extent, this error message will never occur here. | Enter correct motor nameplate data and set a matching motor control type in C00006:<br>• Motor nameplate data asynchronous motor → motor control type must be ASM, SLVC or VFCplus servo control.<br>• Motor nameplate data synchronous motor → motor control type must be PSM, SLPSM or VFCplus servo control. |

### Id8: Speed encoder has not been set [xx.0123.00061]

| **Response** (Lenze setting printed in bold) | **Setting:** C00571/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| When being In controller enable status, the device has detected that a motor control type with feedback has beent set in C00006 , but no speed sensor has been set in C00495. | Set the speed sensor in C00495.<br>**Note:** The error can only be reset if the settings in C00006 and C00495 match. |

### Sd8: Encoder angular drift monit. [xx.0123.00062]

| **Response** (Lenze setting printed in bold) | **Setting:** C00603/3 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| • Encoder signal interferences (EMC).<br>• Encoder cable interrupted.<br>• Encoder defective.<br>• Faulty parameter setting of the encoder. | • Check encoder cable, use shorter encoder cable if required.<br>• Check encoder.<br>• Check parameter setting.<br>• If required, switch off monitoring (C00603/3). |

### oC12: I2xt overload - brake resistor [xx.0123.00065]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Too frequent and too long braking processes. | Check drive dimensioning. |

### oC11: Clamp operation active [xx.0123.00071]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device indicates that the "CLAMP" overcurrent limitation has been activated.<br>• A permanent clamp operation causes an overload disconnection. | Reduce setpoint generation dynamics or motor load. |

#### Id5: Pole position identification error [xx.0123.00074]

| **Response** (Lenze setting printed in bold) | **Setting:** C00643/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| With 360° pole position identification:<br>• The rotor position detected via the encoder system does not comply with the controlled output position (plausibility check).<br>With pole position identification with minimum movement:<br>• The encoder system has detected a movement that is greater than the permitted movement set in C00645/1. | With 360° pole position identification:<br>• Check parameter setting of the pole position identification.<br>• If required, adapt error tolerance for plausibility check in C00645/2.<br>With pole position identification with minimum movement:<br>• If required, adjust permitted movement in C00645/1. |

#### Id6: Resolver ident. error [xx.0123.00075]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| • A controller inhibit was set during resolver error identification.<br>• A time-out occurred while the algorithm was being processed. This error may occur if more than 60 seconds pass by between the setting of the "Resolver error identification" device command and the enable of the inverter.<br>• The setpoint speed was too small (\|nset\| < 500 rpm).<br>• The setpoint speed was not traversed for at least 1 second. | • Reduce the gain of the speed controller.<br>• Check the shielding of the motor and encoder.<br>• Check encoder/encoder connection.<br>• Controller is enabled after the "Resolver error identification" device command has been set.<br>• Ensure that the speed profile is traversed for at least 1 second at constant setpoint speed (\|nset\| > 500 rpm). |

#### oC13: Maximum current for Fch exceeded [xx.0123.00090]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device has detected a motor current which exceeds the maximum current limit at permanent switching frequency of the inverter.<br>• If a permanent switching frequency inverter is set, a certain limit arises for the maximum current, depending on the setting. If this current limit is exceeded due to a load impulse or overload, an error message is displayed. | • Observe the maximum current setting depending on the set switching frequency of the inverter.<br>• Reduce the required load or setting of the dynamic switching frequency if necessary. |

#### ot2: Speed controller output limited [xx.0123.00093]

| **Response** (Lenze setting printed in bold) | **Setting:** C00567 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The output of the speed controller has reached the internal limit value. In this status, the speed controller is not able anymore to correct the system deviation.<br>• Only during "Closed loop" operation or with vector control (SLVC). | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint generation dynamics if necessary.<br>▸ Motor control |

**FC01: Switching frequency reduction [xx.0123.00094]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00590 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Load-dependent switching frequency reduction | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint generation dynamics if necessary.<br>▸ Motor control |

**FC02: Maximum speed for Fchop [xx.0123.00095]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00588 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Maximum speed for chopper frequency has been reached.<br>• The maximum speed has been exceeded depending on the switching frequency. | Select the correct maximum speed as a function of the switching frequency.<br>▸ Motor control: Determine speed limits |

**oC14: Direct-axis current controller limitation [xx.0123.00096]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00570/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Direct-axis current controller limitation is active. | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint generation dynamics if necessary.<br>▸ Motor control |

**oC15: Cross current controller limitation [xx.0123.00097]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00570/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Cross current controller limitation is active. | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint generation dynamics if necessary.<br>• Check parameter setting of the current controller with regard to the motor controllers (e.g. reduce Vp).<br>▸ Motor control |

**oC16: Torque controller limitation [xx.0123.00098]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00570/3 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Actuator limitation according to speed controller. | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint generation dynamics if necessary.<br>▸ Motor control |

#### FC03: Field controller limitation [xx.0123.00099]

| Response (Lenze setting printed in bold) | Setting: C00570/4 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The output of the field controller has reached its maximum limit value. The drive is at the torque limit in the field weakening range. | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint from the field weakening range if necessary.<br>▸ Motor control |

#### oC6: I2xt overload - motor [xx.0123.00105]

| Response (Lenze setting printed in bold) | Setting: C00606 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Thermal overload of the motor. | Only self-ventilated motors can be monitored using the I2xt function.<br>• Check whether is it a self-ventilated motor. If not, set C00606 to "0: No Reaction".<br>• Observe load requirements.<br>• Correct dimensioning if necessary.<br>• For VFCplus control type: Check Vmin boost (C00016).<br>▸ Set Vmin boost |

#### LP1: Motor phase failure [xx.0123.00145]

| Response (Lenze setting printed in bold) | Setting: C00597 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Motor phase failure - power section<br>• This error message is displayed if a motor phase carries less current of one half-wave than set in C00599. | • Check the motor connections and the corresponding plug connector on the device and, if necessary, the motor terminal box.<br>• Check the trigger threshold (C00599). |

#### Sd10: Speed limit - feedback system 12 [xx.0123.00200]

| Response (Lenze setting printed in bold) | Setting: C00607 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Maximally permissible speed of the feedback system connected to DI1/DI2 reached. | Reduce speed of the rotation shaft/feedback system.<br>$n_{encoder}$ <= ($f_{max}$ x 60) / encoder increments<br>(for $f_{max}$ = 10 kHz) |

#### Sd11: Speed limit for feedback system 67 [xx.0123.00201]

| Response (Lenze setting printed in bold) | Setting: C00607 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Maximally permissible speed of the feedback system connected to DI6/DI7 reached. | Reduce speed of the rotation shaft/feedback system.<br>$n_{encoder}$ <= ($f_{max}$ x 60) / encoder increments<br>(for $f_{max}$ = 10 kHz) |

### Sd3: Open circuit HTL 2-fold or 4-fold [xx.0123.00205]

| **Response** (Lenze setting printed in bold) | **Setting:** C00586 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| • HTL encoder cable interrupted.<br>• HTL encoder is defective.<br>Note: The reason can also be a very dynamic acceleration or an approach against a blocked motor shaft, e.g. with a a closed holding brake or when referencing to positive stop (mode 14/15) and a waiting time (C01223) >100 ms. | • Check HTL encoder cable.<br>• Check HTL encoder.<br>• Check related terminals.<br>• Switch off monitoring (C00586 = "0: No reaction") when the HTL encoder is not used. |

### Sd13: Inaccuracy SinCos [xx.0123.00206]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☒ **6: Information** |

| Cause | Remedy |
|---|---|
| Only in case of "Multi-Encoder" speed encoder selection (C00495 = 3):<br>A plausibility check for the digital and analog signals of the absolute value encoder (Hiperface) or sin/cos encoder has responded.<br>• The reason for this are EMC interferences.<br>• This information is given since angle errors (position error) can occur in such a case. | Eliminate EMC interference. |

### Sd15: Open circuit TL 4-fold [xx.0123.00207]

| **Response** (Lenze setting printed in bold) | **Setting:** C00605/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| From version 15.00.00 onwards, Sd15 is triggered if due to different signal levels (4-fold evaluation of the digital inputs) an open circuit has been detected. The 4-fold evaluation gets active if C00495/1 or C00490/0 with "5: Encoder signal FreqIn1267" is selected as speed or position encoder. | • Check HTL encoder cable.<br>• Check HTL encoder.<br>• Check wiring of the input terminals for open circuit or loose contact. |

### Sd16: Hiperface-SinCos deviation [xx.0123.00208]

| **Response** (Lenze setting printed in bold) | **Setting:** C00603/8 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The deviation between the position of a Hiperface encoder detected from the evaluation of the SinCos track and the position of the Hiperface track detected in parallel is monitored. From version 15.00.00 onwards, Sd16 is triggered if the electrical deviation is higher than 45° (depending on the number of motor pole pairs). Monitoring is only active as long as the motor speed is below 100 rpm for at least 128 ms. Above this speed, no evaluation is possible. | |

**An01: AIN1_I < 4 mA [xx.0125.00001]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00598/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☑ 2: Trouble ☒ **3: TroubleQuickStop** ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Open-circuit monitoring for analog input 1 has tripped.<br>• Only if the analog input has been configured as a current loop of 4 ... 20 mA (C00034/1 = 2). | • Check wiring of the analog X3/A1I input terminal for open circuit.<br>• Check minimum current values of the signal sources. |

**An02: AIN2_I < 4 mA [xx.0125.00002]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00598/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☑ 2: Trouble ☒ **3: TroubleQuickStop** ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Open-circuit monitoring for analog input 2 has tripped.<br>• Only if the analog input has been configured as a current loop of 4 ... 20 mA (C00034/2 = 2). | • Check wiring of the analog X3/A2I input terminal for open circuit.<br>• Check minimum current values of the signal sources. |

**Ab01: Axis bus time-out [xx.0126.00001]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00591/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The axis bus data error monitoring has been triggered as the monitoring time for data errors set in C02431/3 has been exceeded.<br>• The current number of incorrectly transmitted telegrams is displayed in C02438/1.<br>• The current number of incorrectly received telegrams is displayed in C02438/2.<br>• The current axis bus error status is displayed in C02436. | • Check wiring and bus terminating resistor.<br>• Assign different IDs to nodes.<br>• Eliminate electrical interference (e.g. EMC). |

**Ab02: Axis bus IO error [xx.0126.00002]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00591/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| An axis bus node has put the IO axis bus into the "error" status with parameterised "master/slave" function ("release cord" principle).<br>• In the "Error" status, all nodes start their adjustable response, e.g. a synchronised braking of the drive system or the master only brakes the network to standstill. | Remove the fault of the node which pulled the "release cord".<br>**Note:** The "error" status can only be reset by the node defined as "master". |

**CE04: MCI communication error [xx.0127.00002]**

| **Response** (Lenze setting printed in bold) | **Setting:** C01501/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| Communication error with extension module in slot 1. | • Eliminate EMC interference.<br>• Switch off inverter, plug in module correctly, switch on inverter again.<br>• Switch the mains or restart inverter.<br>• Replace module/inverter.<br>• Please contact Lenze if the problem occurs again. |

**CE0F: MCI control word [xx.0127.00015]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00594/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Bit 14 ("SetFail") of the wMciCtrl control word of the LS_DriveInterface system block has been set. | Trace back signal source on the bus (e.g. PROFIBUS) that sets bit 14 ("SetFail"). |

**CE4: CAN bus off [xx.0131.00000]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00592/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: "Bus off" status<br>• Received too many faulty telegrams.<br>• Damaged cable (e.g. loose contact).<br>• Two nodes with the same ID. | • Check wiring and bus terminating resistor.<br>• Set identical baud rate for each bus node.<br>• Assign different IDs to nodes.<br>• Eliminate electrical interference (e.g. EMC). |

**CA06: CAN CRC error [xx.0131.00006]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00592/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: A faulty CAN telegram has been detected. | • Check wiring and bus terminating resistor.<br>• Eliminate electrical interference (e.g. EMC). |

**CA07: CAN bus warning [xx.0131.00007]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00592/3 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: Incorrect transmission or reception of more than 96 CAN telegrams.<br>• The current number of incorrectly transmitted CAN telegrams is displayed in C00372/1.<br>• The current number of incorrectly received CAN telegrams is displayed in C00372/2.<br>• The current CAN error status is displayed in C00345. | • Check wiring and bus terminating resistor.<br>• Set identical baud rate for each bus node.<br>• Assign different IDs to nodes.<br>• Eliminate electrical interference (e.g. EMC). |

### CA08: CAN bus stopped [xx.0131.00008]

| Response (Lenze setting printed in bold) | Setting: C00592/4 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: The device has received the "Stop Remote Node" NMT telegram. | Check CAN master (NMT master). |

### CA0b: CAN HeartBeatEvent [xx.0131.00011]

| Response (Lenze setting printed in bold) | Setting: C00592/5 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: Cyclic node monitoring<br>• Being a Heartbeat consumer, the device has not received a Heartbeat telegram from Heartbeat producer 1 ... 15 within the defined time.<br>• The current states of the Heartbeat producers are displayed in C00347/1...15. | • Reactivate Heartbeat producers by mains switching, restarting the inverter, or a CAN Reset Node.<br>• Reparameterise CAN Heartbeat producer time or switch off consumer monitoring and reset error status if latched.<br>▸ Heartbeat protocol |

### CA0F: CAN control word [xx.0131.00015]

| Response (Lenze setting printed in bold) | Setting: C00594/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Bit 14 ("SetFail") in the wCANControl control word of the LS_DriveInterface system block has been set. | Trace back signal source on the CAN bus that sets bit 14 ("SetFail"). |

### CE1: CAN RPDO1 [xx.0135.00001]

| Response (Lenze setting printed in bold) | Setting: C00593/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: Time monitoring for RPDO1 has been triggered.<br>• RPDO1 has not been received within the monitoring time set in C00357/1 or was faulty. | • Set correct telegram length for CAN master (transmitter).<br>• Eliminate electrical interference (e.g. EMC).<br>• Adjust monitoring time C00357/1 or switch off time monitoring. |

### CE2: CAN RPDO2 [xx.0135.00002]

| Response (Lenze setting printed in bold) | Setting: C00593/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: Time monitoring for RPDO2 has been triggered.<br>• RPDO2 has not been received within the monitoring time set in C00357/2 or was faulty. | • Set correct telegram length for CAN master (transmitter).<br>• Eliminate electrical interference (e.g. EMC).<br>• Adjust monitoring time C00357/2 or switch off time monitoring. |

#### CE3: CAN RPDO3 [xx.0135.00003]

| **Response** (Lenze setting printed in bold) | **Setting:** C00593/3 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: Time monitoring for RPDO3 has been triggered.<br>• RPDO3 has not been received within the monitoring time set in C00357/3 or was faulty. | • Set correct telegram length for CAN master (transmitter).<br>• Eliminate electrical interference (e.g. EMC).<br>• Adjust monitoring time C00357/3 or switch off time monitoring. |

#### CE5: CAN RPDO4 [xx.0135.00004]

| **Response** (Lenze setting printed in bold) | **Setting:** C00593/4 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: Time monitoring for RPDO4 has been triggered.<br>• RPDO4 has not been received within the monitoring time set in C00357/4 or was faulty. | • Set correct telegram length for CAN master (transmitter).<br>• Eliminate electrical interference (e.g. EMC).<br>• Adjust monitoring time C00357/4 or switch off time monitoring. |

#### CI01: Module missing/incompatible [xx.0140.00013]

| **Response** (Lenze setting printed in bold) | **Setting:** C01501/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The optional communication module has been removed or there is a connection problem or incompatibility with the standard device. | • Check connection between the communication module and standard device.<br>• Check if the module is plugged in correctly.<br>• In case of an incompatibility, either the module or the software of the standard device is out of date. In this case, please contact Lenze. |

#### PS01: No memory module [xx.0144.00001]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Memory module is either not available or not snapped into place correctly. | • If a memory module has been provided: Plug the memory module into the slot of the standard device intended for this purpose.<br>• If a memory module has been provided: Check if the memory module has been plugged-in correctly. |

**PS02: Par. set invalid [xx.0144.00002]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The parameter set stored in the memory module is invalid. The reason for this can be as follows:<br>• Incomplete storage of the parameter set due to voltage failure.<br>• The plugged-in module stems from a device with new firmware (compare C00099) or from a different device type (e.g. 8400 BaseLine). | The error can only be removed by loading the Lenze setting with the C00002/1 = "1: On / start" device command.<br>• In order to prevent the error, do not switch off the voltage during the saving process.<br>• If the parameter set is to be transferred from one device with a higher version to a device with a lower version, use the "copy parameter set" function of the keypad. Make sure that you do not use functions that are not available in the older device. |

**PS03: Par. set device invalid [xx.0144.00003]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The parameter set saved to the memory module is incompatible to the standard device.<br>• An incompatibility of the parameter set is caused e.g. when the memory module of an 8400 HighLine is plugged into an 8400 StateLine or the parameter set in the memory module has a higher version than expected by the standard device.<br>If the parameter set stored in the memory module is compatible with the standard device but has a different (lower) version, this message is only output as "Information". The message can be eliminated by saving the parameter set again.<br>**Note:**<br>If you save the parameter set to a higher device version, you can no longer load this parameter set to a lower device version.<br>▸ Replacement of the inverter | When the memory modules are replaced, observe the compatibility:<br>• OK: StateLine V2.0 to StateLine V3.0<br>• OK: StateLine V2.0 to HighLine V2.0<br>• Not OK: HighLine Vx.x to StateLine Vx.x<br>• Not OK: StateLine V3.0 to StateLine < V3.0 |

**PS04: Par. set Mci invalid [xx.0144.00004]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The parameter set saved to the communication module is incompatible to the standard device.<br>• An incompatibility of the parameter set is caused e.g. when the MCI module parameters in the memory module do not match the plugged communication module. | When the memory modules are replaced, observe the compatibility:<br>• Not OK: Profibus V1.0 to EtherCAT V1.0 |

**PS07: Par. memory module invalid [xx.0144.00007]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The parameter set saved to the memory module is invalid.<br>• The error occurs while loading the parameter set.<br>• The memory module plugged in the device lacks a code or a code is incorrect. | Please contact Lenze. |

**PS08: Par. device invalid [xx.0144.00008]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The parameter set in the device is invalid.<br>• The error occurs while loading the parameter set.<br>• One code in the device is incorrect. | Please contact Lenze. |

**PS09: Par. format invalid [xx.0144.00009]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The code format is invalid.<br>• The error occurs while loading the parameter set. | Please contact Lenze. |

**PS10: Memory module binding invalid [xx.0144.00010]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Device personalisation is active: The binding ID of the memory module does not comply with the binding ID of the inverter. | • Use memory module/ inverter with matching binding IDs.<br>• Contact machine manufacturer.<br>**Note:**<br>It is not possible for Lenze to modify a replacement device via special accesses in such a way that it cooperates with a personalised memory module. |

**PS11: Lenze setting loaded [xx.0144.00011]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| | |

**PS12: Parameter sets loaded [xx.0144.00012]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| | |

**PS13: Parameter sets saved [xx.0144.00013]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| | |

**dF01: FW updated [xx.0145.00001]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| | |

**dF14: SW-HW invalid [xx.0145.00014]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Please contact Lenze. |

**dF15: DCCOM CU2 error [xx.0145.00015]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Please contact Lenze. |

**dF18: BU RCOM error [xx.0145.00024]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Please contact Lenze. |

**dF25: CU RCOM error [xx.0145.00025]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Please contact Lenze. |

**dF26: Appl. watchdog [xx.0145.00026]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00580/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Time-out of the application. The required computing time of the application exceeds the available computing time. | Reduction of the function block interconnection or the complexity of the application. |

**dF21: BU watchdog [xx.0145.00033]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Device error | Please contact Lenze. |

**dF22: CU watchdog [xx.0145.00034]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Device error | Please contact Lenze. |

**dF10: AutoTrip reset [xx.0145.00035]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00189 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| Too frequent auto-trip reset. | • Check the error cause that activates the auto-trip reset.<br>• Eliminate error cause and reset (acknowledge) error manually afterwards. |

**dF50: Retain error [xx.0145.00050]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| An error has occurred when accessing retain data.<br>• Either caused by an internal hardware error or by lack of mains switching after a firmware download. | Mains switching<br>• Please contact Lenze if the problem occurs again. |

**dF51: CuCcr error [xx.0145.00051]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Device error | Mains switching<br>• Please contact Lenze if the problem occurs again. |

#### dF52: BuCcr error [xx.0145.00052]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Device error | Mains switching<br>• Please contact Lenze if the problem occurs again. |

#### Ck01: Pos. HW limit switch [xx.0184.00001]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: The hardware limit switch in positive traversing direction has tripped.<br>• The *bLimitSwitchPos* input for travel range monitoring via positive hardware limit switch has been set to FALSE (fail-safe). | Reset error message and retract limit switch. |

#### Ck02: Neg. HW limit switch [xx.0184.00002]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: The hardware limit switch in negative traversing direction has tripped.<br>• The *bLimitSwitchNeg* input for travel range monitoring via negative hardware limit switch has been set to FALSE (fail-safe). | Reset error message and retract limit switch. |

#### Ck15: Error status sign. brake [xx.0184.00005]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| MCK: The status monitoring of the holding brake control has tripped. | • Check configuration of the *bMBrakeApplied* input for status detection of the brake (via a switching contact at the brake).<br>• Check wiring/function of the switching contact.<br>• Adapt waiting time (C02589/3).<br>• Deactivate status monitoring (via bit 5 in C02582). |

#### Ck03: Pos. SW limit position [xx.0184.00007]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/3 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: The device has detected that the position is outside the positive software limit position (C01229/1). | • Increase permissible traversing range (change setting of the software limit positions).<br>• Deactivate software limit position monitoring. |

**Ck04: Neg. SW limit position [xx.0184.00008]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/4 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: The device has detected that the position is outside the negative software limit position (C01229/2). | • Increase permissible traversing range (change setting of the software limit positions).<br>• Deactivate software limit position monitoring. |

**Ck14: Target position outside SW limit position [xx.0184.00015]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/14 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☒ **4: WarningLocked** ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: It has been attempted to position a target outside the software limit positions (C01229/1 and C01229/2). | • Select a target within the software limit positions.<br>• Increase permissible traversing range (change setting of the software limit positions).<br>• Deactivate software limit position monitoring. |

**Ck16: Time overflow manual operation [xx.0184.00064]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| PC manual control: The connection monitoring has tripped.<br>• The online connection between the PC and the inverter has been interrupted for a longer period of time than the timeout set in C00464/1. | • Check communication link between PC and inverter.<br>• Check voltage supply/function of the inverter.<br>• Adjust the timeout (C00464/1). |

**Ck05: Following error 1 [xx.0184.00153]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/5 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: Following error limit 1 (C01215/1) has been exceeded. | • Optimise control mode.<br>• Increase following error limit.<br>• Deactivate following error monitoring. |

**Ck06: Following error 2 [xx.0184.00154]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/6 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: Following error limit 2 (C01215/2) has been exceeded. | • Optimise control mode.<br>• Increase following error limit.<br>• Deactivate following error monitoring. |

### Ck07: Travel range limit exceeded [xx.0184.00155]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/7 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: The maximum travel distance has been exceeded.<br>• The maximum travel distance is displayed in C01213/1. | • Check profile parameters.<br>• Deactivate travel range limit monitoring. |

### Ck08: Home position unknown [xx.0184.00156]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/8 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☒ **4: WarningLocked** ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: Home position is unknown. | Perform homing. |

### Ck09: Positioning mode invalid [xx.0184.08005]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/9 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☒ **4: WarningLocked** ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: The positioning mode defined via the *wPosProfileMode* input is not supported. | Define a valid positioning mode. |

### Ck10: Implausible profile data [xx.0184.08007]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/10 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☒ **4: WarningLocked** ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| Cause 1: The profile data result in a braking distance that is longer than the distance to be travelled.<br>• a: Occurs if a small specified distance within a few increments is to be positioned with final speed and S rounding.<br>• b: It has been detected that the braking distance required for sizing the profile phases correctly regarding the S rounding and overchange is longer than the selected distance in the first cycle when *bExecute* = TRUE.<br>Cause 2: Reversing process in overchange profile linkage<br>• If the linkage of two profiles with final speed causes a reversing process in the second profile and this is to smooth another sequence profile with final speed too, the error is set and ramped down to standstill.<br>Cause 3: Final speed at overchange is higher than maximum profile speed<br>Cause 4: Traversing, accelerating or braking is not possible due to 0-parameters for speed, acceleration or deceleration. | On cause 1a:<br>• Set the S rounding to zero for the respective profile and go to the next profile with final speed zero.<br>On cause 1b:<br>• The error can be avoided with C02868/Bit02!<br>On cause 2:<br>• The continuous profile linkage with overchange must not provide any reversing process due to the profile selection. Here, the profile which causes a reversing must be defined with the final speed 0.<br>• From version 16.00.00 onwards, the following modes with final speed are possible via C02868/Bit02:<br>  • Reversing<br>  • Profiles where the final speed cannot be reached<br>On cause 3:<br>• Profiles in an overchange profile linkage with final speeds higher than their max. profile speeds are not supported. The final speed of a profile is limited internally to the max. profile speed in the profile data set.<br>• The error can be avoided with C02868/Bit02!<br>On cause 4:<br>• Ensure that the corresponding profile parameters for speed, acceleration and deceleration are set to non-zero when starting a traversing process. |

**Ck11: Invalid operating mode [xx.0184.08009]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/11 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| MCK: The operating mode defined via the *wMckOperationMode* input is not supported. | Define a valid operating mode. |

**Ck12: Invalid profile number [xx.0184.08014]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/12 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☒ **4: WarningLocked** ☑ 5: Warning ☑ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| MCK: The positioning profile number in the positioning operating mode specified via the *wPosProfileNo* input is invalid. | Define a valid profile number. |

**Ck13: Error - MCKCtrlInterface function block [xx.0184.08015]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/13 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| MCK: An error in the L_MckCtrlInterface_1 function block has occurred. | Check the configuration and parameterisation of the L_MckCtrlInterface_1 FB. In this regard, also observe the status messages of the FB (*wFailState* or C01299 output). |

**dH09: EEPROM power section [xx.0400.00009]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| **Cause** | **Remedy** |
|---|---|
| Device error | Please contact Lenze. |

**dH10: Fan failure [xx.0400.00016]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00566 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| The device fan has failed.<br>Possible causes:<br>• The short-circuit check of the fan connection has tripped.<br>• The speed monitoring of the fan has tripped. | • Check the fan for short-circuit.<br>• Clean the fan. |

**dH68: Adjustment data error CU [xx.0400.00104]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| **Cause** | **Remedy** |
|---|---|
| Device error | Please contact Lenze. |

**dH69: Adjustment data error BU [xx.0400.00105]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Please contact Lenze. |

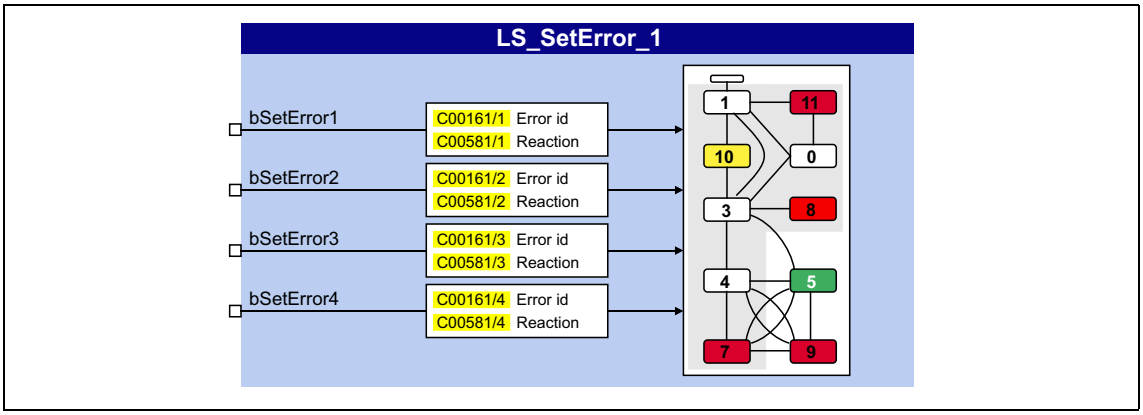
**dH70: ControlUnit is unequal to BaseUnit [xx.0400.00106]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Please contact Lenze. |

## 10.10 "LS_SetError_1" system block

This system block is used to implement error handling within the application.

- The application can trip up to four different user error messages with parameterisable error IDs and error responses via the four boolean inputs.
- If several inputs are set to TRUE at the same time, the input with the lowest number will trip the error message.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bSetError1<br>BOOL | Input for tripping "User error 1"<br>• Error subject number: 980<br>• Error number: (C00581/1 x 0x0400000) + (980 x 0x10000) + (C00161/1) |
| bSetError2<br>BOOL | Input for tripping "User error 2"<br>• Error subject number: 981<br>• Error number: (C00581/2 x 0x0400000) + (981 x 0x10000) + (C00161/2) |
| bSetError3<br>BOOL | Input for tripping "User error 3"<br>• Error subject number: 982<br>• Error number: (C00581/3 x 0x0400000) + (982 x 0x10000) + (C00161/3) |
| bSetError4<br>BOOL | Input for tripping "User error 4"<br>• Error subject number: 983<br>• Error number: (C00581/4 x 0x0400000) + (983 x 0x10000) + (C00161/4) |

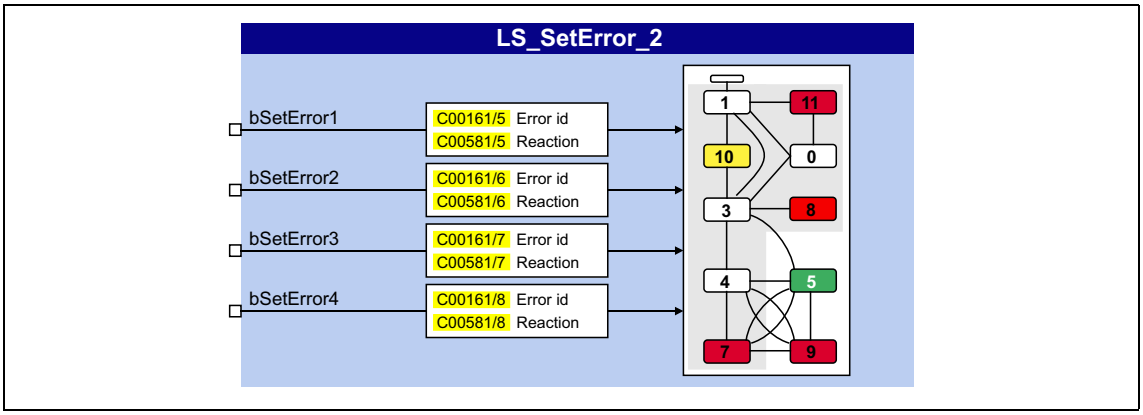
### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00161/1...4 | 0 | ... | 65535 | Error ID for user errors 1 ... 4 |
| C00581/1...4 | | | | Response to user errors 1 ... 4 |
| | 0 | No Reaction | | |
| | 1 | Fault | | |
| | 2 | Trouble | | |
| | 3 | TroubleQuickStop | | |
| | 4 | WarningLocked | | |
| | 5 | Warning | | |
| | 6 | Information | | |

## 10.11 "LS_SetError_2" system block

This system block is used to implement error handling within the application.

- The application can trip up to four different user error messages with parameterisable error IDs and error responses via the four boolean inputs.
- If several inputs are set to TRUE at the same time, the input with the lowest number will trip the error message.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bSetError1<br>BOOL | Input for tripping "User error 5"<br>• Error subject number: 984<br>• Error number: (C00581/5 x 0x0400000) + (984 x 0x10000) + (C00161/5) |
| bSetError2<br>BOOL | Input for tripping "User error 6"<br>• Error subject number: 985<br>• Error number: (C00581/6 x 0x0400000) + (985 x 0x10000) + (C00161/6) |
| bSetError3<br>BOOL | Input for tripping "User error 7"<br>• Error subject number: 986<br>• Error number: (C00581/7 x 0x0400000) + (986 x 0x10000) + (C00161/7) |
| bSetError4<br>BOOL | Input for tripping "User error 8"<br>• Error subject number: 987<br>• Error number: (C00581/8 x 0x0400000) + (987 x 0x10000) + (C00161/8) |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00161/5...8 | 0 | ... | 65535 | Error ID for user errors 5 ... 8 |
| C00581/5...8 | | | | Response to user errors 5 ... 8 |
| | 0 | No Reaction | | |
| | 1 | Fault | | |
| | 2 | Trouble | | |
| | 3 | TroubleQuickStop | | |
| | 4 | WarningLocked | | |
| | 5 | Warning | | |
| | 6 | Information | | |

# 11 Oscilloscope function

This function extension is available from version 12.00.00!

The oscilloscope function integrated in the 8400 TopLine can be used as support for commissioning, maintenance, and troubleshooting. It is operated via a user interface in the engineering tool.

**Typical applications**

- Graphical representation of measured values (e.g. speed setpoint, actual speed and torque)
- Detection of process values without additional measuring instruments (e.g. oscilloscope, voltmeter and ammeter)
- Convenient documentation for fine tuning of control circuits or parameter changes of the inverter

**Special features**

- Recording and saving of measured values in the 8400 TopLine
- Simultaneous measuring on eight independent channels
- Measuring of fast and slow signals by means of adjustable sample rate
- Triggering on a channel, an application variable or system event
- Detecting measured values before and after a trigger event
- Transfer of the measured values to the Engineering PC for the graphical representation and evaluation in the engineering tool
- The measured values represented in the form of interpolated curves can be optionally shown and hidden, represented in any colour or overlaid with the signal characteristics of other variables recorded.
- Cursor and zoom function for the measurement analysis
- Saving & loading of oscilloscope configurations on the Engineering PC
- Export of measured values via the clipboard for further processing
- Linking channel values with arithmetic operations (addition, subtraction and multiplication)
- Easy signal analysis by frequency transformation of time signals with FFT ("Fast Fourier Transformation").

**Functional description**

In the engineering tool, you set the trigger condition and sample rate and select the signal sources to be recorded via the oscilloscope user interface when an online connection to the 8400 TopLine has been established. In this case, "signal sources" are the internal output signals of the function, system, application and port blocks.

The changed configuration is only then taken over by the device when pressing the "start" button in the toolbar.

The values are validated each time the entry is confirmed by the "Return" key. If invalid settings are found during the check, the oscilloscope triggers an error.

With an online connection, the measured 8400 TopLine values are transferred to the engineering tool and graphically represented on the oscilloscope user interface as soon as the measurement has been completed.

## 11.1 Technical data

| Oscilloscope function of 8400 TopLine | |
|---|---|
| Number of channels | 1 ... 8 |
| Depth of the measured value memory | Max. 8192 measured values, depending on the number of channels and the size of the signal sources to be recorded |
| Data width of a channel | Max. 32 bits, corresponding to the data type of the signal sources to be recorded |
| Sample rate | 1 ms or a multiple thereof |
| Time base | 5, 10, 20, 50, 100, 200, 500 ms or 1 s |
| Trigger level | Corresponding to the value range of the signal sources to be triggered |
| Trigger selection | The trigger is activated if the trigger value set for the respective channel is fallen short of or exceeded. The trigger value "must actually pass" the threshold. |
| Trigger delay | - 200 % ... + 200 % |
| Trigger source | Channel 1 ... 8 |

## 11.2 Operation

This chapter describes step-by-step how to record the signal characteristics of 8400 TopLine signal sources and represent, analyse, document and process them in the oscilloscope.

**Note!**

Recording can only be started when an online connection has been established to the 8400 TopLine.

### 11.2.1 User interface

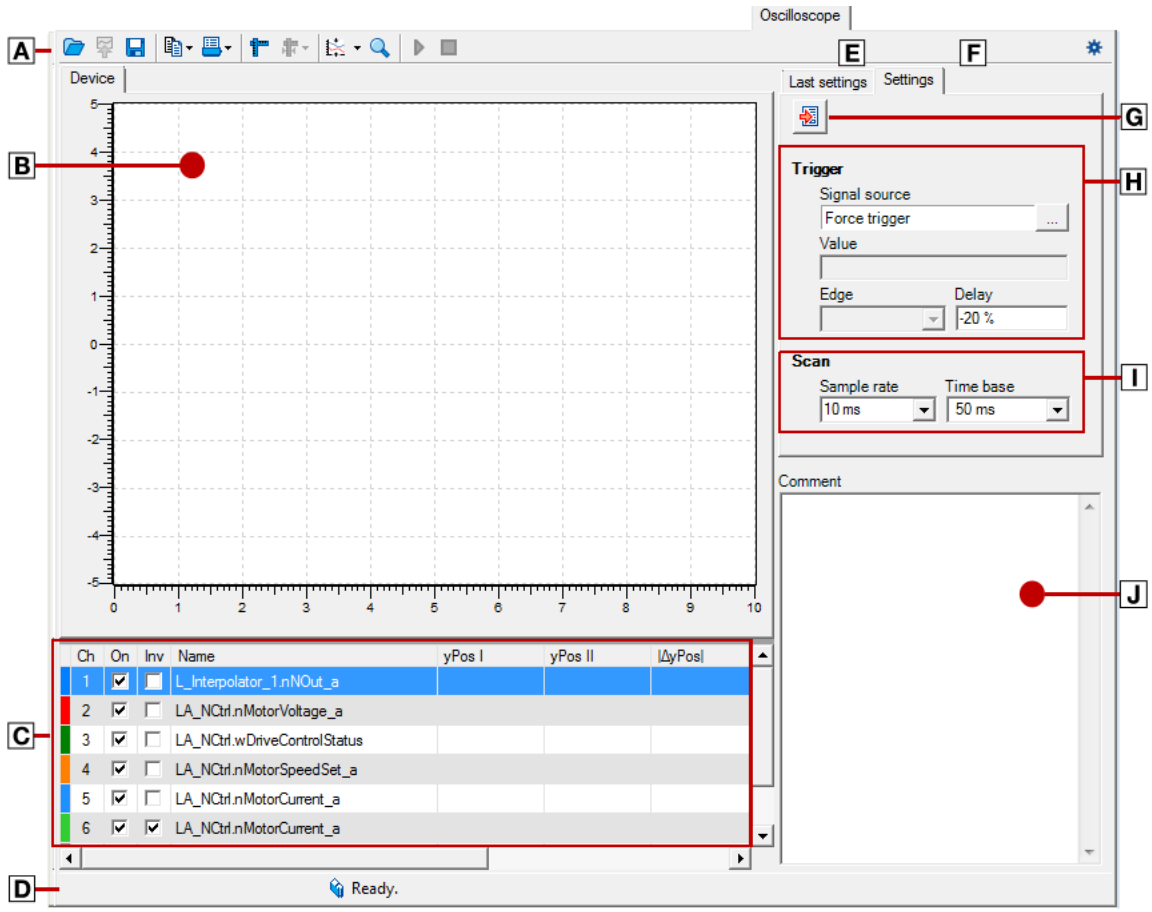
The oscilloscope user interface is available in the following Lenze engineering tools:

| | | |
|---|---|---|
| ☑ »Engineer« from version 2.16 | ☐ »PLC Designer« | ☑ »EASY Starter« from version 1.9 |

**How to go to the oscilloscope user interface:**

1. Go to the *Project view* and select the servo inverter.
2. Select the **Oscilloscope** tab from the *Workspace*.

The oscilloscope user interface contains the following control and function elements:




- A Oscilloscope toolbar
- B Oscillogram selection
- C Channel list
  - Selecting signal sources to be recorded
- D Status bar
- E Last settings
- F Settings
- G Import settings from a loaded oscillogram
- H Trigger settings
- I Input fields for sampling rate and time base
- J Input field for comments

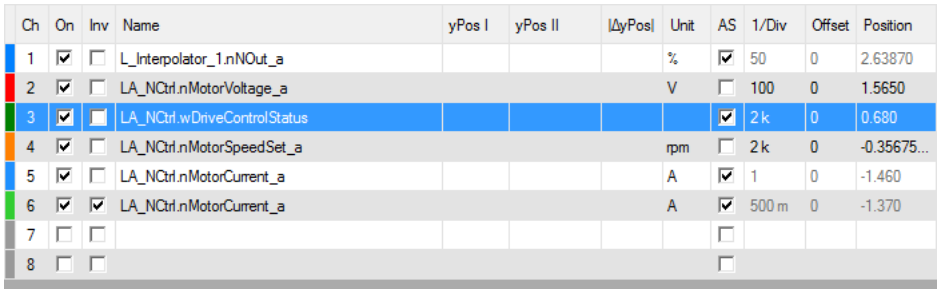
### Oscilloscope toolbar

| Symbol | Function |
|---|---|
|  | Loading the oscillogram file (📖 769) |
|  | Upload recorded oscillogram from device<br>• Transfer values from the measured value memory of the 8400 TopLine to the Engineering PC.<br>• Only possible when an online connection has been established to the 8400 TopLine. |
|  | Saving an oscillogram in a file (📖 768) |
|  | Copy to clipboard: Copy as text \| Copy as picture \| Copy as table<br>• For documentation purposes, it is possible to copy the measured value of an oscillogram set as a table or, alternatively, the oscilloscope user interface as a picture, to the clipboard for use in other programs. |
|  | Printer settings \| Print view \| Print |
|  | Display cursor |
|  | Automatically scale vertically<br>• Set all Y positions to zero |
|  | Activate zoom function<br>▸ Adjusting the representation (📖 763) |
|  | Start recording<br>• The settings are transferred to the device and the recording is started. |
|  | Stop recording |
|  | Oscilloscope settings<br>• Cyclic recording of oscillograms<br>• After recording is completed, always load oscillograms without further inquiry |

### 11.2.2 Selecting signal sources to be recorded

The oscilloscope supports up to eight channels. Hence, the **channel list** can maximally record eight signal sources.

The **channel list** serves to configure the signal sources to be recorded. Four signal sources have already been selected in the default setting:



| Ch | On | Inv | Name | yPos I | yPos II | \|ΔyPos\| | Unit | AS | 1/Div | Offset | Position |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ☑ | ☐ | L_Interpolator_1.nNOut_a | | | | % | ☑ | 50 | 0 | 2.63870 |
| 2 | ☑ | ☐ | LA_NCtrl.nMotorVoltage_a | | | | V | ☐ | 100 | 0 | 1.5650 |
| 3 | ☑ | ☐ | LA_NCtrl.wDriveControlStatus | | | | | ☑ | 2 k | 0 | 0.680 |
| 4 | ☑ | ☐ | LA_NCtrl.nMotorSpeedSet_a | | | | rpm | ☐ | 2 k | 0 | -0.35675... |
| 5 | ☑ | ☐ | LA_NCtrl.nMotorCurrent_a | | | | A | ☑ | 1 | 0 | -1.460 |
| 6 | ☑ | ☑ | LA_NCtrl.nMotorCurrent_a | | | | A | ☑ | 500 m | 0 | -1.370 |
| 7 | ☐ | ☐ | | | | | | ☐ | | | |
| 8 | ☐ | ☐ | | | | | | ☐ | | | |

| Name | Meaning |
|---|---|
| - | Curve colour for representation in the oscillogram<br>• A double-click on the coloured area of the channel serves to set a user-defined colour. |
| Ch | Channel number |
| On | Curve visible / invisible |
| Inv | Inversion yes / no |
| Name | Designation of the signal source |
| yPos I | y position of cursor I |
| yPos II | y position of cursor II |
| \|ΔyPos\| | Difference of the y positions of both cursors<br>• Difference = \|yPos I - yPos II\| |
| Unit | Unit of the signal source |
| AS | Select / deselect channel for automatic scaling |
| 1/Div | Vertical scaling factor |
| Offset | Offset value<br>• The offset value is subtracted from the recorded raw value before scaling is executed. This serves, for instance, to make very slight value fluctuations visible within a continuously very high recording value (e.g. harmonics with low amplitude). |
| Position | Position value<br>• With regard to the vertical curve scale (- 5 ... + 5), the position value determines the vertical position of the zero point of the Y axis of a curve. |

**How to select a signal source for recording:**

1. Go to the **channel list** and click a line not pre-assigned so far in order to open the *Select signal source* dialog box.
   - Double-clicking an already assigned selection serves to assign it to another signal source.
2. Select a new signal source in the *Select signal source* dialog box.
3. Click the **OK** button.
   - The dialog box is closed and the selection is accepted.

**How to delete a selection:**

1. Click the signal source to be removed from the **channel list**.
2. Right-click the *context menu* to open it.
3. Select the **Remove signal source** command in the *context menu*.

You can add so many signal sources for recording until all eight channels are assigned.

### 11.2.3 Selecting the recording time/sample rate

**How to define the sampling rate and time base for recording:**

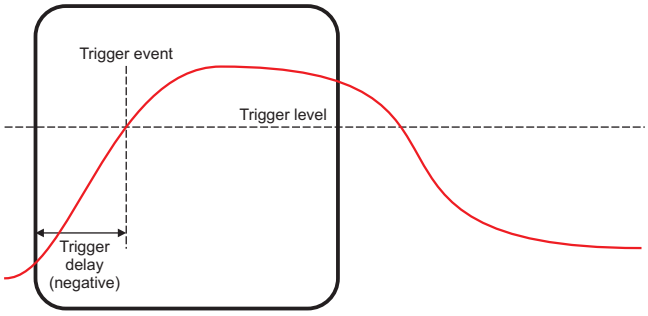
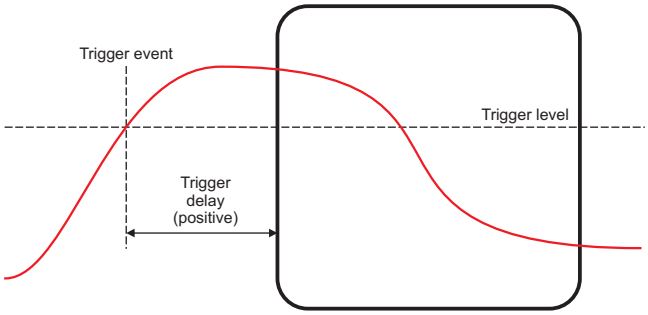
1. Select the desired time base from the **time base** list field.
   - The current time base setting multiplied by ten results in the recording time.
   - Since the measured value memory of the 8400 TopLine has a limited capacity, usually a compromise is made between sample rate and recording time.
2. Enter the desired sampling rate in [ms] in the **sampling rate** input field.

### 11.2.4 Defining the trigger condition

The trigger condition serves to define the starting time of recording in the 8400 TopLine. The oscilloscope provides various trigger conditions by means of which recording of the measured values can be controlled.

If the **Last settings** tab is in the foreground, click the **Settings** tab to show the input fields for configuring the trigger condition. The **Settings** tab includes the button which serves to import settings from a loaded oscillogram.

| Setting | Function |
|---|---|
| **Signal source** | Selection of the trigger source |
| Variable | Selection of an application variable as signal source |
| Channel | The oscilloscope triggers on a channel configured in the **channel list**. |

| Setting | Function |
|---|---|
| System event | The following system events serve as trigger source:<br>• Fault,<br>• Warning<br>• Error |
| Direct trigger | No trigger condition, recording takes place immediately after clicking the symbol ▶ in the *oscilloscope toolbar*. |
| **Value** | Value from which on triggering is activated. |
| **Delay** | Delay between recording and trigger event.<br>Unit: [%] |
| Trigger delay | Select a negative delay time to detect signals **prior to** the trigger event.<br>  <br>• In the oscillogram, the trigger time is marked by a dashed line.<br>• When triggering on occurrence of an event, it is thus possible to detect the values that have caused the event. |
| | Select a positive delay time to detect signals occurring a certain time **after** the trigger event.<br>  |
| **Edge** | Three trigger modes are available: |
| Positive edge | For trigger activation, the selected trigger value must first be fallen short of and then be exceeded. |
| Negative edge | For trigger activation, the selected trigger value must first be exceeded and then be fallen short of. |
| Change | When triggering to a BOOL signal source:<br>• Trigger activation requires a state change.<br>When triggering to a signal source of a different type:<br>• For trigger activation, the current value must be different than the last value. |

### 11.2.5 Starting recording

**Note!**

Recording can only be started when an online connection has been established to the 8400 TopLine.

Press the button ▶ included in the *oscilloscope toolbar* to activate the recording. Recording is started depending on the settings for triggering the signal source.

To obtain a maximum sample rate when recording the signal source values, the data are first stored in the measured value memory of the 8400 TopLine and then transferred to the Engineering PC. The current recording status is displayed in the status bar.

### 11.2.6 Cyclic recording

**How to record oscillograms cyclically:**

1. Go to the *oscilloscope toolbar* and click the icon to open the **Oscilloscope settings** dialog box.
2. In order that the recording is restarted automatically after uploading an oscillogram from the device, set the respective tick.
   - Cyclic recording is only possible for time base values ≥ 500 ms.

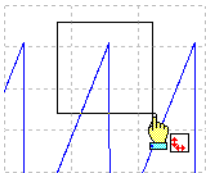
For monitoring certain situations, it is possible to keep the bigger view of the interesting part of a curve even after the cyclic update - as originally zoomed.

### 11.2.7 Adjusting the representation

After the variable values have been recorded and the oscillogram has been transferred to the PC, it is visualised in the oscillograph. If required, the representation can now be adjusted by using the zoom or the automatic scaling function.

#### Zoom function

Go to the *oscilloscope toolbar* and click the icon to activate the zoom function.

| Zoom function | Procedure | |
|---|---|---|
| Zoom selection |  | Hold down the left mouse button and draw the oscillogram section to be zoomed:<br>  <br>• While being drawn, the selection is shown with a frame.<br>• When the left mouse button is released, the selection is zoomed in the oscillogram. |
| Horizontal stretching |  | Hold down the left mouse button and move the mouse pointer on the horizontal scale to the left to stretch the shown selection from the right edge. |
| |  | Hold down the right mouse button and move the mouse pointer on the horizontal scale to the right to stretch the shown selection from the left edge. |
| | Moving the mouse pointer in opposite direction continuously reduces the stretching. | |

| Zoom function | Procedure | |
|---|---|---|
| Vertical stretching |  | Hold down the left mouse button and move the mouse pointer on the vertical scale to the bottom to stretch the shown selection from the top. |
| |  | Hold down the right mouse button and move the mouse pointer on the vertical scale to the top to stretch the shown selection from the bottom. |
| | Moving the mouse pointer in opposite direction continuously reduces the stretching. | |
| Return to original representation |  | Click the right mouse button in the oscillogram to return step by step to the original representation. |

### Automatic scaling function

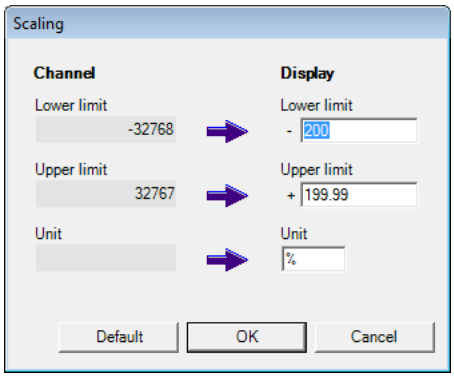
Use the automatic scaling function to automatically scale and reposition the representation of selectable signal characteristics in the oscillogram and reset the offset to "0".

**How to carry out automatic scaling:**

1. Activate the automatic scaling for each channel in the **channel list** by setting a tick in the "AS" column.
2. Go to the *oscilloscope toolbar* and click the icon to activate the automatic scaling function for the activated channels.
3. Click the **OK** button.
   - The dialog box is closed and the selected channels/signal sources are scaled automatically.
4. Go to the *oscilloscope toolbar* and click the arrow next to the icon to set all the displayed characteristics to the Y position "0" if required. Thus, the characteristics are displayed overlapping.

**How to execute the scaling manually for each channel:**

1. Go to the **channel list** and double-click the entry to be changed in the "Unit" column (double-clicking an empty line has no effect.)
   - The dialog for entering the scaling opens which allows for entering the lower and upper limit and the unit, as can be seen from the example:



2. Enter the lower limit, the upper limit and the unit.
   - On the right side, you can enter any scaling values as upper and lower limit.
   - On the left side of the dialog, the limits of the data type of the accepted value is displayed.
3. Click the **OK** button.
   - The dialog box is closed.

### 11.2.8 Cursor function: Reading individual measured values

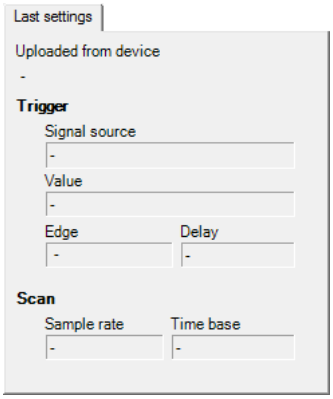
In addition to the zoom and scaling function, the oscilloscope offers a "cursor function" that can be used to display individual measured values of a selectable channel or the difference between two measured values.

**How to use the cursor function:**

1. Go to the *oscilloscope toolbar* and click the icon to activate the cursor function.
   - Another button is displayed which enables a centering of two vertical measuring lines that are independent of each other and can be moved.
   - The status bar displays the position of both measuring lines and the difference between them.
2. Select the channel for which individual measured values are to be indicated from the **Channel** list field.
3. Hold down the left mouse button and drag the red vertical measuring line to the desired position.
   - The active measuring line is represented by a continuous line, the inactive measuring line is represented by a dashed line.
   - If you click the inactive measuring line, it automatically becomes active.
   - The value measured at the position of the active measuring line is indicated in the Value group field.
   - The difference between the values measured at the two measuring lines is indicated in the Differential value group field.
   - Comparing peak values: Several values displayed in the oscillogram can be compared by means of a horizontal measuring line. This measuring line is automatically generated based on the current cursor position and thus cannot be moved separately.

### 11.2.9 Last settings

All information contained in the **Last settings** tab refer to the oscillogram loaded last in the device:




The contents cannot be changed.

## 11.3 Managing oscillograms

If several oscillograms are loaded in the oscilloscope at the same time, the oscillogram to be displayed is selected via the respective tab below the toolbar. Generally, the following oscillograms can be distinguished:

**Device oscillogram**

The device oscillogram is the only oscillogram which serves to establish a connection to the target system to execute an oscilloscope measurement.

**MERGE oscillogram**

If two or more oscillograms are loaded in the oscilloscope, a "MERGE" tab is available.

- In the MERGE tab, several characteristics from the currently loaded oscillograms can be overlaid, e.g. to compare signal characteristics from different recordings. ▸ Overlay function (📖 770)

**Loaded oscillogram**

An oscillogram loaded from a file.

### 11.3.1 Commenting the oscillogram

The **Comment** text field serves to enter a comment on the selected oscillogram.

- If you execute the Saving an oscillogram in a file command, the comment is saved together with the oscillogram in the file.

### 11.3.2 Saving an oscillogram in a file

After the signal sources to be recorded have been selected and the required settings have been entered, you can save the configuration and recording, if already executed, for future use in the project or export them to a file.

**Note!**

The reuse of a saved configuration is only reasonable for controllers of the same type, as otherwise due to a scaling of the oscilloscope channels that is not adapted, incorrect values are displayed!

**How to save an oscillogram in the project:**

1. Click the icon in the *oscilloscope toolbar*.
   - The *Save oscillogram file* dialog box is displayed.
2. Define a file name in the **File name** input field.
3. Click the **Filing in the project** button.
   - The dialog box is closed and the current oscillogram is filed in the project.

**Note!**

The oscillogram is only saved if the entire project is saved!

**How to save an oscillogram as external file:**

1. Click the icon in the *oscilloscope toolbar*.
   - The *Save oscillogram file* dialog box is displayed.
2. Press the **Save as external file...** button.
   - A new window opens in which you have to define the directory and file name for the oscillogram to be saved.
3. Click the **Save** button.
   - The dialog box is closed and the current oscillogram is saved.

### 11.3.3 Loading the oscillogram file

Configurations/oscillograms already saved can be reloaded into the oscilloscope any time, e.g. for the overlay function.

**Note!**

The reuse of a saved configuration is only possible for controllers of the same type, as otherwise due to a scaling of the oscilloscope channels that is not adapted, incorrect values are displayed!

**How to load an oscillogram file from the project:**

1. Click the icon in the *oscilloscope toolbar*.
   - The *Load oscillogram file* dialog box is displayed.
2. Select the file to be loaded from the upper list field.
3. Click the **OK** button.
   - The dialog box is closed and the oscillogram file is accepted.

**How to load an oscillogram file from an external file:**

1. Click the icon in the *oscilloscope toolbar*.
   - The *Load oscillogram file* dialog box is displayed.
2. Press the **Load from external file...** button.
   - A new window opens in which you have to select the directory and file name for the oscillogram to be saved.
3. Click **Open**.
   - The dialog box is closed and the oscillogram file is loaded.
   - The oscillogram is displayed on an additionally appearing tab.
   - If the configuration to be loaded contains signal sources that are no longer available in the controller, these variables are automatically removed from the configuration.

### 11.3.4 Overlay function

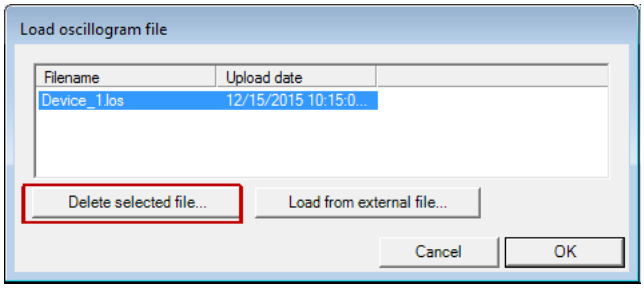
The overlay function serves to lay several characteristics from the currently loaded oscillogram files on top of each other, e.g. to compare signal characteristics from different recordings.

- If two or more oscillograms are loaded in the oscilloscope, e.g. the device oscillogram and an oscillogram previously saved in the project, a "MERGE" tab is available.
- If the MERGE tab is selected, the desired characteristics to be overlaid or compared can be selected from the loaded files in the **Channel list**.
- If a device oscillogram is used in the MERGE tab, an update is carried out in the MERGE oscillogram in case of a renewed recording.
- Removing signal sources from the device oscillogram causes the characteristics in the MERGE oscillogram to be deleted.

### 11.3.5 Deleting an oscillogram file saved in the project

**How to delete an oscillogram file saved in the project:**

1. Click the icon in the *oscilloscope toolbar*.
   - The *Load oscillogram file* dialog box is displayed.




2. Select one or several oscillogram files from the upper list field.
3. Press **Delete selected files...** button.
   - The selected file(s) are deleted and the dialog box is closed.

# 12 System bus "CAN on board"

The inverter has an integrated CANopen interface ("CAN on board") which is used to exchange i.a. process data and parameter values between the nodes. Furthermore, other modules can be connected via this interface such as decentralised terminals, operator and input devices (HMIs), as well as external controls and host systems.

The interface transfers CAN objects following the CANopen communication profile (CiA DS301, version 4.02) developed by the umbrella organisation of CiA (CAN in Automation) in conformity with the CAL (CAN Application Layer).

**Tip!**

- In the »Engineer« parameter list and in the keypad, category **CAN**, you can find the parameters relevant for the CANopen interface classified in different subcategories.
- Information on CAN communication modules and CANopen interfaces of other Lenze devices is provided in the "CAN" communication manual in the Lenze library.

## 12.1 General information

For many years, the system bus (CAN) based on the CANopen communication profile has been integrated in Lenze inverters. Due to the lower number of data objects available, the functionality and compatibility of the previous system bus are lower as compared to CANopen. For parameter setting, two parameter data channels are always available to the user while CANopen provides only one active parameter channel.

The system bus (CANopen) of the Inverter Drives 8400 is a further development of the system bus (CAN) including the following properties:

- Full compatibility with CANopen DS301 V4.02.
- Support of the "Heartbeat" NMT slave function (DS301 V4.02)
- 1 or 2 parameterisable parameter data channels for transmitting SDOs (*Service Data Objects*)
- Up to four parameterisable process data channels (dependent on the device version) for transmitting PDOs (*Process Data Objects*)
  - All process data channels are functionally equivalent
  - Monitoring of the receive PDOs for data reception
- Adjustable error response to ...
  - physical CAN errors (frame, bit, ACK error)
  - bus-stop, bus working
  - absent PDOs
- Telegram counters for SDOs and PDOs
- Bus status diagnostics
- Boot-up telegram generation
- Emergency telegram generation
- Reset node telegram generation (for master configuration)
- Sync telegram generation and response to sync telegrams:
  - Data transmission/reception
  - Device-internal time base synchronisation
- Abort codes
- All CAN on board functions can be parameterised via codes
- Object directory (all mandatory functions, optional functions, indexes)

### 12.1.1 General data and application conditions

| Range | Values |
|---|---|
| Communication profile | CANopen, DS301 V4.02 |
| Communication medium | DIN ISO 11898 |
| Network topology | Line terminated at both ends |
| Adjustable node addresses (max. number of nodes) | Depending on the number of SDO channels set in C00366:<br>• 1 SDO: Node address 1 ... 127 (max. 127 nodes)<br>• 2 SDO: Node address 1 ... 63 (max. 63 nodes)<br>• adjustable via DIP switches or via code C00350. |
| Adjustable baud rates | 20, 50, 125, 250, 500, 1000 kbps<br>• adjustable via DIP switches or via code C00351. |

| Range | Values |
|---|---|
| Process data | For "BaseLine C" version:<br>• Max. 2 transmit PDOs (TPDOs) with 1 ... 8 bytes (adjustable)<br>• Max. 2 receive PDOs (RPDOs) with 1 ... 8 bytes (adjustable) |
| | For "StateLine C" version:<br>• Max. 4 transmit PDOs (TPDOs) with 1 ... 8 bytes (adjustable)<br>• Max. 4 receive PDOs (RPDOs) with 1 ... 8 bytes (adjustable)<br>(Process data channel 4 is available from version 15.00.00!) |
| Parameter data | Max. 2 server SDO channels with 1 ... 8 bytes<br>• Because of the 2 server SDO channels, the address range from 1 ... 63 is available |
| Transfer mode for TPDOs | • in case of data change (including adjustable blocking time)<br>• Time-controlled, 1 to x ms<br>• After the reception of 1 to 240 sync telegrams |

### 12.1.2 Supported protocols

| Protocols | |
|---|---|
| Standard PDO protocols | PDO write<br>PDO read |
| SDO protocols | SDO download<br>SDO download initiate<br>SDO download segment |
| | SDO upload<br>SDO upload initiate<br>SDO upload segment |
| | SDO abort transfer |
| | SDO block download<br>SDO block download initiate<br>SDO block download end |
| | SDO block upload<br>SDO block upload initiate<br>SDO block upload end |
| NMT protocols | Start remote node (master and slave) |
| | Stop remote node (slave) |
| | Enter pre-operational (slave) |
| | Reset node (slave and local device) |
| | Reset communication protocol (slave) |
| Monitoring protocols | Heartbeat (heartbeat producer and heartbeat consumer)<br>• Up to 15 Heartbeat Producers can be monitored. |
| | Emergency telegram (to master) |
| More protocols | Transmitting and receiving a sync telegram<br>• Synchronisation of the internal time base to the reception of the CAN sync telegram is possible. ▸ Synchronisation of the internal time base |

### 12.1.3 Communication time

The communication time is the time between the start of a request and the arrival of the corresponding response.



**Tip!**

The communication times in the CAN network depend on:

- the processing time in the device
- the telegram runtime (baud rate/telegram length)
- the bus load (especially if the bus is charged with PDOs and SDOs at a low baud rate)

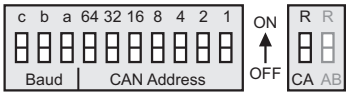
**Processing time in the 8400 inverter**

No dependencies exist between parameter data and process data.

- Parameter data: approx. 5 ms (typical value)
  - For parameters concerning the motor control (e.g. C00011), the processing time may be longer (up to 30 ms).
- Process data: 1 ms

## 12.2 Possible settings via DIP switch

The following settings for the "CAN on board" system bus can be made via the front panel DIP switches:

| DIP switch | | Possible settings/detailed information |
|---|---|---|
|  | a ... c | ▸ Setting the baud rate |
| | 1 ... 64 | ▸ Setting the node address |
| | CA | ▸ Activating the bus terminating resistor |

Lenze setting: All DIP switches are in the "OFF" position

**Note!**

- The DIP switch settings are accepted if a node address is unequal zero when the device or the 24-V supply is switched on by the DIP address.
- If all DIP switches are OFF when the device or the 24 V supply is switched on, the setting of the baud rate and node address are read out of the parameter set/parameter.

**Tip!**

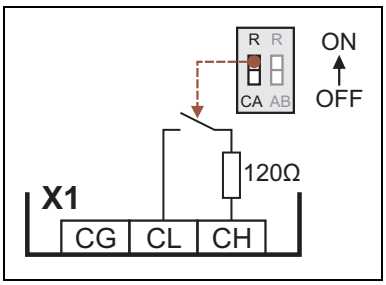
The current DIP switch settings are displayed in code C00349.

Bit 15 indicates that the setting of the DIP switches has been accepted when the device or the 24V supply has been switched on.

### 12.2.1 Activating the bus terminating resistor

The system bus must be terminated between CAN low and CAN high at the first and last physical node each by a resistor (120 Ω).

The 8400 inverter is provided with an integrated bus terminating resistor, which can be activated via the DIP switch labelled with "CA":




[12-1] Activation of the integrated bus terminating resistor

- OFF = bus terminating resistor is inactive
- ON = bus terminating resistor is active

### 12.2.2 Setting the baud rate

The baud rate can be set via code C00351 or with the DIP switches a to c:

**Note!**

- All DIP switches (a ... c, 1 ... 64) = OFF (Lenze setting):
  - At switching on, the settings under code C00350 (node address) and C00351 (baud rate) will become active.
- Preset baud rate: 500 kbps

| DIP switch position | | | Baud rate |
|---|---|---|---|
| c | b | a | |
| **ON** | OFF | **ON** | 20 kbps |
| OFF | **ON** | **ON** | 50 kbps |
| OFF | **ON** | OFF | 125 kbps |
| OFF | OFF | **ON** | 250 kbps |
| OFF | OFF | OFF | **500 kbps** |
| **ON** | OFF | OFF | 1000 kbps |

### 12.2.3 Setting the node address

The node address can be set via code C00350 or with the DIP switches 1 to 64.

- The labelling on the housing corresponds to the values of the individual DIP switches for determining the node address.
- The valid address range depends on the number of SDO channels set in C00366:
  - 1 SDO (Lenze setting): 1 ... 127
  - 2 SDO: 1 ... 63

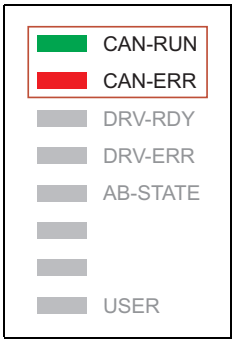
**Note!**

- The addresses of the nodes must differ from each other.
- All DIP switches (a ... c, 1 ... 64) = OFF (Lenze setting):
  - At switching on, the settings under code C00350 (node address) and C00351 (baud rate) will become active.

**Example: Setting of the node address 23**

| DIP switch | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| Switch position | OFF | OFF | **ON** | OFF | **ON** | **ON** | **ON** |
| Value | 0 | 0 | 16 | 0 | 4 | 2 | 1 |
| Node address | **= Sum of the values = 16 + 4 + 2 + 1 = 23** | | | | | | |

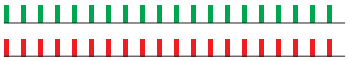
## 12.3 LED status displays for the system bus




Information about the status of the system bus can be obtained quickly via LED displays "CAN-RUN" and "CAN-ERR" on the front of the inverter.

The meaning can be seen from the tables below.

### Inverter is not (yet) active on the system bus

| LED display | Meaning |
|---|---|
|  (CAN-ERR is permanently lit) | Inverter is not active on the system bus / Bus Off |
|  (CAN-RUN and CAN-ERR flicker) | Automatic detection of baud rate is active |

### Inverter is active on the system bus

- LED "CAN-RUN" signals the CANopen state:

| LED display | CANopen state |
|---|---|
|  (CAN-RUN is blinking every 0.2 seconds) | Pre-Operational |
|  (CAN-RUN is permanently lit) | Operational |
|  (CAN-RUN is blinking every second) | Stopped |

- LED "CAN-ERR" signals a CANopen error:

| LED display | CANopen error |
|---|---|
|  (CAN-ERR is blinking once, then off for 1 second) | Warning Limit reached |
|  (CAN-ERR is blinking twice, then off for 1 second) | Node Guard Event |
|  (CAN-ERR is blinking three times, then off for 1 second) | Sync Message Error<br>(only possible in the "Operational" state) |

## 12.4 Going online via the system bus

The integrated system bus interface (CAN on board, X1 terminal) can also be used for the communication between the »Engineer« and the inverter, alternatively to the USB diagnostic adapter.

- Lenze offers the following communication accessories for connection to the PC:

| Communication accessories | PC interface |
|---|---|
| **PC system bus adapter 2173**<br>incl. connection cable and voltage supply adapter<br>• for DIN keyboard connection (EMF2173IB)<br>• for PS/2 keyboard connection (EMF2173IBV002)<br>• for PS/2 keyboard connection with electrical isolation (EMF2173IBV003) | Parallel interface<br>(LPT port) |
| **PC system bus adapter 2177**<br>incl. connection cable (EMF2177IB) | USB<br>(Universal Serial Bus) |

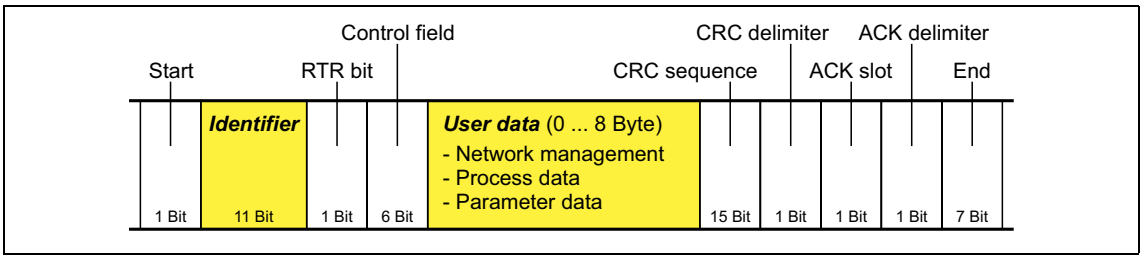
**Note!**

- For detailed information about the PC system bus adapter, please see the "CAN Communication Manual".
- Please observe the documentation for the PC system bus adapter!
- In the »Engineer«, go to the *Device assignment offline devices* dialog box and select the "System bus CAN" entry from the **Bus connection** list field to establish an online connection.

## 12.5 Reinitialising the CANopen interface

The C00002/26 = "1: On / start" device command reinitialises the CANopen interface of the inverter ("Reset node"), which is required after e.g. changing the data transfer rate, the node address or the identifiers, respectively.

## 12.6 Structure of the CAN data telegram




[12-2] Basic structure of the CAN telegram

The following subchapters provide a detailed description of the identifier and the user data. The other signals refer to the transfer characteristics of the CAN telegram whose description is not included in the scope of this documentation.

**Tip!**

Please visit the homepage of the CAN user organisation CiA (CAN in automation) for further information:
http://www.can-cia.org

### 12.6.1 Identifier

The principle of the CAN communication is based on a message-oriented data exchange between a transmitter and many receivers. All nodes can transmit and receive quasi-simultaneously.

The identifier, also called COB-ID (abbr. for communication object identifier), is used to control which node is to receive a transmitted message. In addition to the addressing, the identifier contains information on the priority of the message and the type of user data.

The identifier consists of a basic identifier and the node address of the node to be addressed:

**Identifier (COB-ID) = basic identifier + node address (node ID)**

Exception: The identifier for process data/heartbeat/emergency objects as well as network management and sync telegrams is freely assigned by the user (either manually or automatically by the network configurator), or is permanently assigned.

### Node address (node ID)

Every node of the system bus network must be assigned to a node address (also called node ID) within the valid address range (1 ... 127) for unambiguous identification.

- Assigning a node address more than once within a network is impermissible.
- The own node address can be configured via the DIP switches or via code C00350. ▸ Setting the node address (📖 777)

### Identifier assignment

The system bus is message-oriented instead of node-oriented. Every message has an unambiguous identification, the identifier. For CANopen, node-oriented transfer is achieved by the fact that every message has only one transmitter.

- The basic identifiers for network management (NMT) and the basic SDO channel (SDO1) are defined in the CANopen protocol and cannot be changed.
- The basic identifiers of the PDOs are preset in the Lenze setting according to the "Predefined Connection Set" of DS301 V4.02.

| Object | | Basis identifier (CANBaseID) | | Direction |
|---|---|---|---|---|
| Network management (NMT) | | 0 | 0x0000 | |
| Sync [1] | | 128 | 0x0080 | |
| Emergency [1] | | 128 | 0x0080 | Device → system bus |
| PDO1 (Process data channel 1) | TPDO1 | 384 | 0x0180 | Device → system bus |
| | RPDO1 | 512 | 0x0200 | System bus → device |
| PDO2 (Process data channel 2) | TPDO2 | 641 | 0x0281 | Device → system bus |
| | RPDO2 | 640 | 0x0280 | System bus → device |
| PDO3 (Process data channel 3) | TPDO3 | 769 | 0x0301 | Device → system bus |
| | RPDO3 | 768 | 0x0300 | System bus → device |
| PDO4 [2] (Process data channel 4) | TPDO4 | 897 | 0x0381 | Device → system bus |
| | RPDO4 | 896 | 0x0380 | System bus → device |
| SDO1 (Parameter data channel 1) | TSDO1 | 1408 | 0x0580 | Device → system bus |
| | RSDO1 | 1536 | 0x0600 | System bus → device |
| SDO2 (Parameter data channel 2) | TSDO2 | 1472 | 0x05C0 | Device → system bus |
| | RSDO2 | 1600 | 0x0640 | System bus → device |
| Heartbeat | | 1792 | 0x0700 | Device → system bus |
| Boot-up | | 1792 | 0x0700 | Device → system bus |

[1] If you set the sync identifiers manually, observe the use of the emergency telegram, since it has the same COB-ID.
[2] From version 15.00.00

**Tip!**

If required, the identifiers of the PDOs and the identifiers for the sync telegram can be changed via parameters/indices.

- ▸ Identifiers of the process data objects (📖 801)
- ▸ Synchronisation of PDOs via sync telegram (📖 805)

The active identifiers of the PDOs are displayed in C00355/x.

### 12.6.2 User data

All nodes communicate by exchanging data telegrams via the system bus. The user data area of the CAN telegram either contains network management data or parameter data or process data:

**Network management data**

(NMT data)

- Control information on start, stop, reset, etc. of communication to specific nodes or to all nodes of the CAN network.

**Process data**

(PDOs – process data objects)

- Process data are transferred via the process data channel.
- Process data can be used to control the inverter.
- Process data are not saved in the inverter.
- Process data are transmitted between host system and nodes to ensure continuous exchange of current input and output data.
- Process data usually are unscaled/scalable raw data.
- Process data are, for instance, setpoints and actual values.
- The exact meaning of the PDO file contents is determined via the function block editor (FB Editor) in the I/O level or via the PDO mapping.

**Parameter data**

(SDOs – service data objects)

- Parameter data are the CANopen indexes or, in case of Lenze devices, the codes.
- Parameters are, for instance, used for one-off plant setting during commissioning or when the material is changed on a production machine.
- Parameter data are transmitted as SDOs via the parameter data channel. They are acknowledged by the receiver, i.e. the transmitter gets a feedback about the transmission being successful or not.
- The parameter data channel enables access to all Lenze codes and CANopen indexes.
- Parameter changes are automatically saved to the inverter until mains switching.
- In general, the parameter transfer is not time-critical.
- Parameter data are, for instance, operating parameters, diagnostic information and motor data as well as control information on the interconnection of function blocks in the I/O level of the FB Editor.

## 12.7 Communication phases/network management

Regarding communication via the system bus, the inverter distinguishes between the following statuses:

| Status | Explanation |
|---|---|
| **"Initialisation"** (Initialisation) | After switch-on, an initialisation run is carried out.<br>• During this phase, the inverter is not involved in the data exchange via the bus.<br>• The standard values are re-written to all CAN-relevant parameters.<br>• After initialisation is completed, the inverter is automatically set to the "Pre-Operational" status. |
| **"Pre-Operational"** (before being ready for operation) | Parameter data can be received, process data are ignored. |
| **"Operational"** (ready for operation) | Parameter data and process data can be received! |
| **"Stopped"** (stopped) | Only network management telegrams can be received. |

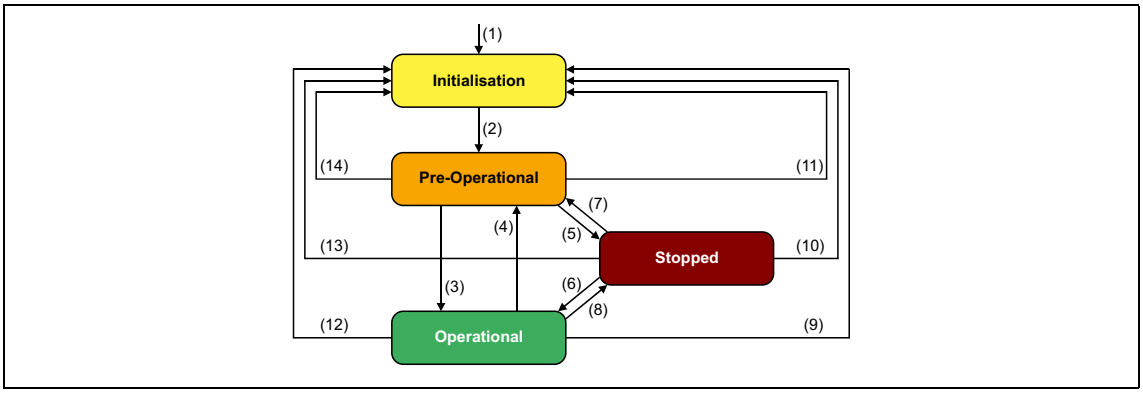
| Communication object | Initialisation | Pre-Operational | Operational | Stopped |
|---|---|---|---|---|
| PDO | | | ● | |
| SDO | | ● | ● | |
| Sync | | ● | ● | |
| Emergency | | ● | ● | |
| Boot-up | ● | | | |
| Network management (NMT) | | ● | ● | ● |

**Tip!**

Part of the initialisation or the entire initialisation can be carried out anew in every status by transferring the corresponding network management telegrams.

The current CAN status is displayed in C00359 for diagnostic purposes.

### 12.7.1 Status transitions

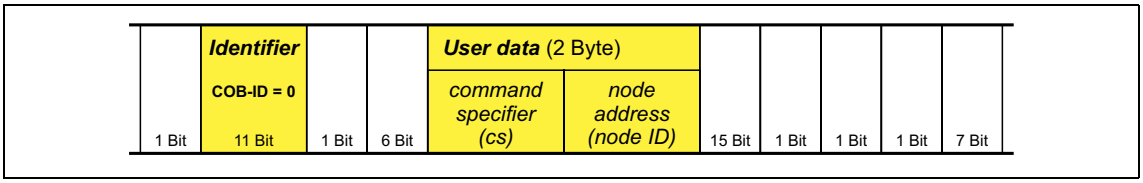



[12-3] NMT status transitions in the CAN network

| Transition | NMT command | Status after change | Effects on process/parameter data after status change |
|---|---|---|---|
| (1) | - | Initialisation | Initialisation starts automatically when the mains is switched on.<br>• During initialisation, the inverter is not involved in the data exchange.<br>• After the initialisation is completed, the node sends a boot-up message with an individual identifier and automatically changes to the "pre-operational" status. |
| (2) | - | Pre-Operational | In this phase, the master determines the way in which the node(s) takes/take part in communication. |
| | From here, the master changes the statuses for the entire network.<br>• A target address included in the NMT command defines the receiver(s).<br>• If the 8400 inverter is configured as CAN master, the status is automatically changed to "Operational" after a waiting time has expired (C00356/1), and the 0x0100 ("Start remote node") NMT command is transmitted to all nodes.<br>• Data can only be exchanged via process data objects if the status is "Operational"! | | |
| (3), (6) | 0x01 xx<br>Start remote node | Operational | Network management/sync/emergency telegrams as well as process data (PDO) and parameter data (SDO) are active.<br>Optional: When the status is changed, event and time-controlled process data (PDOs) are transmitted once. |
| (4), (7) | 0x80 xx<br>Enter Pre-Operational | Pre-Operational | Network management/sync/emergency telegrams and parameter data (SDO) are active. |
| (5), (8) | 0x02 xx<br>Stop remote node | Stopped | Only network management telegrams can be received. |
| (9), (10), (11) | 0x81 xx<br>Reset node | Initialisation | All CAN-relevant parameters (CiA DS 301) are initialised with the saved values. |
| (12), (13), (14) | 0x82 xx<br>Reset communication | | All CAN-relevant parameters (CiA DS 301) are initialised with the saved values. |
| | Meaning of the node address in the NMT command:<br>• xx = 0x00: If this assignment is selected, the telegram addresses all nodes (broadcast telegram). The status of all nodes can be changed at the same time.<br>• xx = Node ID: If a node address is specified, only the status of the node with the corresponding address changes. | | |

### 12.7.2 Network management telegram (NMT)

The telegram for the network management contains identifier "0" and the command included in the user data which consists of the command byte and the node address:




[12-4] Network management telegram for changing over the communication phases

| Command specifier (cs) | | NMT command |
|---|---|---|
| dec | hex | |
| 1 | 0x01 | Start remote node |
| 2 | 0x02 | Stop remote node |
| 128 | 0x80 | Enter Pre-Operational |
| 129 | 0x81 | Reset node |
| 130 | 0x82 | Reset communication |

The communication phases are changed over by a node, the CAN master, for the entire network. The CAN master can also be a inverter. ▸ Parameterising the inverter as CAN master (📖 786)

Meaning of the node address in the user data:

- node ID = "0": The telegram addresses all nodes (broadcast telegram). The status of all nodes can be changed at the same time.
- node ID = "1" ... "127": If a node address is specified, only the status of the node with the corresponding address changes.

**Example :**

Data can only be exchanged via process data objects if the status is "Operational".If the CAN master is supposed to switch all nodes connected to the bus from the "Pre-Operational" communication status to the "Operational" communication status, the identifier and user data in the transmission telegram must be set as follows:

- Identifier: 0x00 (network management)
- User data: 0x0100 ("Start remote node" NMT command to all nodes)

### 12.7.3 Parameterising the inverter as CAN master

If the initialisation of the system bus and the associated status change from "Pre-Operational" to "Operational" is not effected by a superimposed host system, the inverter can instead be defined to be a "quasi" master to execute this task.

The inverter is configured as CAN master in C00352.

- Being the CAN master, the inverter sets all nodes connected to the bus (broadcast telegram) to the "Operational" communication status with the "Start remote node" NMT telegram. Only in this communication status, data can be exchanged via process data objects.
- A delay time can be set in C00356/1 which must expire after mains switching before the inverter transmits the "Start remote node" NMT telegram.

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00352 | CAN slave/master | slave | |
| C00356/1 | CAN delay boot-up - Operational | 3000 | ms |

**Note!**

The changes of the master/slave operation in C00352 will not be activated until

- another mains switching of the inverter

or

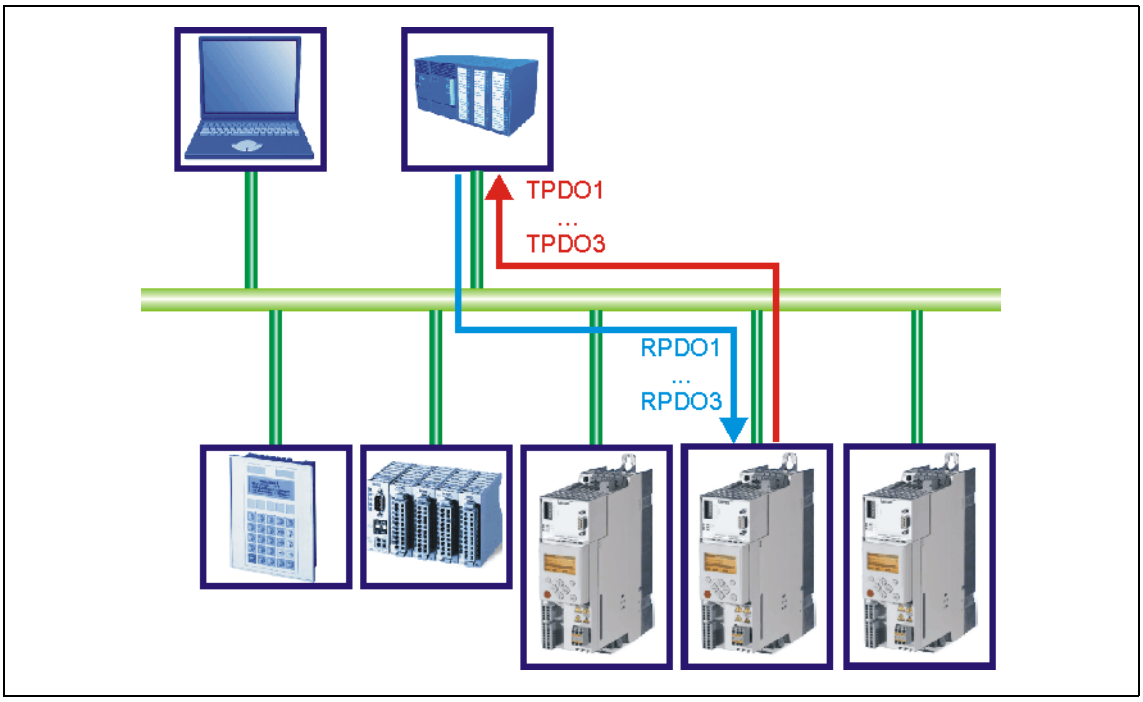
- the "Reset node" or "Reset communication" NMT telegram has been transmitted to the inverter.

The "CAN reset node" device command (C00002/26) is provided as an alternative to the "Reset node" NMT telegram for the reinitialisation of the CAN-specific device parameters.

**Tip!**

Master functionality is only required during the initialisation phase of the drive system.

## 12.8 Process data transfer




[12-5] PDO data transfer from / to the higher-level host system

For transmitting process data, up to four separated process data channels (PDO1 ... PDO4) are available depending on the device version and software version.

**Definitions**

- Process data telegrams between the host system and the devices are distinguished in terms of direction as follows:
  - Process data telegrams to the device (RPDO)
  - Process data telegrams from the device (TPDO)
- The CANopen process data objects are designated as seen from the node's view:
  - Receive PDOs (RPDOx): Process data object received by a node
  - Transmit PDOs (TPDOx): Process data object sent by a node

**Note!**

Data can only be exchanged via process data objects if the status is "Operational"!

▸ Communication phases/network management (📖 783)

### 12.8.1 Available process data objects

The number of available process data objects for inverters of the 8400 series depends on the device version:

| Process data object | Version | |
|---|---|---|
| | BaseLine C | StateLine C<br>HighLine C<br>TopLine C |
| RPDO1 \| Port block "LP_CanIn1" | ● | ● |
| RPDO2 \| "LP_CanIn2" port block | ● | ● |
| RPDO3 \| "LP_CanIn3" port block | | ● |
| RPDO4 \| "LP_CanIn4" port block | | ● (from version 15.00.00) |
| TPDO1 \| "LP_CanOut1" port block | ● | ● |
| TPDO2 \| "LP_CanOut2" port block | ● | ● |
| TPDO3 \| "LP_CanOut3" port block | | ● |
| TPDO4 \| "LP_CanOut4" port block | | ● (from version 15.00.00) |

#### Receive PDOs (RPDOs)

The process data objects transmitted from the system bus to the drive are processed via the LP_CanIn1 ... LP_CanIn4 port blocks.

- Every port block provides 4 words (2 bytes/word). The data of every first word are provided in a bit decoded manner (bit 0 ... 15).
- The first word of the LP_CanIn1 port block is defined as control word *wCtrl*. The *wCtrl* control word does not have a permanent connection to the device control and can be used as required. The predefined assignment of the *wCtrl* control word in the C00007 = "30: CAN" control mode depends on the technology application selected in C00005:
  - **TA "Actuating drive speed":**
    Process data assignment for fieldbus communication (📖 458)
  - **TA "Table positioning":**
    Process data assignment for fieldbus communication (📖 521)
  - **TA "Abschaltpositionierung":**
    Process data assignment for fieldbus communication (📖 547)

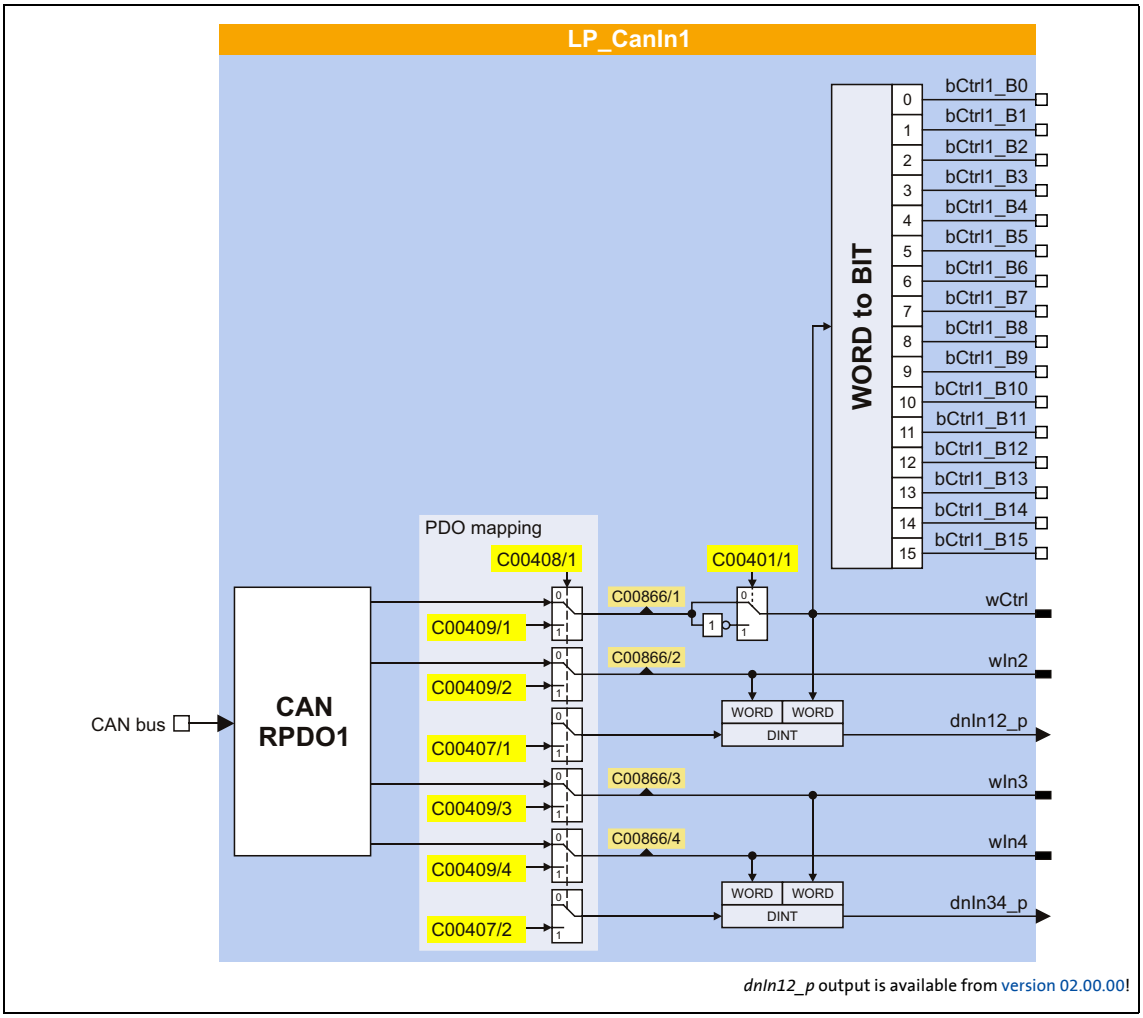
#### Transmit PDOs (TPDOs)

The process data transmitted from the drive to the system bus are processed via the LP_CanOut1 ... LP_CanOut4 port blocks.

- Every port block receives 4 words (2 bytes/word). The data of every first word are transmitted bit by bit (bit 0 ... 15).
- The first word of the LP_CanOut1 port block is defined as the *wState* status word. The *wState* status word does not have a permanent connection to the device control and can be used as required.
  - Vordefinierte Belegung siehe wDeviceStatusWord status word der Antriebsschnittstelle.

### 12.8.1.1 RPDO1 | Port block "LP_CanIn1"

The **LP_CanIn1** port block maps process data object RPDO1 in the FB Editor.



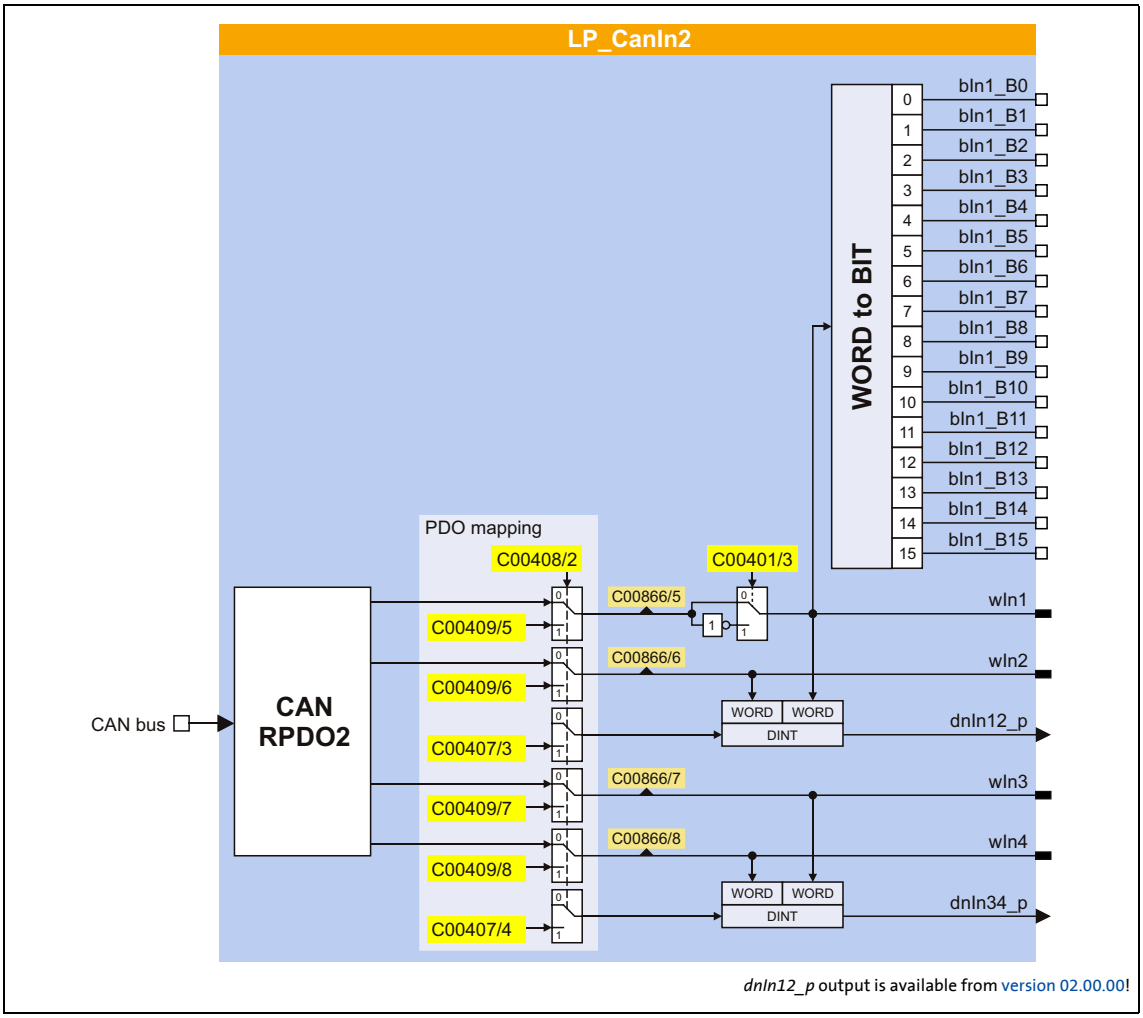

**Short overview of the parameters for LP_CanIn1:**

| Parameters | Info | Lenze setting |
|---|---|---|
| C00401/1 | LP_CanIn1: Inversion bCtrl1_B0..15 | 0x0000 |
| C00866/1 | LP_CanIn1: wCtrl | - |
| C00866/2 | LP_CanIn1: wIn2 | - |
| C00866/3 | LP_CanIn1: wIn3 | - |
| C00866/4 | LP_CanIn1: wIn4 | - |
| **PDO mapping** | | |
| C00408/1 | LP_CanIn1: Mapping selection | CanIn |
| C00409/1 | LP_CanIn1: wCtrl MapVal | 0 |
| C00409/2 | LP_CanIn1: wIn2 MapVal | 0 |
| C00409/3 | LP_CanIn1: wIn3 MapVal | 0 |
| C00409/4 | LP_CanIn1: wIn4 MapVal | 0 |
| Greyed out = display parameter | | |

| Parameters | Info | Lenze setting |
|---|---|---|
| C00407/1 | LP_CanIn1: dwIn12 MapVal<br>• From version 12.00.00<br>• The mapping for the double word is ORed with the mapping setting in C00409/1 and C00409/2. | 0 |
| C00407/2 | LP_CanIn1: dwIn34 MapVal<br>• From version 12.00.00<br>• The mapping for the double word is ORed with the mapping setting in C00409/3 and C00409/4. | 0 |
| Greyed out = display parameter | | |

### 12.8.1.2 RPDO2 | "LP_CanIn2" port block

The **LP_CanIn2** port block maps process data object RPDO2 in the FB Editor.



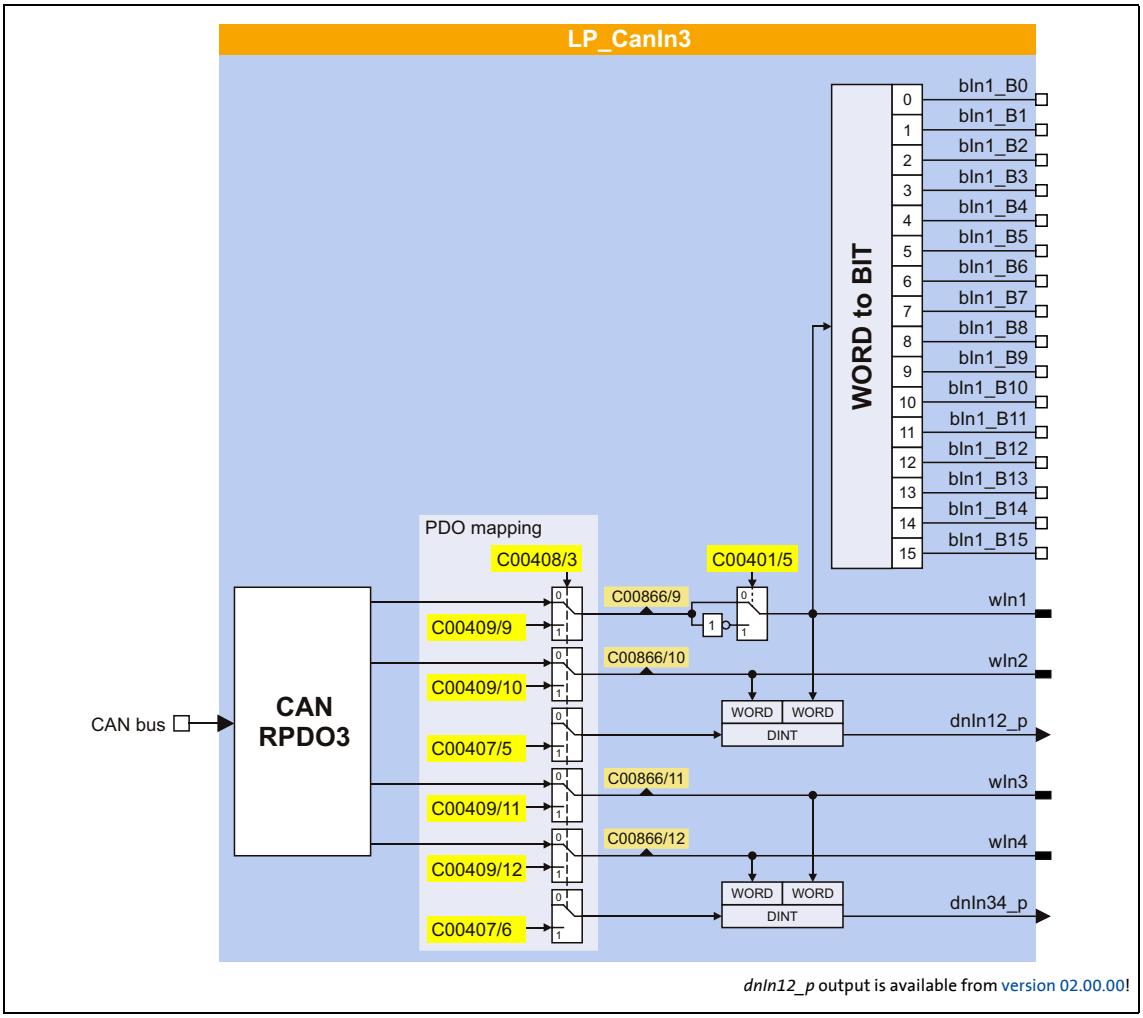

**Short overview of the parameters for LP_CanIn2:**

| Parameters | Info | Lenze setting |
|---|---|---|
| C00401/3 | LP_CanIn2: Inversion bIn1_B0..15 | 0x0000 |
| C00866/5 | LP_CanIn2: wIn1 | - |
| C00866/6 | LP_CanIn2: wIn2 | - |
| C00866/7 | LP_CanIn2: wIn3 | - |
| C00866/8 | LP_CanIn2: wIn4 | - |
| **PDO mapping** | | |
| C00408/2 | LP_CanIn2: Mapping selection | CanIn |
| C00409/5 | LP_CanIn2: wIn1 MapVal | 0 |
| C00409/6 | LP_CanIn2: wIn2 MapVal | 0 |
| C00409/7 | LP_CanIn2: wIn3 MapVal | 0 |
| C00409/8 | LP_CanIn2: wIn4 MapVal | 0 |
| Greyed out = display parameter | | |

| Parameters | Info | Lenze setting |
|---|---|---|
| C00407/3 | LP_CanIn2: dwIn12 MapVal<br>• From version 12.00.00<br>• The mapping for the double word is ORed with the mapping setting in C00409/5 and C00409/6. | 0 |
| C00407/4 | LP_CanIn2: dwIn34 MapVal<br>• From version 12.00.00<br>• The mapping for the double word is ORed with the mapping setting in C00409/7 and C00409/8. | 0 |
| Greyed out = display parameter | | |

### 12.8.1.3 RPDO3 | "LP_CanIn3" port block

The **LP_CanIn3** port block maps process data object RPDO3 in the FB Editor.




**Short overview of the parameters for LP_CanIn3:**

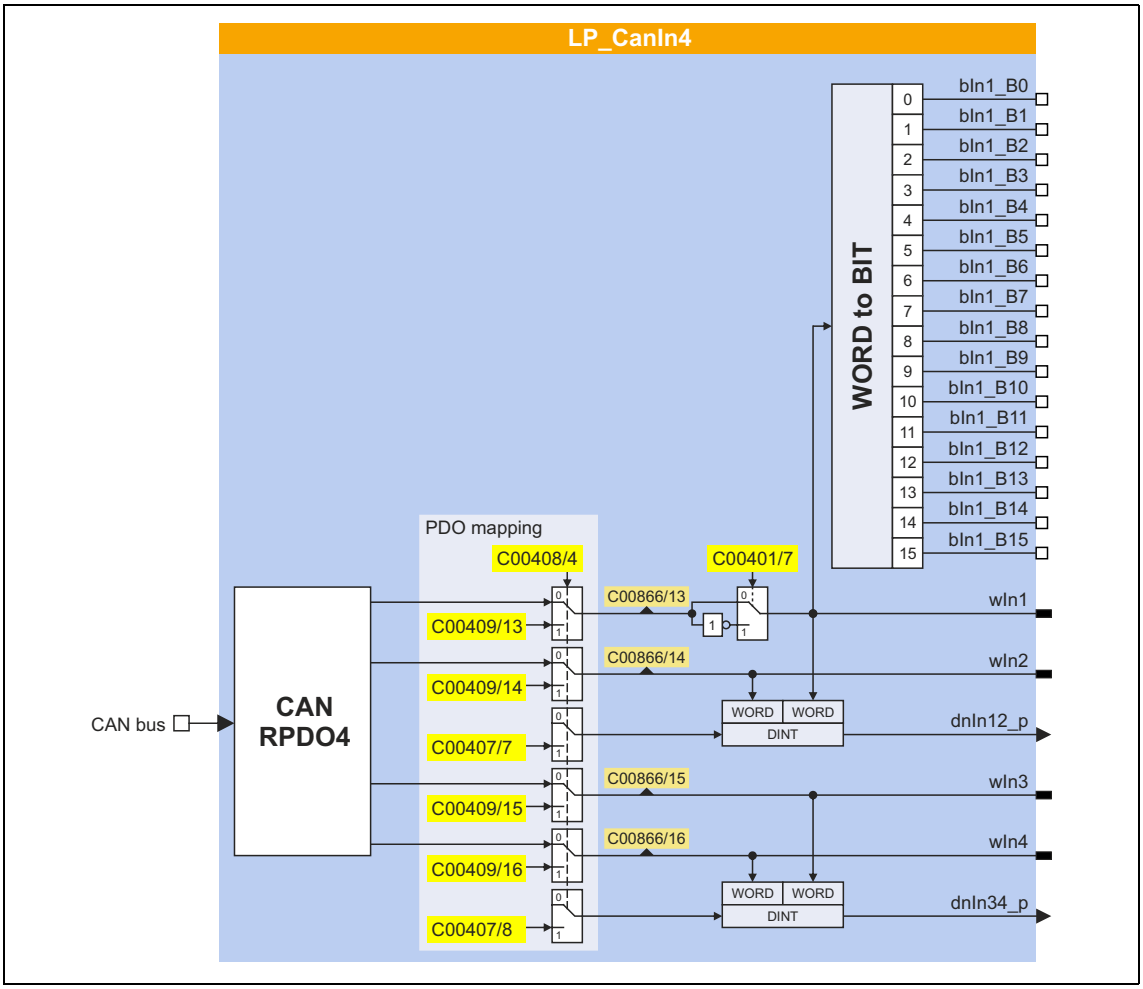
| Parameters | Info | Lenze setting |
|---|---|---|
| C00401/5 | LP_CanIn3: Inversion bIn1_B0..15 | 0x0000 |
| C00866/9 | LP_CanIn3: wIn1 | - |
| C00866/10 | LP_CanIn3: wIn2 | - |
| C00866/11 | LP_CanIn3: wIn3 | - |
| C00866/12 | LP_CanIn3: wIn4 | - |
| **PDO mapping** | | |
| C00408/3 | LP_CanIn3: Mapping selection | CanIn |
| C00409/9 | LP_CanIn3: wIn1 MapVal | 0 |
| C00409/10 | LP_CanIn3: wIn2 MapVal | 0 |
| C00409/11 | LP_CanIn3: wIn3 MapVal | 0 |
| C00409/12 | LP_CanIn3: wIn4 MapVal | 0 |
| Greyed out = display parameter | | |

| Parameters | Info | Lenze setting |
|---|---|---|
| C00407/5 | LP_CanIn3: dwIn12 MapVal<br>• From version 12.00.00<br>• The mapping for the double word is ORed with the mapping setting in C00409/9 and C00409/10. | 0 |
| C00407/6 | LP_CanIn3: dwIn34 MapVal<br>• From version 12.00.00<br>• The mapping for the double word is ORed with the mapping setting in C00409/11 and C00409/12. | 0 |
| Greyed out = display parameter | | |

### 12.8.1.4 RPDO4 | "LP_CanIn4" port block

This function extension is available from version 15.00.00!

The **LP_CanIn4** port block maps process data object RPDO4 in the FB Editor.



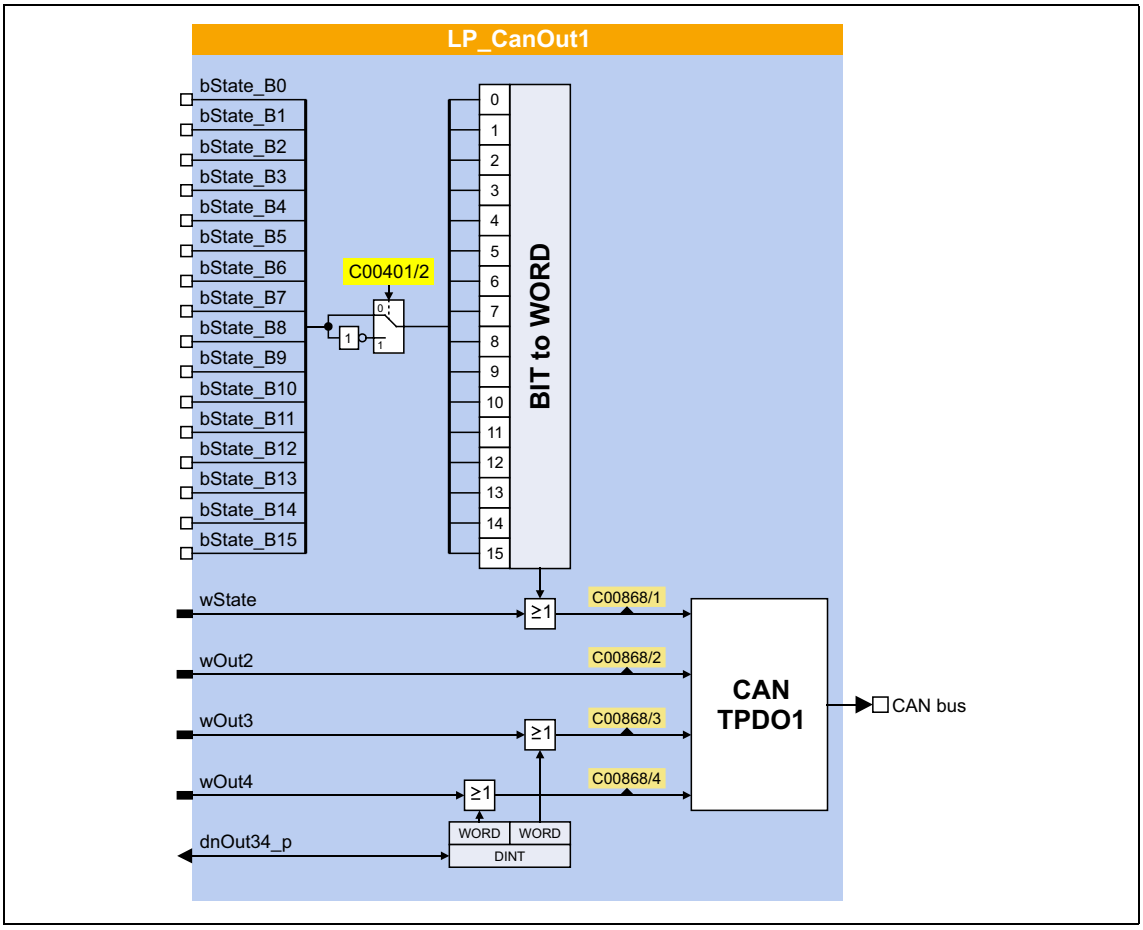

**Short overview of the parameters for LP_CanIn4:**

| Parameters | Info | Lenze setting |
|---|---|---|
| C00401/7 | LP_CanIn4: Inversion bIn1_B0..15 | 0x0000 |
| C00866/13 | LP_CanIn4: wIn1 | - |
| C00866/14 | LP_CanIn4: wIn2 | - |
| C00866/15 | LP_CanIn4: wIn3 | - |
| C00866/16 | LP_CanIn4: wIn4 | - |
| **PDO mapping** | | |
| C00408/4 | LP_CanIn4: Mapping Selection | CanIn |
| C00409/13 | LP_CanIn4: wIn1 MapVal | 0 |
| C00409/14 | LP_CanIn4: wIn2 MapVal | 0 |
| C00409/15 | LP_CanIn4: wIn3 MapVal | 0 |
| C00409/16 | LP_CanIn4: wIn4 MapVal | 0 |
| Greyed out = display parameter | | |

| Parameters | Info | Lenze setting |
|---|---|---|
| C00407/7 | LP_CanIn4: dwIn12 MapVal<br>• The mapping for the double word is ORed with the mapping setting in C00409/13 and C00409/14. | 0 |
| C00407/8 | LP_CanIn4: dwIn34 MapVal<br>• The mapping for the double word is ORed with the mapping setting in C00409/15 and C00409/16. | 0 |
| Greyed out = display parameter | | |

### 12.8.1.5 TPDO1 | "LP_CanOut1" port block

The **LP_CanOut1** port block maps process data object TPDO1 in the FB Editor.

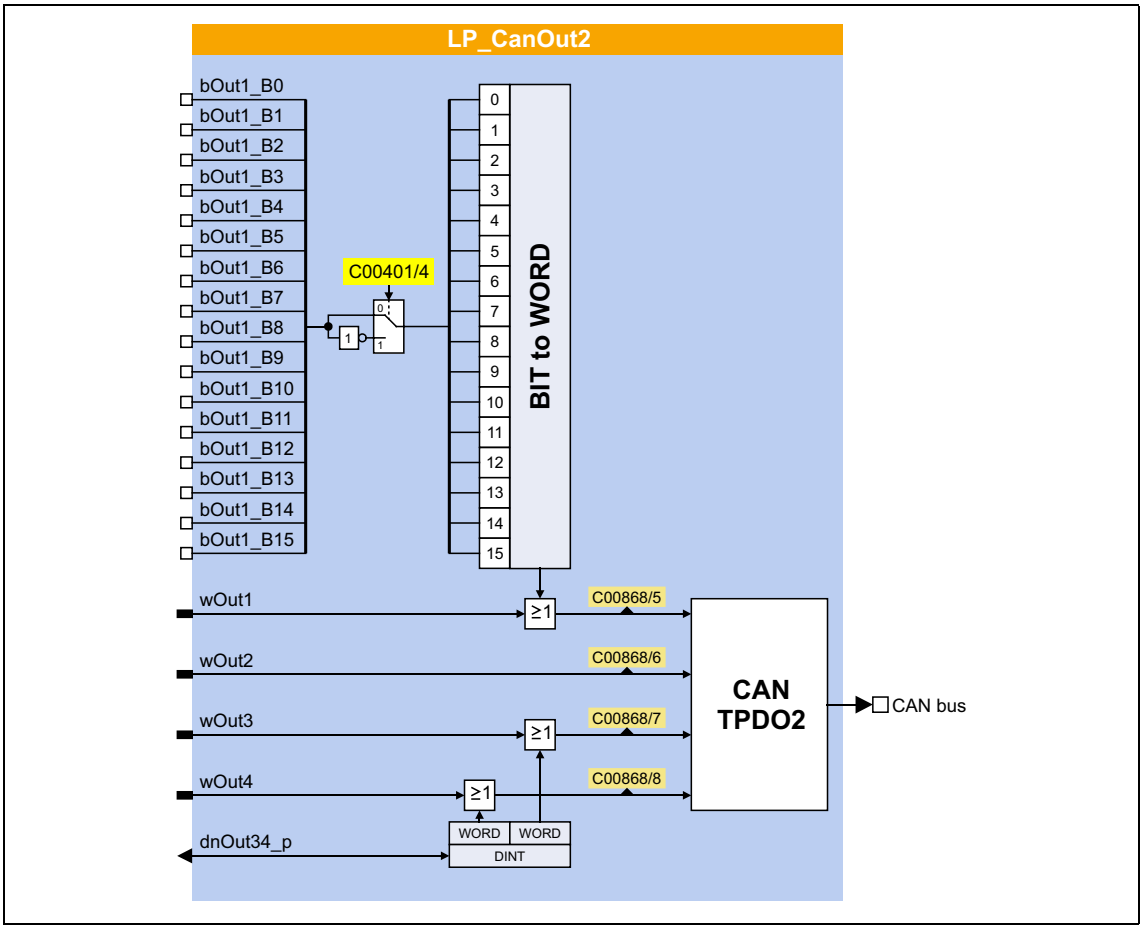



**Short overview of the parameters for LP_CanOut1:**

| Parameters | Info | Lenze setting |
|---|---|---|
| C00401/2 | LP_CanOut1: Inversion bState_B0..15 | 0x0000 |
| C00868/1 | LP_CanOut1:wState | - |
| C00868/2 | LP_CanOut1:wOut2 | - |
| C00868/3 | LP_CanOut1:wOut3 | - |
| C00868/4 | LP_CanOut1: wOut4 | - |
| Greyed out = display parameter | | |

### 12.8.1.6 TPDO2 | "LP_CanOut2" port block

The **LP_CanOut2** port block maps process data object TPDO2 in the FB Editor.

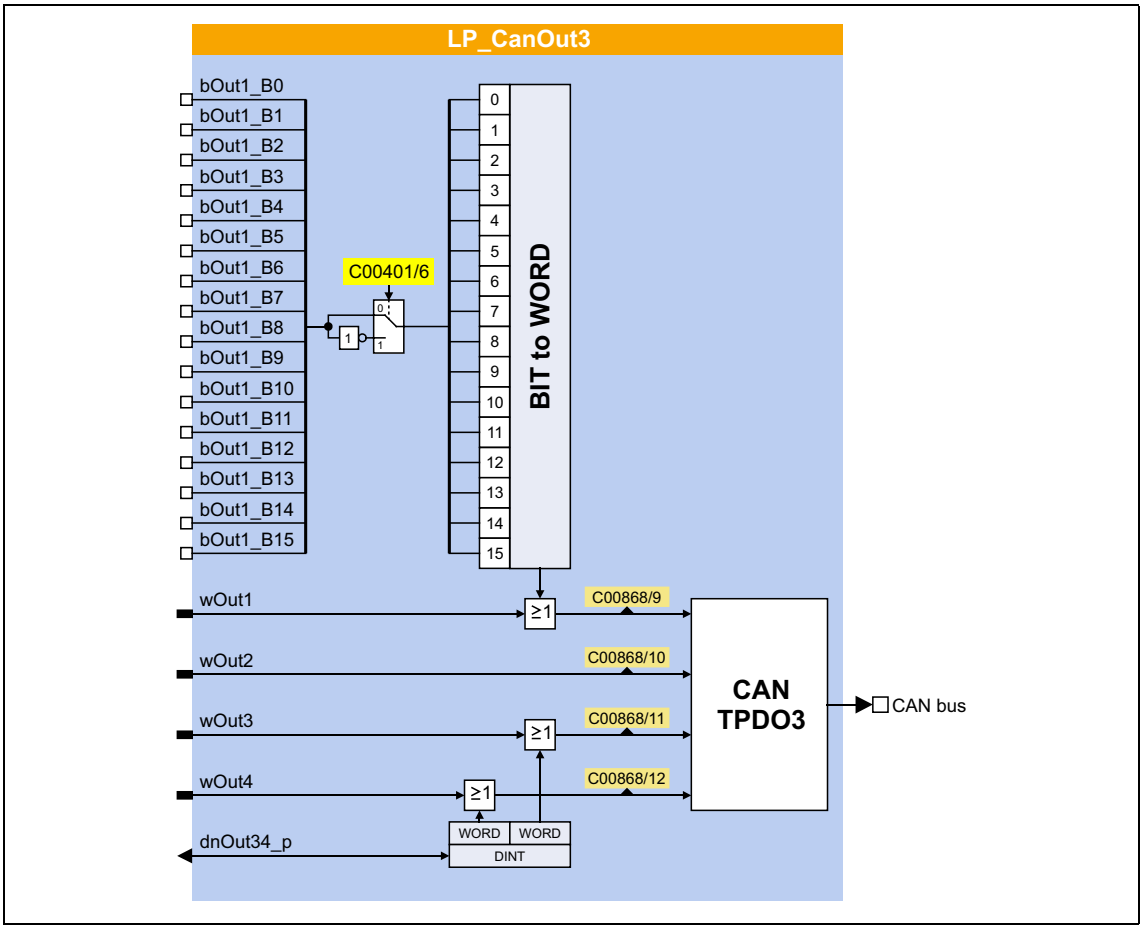



**Short overview of the parameters for LP_CanOut2:**

| Parameters | Info | Lenze setting |
|---|---|---|
| C00401/4 | LP_CanOut2: Inversion bOut1_B0..15 | 0x0000 |
| C00868/5 | LP_CanOut2: wOut1 | - |
| C00868/6 | LP_CanOut2: wOut2 | - |
| C00868/7 | LP_CanOut2: wOut3 | - |
| C00868/8 | LP_CanOut2: wOut4 | - |
| Greyed out = display parameter | | |

### 12.8.1.7 TPDO3 | "LP_CanOut3" port block

The **LP_CanOut3** port block maps process data object TPDO3 in the FB Editor.



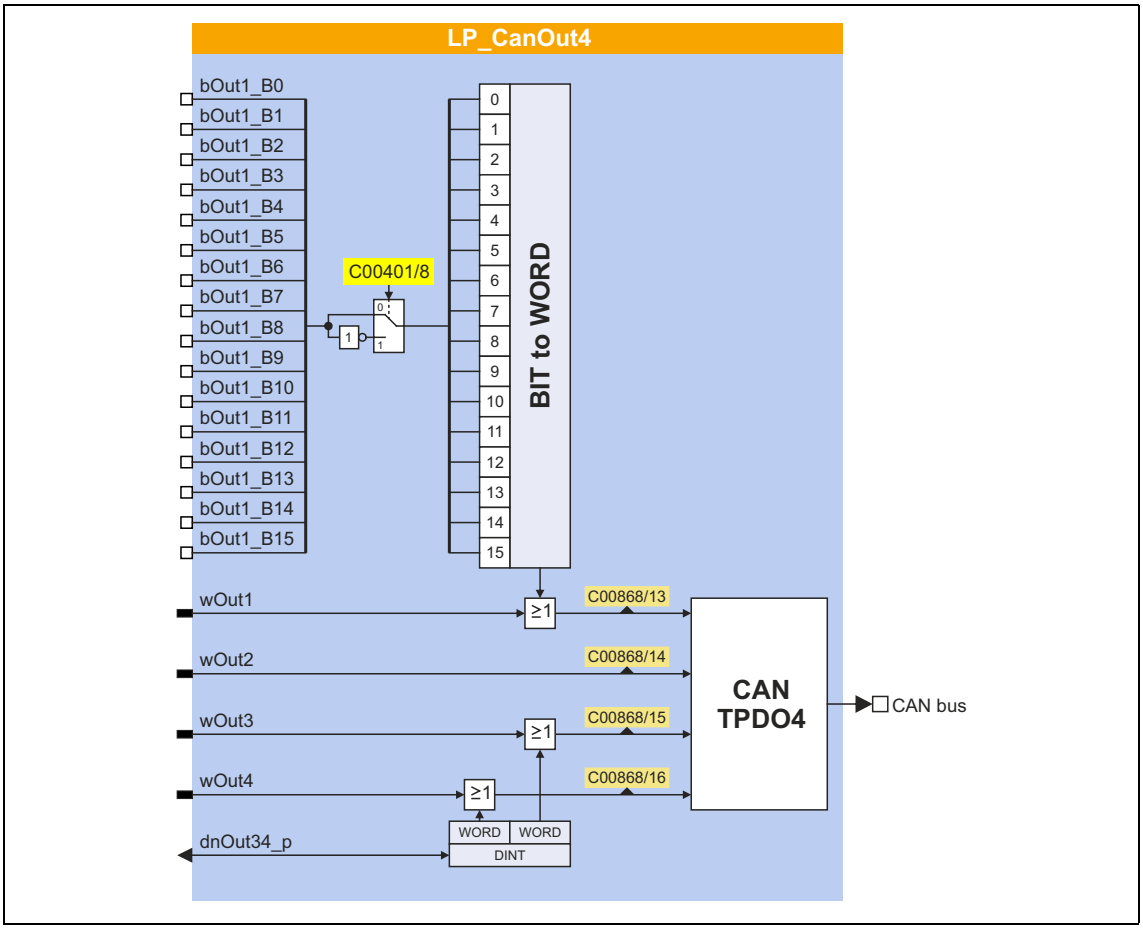

**Short overview of the parameters for LP_CanOut3:**

| Parameters | Info | Lenze setting |
|---|---|---|
| C00401/6 | LP_CanOut3: Inversion bOut1_B0..15 | 0x0000 |
| C00868/9 | LP_CanOut3: wOut1 | - |
| C00868/10 | LP_CanOut3: wOut2 | - |
| C00868/11 | LP_CanOut3: wOut3 | - |
| C00868/12 | LP_CanOut3: wOut4 | - |
| Greyed out = display parameter | | |

### 12.8.1.8 TPDO4 | "LP_CanOut4" port block

This function extension is available from version 15.00.00!

The **LP_CanOut4** port block maps process data object TPDO4 in the FB Editor.




**Short overview of the parameters for LP_CanOut4:**

| Parameters | Info | Lenze setting |
|---|---|---|
| C00401/8 | LP_CanOut4: Inversion bOut1_B0..15 | 0x0000 |
| C00868/13 | LP_CanOut4: wOut1 | - |
| C00868/14 | LP_CanOut4: wOut2 | - |
| C00868/15 | LP_CanOut4: wOut3 | - |
| C00868/16 | LP_CanOut4: wOut4 | - |
| Greyed out = display parameter | | |

### 12.8.2 Identifiers of the process data objects

For the process data objects PDO1 ... PDO4, three identifier assignment procedures are available in C00353/1...4 :

- Identifier (COB-ID) = node address (C00350) + CANBaseID (Lenze setting)
- Identifier (COB-ID) = node address (C00350) + LenzeBaseID
- Identifier (COB-ID) = C0354/x

**Lenze setting: Basic identifiers according to the "Predefined Connection Set" of DS301 V4.02**

In the Lenze setting, the identifiers for the process data objects PDO1 ... PDO4 consist of the node address set in C00350 and a basic identifier (CANBaseID) which corresponds to the "Predefined Connection Set" of DS301 V4.02:

| Object | | | Basis identifier (CANBaseID) | |
|---|---|---|---|---|
| PDO1 | TPDO1 | COB-ID = node address (C00350) + | 384 | 0x180 |
| | RPDO1 | | 512 | 0x200 |
| PDO2 | TPDO2 | | 640 | 0x280 |
| | RPDO2 | | 768 | 0x300 |
| PDO3 | TPDO3 | | 896 | 0x380 |
| | RPDO3 | | 1024 | 0x400 |
| PDO4 | TPDO4 | | 1152 | 0x480 |
| | RPDO4 | | 1280 | 0x500 |

**Tip!**

The active identifiers (COB-ID) are displayed in C00355/x.

**OPTIONAL: Use the basic identifier according to Lenze definition (LenzeBaseID)**

If the basic identifiers shall be used instead according to the Lenze definition, go to C00353/x and select "0: COBID = C0350 + LenzeBaseID" for the coresponding PDO.

The basic Lenze identifiers which differ from the "Predefined Connection Set" are highlighted in colour in the following table:

| Object | | | Basic identifier (LenzeBaseID) | |
|---|---|---|---|---|
| PDO1 | TPDO1 | COB-ID = node address (C00350) + | 384 | 0x180 |
| | RPDO1 | | 512 | 0x200 |
| PDO2 | TPDO2 | | 640 | 0x280 |
| | RPDO2 | | 641 | 0x281 |
| PDO3 | TPDO3 | | 768 | 0x300 |
| | RPDO3 | | 769 | 0x301 |
| PDO4 | TPDO4 | | 896 | 0x380 |
| | RPDO4 | | 897 | 0x381 |

**OPTIONAL: Set identifier individually**

When "2: COBID = C0354/x" in C00353/x, the identifiers for the PDOs can be individually set via the Lenze codes and CANopen indexes listed in the table below. That way, identifiers independent of the node address can be set for specific PDOs.

- If identifiers are assigned individually, all PDOs must have basic identifier values in the range of 385 ... 1407.

| Object | | | Lenze code | CANopen index | Default setting |
|---|---|---|---|---|---|
| PDO1 | TPDO1 | COB-ID = | C00354/1 | I-1400/1 | 0x201 |
| | RPDO1 | | C00354/2 | I-1800/1 | 0x181 |
| PDO2 | TPDO2 | | C00354/3 | I-1401/1 | 0x301 |
| | RPDO2 | | C00354/4 | I-1801/1 | 0x281 |
| PDO3 | TPDO3 | | C00354/5 | I-1402/1 | 0x401 |
| | RPDO3 | | C00354/6 | I-1802/1 | 0x381 |
| PDO4 | TPDO4 | | C00354/7 | I-1403/1 | 0x501 |
| | RPDO4 | | C00354/8 | I-1803/1 | 0x481 |

**Note!**

After a node address change (C00350) and a CAN reset node afterwards, the subcodes of C00354 automatically resume the values which result from the respective basic identifier and the set node address.

### 12.8.3 Transmission type

Process data objects can be transmitted in an event-controlled or time-controlled manner. The below table shows that it is possible to combine the different methods by means of logic operations (AND, OR):

- Event-controlled
  The PDO is sent when a special device-internal event has occurred, e.g. when the data contents of the TPDO have changed or when a transmission cycle time has elapsed
- Synchronous transmission
  A TPDO (or RPDO) is transmitted (or received) after the device has received a sync telegram (COB-ID 0x80).
- Cyclic transmission
  The cyclic transmission of PDOs takes place when the transmission cycle time has elapsed.
- Polled via RTR
  A TPDO is transmitted when another device requests it by means of a data request telegram (RTR remote transmit request). For this purpose, the data requester (e.g. the master) sends the data request telegram with the COB-ID of the TPDO requested to be sent. The receiver recognises the RTR and transmits the corresponding PDO.

| Transmission type | PDO transmission | | | Logic combination of different transmission types |
|---|---|---|---|---|
| | cyclic | synchronous | event-controlled | |
| 0 | | ● | ● | AND |
| 1 ... 240 | | ● | | - |
| 254, 255 | ● | | ● | OR |

| Transmission type | Description |
|---|---|
| 0 | Synchronous and acyclic:<br>The PDO is transmitted on an event-controlled basis with every sync (e.g. when a bit change occurs in the PDO). |
| 1 ... 240 | Synchronous and cyclic (sync-controlled with response):<br>• Selection n = 1: The PDO is transmitted with every sync.<br>• Selection 1 < n ≤ 240: The PDO is transmitted with every n-th sync.<br>From version 16.00.00 onwards, the transmit PDOs are also sent when the sync telegram is generated. Precondition: the sync Tx identifier and sync Rx identifier have the same value. |
| 241 ... 251 | Reserved |
| 252 | Synchronous - RTR only |
| 253 | Asynchronous - RTR only |
| 254, 255 | Asynchronous - manufacturer-specific / device profile-specific:<br>If this value is entered, the PDO transmission is event-controlled or cyclic.<br>(Note: The values "254" and "255" have the same meaning).<br>For a cyclic transmission, a cycle time must be entered for the respective PDO. In this case, cyclic transmission takes place in addition to event-controlled transmission. |

The communication parameters such as the transmission mode and cycle time can be set freely for every PDO and independently of the settings of other PDOs:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| **CAN1_OUT** | | | |
| C00322/1 | Transmission mode | 254 | |
| C00324/2 | Blocking time | 0 | ms |
| C00356/5 | Cycle time | 0 | ms |
| C00358/1 | Data length | 8 | Bytes |
| **CAN2_OUT** | | | |
| C00322/2 | Transmission mode | 254 | |
| C00324/3 | Blocking time | 0 | ms |
| C00356/2 | Cycle time | 0 | ms |
| C00358/2 | Data length | 8 | Bytes |
| **CAN3_OUT** | | | |
| C00322/3 | Transmission mode | 254 | |
| C00324/4 | Blocking time | 0 | ms |
| C00356/3 | Cycle time | 0 | ms |
| C00358/3 | Data length | 8 | Bytes |
| **CAN4_OUT** (from version 15.00.00 | | | |
| C00322/4 | Transmission mode | 254 | |
| C00324/5 | Blocking time | 0 | ms |
| C00356/6 | Cycle time | 0 | ms |
| C00358/4 | Data length | 8 | Bytes |
| **CAN1_IN ... CAN4_IN** | | | |
| C00323/1...4 | Transmission mode CAN1_IN ... CAN4_IN<br>• In the case of the RPDO serves as monitoring setting in the case of sync-controlled PDOs. | 254 | |

### Blocking time

In C00324/x a "blocking time" can be set which defines the shortest transmission cycle with the transmission type"asynchronous - manufacturer-specific/device profile-specific".

Example: Cycle time = 500 ms, blocking time = 100 ms, sporadic data change:

- With a sporadic data change < 500 ms, quickest transmission takes place every 100 ms due to the set blocking time (event-controlled transmission). The transmission cycle timer is reset to 0 if the transmission has been activated in an event-controlled way.
- In the case of a sporadic data change > 500 ms, due to the cycle time set, transmission takes place every 500 ms (cyclic transmission).

**Tip!**

The communication parameters can also be set via the following CANopen objects:

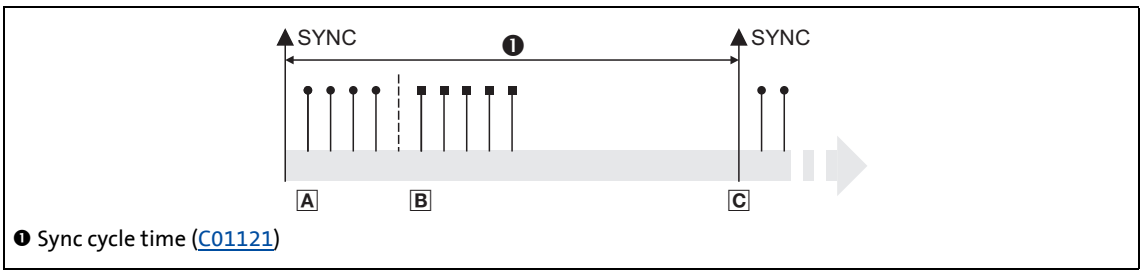
- I-1400 ... I-1403: Communication parameters for RPDO1 ... RPDO4
- I-1800 ... I-1803: Communication parameters for TPDO1 ... TPDO4

### 12.8.4 Synchronisation of PDOs via sync telegram

During cyclic transmission, one or more PDOs are transmitted/received in fixed time intervals. An additional specific telegram, the so-called sync telegram, is used for synchronising cyclic process data.

- The sync telegram is the trigger point for the transmission of process data from the slaves to the master and for the acceptance of process data from the master in the slaves.
- For sync-controlled process data processing, the sync telegram must be generated accordingly.
- The response to a sync telegram is determined by the transmission type selected.
  ▸ Transmission type (📖 803)

#### Basic workflow




❶ Sync cycle time (C01121)

[12-6] Sync telegram

A. After the sync telegram has been received, the slaves transmit the synchronous process data to the master (TPDOs). The master reads them as process input data.
B. When the transmission process is completed, the slaves receive (RPDOs) the process output data (of the master).
   - All other telegrams (e.g. parameters or event-controlled process data) are accepted acyclically by the slaves after the transmission is completed.
   - Illustration [12-6] does not include acyclic data. However, they need to be considered when dimensioning the cycle time.
C. The data are accepted in the slave with the next sync telegram if the Rx mode is set to 1 ... 240. If the Rx mode is 254 or 255, the data are accepted in the next device cycle, irrespective of the sync telegram.

#### Short overview: Parameters for the synchronisation via sync telegram

| Parameters | Info | Lenze setting | | Assignment | |
|---|---|---|---|---|---|
| | | Value | Unit | Sync master | Sync slave |
| C00367 | CAN SYNC Rx identifier | 0x0080 | | | ● |
| C00368 | CAN SYNC Tx identifier | 0x0080 | | ● | |
| C00369 | CAN sync transmission cycle time | 0 | ms | ● | |

#### Related topics:

▸ Synchronisation of the internal time base (📖 880)

### 12.8.5 Monitoring of the RPDOs for data reception

For RPDO1 ... RPDO4 each, a monitoring time can be parameterised within which the RPDO must arrive. If the RPDO is not received within the monitoring time or not with the configured sync, the response parameterised for each RPDO takes place.

**Short overview: Parameters for RPDO monitoring**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00357/1...4 | CAN1...4_IN monitoring time | 3000 | ms |
| C00593/1...4 | Resp. to CAN1...4_IN monitoring | No response | |

### 12.8.6 Configuring exception handling of the CAN PDOs

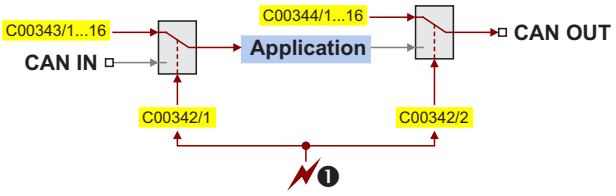
Exception handling for the CAN PDOs in the event of an error can be set via decoupling configuration and decoupling values.

- Bit coded selection is carried out in C00342/1 for the process data words read by the bus, defining the events that will trigger decoupling.
- Bit coded selection is carried out in C00342/2 for the process data words output by the application, defining the events that will trigger decoupling.

| Bit | Event |
|---|---|
| Bit 0 ☐ | BusOff_MsgErr |
| Bit 1 ☐ | Warning |
| Bit 2 ☐ | NodeStopped |
| Bit 3 ☐ | HeartBeatEvent |
| Bit 4 ☐ | CAN1_In_Überw. |
| Bit 5 ☐ | CAN2_In_Überw. |
| Bit 6 ☐ | CAN3_In_Überw. |
| Bit 7 ☐ | CAN4_In_Überw. |
| Bit 8 ☐ | Reserved |
| Bit 9 ☐ | Reserved |
| Bit 10 ☐ | Reserved |
| Bit 11 ☐ | Reserved |
| Bit 12 ☐ | Reserved |
| Bit 13 ☐ | Reserved |
| Bit 14 ☐ | Trouble |
| Bit 15 ☐ | Fault |

Finally, the following parameters define the value that the process data words are to have when they are decoupled:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00343/1 | LP_CanIn1:wCtrl DiscVal | 0 | |
| C00343/2...4 | LP_CanIn1:wIn2...wIn4 DiscVal | 0 | |
| C00343/5...8 | LP_CanIn2:wIn1...wIn4 DiscVal | 0 | |
| C00343/9...12 | LP_CanIn3:wIn1...wIn4 DiscVal | 0 | |
| C00343/13...16 | LP_CanIn4:wIn1...wIn4 DiscVal | 0 | |
| C00344/1 | LP_CanOut1:wState DiscVal | 0 | |
| C00344/2...4 | LP_CanOut1:wOut2...wOut4 DiscVal | 0 | |
| C00344/5...8 | LP_CanOut2:wOut1...wOut4 DiscVal | 0 | |
| C00344/9...12 | LP_CanOut3:wOut1...wOut4 DiscVal | 0 | |
| C00344/13...16 | LP_CanOut4:wOut1...wOut4 DiscVal | 0 | |



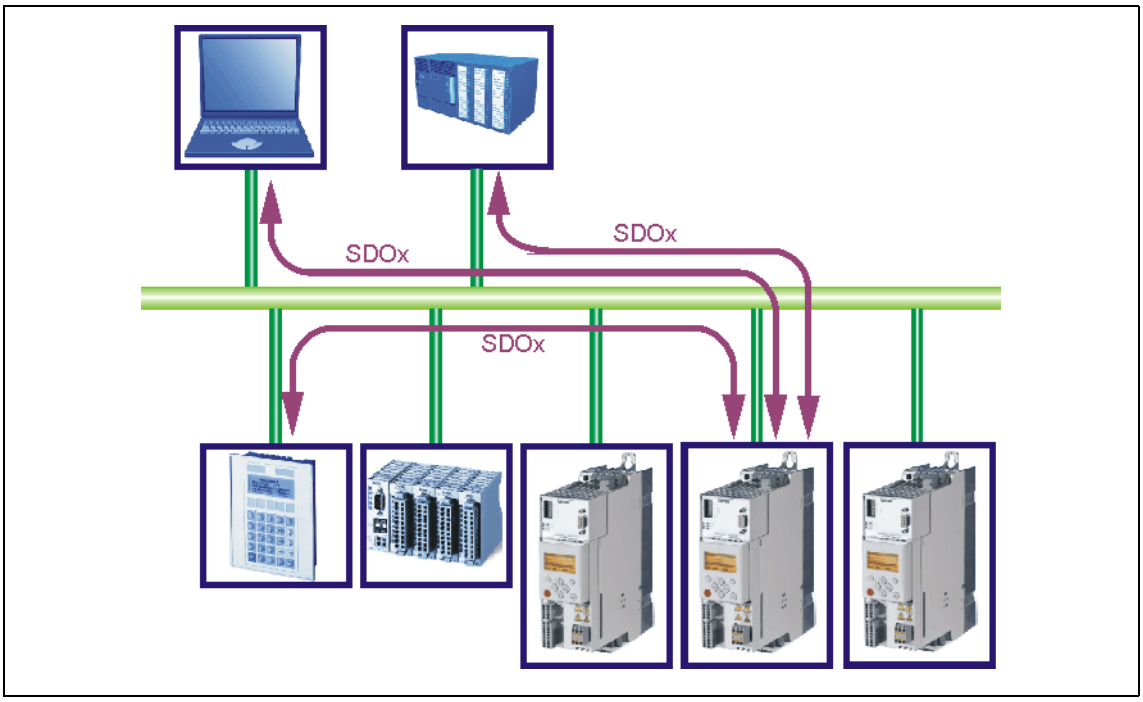

❶ Exception event (an error has occurred)

[12-7] General signal flow in the event of a configured exception

**Related topics:**

- Configuring exception handling of the output terminals (📖 428)

## 12.9 Parameter data transfer




[12-8] Parameter data transfer via the available parameter data channels

Parameters are values stored in codes on Lenze controllers.

Two parameter data channels are available for parameter setting, enabling the simultaneous connection of different devices for configuration purposes.

Parameter data are transmitted via the system bus as SDOs (*Service Data Objects*) and acknowledged by the receiver. The SDO enables read and write access to all device parameters and to the CANopen object directory integrated in the device. Indices (e.g. 0x1000) ensure access to device parameters and functions included in the object directory. To transfer SDOs, the information contained in the user data must comply with the CAN SDO protocol.

**Note!**

In the Lenze setting, only the parameter data channel 1 is activated according to CANopen.

- In C00366, set "2 SDO Lenze" to activate both parameter data channels.

### 12.9.1 Identifiers of the parameter data objects

In the Lenze setting, the basic identifiers of the SDOs are preset according to the "Predefined Connection Set".

The identifiers of the parameter data objects SDO1 and SDO2 result from the basic identifier and the node address set under code C00350:

**Identifier = basic identifier + node address**

| Object | | Direction | | Lenze-Base-ID | | CANopen-Base-ID | |
|---|---|---|---|---|---|---|---|
| | | from device | to device | dec | hex | dec | hex |
| SDO1 (Parameter data channel 1) | TSDO1 | ● | | 1408 | 580 | 1408 | 580 |
| | RSDO1 | | ● | 1536 | 600 | 1536 | 600 |
| SDO2 (Parameter data channel 2) | TSDO2 | ● | | 1472 | 5C0 | 1472 | 5C0 |
| | RSDO2 | | ● | 1600 | 640 | 1600 | 640 |
| Heartbeat | | ● | | 1792 | 700 | 1792 | 700 |
| Boot-up | | ● | | 1792 | 700 | 1792 | 700 |

### 12.9.2 User data

**Structure of the user data of the parameter data telegram**

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| | LOW byte | HIGH byte | | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |

**Note!**

For the user data, the Motorola format is used.

▸ Parameter data telegram examples (📖 815)

The following subchapters provide detailed information on user data.

### 12.9.2.1 Command

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| | LOW byte | HIGH byte | | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |

The following commands can be transmitted or received for writing and reading the parameters:

| Command | 1st byte | | Data length | Info |
|---|---|---|---|---|
| | hex | dec | | |
| Write request | 0x23 | 35 | 4 bytes | Writing of a parameter to the inverter. |
| | 0x2B | 43 | 2 bytes | |
| | 0x2F | 47 | 1 byte | |
| | 0x21 | 33 | Block | |
| Write response | 0x60 | 96 | 4 bytes | Inverter acknowledges a write request. |
| Read request | 0x40 | 64 | 4 bytes | Reading of a parameter from the inverter. |
| Read response | 0x43 | 67 | 4 bytes | Inverter's response to a read request with the current parameter value. |
| | 0x4B | 75 | 2 bytes | |
| | 0x4F | 79 | 1 byte | |
| | 0x41 | 65 | Block | |
| Error response | 0x80 | 128 | 4 bytes | Response from the inverter when the read/write request could not be executed correctly. ▸ Error messages (📖 813) |

More precisely, the command byte comprises the following information:

| Command | 1st byte | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command specifier (cs) | | | Toggle (t) | Length* | | e | s |
| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Write request | 0 | 0 | 1 | 0 | 0/1 | 0/1 | 1 | 1 |
| Write response | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Read request | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read response | 0 | 1 | 0 | 0 | 0/1 | 0/1 | 1 | 1 |
| Error response | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Bit coding of the length: 00 = 4 bytes, 01 = 3 bytes, 10 = 2 bytes, 11 = 1 byte
e: expedited (shortened block service)
s: segmented (normal block service)

**Tip!**

More commands are defined in CANopen specification DS301, V4.02 (e.g. segmented transfer).

### 12.9.2.2 Addressing by means of index and subindex

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| | LOW byte | HIGH byte | | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |

A parameter (a Lenze code) is addressed as per the following formula:
**Index = 24575 - (Lenze code number)**

**Example**

The C00011 parameter (motor reference speed) is to be addressed.

Calculation:

- Index:
  - Decimal: 24575 - 11 = 24564
  - Hexadecimal: 0x5FFF - 0xB = 0x5FF4
- Subindex: 0x00 (subindex 0 since the parameter does not have any subcodes)

Entries:

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| | 0xF4 | 0x5F | 0x00 | | | | |

### 12.9.2.3 Data 1 ... Data 4

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| | LOW byte | HIGH byte | | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |

Maximally 4 bytes are available for parameter value entries. Depending on the data format, they are assigned as follows:

| 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|
| Parameter value (1 byte) | 0x00 | 0x00 | 0x00 |
| Parameter value (2 bytes) | | 0x00 | 0x00 |
| LOW byte | HIGH byte | | |
| Parameter value (4 bytes) | | | |
| LOW word | | HIGH word | |
| LOW byte | HIGH byte | LOW byte | HIGH byte |

**Note!**

The "Factor" column of the Table of attributes contains a so-called scaling factor for all Lenze parameters. The scaling factor is relevant to the transfer of parameter values which have one or more decimal positions in the parameter list.

If the scaling factor is > 1, the value must be multiplied by the indicated scaling factor prior to transmission to be able to transfer the value as an integer. At the SDO client end, the integer must be divided by the scaling factor to obtain the original value including decimal positions again.

**Example**

A value of "123.45" is to be transmitted for a code, unit: "%" (e.g. C00039/1: "Fixed setpoint-JOG1").

In inverters of the 8400 series, parameters with the "%" unit have two decimal positions and hence a scaling factor of "100".

Calculation:

- Value to be transmitted = scaling factor x value
- Data $_{(1 \ldots 4)}$ = 100 x 123.45 = 12345 (0x00 00 30 39)

Entries:

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| | | | | 0x39 | 0x30 | 0x00 | 0x00 |

### 12.9.2.4 Error messages

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Error code | | | |
| 0x80 (128) | LOW byte | HIGH byte | | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |

In the event of an error, the addressed node generates a telegram with the "Error response" (0x80) command.

- The telegram includes the index and subindex of the code where the error occurred.
- The error code is entered in bytes 5 ... 8.
  - The error codes are standardised according to DS301, V4.02.
  - The representation of the error codes is provided in reverse read direction (see example below).

**Example**

Representation of error code "0x06 04 00 41" in bytes 5 ... 8:

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Error code | | | |
| | | | | 0x41 | 0x00 | 0x04 | 0x06 |

### Meaning of the error codes

The error codes are standardised acc. to DS301, V4.02.

| Error code | Explanation |
|---|---|
| 0x0503 0000 | Toggle bit not changed |
| 0x0504 0000 | SDO protocol expired |
| 0x0504 0001 | Invalid or unknown client/server command specifier |
| 0x0504 0002 | Invalid block size (only block mode) |
| 0x0504 0003 | Invalid sequence number (only block mode) |
| 0x0504 0004 | CRC error (only block mode) |
| 0x0504 0005 | Not sufficient memory |
| 0x0601 0000 | Object access not supported |
| 0x0601 0001 | Attempt to read a write-only object |
| 0x0601 0002 | Attempt to write to a read-only object |
| 0x0602 0000 | Object not listed in object directory |
| 0x0604 0041 | Object not mapped to PDO |
| 0x0604 0042 | Number and length of objects to be transferred longer than PDO length. |
| 0x0604 0043 | General parameter incompatibility |
| 0x0604 0047 | General internal device incompatibility |
| 0x0606 0000 | Access denied because of hardware error |
| 0x0607 0010 | Unsuitable data type, unsuitable service parameter length |
| 0x0607 0012 | Unsuitable data type, service parameter length exceeded |
| 0x0607 0013 | Unsuitable data type, service parameter length not long enough |
| 0x0609 0011 | Subindex does not exist |
| 0x0609 0030 | Parameter value range exceeded |
| 0x0609 0031 | Parameter values too high |
| 0x0609 0032 | Parameter values too low |
| 0x0609 0036 | Maximum value falls below minimum value |
| 0x0800 0000 | General error |
| 0x0800 0020 | Data cannot be transferred/saved for application. |
| 0x0800 0021 | Data cannot be transferred/saved for application due to local control. |
| 0x0800 0022 | Data cannot be transferred/saved for application due to current device status. |
| 0x0800 0023 | Dynamic generation of object directory failed or no object directory available (e.g. object directory generated from file, generation not possible because of a file error). |

### 12.9.3 Parameter data telegram examples

#### 12.9.3.1 Read parameters

Task: The heatsink temperature of 43 °C (code C00061, data format INTEGER16, scaling factor 1) of the inverter with node address "5" is to be read.

**Telegram to drive**

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x0605 | 0x40 | 0xC2 | 0x5F | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

| Explanations on the telegram to the drive | |
|---|---|
| Identifier | = 1536 + node address = 1536 + 5 = 1541 = 0x0605<br>(1536 = SDO1 basic identifier to the inverter) |
| Command | = 0x40 = "Read request" (read request of a parameter from the inverter) |
| Index | = 24575 - code number = 24575 - 61 = 24514 = 0x5FC2 |
| Subindex | = 0 (code C00061 does not have any subcodes) |

**Response telegram from drive (if data have been correctly transmitted)**

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | - | - |
| 0x0585 | 0x4B | 0xC2 | 0x5F | 0x00 | 0x2B | 0x00 | - | - |

| Explanations on the telegram from the drive | |
|---|---|
| Identifier | = 1408 + node address = 1408 + 5 = 1413 = 0x0585<br>(1408 = SDO1 basic identifier from the inverter) |
| Command | = 0x4B = "Read Response" (response to the read request with current value) |
| Index | as in telegram to the drive |
| Subindex | |
| Data 1 ... 2 | = 0x002B = 43 [°C] |

### 12.9.3.2 Write parameters

Task: The rated current of the connected motor is to be entered with $I_N$ = 10.20 A (code C00088) into the inverter with node address "2".

| Data 1 ... 4 | Calculation |
|---|---|
| Value for motor current, (data type U16; display factor 1/100) | 10.20 x 100 = 1020 (0x03 FC) |

**Telegram to drive**

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x0602 | 0x23 | 0xA7 | 0x5F | 0x00 | 0xFC | 0x03 | 0x00 | 0x00 |

| Explanations on the telegram to the drive | |
|---|---|
| Identifier | = 1536 + node address = 1536 + 2 = 1538 = 0x0602<br>(1536 = SDO1 basic identifier to the inverter) |
| Command | = 0x23 = "Write request" (write request of a parameter to the inverter) |
| Index | = 24575 - code number = 24575 - 88 = 24487 = 0x5FA7 |
| Subindex | = 0 (code C00088 does not have any subcodes) |
| Data 1 ... 4 | = 10.20 x 100 = 1020 = 0x000003FC<br>(motor current value; data type U32; display factor 1/100) |

**Response telegram from drive (if data have been correctly transmitted)**

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x0582 | 0x60 | 0xA7 | 0x5F | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

| Explanations on the telegram from the drive | |
|---|---|
| Identifier | = 1408 + node address = 1408 + 2 = 1410 = 0x0582<br>(1408 = SDO1 basic identifier from the inverter) |
| Command | = 0x60 = "Write response" (Acknowledgement of the write access from the inverter) |
| Index | as in telegram to the drive |
| Subindex | |

### 12.9.3.3 Read block parameters

Task: The firmware version (code C00099) is to be read from the parameter set of the inverter with node address "12". The firmware version has a length of 11 ASCII characters which are transmitted as a block parameter. Depending on the block, the data width from the 2nd to 8th byte is assigned within the user data.

**Telegram 1 to the drive: Read request**

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x060C | 0x40 | 0x9C | 0x5F | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

| Explanations on the telegram to the drive | |
|---|---|
| Identifier | = 1536 + node address = 1536 + 12 = 1548 = 0x060C<br>(1536 = SDO1 basic identifier to the inverter) |
| Command | = 0x40 = "Read request" (read request of a parameter from the inverter) |
| Index | = 24575 - code number = 24575 - 99 = 24476 = 0x5F9C |
| Subindex | = 0 (code C00099 does not have any subcodes) |

**Response telegram 1 from the drive: Indication of the block length (11 characters)**

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x058C | 0x41 | 0x9C | 0x5F | 0x00 | 0x0B | 0x00 | 0x00 | 0x00 |

| Explanations on the telegram from the drive | |
|---|---|
| Identifier | = 1408 + node address = 1408 + 12 = 1420 = 0x058C<br>(1408 = SDO1 basic identifier from the inverter) |
| Command | = 0x41 = "Read response" (response is block telegram) |
| Index | as in telegram to the drive |
| Subindex | |
| Data 1 ... 4 | = 0x0000000B = data length of 11 characters in the ASCII format |

### Telegram 2 to the drive: Request of the 1st data block

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
| | Command | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 |
| 0x060C | 0x60 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

| Explanations on the telegram to the drive | |
|---|---|
| Command | = 0x60 = "Read segment request" (request: read data block)<br>• Bit 4 = 0 (toggle bit) |
| | **Influence of the toggle bit on the request command**<br>The single blocks are toggled one after another, i.e. first the request is made with the "0x60" command (= 0b011**0***0000), then with the "0x70" command (= 0b011**1***0000), and then again with the "0x60" command, etc.<br>* Toggle bit |

### Response telegram 2 from the drive: Transmission of the 1st data block

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
| | Command | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 |
| 0x058C | 0x00 | 0x30 | 0x31 | 0x2E | 0x30 | 0x30 | 0x2E | 0x30 |
| | | $0_{asc}$ | $1_{asc}$ | $\cdot_{asc}$ | $0_{asc}$ | $0_{asc}$ | $\cdot_{asc}$ | $0_{asc}$ |

| Explanations on the telegram to the drive | |
|---|---|
| Command | = 0x00 = 0b00000000<br>• Bit 4 = 0 (toggle bit) |
| | **Influence of the toggle bit on the transmission command**<br>• The 1st response of the inverter in the command byte is "0b000**0***0000" if bytes 2 ... 8 are completely filled with data and other telegrams are following.<br>• The 2nd response of the inverter in the command byte is "0b00011**1***0000" if bytes 2 ... 8 are completely filled with data and other telegrams are following.<br>* Toggle bit |
| Data 1 ... 7 | = "01.00.0" (ASCII representation) |

### Telegram 3 to the drive: Request of the 2nd data block

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
| | Command | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 |
| 0x060C | 0x70 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

| Explanations on telegram 3 to the drive | |
|---|---|
| Command | = 0x70 = "Read segment request" (request: read data block)<br>• Bit 4 = 1 (toggle bit) |

### Response telegram 3 from the drive: Transmission of the 2nd data block including end identifier

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
| | Command | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 |
| 0x058C | 0x17 | 0x30 | 0x2E | 0x30 | 0x30 | 0x00 | 0x00 | 0x00 |
| | | $0_{asc}$ | $._{asc}$ | $0_{asc}$ | $0_{asc}$ | - | - | - |

| Explanations on telegram 3 from the drive | |
|---|---|
| Command | = 0x17 = 0b00010111:<br>• Bit 0 = 1 (end of transmission)<br>• Bit 1 ... bit 3 = 0b011 (3 bytes do not contain any data)<br>• Bit 4 = 1 (toggle bit) |
| | **Influence of the final bit and the residual data length on the transmission command**<br>• The end of transmission is signalled via the set final bit 0.<br>• Bits 1 ... 3 reveal the number of bytes that do not contain any data anymore.<br>* Toggle bit |
| Data 1 ... 7 | = "0.00" (ASCII representation)<br>The result of the data block transmission is: "01.00.00.00" |

## 12.10 Monitoring

### 12.10.1 Integrated error detection

If a node detects an error, it rejects the telegram bits received so far and transmits an error flag. The error flag consists of 6 consecutive bits with the same logic value.

The following errors are detected:

| Error | Description |
|---|---|
| **Bit error** | The sending node follows the transmission on the bus and interrupts the transmission if it receives a different logic value than the value transmitted. With the next bit, the sending node starts the transmission of an error flag.<br>In the arbitration phase, the transmitter only detects a bit error if a dominantly sent bit is received as recessive bit. In the ACK slot as well, the dominant overwriting of a recessive bit is not indicated as a bit error. |
| **Stuff-bit error** | If more than 5 consecutive bits have the same logic value before the ACK delimiter in the telegram, the previously transmitted telegram will be rejected and an error flag will be sent with the next bit. |
| **CRC error** | If the received CRC checksum does not correspond to the checksum calculated in the bus controller, the bus controller will send an error flag after the ACK delimiter and the previously transmitted telegram will be annulled. |
| **Acknowledgement error** | If the sent ACK slot recessively sent by the transmitting node is not dominantly overwritten by a receiver, the transmitting node will cancel the transmission. The transmitting node will annul the transmitted telegram and will send an error flags with the next bit. |
| **Format error** | If a dominant bit is detected in the CRC delimiter, in the ACK delimiter or in the first 6 bits of the EOF field, the received telegram will be rejected and an error flag will be sent with the next bit. |

**Tip!**

The errors mentioned before indicate that a physical error has occurred in the bus system.

Possible causes are:

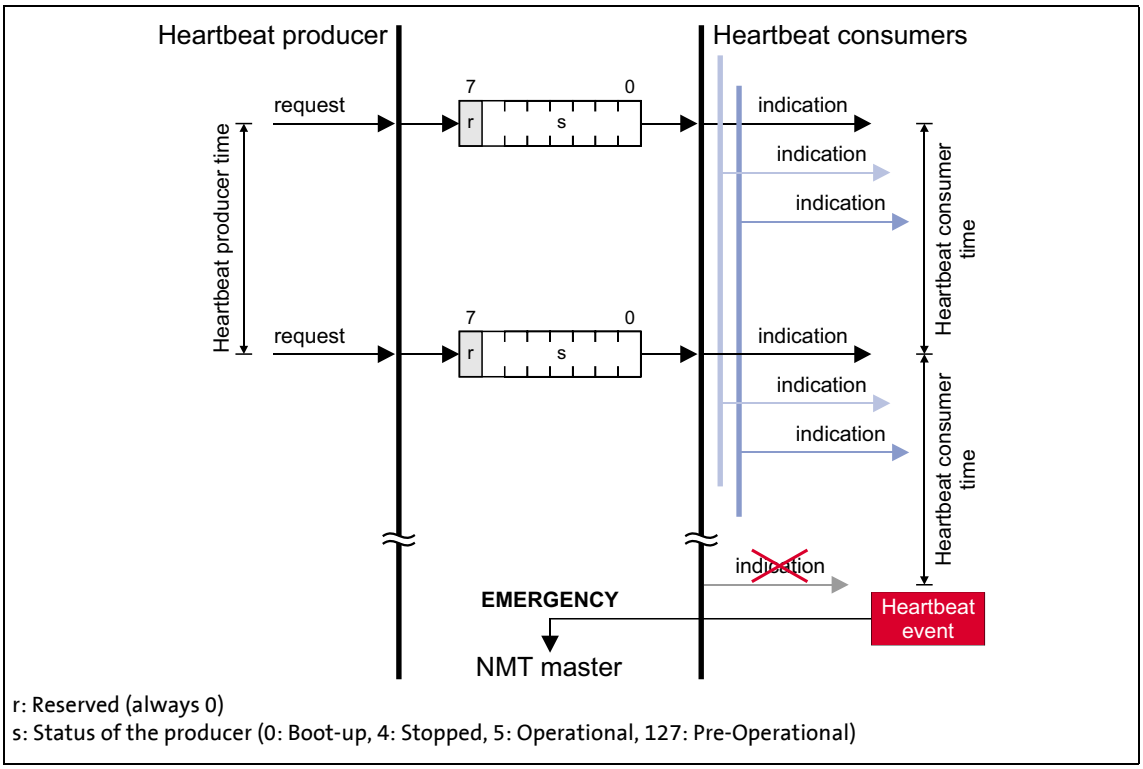
- Several nodes with identical node address
- Wrong baud rate of one or several nodes
- Too high cable length
- Too many or no terminating resistors
- Too high bus load/too many data telegrams
  (e.g. since a node permanently transmits event-controlled due to data changes of an analog signal/actual value.)
- EMC interferences on the system bus
  (e.g. since the CAN bus cable next to the motor cable is unshielded.)

C00364 displays whether such an error is active.

### 12.10.2 Heartbeat protocol

The heartbeat protocol can be used for node monitoring purposes within a CAN network.

**Basic workflow**




[12-9] Heartbeat protocol

1. A heartbeat producer cyclically transmits a so-called heartbeat telegram to one or more consumers.
2. The consumer(s) monitor the heartbeat telegram for arrival on a regular basis.

#### 12.10.2.1 Telegram structure

- The heartbeat telegram of the producer has the following identifier:
  Identifier (COB-ID) = 1792 + producer's node address
- The user data (1 byte) contain the status (s) of the producer:

| Heartbeat producer status | | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Communication status | Decimal value (s) | (r) | Producer status (s) | | | | | | |
| | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Boot-up | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Stopped | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Operational | 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Pre-Operational | 127 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

### 12.10.2.2 Parameter setting

**Short overview of the parameters for the "Heartbeat" monitoring function:**

| Parameters | Info | Lenze setting | | Assignment | |
|---|---|---|---|---|---|
| | | Value | Unit | Consumer | Producer |
| C00347/1...n | CAN status of the heartbeat producer 1 ... n | - | | ● | |
| C00381 | Heartbeat producer time | 0 | ms | | ● |
| C00385/1...n | CAN node address of the heartbeat producer 1 ... n | 0 | | ● | |
| C00386/1...n | Heartbeat consumer time for the heartbeat producer 1 ... n | 0 | ms | ● | |
| C00592/5 | Resp. to heartbeat event | No response | | ● | |
| Greyed out = display parameter | | | | | |

**Heartbeat producer time**

Time interval for the transmission of the heartbeat telegram to the consumer(s).

- Parameterisable in C00381 or via object I-1017. The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is sent automatically as soon as a time > 0 ms is set.

**Heartbeat consumer time**

Monitoring time for the nodes (producers) to be monitored.

- Parameterisable in C00386/1...n or via object I-1016.
- The parameterised time is rounded down to an integer multiple of 5 ms and must have a greater value than the heartbeat producer time of the node to be monitored.
- The maximum number of the nodes to be monitored depends on the device version:
  - "BaseLine C": 1 Heartbeat Producer can be monitored.
  - "StateLine C": Up to 7 Heartbeat Producers can be monitored.
  - "HighLine C": Up to 15 Heartbeat Producers can be monitored.
  - "TopLine C": Up to 15 Heartbeat Producers can be monitored.
- The node address(es) of the nodes to be monitored is/are set in C00385/1...n or via object I-1016, too.

**Heartbeat event**

The "Heartbeat event" is activated in the consumer if it does not receive any heartbeat telegram from the producer within the heartbeat consumer time:

- The consumer changes from the "Operational" communication status to the "Pre-Operational" communication status.
- The NMT master receives an emergency telegram containing emergency error code 0x8130.
- The response parameterised in C00592/5 is activated (Lenze setting: "No response").

**Note!**

The heartbeat monitoring will not start until the first heartbeat telegram of a monitored producer has been received successfully and the "Pre-Operational" NMT status has been assumed.

The boot-up telegram counts as the first heartbeat telegram.

### 12.10.2.3 Commissioning example

**Task**

An 8400 inverter (node 2) which is configured as heartbeat consumer is to monitor another 8400 inverter (heartbeat producer, node 1).

- The heartbeat producer is to transmit a heartbeat telegram to the heartbeat consumer every 10 ms.
- The heartbeat consumer monitors the heartbeat telegram for arrival. A response is to be activated in the event of an error.

**Parameterising the heartbeat producer (node 1)**

1. Set the heartbeat producer time (C00381) to 10 ms.

**Parameterising the heartbeat consumer (node 2)**

1. Set the CAN node address of the producer in C00385/1.
2. Set the heartbeat consumer time in C00386/1.
   - Note: The heartbeat consumer time must be greater than the heartbeat producer time of the node to be monitored set in C00381.
3. Set the desired response in C00592/5 which is to be activated should a heartbeat event in the consumer occur.

**Tip!**

C00347/1...n displays the heartbeat status of the monitored nodes.

**Heartbeat telegram**

- The heartbeat telegram of the producer has the following identifier:
  Identifier (COB-ID) = 1792 + producer's node address = 1792 + 1 = 1793 = 0x701

### 12.10.3 Emergency telegram

If the error status changes because an internal device error occurs or has been eliminated, the NMT master receives an emergency telegram once with the following structure:

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Emergency error codes | | Error register | Manufacturer-specific error message | | | | |
| LOW byte | HIGH byte | I-1001 | 0x00 (Reserved) | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |
| See table below | | | • For emergency error code 0xF000: Lenze error number (value displayed in C00168)<br>• All other emergency error codes have a value of "0" here. | | | | |

| Emergency error codes | Error register | Cause |
|---|---|---|
| 0x0000 | 0xXX | One of several errors eliminated |
| | 0x00 | One error has been eliminated (error-free status afterwards) |
| 0x3100 | 0x01 | Supply voltage of standard device faulty or failed |
| 0x8100 | 0x11 | Communication error (warning) |
| 0x8130 | 0x11 | Life guarding error or heartbeat error |
| 0x8150 | 0x11 | Collision of identifiers (COB-IDs): An identifier parameterised for reception is also used for transmission. |
| 0x8210 | 0x11 | PDO length shorter than expected |
| 0x8220 | 0x11 | PDO length greater than expected |
| 0x8700 | 0x11 | Monitoring of the sync telegram |
| 0xF000 | 0x01 | Generic error<br>• An error with a "Fault", "Trouble", "TroubleQSP", "Warning", or "SystemFault" error response occurred in the standard device.<br>• Error message is the Lenze error number (C00168). |

The Short overview (A-Z) of error messages of the operating system includes a list of more emergency error codes. (📖 721)

#### Example

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Emergency error codes | | Error register | Manufacturer-specific error message | | | | |
| 0x00 | 0xF0 | 0x01 | 0x00 (Reserved) | Lenze error number | | | |
| Generic error | | | | ▸ Error messages of the operating system<br>Corresponding error-free message: Value "0x00000000" | | | |

**Tip!**

A detalied description can be found in CAN specification DS301, V4.02.

## 12.11 Implemented CANopen objects

Lenze devices can both be parameterised with Lenze codes and manufacturer-independent "CANopen objects". A completely CANopen-compliant communication can only be achieved by using CANopen objects for parameter setting. The CANopen objects described in this chapter are defined in the CAN specification DS301 V4.02.

Many CANopen objects can be mapped on Lenze codes. In the following table, the corresponding Lenze codes are listed in the column "Relationship to Lenze codes".

**Note!**

Some of the terms used here derive from the CANopen protocol.

### Overview of CANopen indices and their relationship to Lenze codes

| CANopen object | | | Relationship to Lenze code |
|---|---|---|---|
| Index | Subindex | Name | |
| I-1000 | 0 | Device type | - |
| I-1001 | 0 | Error register | - |
| I-1003 | Predefined error field | | |
| | 0 | Number of errors | - |
| | 1 ... 10 | Standard error field | - |
| I-1005 | 0 | COB-ID SYNC message | C00367 |
| | | | C00368 |
| I-1006 | 0 | Communication cycle period | C00369 |
| I-1014 | 0 | COB-ID EMCY | - |
| I-1016 | Consumer heartbeat time | | |
| | 0 | Highest subindex supported | - |
| | 1 ... n | Consumer heartbeat time<br>• "BaseLine C" version: n = 1<br>• "StateLine C" version: n = 7<br>• "HighLine C" version: n = 15<br>• "TopLine C" version: n = 15 | C00385/1...n<br>C00386/1...n |
| I-1017 | 0 | Producer heartbeat time | C00381 |
| I-1018 | Identity object | | |
| | 0 | Highest subindex supported | - |
| | 1 | Vendor ID | - |
| | 2 | Product code | - |
| | 3 | Revision number | - |
| | 4 | Serial number | - |
| I-1200 | SDO1 server parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID client –> server (rx) | - |
| | 2 | COB-ID server –> client (tx) | - |
| I-1201 | SDO2 server parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID client –> server (rx) | - |
| | 2 | COB-ID server –> client (tx) | - |

| CANopen object | | | Relationship to Lenze code |
|---|---|---|---|
| Index | Subindex | Name | |
| I-1400 | RPDO1 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by RPDO | C00355/1 |
| | 2 | Transmission type | C00323/1 |
| I-1401 | RPDO2 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by RPDO | C00355/3 |
| | 2 | Transmission type | C00323/2 |
| I-1402 | RPDO3 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by RPDO | C00355/5 |
| | 2 | Transmission type | C00323/3 |
| I-1403 | RPDO4 communication parameter (from version 15.00.00) | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by RPDO | C00355/7 |
| | 2 | Transmission type | C00323/4 |
| I-1600 | RPDO1 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00409/1...4<br>C00866/1...4 |
| I-1601 | RPDO2 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00409/5...8<br>C00866/5...8 |
| I-1602 | RPDO3 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00409/9...12<br>C00866/9...12 |
| I-1603 | RPDO4 mapping parameter (from version 15.00.00) | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00409/13...16<br>C00866/13...16 |
| I-1800 | TPDO1 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by TPDO | C00355/2 |
| | 2 | Transmission type | C00322/1 |
| | 3 | Inhibit time | C00324/2 |
| | 5 | Event timer | C00356/5<br>C00369 |

| CANopen object | | | Relationship to Lenze code |
|---|---|---|---|
| Index | Subindex | Name | |
| I-1801 | TPDO2 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by TPDO | C00355/4 |
| | 2 | Transmission type | C00322/2 |
| | 3 | Inhibit time | C00324/3 |
| | 5 | Event timer | C00356/2<br>C00369 |
| I-1802 | TPDO3 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by TPDO | C00355/6 |
| | 2 | Transmission type | C00322/3 |
| | 3 | Inhibit time | C00324/4 |
| | 5 | Event timer | C00356/3<br>C00369 |
| I-1803 | TPDO4 communication parameter (from version 15.00.00) | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by TPDO | C00355/8 |
| | 2 | Transmission type | C00322/4 |
| | 3 | Inhibit time | C00324/5 |
| | 5 | Event timer | C00356/4<br>C00369 |
| I-1A00 | TPDO1 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00868/1...4 |
| I-1A01 | TPDO2 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00868/5...8 |
| I-1A02 | TPDO3 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00868/9...12 |
| I-1A03 | TPDO4 mapping parameter (from version 15.00.00) | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00868/13...16 |

### I-1000

| Index<br>**I-1000** | Name:<br>**Device type** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Display range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Device type | 0 | 0 | | 4294967295 | ro | U32 |

The CANopen index I-1000 specifies the profile for this device. Furthermore, additional information defined in the device profile itself can be stored here.

| 8th byte | 7th byte | 6. byte | 5th byte |
|---|---|---|---|
| **Data 4** | **Data 3** | **Data 2** | **Data 1** |
| HIGH word | | LOW word | |
| HIGH byte | LOW byte | HIGH byte | LOW byte |
| **Additional information** | | **ECAT: Device Profile Number** | |

[12-1] Data telegram assignment

In case of 8400 series inverters, the four bytes contain the following values:

- 5th and 6th byte: The data content is 0x0000, i.e. no profile definition.
- 7th byte: The data content specifies the device type: Here the value is 0x00 for inverters.
- 8th byte: The data content is 0x00.

The data content for the 8400 inverter thus is: 00 00 00 00

### I-1001

| Index:<br>**I-1001** | Name:<br>**Error register** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Display range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Error register | - | 0 | | 255 | ro | U8 |

Error register

The error status in the data byte (U8) is bit coded. The following error states are coded in the data byte (U8):

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Error status |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | No error |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Device error message |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Communication error |

**I-1003**

| Index: **I-1003** | Name: **Predefined error field** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of errors | 0 | 0 | | 255 | rw | U8 |
| 1 ... 10: Standard error field | - | 0 | | 4294967295 | ro | U32 |

Error history

This object indicates that an error has occurred in the module and in the standard device.

| Subindex | Meaning |
|---|---|
| 0 | Number of saved error messages |
| 1 ... 10 | Display of the error list<br>The error messages (U32) consist of a 16-bit error code and a manufacturer-specific information field comprising 16 bits. |

**Note!**

The values in the "standard error field" under subindex 1 ... 10 will be deleted if the subindex "number of recorded errors" is overwritten with the value "0".

| Emergency error codes | Cause | Entry in the error register (I-1001) |
|---|---|---|
| 0x0000 | One of several errors eliminated | 0xXX |
| | Elimination of one single error (afterwards no more errors) | 0x00 |
| 0x1000 | Standard device is in error status (error response "fault", "message", "warning", "error", "quick stop by trouble", or "system error") | 0x01 |
| 0x3100 | Supply voltage of standard device faulty or failed | 0x01 |
| 0x8100 | Communication error (warning) | 0x11 |
| 0x8130 | Life guard error or heartbeat error | 0x11 |
| 0x8150 | Collision of COB-IDs: An ID parameterised for reception is also used for transmission. | 0x11 |
| 0x8210 | PDO length shorter than expected | 0x11 |
| 0x8220 | PDO length greater than expected | 0x11 |
| 0x8700 | Monitoring of the sync telegram | 0x11 |

**I-1005**

| Index: **I-1005** | Name: **COB-ID SYNC message** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: COB-ID SYNC message | 0x0000 0080<br>or<br>0x8000 0080 | 0 | | 4294967295 | rw | U32 |

This object can be used to activate the generation of sync telegrams and to write the identifier value.

- This object relates to codes C00367 and C00368.

### Creating sync telegrams

Sync telegrams are created by setting bit 30 (see below) to "1". The time between the sync telegrams can be set using the object I-1006.

### Writing identifiers

To receive PDOs, the value 0x80 must be entered in the 11-bit identifier in the Lenze setting (and according to CANopen specification) . This means that all modules are by default set to the same sync telegram.

- If sync telegrams are only to be received by certain communication modules, their identifiers can be entered with values up to and including 0x07FF.
- The identifier can only be changed if the communication module does not send any sync telegrams (bit 30 = "0").
- How to change the identifier:
  - Deactivate identifier (set bit 30 to "0").
  - Change identifier.
  - Activate identifier (set bit 30 to "1").

| 8th byte | | 7th byte | 6. byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **x** | **0/1** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[12-2] Data telegram assignment

### I-1006

| Index: I-1006 | Name: Communication cycle period | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Communication cycle period | 0 µs | 0 | µs | 65535000 | rw | U32 |

Setting the sync telegram cycle time.

- The cycle time can be selected as "1000" or as an integer multiple of it.
- If "0 µs" is set (Lenze setting), no sync telegrams are created.
- This object relates to code C00369.

### I-1014

| Index: I-1014 | Name: COB-ID EMCY | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: COB-ID EMCY | 0x80 + node ID | 0 | | 4294967295 | rw | U32 |

When communication errors occur and are acknowledged or when internal errors occur in the communication module or inverter (e.g. "fault"), the system bus sends an error message. The telegram is sent once for every error. This function can be activated or deactivated with bit 31.

| 8th byte | | 7th byte | 6. byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **0/1** | **0** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[12-3] Data telegram assignment

| Bit | Setting | |
|---|---|---|
| Bit 31 | 0 | Emergency object is valid. |
| | 1 | Emergency object is invalid. |

**Note!**

The identifier can only be changed in the "emergency object invalid" status (bit 31 = 1).

**I-1016**

| Index: **I-1016** | Name: **Consumer heartbeat time** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | BaseLine C: 1<br>StateLine C: 7<br>HighLine C: 15<br>TopLine C: 15 | - (read access only) | | | ro | U16 |
| 1 ... n: Consumer heartbeat time | 0 | 0 | | 65535 | rw | U16 |

Monitoring time for the nodes to be monitored via heartbeat. ▸ Heartbeat protocol (📖 821)

- The parameterised time is rounded down to an integer multiple of 5 ms and must have a greater value than the heartbeat producer time of the node to be monitored.

| Subindex | Meaning | Lenze code |
|---|---|---|
| 0 | Number of nodes to be monitored | |
| 1 ... n | Node ID and heartbeat time of the node to be monitored | Node ID: C00385/x<br>Heartbeat time: C00386/x |

| 8th byte | 7th byte | 6. byte | 5th byte |
|---|---|---|---|
| **Data 4** | **Data 3** | **Data 2** | **Data 1** |
| Bit 31 ... bit 24 | Bit 23 ... Bit 16 | Bit 15 ... Bit 0 | |
| **0**<br>(Reserved) | **Node ID** | **Heartbeat time**<br>in [ms] | |

[12-4] Data telegram assignment

**I-1017**

| Index: **I-1017** | Name: **Producer heartbeat time** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Producer heartbeat time | 0 | 0 | ms | 65535 | rw | U16 |

Time interval for sending the heartbeat telegram to the consumer(s). ▸ Heartbeat protocol (📖 821)

- The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is automatically sent as soon as a time > 0 ms is entered. In this case, the "node guarding" monitoring function is deactivated.
- This object relates to code C00381.

**I-1018**

| Index: **I-1018** | Name: **Identity object** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Display range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | see below | 0 | | 4294967295 | ro | U32 |
| 1: Vendor ID | | | | | | |
| 2: Product code | | | | | | |
| 3: Revision number | | | | | | |
| 4: Serial number | | | | | | |

| Subindex | Meaning | |
|---|---|---|
| 1 | Manufacturer's identification number<br>• The identification number allocated to Lenze by the organisation "CAN in Automation e. V." is "0x0000003B". | |
| 2 | Product code | |
| | 84001 | 8400 BaseLine C |
| | 84002 | 8400 StateLine C |
| | 84003 | 8400 HighLine C |
| | 84004 | 8400 TopLine C |
| 3 | Main and subversion of firmware | |
| 4 | Serial number | |

**I-1200**

| Index: **I-1200** | Name: **SDO1 server parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Display range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 2 | 2 | | 2 | ro | U8 |
| 1: COB-ID client -> server (rx) | node ID + 0x600 | 0 | | 4294967295 | ro | U32 |
| 2: COB-ID server -> client (tx) | node ID + 0x580 | 0 | | 4294967295 | ro | U32 |

Identifiers for SDO server channel 1 (basic SDO channel).

- According to DS301 V4.02, the basic SDO channel can neither be changed nor deactivated.

| Subindex | Meaning |
|---|---|
| 1 | Specification of receive identifier<br>• For SDO server channel 1: node address (C00350) + 0x600 |
| 2 | Specification of send identifier<br>• For SDO server channel 1: node address (C00350) + 0x580 |

| 8th byte | | 7th byte | 6. byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **0** | **0** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[12-5] Data telegram assignment

**I-1201**

| Index: **I-1201** | Name: **SDO2 server parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 3 | - (read access only) | | | ro | U8 |
| 1: COB-ID client -> server (rx) | 0x80000000 | 0 | | 4294967295 | rw | U32 |
| 2: COB-ID server -> client (tx) | 0x80000000 | 0 | | 4294967295 | rw | U32 |

Identifiers for SDO server channel 2.

- The SDO server parameter is only valid, if bit 31 is set to "0" for both transmission directions (subindex 1 and 2).
- In the Lenze setting, the SDO server channels 2 are deactivated (bit 31 = "1").
- The identifier can only be changed if the SDO is invalid (bit 31 = "1").

| Subindex | Meaning |
|---|---|
| 1 | Specification of receive identifier |
| 2 | Specification of send identifier |

| 8th byte | | 7th byte | 6. byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **0/1** | **0** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[12-6] Data telegram assignment

| Bit | Setting | |
|---|---|---|
| Bit 31 | 0 | SDO is valid. |
| | 1 | SDO is invalid. |

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**Example**

Parameter data channel 2 of the inverter with node address 4 shall be activated.

- For this, bit 31 must be set to "0" (≡ "SDO is valid") in subindices 1 and 2 of the object I-1201.
- The master must send the two "write request" commands to the nodes via the basic SDO channel.

### Identifier calculation

- Identifier (COB-ID) = basic identifier + node address (node ID)
- Basic identifier SDO2 from master to drive: 1600 (0x640)
  ➞ Identifier = 0x640 + 0x4 = 0x644
- Basic identifier SDO2 from drive to master: 1472 (0x5C0)
  ➞ Identifier = 0x5C0 + 0x4 = 0x5C4

### Resulting data (data 1 ... data 4)

| 8th byte | | 7th byte | 6. byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| 0 | 0 | Extended identifier = 0 | | 11-bit identifier = 0x644 |
| 0x00 | | 0x00 | 0x06 | 0x44 |

[12-7] Data telegram assignment for subindex 1

| 8th byte | | 7th byte | 6. byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| 0 | 0 | Extended identifier = 0 | | 11-bit identifier = 0x5C4 |
| 0x00 | | 0x00 | 0x05 | 0xC4 |

[12-8] Data telegram assignment for subindex 2

### User data assignment

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x23 | 0x01 | 0x12 | 0x01 | 0x44 | 0x06 | 0x00 | 0x00 |

[12-9] User data assignment for writing to subindex 1

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6. byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x23 | 0x01 | 0x12 | 0x02 | 0xC4 | 0x05 | 0x00 | 0x00 |

[12-10] User data assignment for writing to subindex 2

**I-1400**

| Index: **I-1400** | Name: **RPDO1 communication parameter** | | | | |
|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by RPDO | 0x200 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | - | - (not used for RPDOs) | | | rw | U16 |
| 4: Compatibility entry | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | - | - (not used for RPDOs) | | | rw | U16 |

Communication parameter for receiving process data via RPDO1

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | RPDO1 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x200 + node ID | C00354/1 |
| 2 | RPDO Transmission type according to DS301 V4.02<br>▸ Transmission type (📖 803) | C00323/1 |

| 8th byte | | 7th byte | 6. byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **0/1** | **0/1** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[12-11] Data telegram assignment

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

Description of subindex 1

| Bit no. | Value | Explanation |
|---|---|---|
| 0 ... 10 | 0/1 | 11-bit identifier |
| (11 ... 28)* | 0 | *) The extended identifier (29 bits) is not supported. Any of these bits must be "0". |
| 29* | 0 | |
| 30 | 0 | RTR to this PDO possible (cannot be set) |
| | 1 | RTR to this PDO not possible (Lenze) |
| 31 | 0 | PDO active |
| | 1 | PDO not active |

[12-12] I-1400 ... I-1402, subindex 1

Description of subindex 2

| PDO transmission | | | Transmission type | Explanation |
|---|---|---|---|---|
| cyclic | synchronous | event-controlled | | |
| X | X | | n = 1 ... 240 | When a value n is entered, this PDO will be accepted with every nth SYNC. |
| | | X | n = 254 | PDO will be accepted immediately. |

[12-13] I-1400 ... I-1402, subindex 2

**I-1401**

| Index: **I-1401** | Name: **RPDO2 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by RPDO | 0x300 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | - | - (not used for RPDOs) | | | rw | U16 |
| 4: Compatibility entry | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | - | - (not used for RPDOs) | | | rw | U16 |

Communication parameter for receiving process data via RPDO2

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | RPDO2 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x300 + node ID | C00354/3 |
| 2 | RPDO Transmission type according to DS301 V4.02<br>▸ Transmission type (📖 803) | C00323/2 |

- For data telegram assignment and description of subindices 1 and 2, see object I-1400.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**I-1402**

| Index: **I-1402** | Name: **RPDO3 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by RPDO | 0x400 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | - | - (not used for RPDOs) | | | rw | U16 |
| 4: Compatibility entry | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | - | - (not used for RPDOs) | | | rw | U16 |

Communication parameter for receiving process data via RPDO3

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | RPDO3 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x400 + node ID | C00354/5 |
| 2 | RPDO transmission type according to DS301 V4.02<br>▸ Transmission type (📖 803) | C00323/3 |

- For data telegram assignment and description of subindices 1 and 2, see object I-1400.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**I-1403**

From version 15.00.00

| Index: **I-1403** | Name: **RPDO4 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by RPDO | 0x500 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | - | - (not used for RPDOs) | | | rw | U16 |
| 4: Compatibility entry | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | - | - (not used for RPDOs) | | | rw | U16 |

Communication parameter for receiving process data via RPDO4

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | Identifier RPDO4<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x500 + node ID | C00354/7 |
| 2 | RPDO transmission type according to DS301 V4.02<br>▸ Transmission type (📖 803) | C00323/4 |

- For data telegram assignment and description of subindices 1 and 2, see object I-1400.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**I-1600**

| Index: **I-1600** | Name: **RPDO1 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1600 serves to receive parameter data as RPDO1.

- This object relates to codes C00409/1...4 and C00866/1...4.
- From version 12.00.00 onwards: For 32-bit values, this object relates to codes C00407/1...2.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for RPDO1<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

| 8th byte | 7th byte | 6. byte | 5th byte |
|---|---|---|---|
| **Data 4** | **Data 3** | **Data 2** | **Data 1** |
| Bit 31 ... bit 16 | | Bit 15 ... bit 8 | Bit 7 ... bit 0 |
| **Index** | | **Subindex** | **Length** |

[12-14] Data telegram assignment

IEC 61131 process data words are mapped. Only whole bytes can be mapped (1-byte/mapping entry).

**Related topics:**

- RPDO1 | Port block "LP_CanIn1" (789)

### I-1601

| Index: **I-1601** | Name: **RPDO2 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1601 serves to receive parameter data as RPDO2.

- This object relates to codes C00409/5...8 and C00866/5...8.
- From version 12.00.00 onwards: For 32-bit values, this object relates to codes C00407/3...4.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for RPDO2<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

- For data telegram assignment, see object I-1600.

**Related topics:**

▸ RPDO2 | "LP_CanIn2" port block (📖 791)

### I-1602

| Index: **I-1602** | Name: **RPDO3 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1602 serves to receive parameter data as RPDO3.

- This object relates to codes C00409/9...12 and C00866/9...12.
- From version 12.00.00 onwards: For 32-bit values, this object relates to codes C00407/5...6.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for RPDO3<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

- For data telegram assignment, see object I-1600.

**Related topics:**

▸ RPDO3 | "LP_CanIn3" port block (📖 793)

**I-1603**

From version 15.00.00

| Index: **I-1603** | Name: **RPDO4 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1603 serves to receive parameter data as RPDO4.

- This object relates to codes C00409/13...16 and C00866/13...16.
- For 32-bit values, this object relates to the codes C00407/7...8.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for RPDO4<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

- For data telegram assignment, see object I-1600.

**Related topics:**

- RPDO4 | "LP_CanIn4" port block (📖 795)

**I-1800**

| Index: **I-1800** | Name: **TPDO1 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by TPDO | 0x180 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | 0 ms | 0 | 0.1 ms | 65535 | rw | U16 |
| 4: Reserved | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | 0 ms | 0 | ms | 65535 | rw | U16 |

Communication parameter for sending process data via TPDO1

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | TPDO1 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x180 + node ID | C00354/2 |
| 2 | TPDO transmission type according to DS301 V4.02<br>▸ Transmission type (📖 803) | C00322/1 |
| 3 | Minimum time between sending two identical TPDOs (see DS301 V4.02). | C00324/2 |
| 5 | Cycle time for PDO transmission with transmission type "254". | C00356/5<br>C00369 |

| 8th byte | | 7th byte | 6. byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **0/1** | **0/1** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[12-15] Data telegram assignment

| Bit | Setting | |
|---|---|---|
| Bit 30 | 0 | RTR to this PDO possible (Lenze). |
| | 1 | RTR to this PDO not possible (not adjustable) |
| Bit 31 | 0 | PDO active |
| | 1 | PDO inactive |

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**Subindex 2 - transmission type**

| PDO transmission | | | Transmission type | Explanation |
|---|---|---|---|---|
| cyclic | synchronous | event-controlled | | |
| ● | ● | | n = 1 ... 240 | When a value n is entered, this PDO will be accepted with every nth SYNC. |
| | ● | | n = 252 | On sync, the PDO is filled with new data, but only sent on RTR. |
| | | ● | n = 254, 255 | Event-controlled or cyclic |

**Subindex 3 - inhibit time**

**Note!**

The delay time can only be changed when the PDO is not active (see subindex 1, bit 31 = 1).

The entered value multiplied by 0.1 gives the delay time in [ms]. Only integers will be considered, i.e. fractional numbers will be **rounded down** to integers.

Example :

- Entered value: 26
- Calculated time = 26 x 0.1 [ms] = 2.6 [ms] → delay time = 2 [ms]

**Subindex 5 - event timer**

For cyclic operation (transmission type 254), the cycle time for sending the process data object on the system bus can be set under subindex 5:

The entered value corresponds to the time in [ms].

**I-1801**

| Index: **I-1801** | Name: **TPDO2 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by TPDO | 0x280 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | 0 ms | 0 | 0.1 ms | 65535 | rw | U16 |
| 4: Reserved | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | 0 ms | 0 | ms | 65535 | rw | U16 |

Communication parameter for sending process data via TPDO2

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | TPDO2 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x280 + node ID | C00354/4 |
| 2 | TPDO transmission type according to DS301 V4.02<br>▸ Transmission type (📖 803) | C00322/2 |
| 3 | Minimum time between sending two identical TPDOs (see DS301 V4.02). | C00324/3 |
| 5 | Cycle time for PDO transmission with transmission type "254". | C00356/2<br>C00369 |

- For data telegram assignment and description of subindices, see object I-1800.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**I-1802**

| Index: **I-1802** | Name: **TPDO3 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by TPDO | 0x380 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | 0 ms | 0 | 0.1 ms | 65535 | rw | U16 |
| 4: Reserved | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | 0 ms | 0 | ms | 65535 | rw | U16 |

Communication parameter for sending process data via TPDO3

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | TPDO3 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x380 + node ID | C00354/6 |
| 2 | TPDO transmission type according to DS301 V4.02<br>▸ Transmission type (📖 803) | C00322/3 |
| 3 | Minimum time between sending two identical TPDOs (see DS301 V4.02). | C00324/4 |
| 5 | Cycle time for PDO transmission with transmission type "254". | C00356/3<br>C00369 |

- For data telegram assignment and description of subindices, see object I-1800.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**I-1803**

From version 15.00.00

| Index: **I-1803** | Name: **TPDO4 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by TPDO | 0x480 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | 0 ms | 0 | 0.1 ms | 65535 | rw | U16 |
| 4: Reserved | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | 0 ms | 0 | ms | 65535 | rw | U16 |

Communication parameter for sending process data via TPDO4

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | Identifier TPDO4<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x480 + node ID | C00354/8 |
| 2 | TPDO transmission type according to DS301 V4.02<br>▸ Transmission type (📖 803) | C00322/4 |
| 3 | Minimum time between sending two identical TPDOs (see DS301 V4.02). | C00324/5 |
| 5 | Cycle time for PDO transmission with transmission type "254". | C00356/4<br>C00369 |

- For data telegram assignment and description of subindices, see object I-1800.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**I-1A00**

| Index: **I-1A00** | Name: **TPDO1 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1A00 serves to send parameter data as TPDO1.

- This object relates to code C00868/1...4.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for TPDO1<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

| 8th byte | 7th byte | 6. byte | 5th byte |
|---|---|---|---|
| **Data 4** | **Data 3** | **Data 2** | **Data 1** |
| Bit 31 ... bit 16 | | Bit 15 ... bit 8 | Bit 7 ... bit 0 |
| **Index** | | **Subindex** | **Length** |

[12-16] Data telegram assignment

IEC 61131 process data words are mapped. Only whole bytes can be mapped (1-byte/mapping entry).

**Related topics:**

- TPDO1 | "LP_CanOut1" port block (📖 797)

**I-1A01**

| Index: **I-1A01** | Name: **TPDO2 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1A01 serves to send parameter data as TPDO2.

- This object relates to code C00868/5...8.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for TPDO2<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

- For data telegram assignment, see object I-1A00.

**Related topics:**

- TPDO2 | "LP_CanOut2" port block (📖 798)

**I-1A02**

| Index: **I-1A02** | Name: **TPDO3 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1A02 serves to send parameter data as TPDO3.

- This object relates to code C00868/9...12.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for TPDO3<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

- For data telegram assignment, see object I-1A00.

**Related topics:**

▸ TPDO3 | "LP_CanOut3" port block (📖 799)

**I-1A03**

| Index: **I-1A03** | Name: **TPDO4 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1A03 serves to send parameter data as TPDO4.

- This object relates to code C00868/13...16.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for TPDO4<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

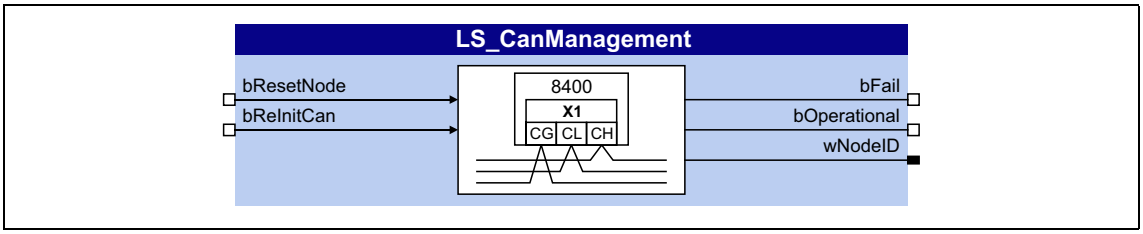
- For data telegram assignment, see object I-1A00.

**Related topics:**

▸ TPDO4 | "LP_CanOut4" port block (📖 800)

## 12.12 Internal interfaces | System block "LS_CANManagement"

The **LS_CANManagement** system block serves to control internal functions of the CAN driver (reset node and re-initialisation) and to display the "Operational" status as well as the node address (analogous to the 9300 ServoPLC and ECS devices).




### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bResetNode<br>BOOL | Reset node | |
| | TRUE | Carry out reset node<br>• If the inverter is configured as CAN master in C00352, the NMT command "Start Remote Node" is sent to all nodes at the bus (broadcast telegram). ▸ Network management telegram (NMT) |
| bReInitCAN<br>BOOL | Reinitialisation | |
| | TRUE | Reinitialise "CAN on board" interface. |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bFail<br>BOOL | Error | |
| | TRUE | An event according to the error configuration in C00341 has occurred |
| bOperational<br>BOOL | "Operational" status signal | |
| | TRUE | The system bus is in the "Operational" status |
| wNodeID<br>WORD | Output of the node address | |

**Note!**

If a "Bus off" error is detected, the "CAN on board" interface will automatically be reinitialised after 1 second.

Hence, 1 second after the "Bus off" has occurred, the inverter will automatically be active again on the system bus ("Auto bus off recovery").

# 13 Fieldbus interface (MCI)

The Inverter Drives 8400 can accommodate plug-in communication modules and can therefore take part in the data transfer of an existing fieldbus system.

When using a communication module, the major advantage for the user is the possibility of parameterising, controlling, and diagnosing the drive system via the available fieldbus.

The following fieldbuses are supported by the 8400 TopLine:

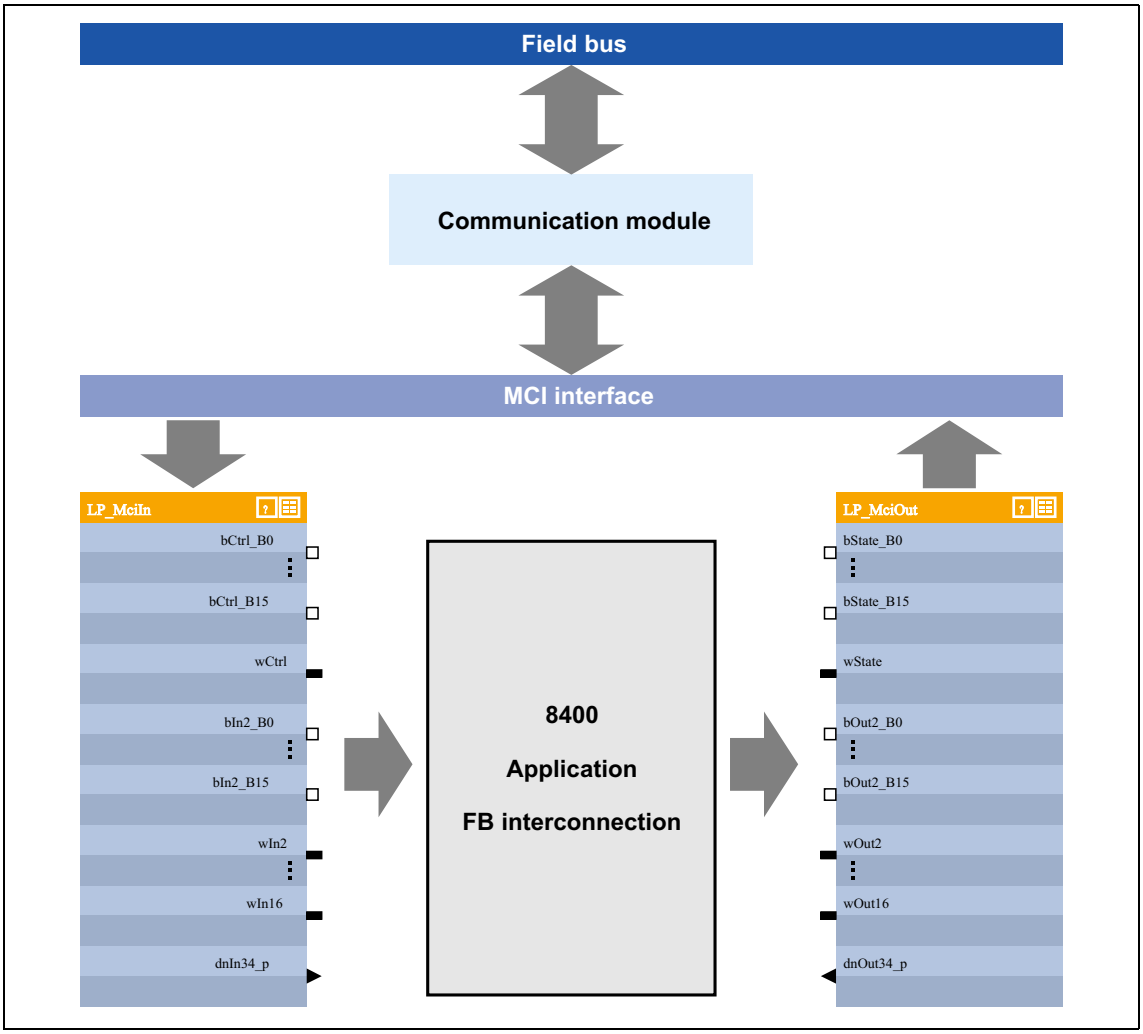
| Fieldbus | Communication module (type designation) |
|---|---|
| CANopen | System bus "CAN on board"(permanently integrated in the standard device) |
| EtherCAT | EtherCAT® (E84AYCET) |
| ETHERNET POWERLINK | Ethernet POWERLINK (E84AYCEC) |
| EtherNet/IP | EtherNet/IP™ (E84AYCEO) |
| INTERBUS | INTERBUS (E84AYCIB) |
| PROFIBUS | PROFIBUS® (E84AYCPM) |
| PROFINET | PROFINET® (E84AYCER) |

Detailed information is provided in the communication manual (KHB) for the respective fieldbus and in the »Engineer« online help.

## 13.1 Process data transfer

The process data serve to control the inverter. Thus the transfer of the process data is time-critical.

- The process cycle is 1 ms, irrespective of the respectively plugged-in bus system and the type of inverter.
- The process data transfer between the host system and the inverters is cyclical.
  - This concerns the continuous exchange of current input and output data.
  - In the case of the 8400 inverter, 16 words per direction are exchanged.
- The master computer can directly access the process data.
  Access to the process data takes place via the port blocks **LP_MciIn** and **LP_MciOut** (see FB interconnection of the »Engineer«).
  These port blocks are also called process data channels.
- The process data are not saved in the inverter.




[13-1] External and internal data transfer between bus system, inverter and function block interconnection

**Voltage supply**

Depending on the complexity and functional range of the fieldbus, the communication modules are supplied by the standard device or an external 24 supply at the module.

The external 24 V voltage supply of the communication module is required if the supply of the standard device fails but the communication via the bus is to continue.

**Parameter setting of the communication modules**

All codes which must be parameterised for establishing the fieldbus communication are saved in the memory module of the inverter.

The archived data can be addressed by all bus systems supported by the inverter.

**Hotplug**

The communication module (MCI module) can be plugged in/out while the inverter is switched on. When the module is plugged in, it is automatically detected and checked for plausibility regarding the function and version.

**Fieldbus-specific device profiles and PDO mapping**

When specific bus systems are used, the inverter is to behave according to a defined, manufacturer-spanning standard. The following definitions have been made for this:

- Definitions of the device state machine (e.g. DSP402, DriveCOM, ProfiDrive etc.)
- Definition of the bit assignment of control and status words
- Definition of signal scaling (on a limited scale)
- Definition of parameter scaling (on a limited scale)
- Definition of the process data mapping

These device profiles are not mapped in the communication module since some definitions have a strong effect on the device-internal behaviour and the device profiles are not uniform regarding this matter.

- The task of the communication modules is
  - to address parameters (SDOs),
  - to transfer PDOs and
  - the signal mapping of the PDOs.
- The process data objects (e.g. the meaning of the control word bits or the speed setpoint stipulated) are interpreted in the inverter.

## 13.2 Control mode "MCI"

"40: MCI" can be selected as a control mode in C00007 in order to quickly and easily set-up inverter control by means of MCI-PDOs via the fieldbus interface.

Given that the technology applications are fundamentally different and have different requirements regarding the signals sent to them, predefined assignment of the MCI-PDOs depends on the technology application selected in C00005:

- **TA "Actuating drive speed":**
  Process data assignment for fieldbus communication (📖 458)
- **TA "Table positioning":**
  Process data assignment for fieldbus communication (📖 521)
- **TA "Abschaltpositionierung":**
  Process data assignment for fieldbus communication (📖 547)

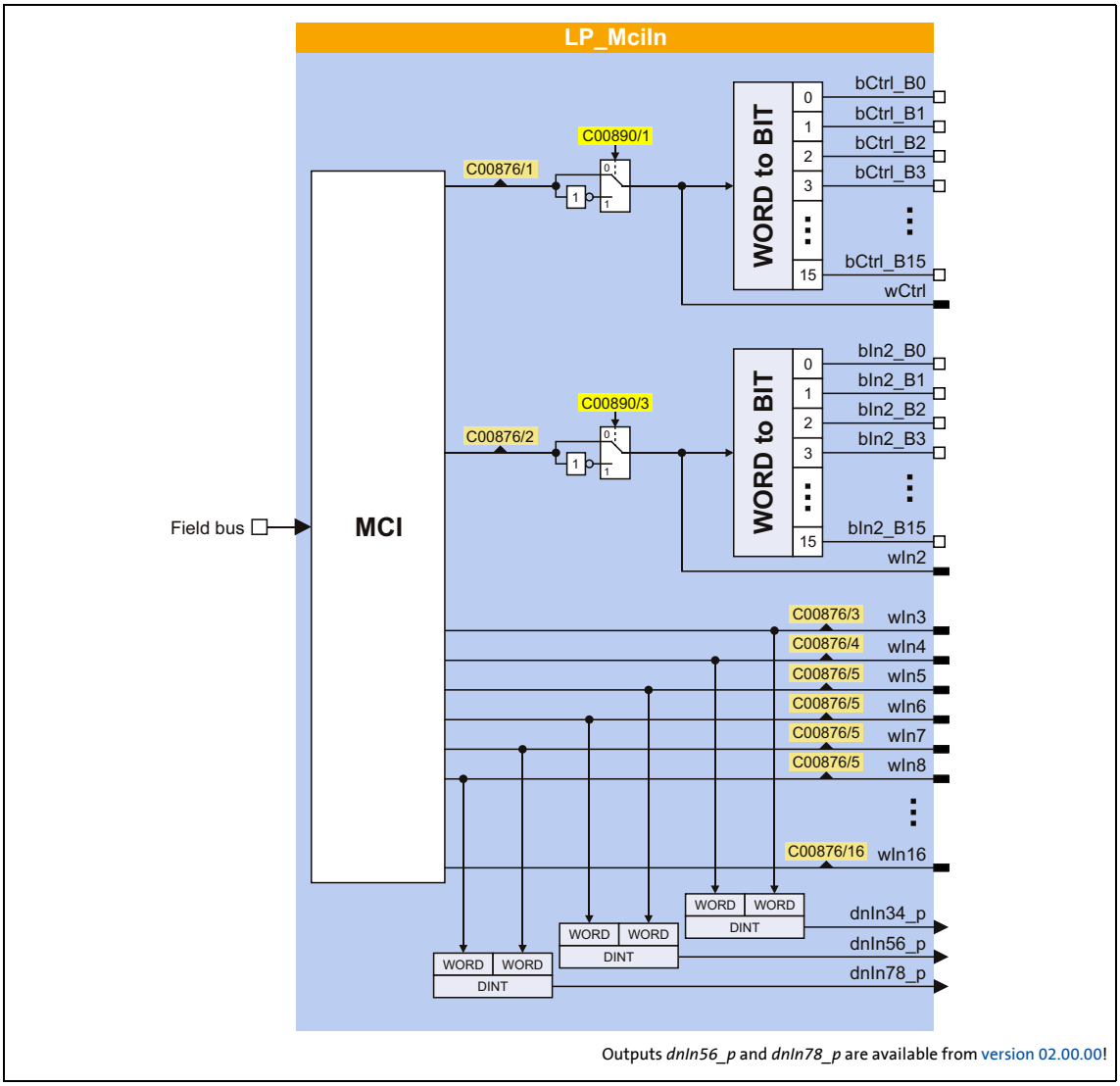


**Tip!**

The predefined assignment of the MCI-PDOs can be parameterised by means of PDO mapping and can be freely configured on the I/O level in the function block editor (FB editor).

### 13.2.1 Port block "LP_MciIn"

The **LP_MciIn** port block maps the received MCI-PDOs in the FB Editor.



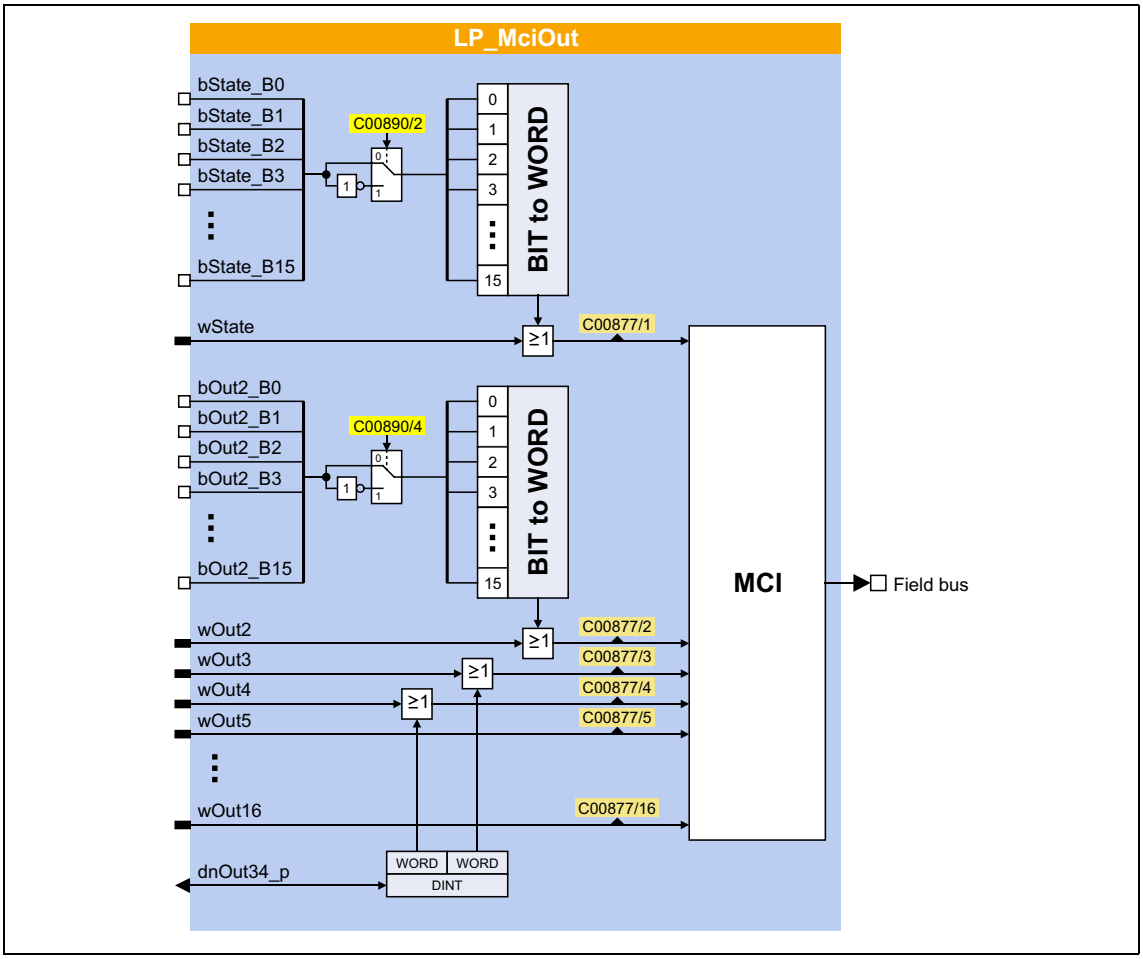

Outputs *dnIn56_p* and *dnIn78_p* are available from version 02.00.00!

**Short overview of the parameters for LP_MciIn:**

| Parameters | Info | Lenze setting |
|---|---|---|
| C00876/1 | LP_MciIn:wCtrl | - |
| C00876/2...16 | LP_MciIn: wIn2 ... wIn16 | - |
| C00890/1 | LP_MciIn: Inversion bCtrl_B0..15 | 0x0000 |
| C00890/3 | LP_MciIn: Inversion bIn2_B0..15 | 0x0000 |
| Greyed out = display parameter | | |

### 13.2.2 Port block "LP_MciOut"

The **LP_MciOut** port block maps the MCI-PDOs to be transmitted in the FB Editor.




**Short overview of the parameters for LP_MciOut:**

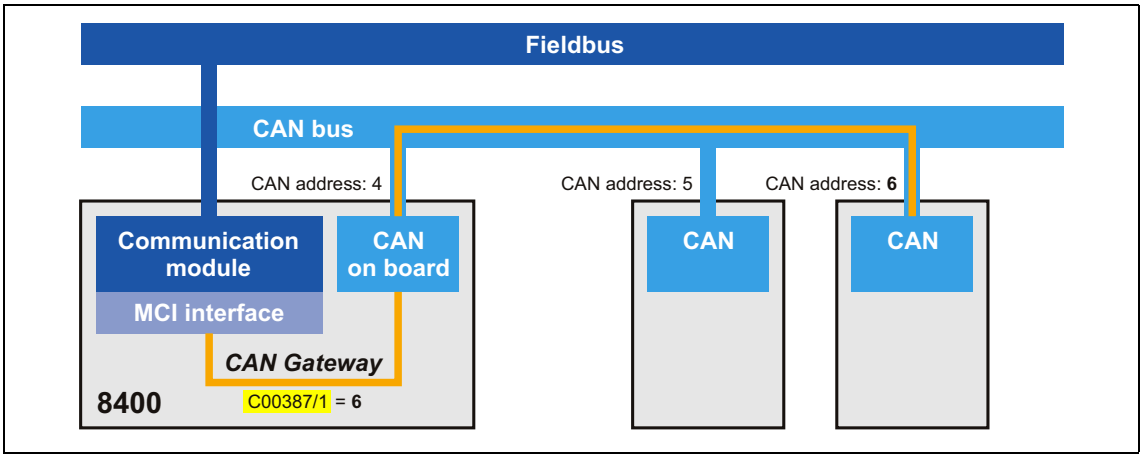
| Parameters | Info | Lenze setting |
|---|---|---|
| C00877/1 | LP_MciOut:wState | - |
| C00877/2...16 | LP_MciOut: wOut2 ... wOut16 | - |
| C00890/2 | LP_MciOut: Inversion bState_B0..15 | 0x0000 |
| C00890/4 | LP_MciOut: Inversion bOut2_B0..15 | 0x0000 |
| Greyed out = display parameter | | |

## 13.3 CAN gateway

This function extension is available from version 12.00.00!

The CAN gateway functionality forwards the read/write requests of the attached communication module to a connected *remote* device via the "CAN on board" system bus. It also receives the responses via this way.

- From the point of view of the attached communication module, the read/write requests are identical to accesses to its own standard device.
- Block services (reading/writing parameters of the "String" data type) can also be executes via the CAN gateway.




[13-2] Functional principle of "CAN gateway"

### Activating the CAN gateway

To activate the CAN gateway, the CAN address (1 ... 127) of the connected *remote* device must be set in C00387/1.

- In the Lenze setting C00387/1 = "0", the function is switched off.
- If the CAN gateway is activated, the own inverter is the client and the *remote* device is the server.
- If the own CAN address is set, the read/write requests are forwarded to the own inverter.
- The parameters of the communication module (C13000 ... C13999) are always processed in the own inverter.
- The parameters C00387/1 (CAN gateway address) and C00350 (CAN node address) are not transferred to the *remote* device.

# 14 Axis bus

This function extension is available from version 02.00.00!

The 8400 TopLine inverter has an integrated axis bus which is used to couple several 8400 TopLine inverters in an axis interconnection. The main task of the axis bus is a simple cross-data exchange from axis to axis.

Axis bus operation uses two transmission media, "data transfer axis bus" and "IO axis bus" that can be used as follows:

- The data transfer axis bus is based on CAN physics and enables a high-performance data transfer of master values and control signals to other 8400 TopLine inverters. For this purpose, a synchronisation of the internal time base of the inverters via the IO axis bus is always required.
- The IO axis bus is a 1-wire bus with open-collector circuitry (5 V isolated). The IO axis bus can either be used to transfer controller errors in the interconnection ("release cord" principle) or as a pure open-collector IO function. In the first case, the inverters' internal time base can be synchronised via the IO axis bus.

A maximum of 62 nodes can be connected to the axis bus.

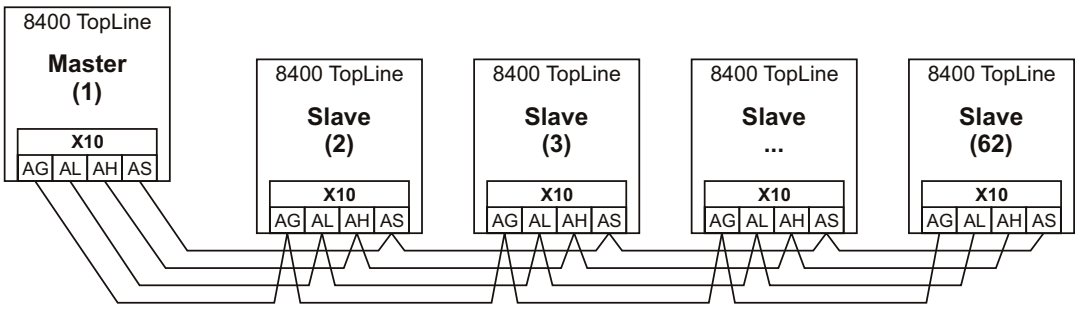



## Stop!

The data transfer axis bus of the 8400 TopLine inverter is designed especially for performance and simplicity. HMIs and other peripherals as well as the »Engineer« are not supported at the axis bus.

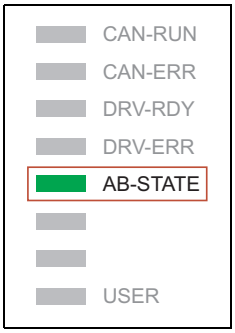
The IO axis bus of the 8400 TopLine inverter is not compatible to the state bus of the 9300/9400 device series due to different voltage levels!

**Axis bus topology**




| | |
|---|---|
| X10/AL, AH | Data transfer axis bus |
| X10/AS | IO axis bus |
| X10/AG | GND, reference potential of axis bus |

## 14.1 LED status displays of the axis bus




Information on the axis bus status can quickly be obtained via the "AB-STATE" LED display on the front of the inverter.

The meaning can be seen from the table below.

| LED display | Meaning |
|---|---|
| OFF | The inverter is disconnected from the axis bus as the node address "0" is set in C02430/1. |
| (blinking - 10 Hz) | The axis bus is in the "boot up" phase. |
| (blinking - 2 Hz) | The axis bus is in the "Stopped" state |
| (blinking - double impulse) | The axis bus is in the "Warning" state.<br>• The error counter for incorrectly sent or received telegrams has a value > 96.<br>• The data transfer continues to run and the error counter is tried to be reduced to 0 by error-free transfers.<br>▸ Axis bus data error monitoring |
| (permanently on) | The axis bus is in the "PDP active" state (normal operation). |

## 14.2 Data transfer axis bus

The data transfer axis bus is based on isolated CAN physics.

- Max. 62 nodes are supported at the bus (1 master and max. 61 slaves).
- Transfer clock = 1 ms
- Baud rate
  - Starting from version 12.00.00, the default setting of the baud rate is 800 kbps.
  - In version 2.00.00 the baud rate is 500 kbps.

**Note!**

The baud rate in an axis interconnection for example involving devices with version 02.00.00 must be consistently set to 500 kbps. The setting is also carried out with C2444/1, bit 15 = TRUE for devices with a higher version!

### 14.2.1 Topologies

There are generally two different topologies for an interconnected operation:

A. All nodes of the network are connected to the higher-level master control via MCI module and the respective fieldbus.
   - The axes can exchange data via the axis bus.
   - Moreover, the CAN bus is available.
   - For synchronous fieldbuses as e.g. EtherCAT, synchronicity of all axes towards the EtherCAT fieldbus is achieved by selecting "MCI" as sync source for the axis bus master. The synchronisation of the slaves on the EtherCAT side has to be switched off then.

B. Only the axis bus master of the network is connected to the higher-level master control via MCI module and the respective fieldbus.
   - For cost reasons, this topology only makes sense if only few information has to be transferred from the master control to the network.

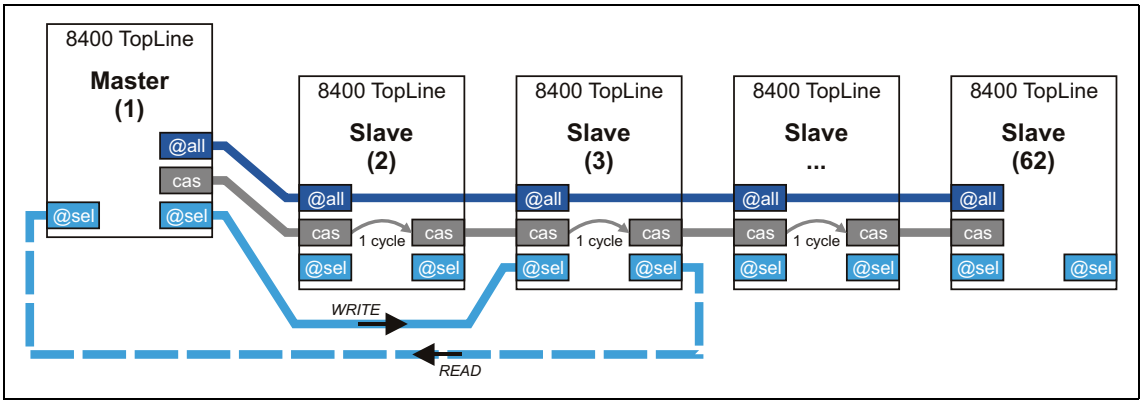
In both cases, the axes can exchange data via the axis bus and the CAN bus is available in addition.

### 14.2.2 Transfer mechanisms

The axis bus supports three different transfer mechanisms simultaneously according to the following methodology:

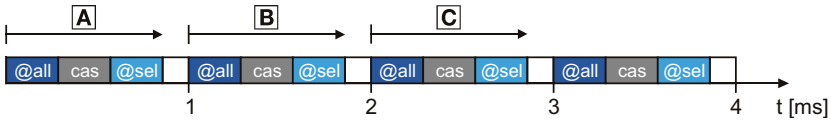
- "@all": Data transfer from master to all slaves
  - A data packet is sent from the master to all slaves in the transmission cycle.
  - Application: Data distribution according to line topology
  - Cycle time = transmission cycle = 1 ms (at baud rate = 500 kbits)
- "cas": Data transfer from node to node (cascade)
  - In the first transmission cycle, a data packet is sent from master to slave 1, in the next transmission cycle from slave 1 to slave 2, etc.
  - Application: Data distribution according to cascade topology
  - Cycle time = transmission cycle * no. of slaves+1

- "@sel": Data transfer from master to a slave
  - A data packet is sent from the master to one selected slave in the transmission cycle.
  - More options: Send to no slave, send to all slaves, and read one selected slave.
  - Application: Data distribution according to line topology for controlling one single node.
  - Cycle time = transmission cycle




[14-1] Transfer mechanisms

The axis bus data is transferred in 1-ms transmission cycle:




| Case 1: | Case 2: |
|---|---|
| A Master sends setpoints<br>B Slave works with new data | A Master requests slave to send<br>B Slave sends as requested<br>C Data is processed by master |

[14-2] Data transfer

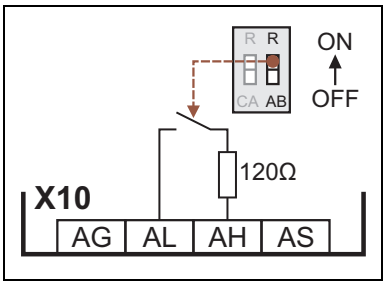
The axis bus data transfer is connected to the master/slave application via the **LS_AxisBus** systems blocks:

- For a data transfer from the master to all slaves (@all) and/or a cascaded data transfer (cas), the systems blocks LS_AxisBusIn and LS_AxisBusOut are available in the FB Editor.
- For a data transfer from the master to a certain slave (@sel), the LS_AxisBusAux system block is available.

### 14.2.3 Activating the bus terminating resistor

The axis bus must be terminated between axis bus low (AL) and axis bus high (AH) at the first and last physical node each by a resistor (120 Ω).

The 8400 inverter is provided with an integrated bus terminating resistor, which can be activated via the DIP switch labelled with "AB":




- OFF = bus terminating resistor is inactive
- ON = bus terminating resistor is active

[14-3] Activation of the integrated bus terminating resistor

### 14.2.4 Parameter setting

**Short overview of the relevant parameters:**

| Parameters | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C01120 | Sync signal source | 0: Off | |
| C02430/1 | Axis bus address<br>• 0 ≡ Nodes disconnected from the axis bus<br>• 1 ≡ Master<br>• 2 ... 62 ≡ slave 1 ... slave 61 | 0 | |
| C02430/2 | Axis bus no. of nodes<br>• Number of inverters connected to the axis bus. | 2 | |
| C02431/1 | Axis bus decel. boot-up | 3000 | ms |
| C02431/2 | Axis bus decel. BusOffRecovery | 1000 | ms |
| C02431/3 | Axis bus monitoring time | 1000 | ms |
| C02440/1 | Axis bus IO function | 0: Off | |
| ▸ Configuring exception handling of the received data | | | |
| C02442/1 | Decoupling inputs from the axis bus<br>• From version 12.00.00 | 0x0000 | |
| C02443/1...11 | Decoupling values<br>• From version 12.00.00 | 0 | |
| ▸ Axis bus data error monitoring | | | |
| C00591/1 | Resp. to axis bus data error<br>• From version 12.00.00 | 1: Fault | |
| **Diagnostic parameter** | | | |
| C02435 | Axis bus status<br>• Display whether the node participates in the axis bus data exchange (PDO active) or is only switched on passively to the axis bus (PDO stopped). | - | |
| C02436 | Axis bus error status | - | |
| C02437 | Axis bus MessageError<br>▸ Integrated error detection | - | |
| C02438/1 | Axis bus Tx_Error<br>• From version 12.00.00<br>• Error counter for faulty sent telegrams. | - | |
| C02438/2 | Axis bus Rx_Error<br>• From version 12.00.00<br>• Error counter for faulty received telegrams. | - | |
| C02438/3 | Axis bus transmitting meter<br>• From version 12.00.00<br>• Display of the entire number of sent telegrams | - | |
| C02438/4 | Axis bus receiving meter<br>• From version 12.00.00<br>• Display of the entire number of received telegrams. | - | |
| Greyed out = display parameter | | | |

### 14.2.4.1 Required settings at the master

For a "safe" data transfer, one inverter must be the master in the network. The master controls and synchronises the network.

Make the following two settings for the inverter which is to be the master in the network:

- Axis bus address (C02430/1) = "1"
  - With this setting, the inverter automatically takes over the control in the network.
  - Only one master is allowed in the network.
- Axis bus IO function (C02440/1) = "1: Master"
  - With this setting, the inverter outputs a synchronisation cycle to the I/O axis bus to which the slaves can orient themselves.

**Optional: Synchronisation of the internal time base of the master**

For the master, a sync signal source (e.g. CAN or MCI) can be selected in C01120.

- Then, the entire axis bus system behaves synchronously with this sync signal.
- The sync signal source "2: AxisBusIO" technically does not make any sense for the axis bus master and is thus ignored (same effect as with setting "0: Off").

**Related topics:**

▸ "Master/slave" function (📖 875)
▸ Synchronisation of the internal time base (📖 880)

### 14.2.4.2 Required settings for the slaves

Make the following settings for every slave inverter in the network:

- Axis bus address (C02430/1) = "2" ... "62"
  - Make sure that all inverters connected to the axis bus have different axis bus addresses.
- Sync signal source (C01120) = "2: AxisBusIO"
  - With this setting, the synchronisation cycle output by the master is used as synchronisation source.
  - Basically, only one source is allowed to synchronise the internal time base.
- Axis bus IO function (C02440/1) = "2: Slave"
  - This setting serves to define the slave.

**Note!**

The selection of the "2: AxisBusIO" sync signal source serves to permanently set the parameters given in the following table for synchronising the internal time base to reasonable values in order to ensure a technically perfect operation of the axis bus!

| Parameters | Info | Setting for "AxisBusIO" signal source | |
|---|---|---|---|
| | | Value | Unit |
| C01121 | Sync cycle time setpoint | 2000 | µs |
| C01122 | Sync phase position | 200 | µs |
| C01123 | Sync window | 40 | µs |

**Related topics:**

- "Master/slave" function (📖 875)

### 14.2.4.3 Boot-up phase

After the devices (master and slave) are switched on and initialised, first a boot-up phase is executed. In this phase, the master already communicates on the axis bus with the slaves which do not participate yet in the data exchange.

Only when all nodes have been synchronised at the axis bus (synchronisation of the slaves with the axis bus IO is compulsory) and the deceleration time set in C02431/1 has been expired, all nodes change to the "PDO active" bus state.

### 14.2.4.4 Disconnect nodes from the axis bus

The setting of the axis bus address (C02430/1) to "0" serves to switch off the corresponding node dynamically from the axis bus.

- The node does not participate in the data exchange anymore.
  - The axis bus LED "AB-STATE" of the node is switched off.
  - C02435 displays the axis bus status "4: Boot-up" for the node.
- The synchronisation and the detection and report of errors via the axis bus IO cable continues to take place.

### 14.2.4.5 Configuring exception handling of the received data

Certain situations require the axis to be decoupled from the received data of the axis bus, e.g. if a data error has occurred. Then it is not possible anymore, for instance, to position the slave axis to a certain target position.

Exception handling for the received data in the event of an error can be set via decoupling configuration and decoupling values in order to e.g. move the slaves to a parking position.

- Bit-coded selection is carried out in C02442 1 defining the events that will trigger decoupling of the received data.

| Bit | Event |
|---|---|
| Bit 0 ☐ | BusOff_MsgErr |
| Bit 1 ☐ | Warning |
| Bit 2 ☐ | NodeStopped<br>• For the master, the *bStop* input at the SB LS_AxisBusAux has been set to TRUE. |
| Bits 3 ... 13 ☐ | Reserved |
| Bit 14 ☐ | Trouble |
| Bit 15 ☐ | Fault |

- Finally, the following parameters define the value that the received data are to have when they are decoupled:

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| Decoupling values for received data at the SB LS_AxisBusIn: | | | |
| C02443/1 | LS_AxisBusIn: wLine1 DiscVal | 0 | |
| C02443/2 | LS_AxisBusIn: wLine2 DiscVal | 0 | |
| C02443/3 | LS_AxisBusIn: wLine3 DiscVal | 0 | |
| C02443/4 | LS_AxisBusIn: wCas1 DiscVal | 0 | |
| C02443/5 | LS_AxisBusIn: wCas2 DiscVal | 0 | |
| C02443/6 | LS_AxisBusIn: wCas3 DiscVal | 0 | |
| C02443/7 | LS_AxisBusIn: wCas4 DiscVal | 0 | |
| Decoupling value for received data at the SB LS_AxisBusAux: | | | |
| C02443/8 | LS_AxisBusAux: wAuxIn1 DiscVal | 0 | |
| C02443/9 | LS_AxisBusAux: wAuxIn2 DiscVal | 0 | |
| C02443/10 | LS_AxisBusAux: wAuxIn3 DiscVal | 0 | |
| C02443/11 | LS_AxisBusAux: wAuxIn4 DiscVal | 0 | |

### 14.2.4.6 Integrated error detection

Just like the system bus ("CAN on board"), the axis bus is also provided with an integrated error detection. If a node detects an error, it rejects the telegram bits received so far and transmits an error flag. The error flag consists of 6 consecutive bits with the same logic value.

The following errors are detected and displayed in C02437:

| Error | Description |
|---|---|
| **Bit error** | The sending node follows the transmission on the bus and interrupts the transmission if it receives a different logic value than the value transmitted. With the next bit, the sending node starts the transmission of an error flag.<br>In the arbitration phase, the transmitter only detects a bit error if a dominantly sent bit is received as recessive bit. In the ACK slot as well, the dominant overwriting of a recessive bit is not indicated as a bit error. |
| **Stuff-bit error** | If more than 5 consecutive bits have the same logic value before the ACK delimiter in the telegram, the previously transmitted telegram will be rejected and an error flag will be sent with the next bit. |
| **CRC error** | If the received CRC checksum does not correspond to the checksum calculated in the bus controller, the bus controller will send an error flag after the ACK delimiter and the previously transmitted telegram will be annulled. |
| **Acknowledgement error** | If the sent ACK slot recessively sent by the transmitting node is not dominantly overwritten by a receiver, the transmitting node will cancel the transmission. The transmitting node will annul the transmitted telegram and will send an error flags with the next bit. |
| **Format error** | If a dominant bit is detected in the CRC delimiter, in the ACK delimiter or in the first 6 bits of the EOF field, the received telegram will be rejected and an error flag will be sent with the next bit. |

**Tip!**

The errors mentioned before indicate that a physical error has occurred in the bus system.

Possible causes are:

- Several nodes with identical node address
- Too high cable length
- Too many or no terminating resistors
- Too high bus load/too many data telegrams
  (e.g. since a node permanently transmits event-controlled due to data changes of an analog signal/actual value.)
- EMC interferences on the axis bus
  (e.g. since the axis bus cable next to the motor cable is unshielded)

### 14.2.4.7 Axis bus data error monitoring

#### Operating mode

There are two error counters, one for incorrectly sent telegrams (C02438/1) and one for incorrectly received telegrams (C02438/2).

- In an error-free normal operation, both error counters are = 0.
- Depending on the error type, the counters are increased by 8 in case of an error detection, and reduced by 1 in case no error has occurred.
  - If one of the two error counters is > 96, a warning is signalled via the LED AB-STATE (double pulse).
  - If an error counter increases or remains unchanged, a watchdog counter is increased in a 1-ms cycle.
  - If an error counter is reduced, the watchdog counter remains unchanged.
  - If the error counters are = 0, the watchdog counter is reset to 0.
- If the watchdog counter reaches the monitoring time set in C02431/3:
  - The error response set in C00591/1 is activated (Lenze setting: "Fault").
  - The error message "Ab01: AchsbusTimeOut" is entered into the logbook.

#### Parameterisation notes for "electrical shaft" application

When an "electrical shaft" is operated via an axis bus or generally via a bus system, data transfer errors may occur due to e.g. EMC interferences. In order to prevent a mechanical offset of the drives in case of a short-time interference only, the L_Interpolator_1 function block should be used in the slave drives. This function block has an internal correction mechanism which provides for an automatic offset correction in case of a missing data telegram. The error monitoring can then be set to an insensitive value (C02431/3 = approx. 10 ... 30 ms) which results in a very robust transfer system.

#### What happens in case of a cable break, loose contact or short circuit of the axis bus cable?

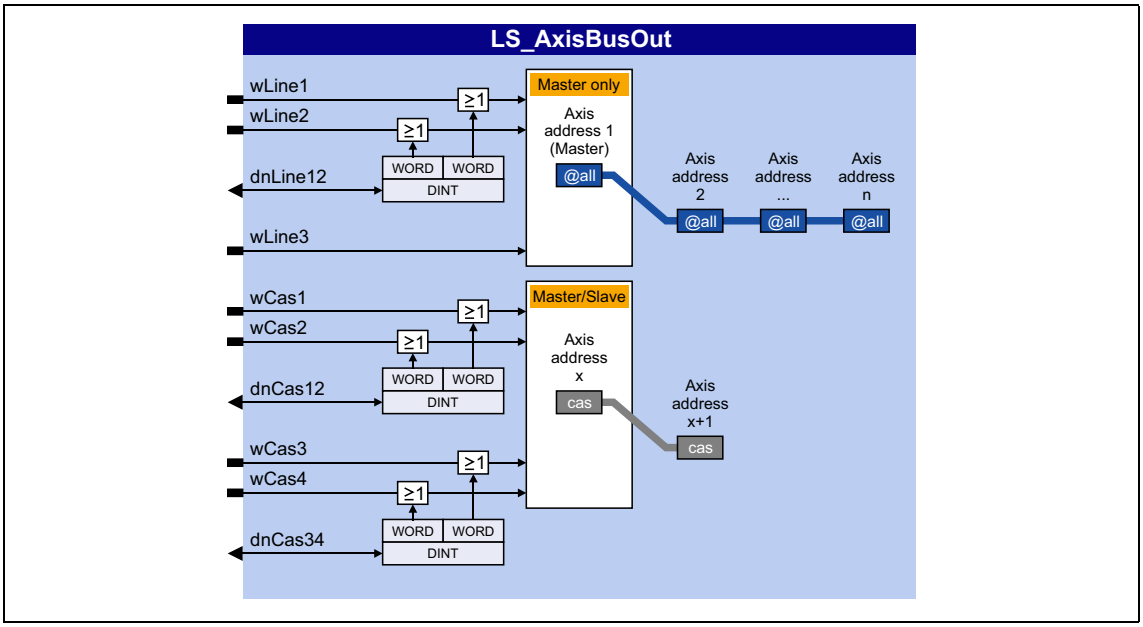
Such severe errors mostly cause a bus-off error in one or even several devices of the system.

- In this case as well, the data error monitoring is triggered.
- The "BusOff" state will be automatically reset after the time set in C02431/2 has expired if the error cause has been removed. Then, the axis bus changes to normal operation again.

### 14.2.5 Internal interfaces | System block "LS_AxisBusOut"

This system block is used to send line data from the master to all slaves (@all) and/or send the cascaded data (cas).

- The line data is sent as process data object (PDO) per transmission cycle to all slaves and forwarded to the application. Moreover, this PDO serves to transmit an internal control word (thus, only three words are available for data transfer).
- The cascaded data is also sent as PDO per transmission cycle from one slave to the next. The control which slave has to send when and at what time a slave in the chain has to accept the data, is executed by the internal control word in the PDO of the line data.

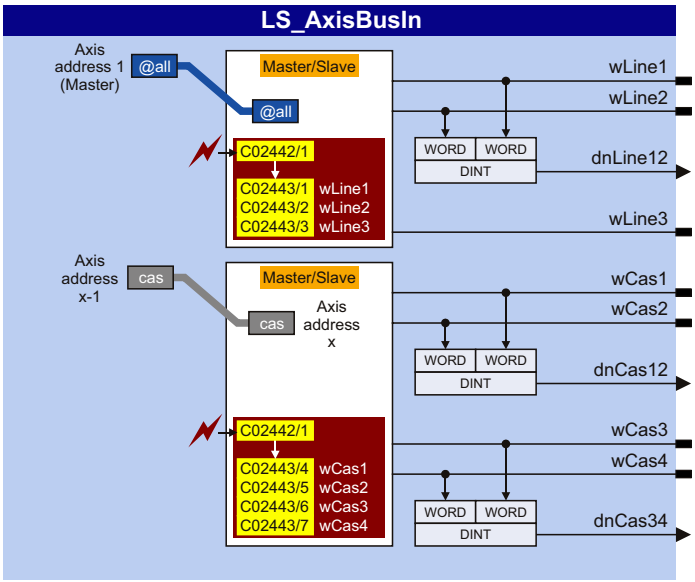



#### Inputs

| Input<br>Data type | Information/possible settings |
|---|---|
| wLine1/wLine2<br>WORD | Line data - word 1 and word 2<br>• Selection as separate words or as double word.<br>• The double word is OR'd with word 1 and word 2. |
| dnLine12<br>DINT | |
| wLine3<br>WORD | Line data - word 3 |
| wCas1/wCas2<br>WORD | Cascaded data - word 1 and word 2<br>• Selection as separate words or as double word.<br>• The double word is OR'd with word 1 and word 2. |
| dnCas12<br>DINT | |
| wCas3/wCas4<br>WORD | Cascaded data - word 3 and word 4<br>• Selection as separate words or as double word.<br>• The double word is OR'd with word 3 and word 4. |
| dnCas34<br>DINT | |

### 14.2.6 Internal interfaces | System block "LS_AxisBusIn"

This system block is used to receive the (line) data from the master (@all) and/or receive the cascaded data (cas).




#### Outputs

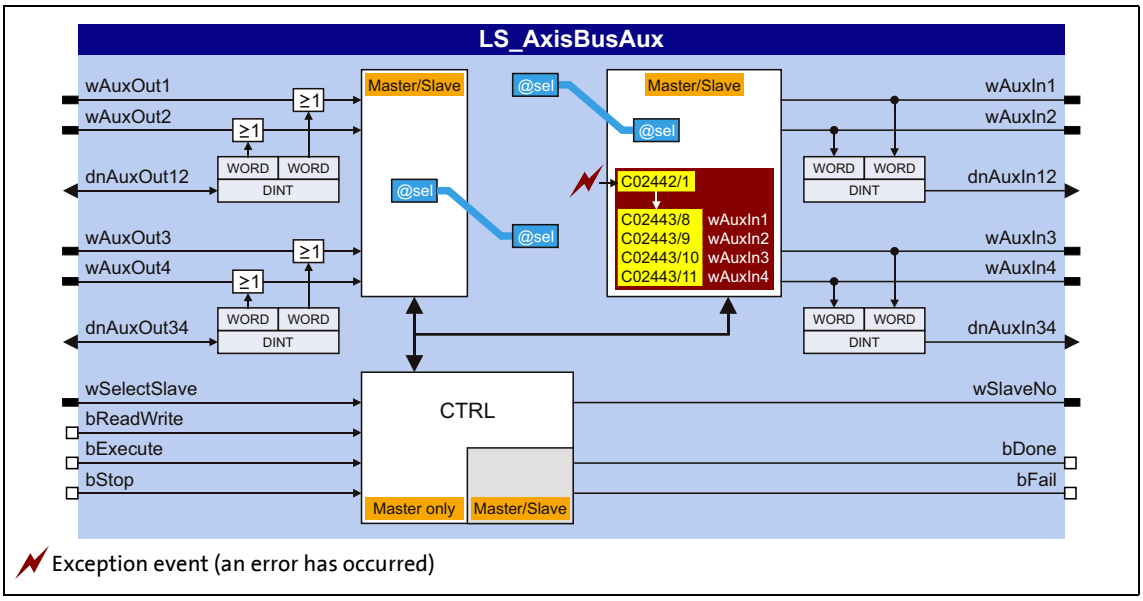
| Out<br>Data type | Value/meaning |
|---|---|
| wLine1/wLine2<br>WORD | Line data - word 1 and word 2 |
| dnLine12<br>DINT | Line data - word 1 and word 2 as double word |
| wLine3<br>WORD | Line data - word 3 |
| wCas1/wCas2<br>WORD | Cascaded data - word 1 and word 2 |
| dnCas12<br>DINT | Cascaded data - word 1 and word 2 as double word |
| wCas3/wCas4<br>WORD | Cascaded data - word 3 and word 4 |
| dnCas34<br>DINT | Cascaded data - word 3 and word 4 as double word |

#### Related topics:

- Configuring exception handling of the received data (📖 867)

### 14.2.7 Internal interfaces | System block "LS_AxisBusAux"

This system block is used for data transfer between the master and one selected slave (@sel). Data can either be received from the slave ("read") or sent to the slave ("write"). By selecting the axis bus address "63", data can also be sent to all slaves at the axis bus.




#### Inputs

| Input<br>Data type | Information/possible settings | |
|---|---|---|
| wAuxOut1/wAuxOut2<br>WORD | Data to be sent - word 1 and word 2<br>• Selection as separate words or as double word.<br>• The double word is OR'd with word 1 and word 2. | |
| dnAuxOut12<br>DINT | | |
| wAuxOut3/wAuxOut4<br>WORD | Data to be sent - word 3 and word 4<br>• Selection as separate words or as double word.<br>• The double word is OR'd with word 3 and word 4. | |
| dnAuxOut34<br>DINT | | |
| The following inputs are only relevant for the master except for *bExecute*.<br>Via these inputs, the master controls the data acceptance of the "@sel" data in the slaves. | | |
| wSelectSlave<br>WORD | Axis bus address of the slave data is to be received of or data is to be sent to.<br>• "0" ≡ data is sent to no slave.<br>• "63" ≡ data is sent to all slaves. | |
| bReadWrite<br>BOOL | Read/write access | |
| | FALSE | Data is sent to selected slave. |
| | TRUE | Data is received by selected slave. |
| bExecute<br>BOOL | Data acceptance | |
| | TRUE | **For the master:** The pending data is accepted and the read/write access is executed. |
| | | **For the slave:** The @sel data received by the master are provided via the outputs *wAuxIn1...4* of this function block. |

| Input<br>Data type | Information/possible settings | |
|---|---|---|
| bStop<br>BOOL | Stop data transfer | |
| | TRUE | The axis bus changes to the "Stopped" state.<br>• The slaves neither accept any data nor send any data anymore.<br>• The master continues to send the line data.<br>• The bit 2 ("NodeStopped") in C02442/1 serves to set whether the last received data remain active or the decoupling values are set.<br>▸ Configuring exception handling of the received data |

### Outputs

| Out<br>Data type | Value/meaning | |
|---|---|---|
| wAuxIn1/wAuxIn2<br>WORD | Received data - word 1 and word 2 | |
| dnAuxIn12<br>DINT | Received data - word 1 and word 2 as double word | |
| wAuxIn3/wAuxIn4<br>WORD | Received data - word 3 and word 4 | |
| dnAuxIn34<br>DINT | Received data - word 3 and word 4 as double word | |
| wSlaveNo<br>WORD | Display which slave has sent its data.<br>• Output only relevant for master. | |
| bDone<br>BOOL | Status signal "Data transfer completed" | |
| | FALSE | The master has sent a new read request to a slave. |
| | TRUE | Data has been received correctly. |
| bFail<br>BOOL | "Fault" status signal | |
| | TRUE | Data has not been received correctly (e.g. bus error).<br>▸ Axis bus data error monitoring |

### Related topics:

▸ Configuring exception handling of the received data (📖 867)

## 14.3 IO axis bus

**Stop!**

The IO axis bus of the 8400 TopLine inverter is not compatible to the state bus of the 9300/9400 device series due to different voltage levels!

In order to use the IO axis bus, first decide whether the IO axis bus is to be used for transferring controller errors in the network ("release cord" principle) or as a pure open-collector IO function. The required function of the IO axis bus must then be set for all nodes in C02440/1.

- "Master/slave" function ("release cord" principle)
  - In this setting, the IO axis bus only knows the two statuses "OK" and "Error".
  - Each node connected to the IO axis bus can set the IO axis bus into the "Error" status.
  - In the "Error" status, all nodes start their adjustable response, e.g. a synchronised braking of the drive system or the master only brakes the network to standstill.
  - The "Error" status can only be reset by the node defined as "master".
  - The internal time base of the inverters can be synchronised via the IO axis bus.
- "IO" function
  - With this setting, the IO axis bus is used as a pure IO transmission medium.
  - Each node connected to the IO axis bus can transmit a TRUE signal to all nodes.

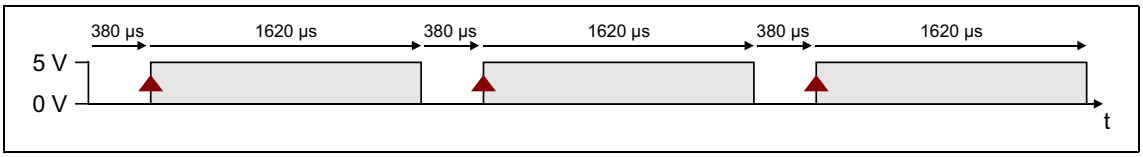
**Short overview of the relevant parameters:**

| Parameters | Info | Lenze setting<br>Value \| Unit |
|---|---|---|
| C00591/2 | Resp. to axis bus IO error<br>• From version 12.00.00 | 1: Fault |
| C01120 | Sync signal source | 0: Off |
| C02440/1 | Axis bus IO function | 0: Off |
| C02444/1 | Axis bus settings (bit coded)<br>• From version 12.00.00<br>• Bit 0 serves to reconfigure the LS_AxisBusIO.*bSetFail_DigOut* input from level-sensitive to edge-sensitive. | 0x0000 |

### 14.3.1 "Master/slave" function

With this setting, the IO axis bus can be used as "release cord", i.e. each node can set all other nodes into an error status.

One IO axis bus node takes over the master role. The master's task is to output a synchronisation cycle to the IO axis bus to which the slaves can orient themselves:

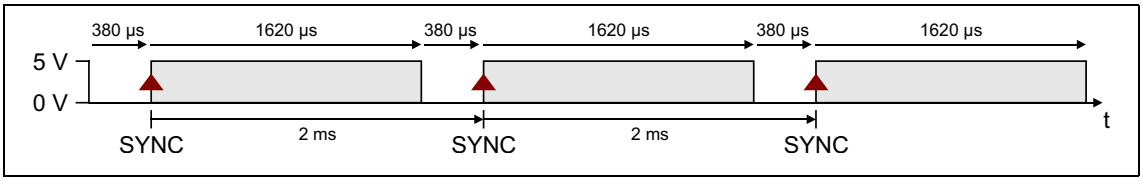



[14-4] IO axis bus: Synchronisation cycle from the master (error-free status)

The inverter is configured as master or slave in C02440/1 or by selecting "1: Master" or "2: Slave".

#### Synchronisation of the internal time base

The internal time base of the slaves can be synchronised via the IO axis bus. The slaves orient themselves to the synchronisation cycle output by the master:




[14-5] Synchronisation cycle

- In order that the IO axis bus is used as synchronisation source, all slaves must be set to "AxisBusIO" in C01120.
  - The internal time base is synchronised based on the HIGH edge in a 2-ms cycle (see illustration above).
  - Internal timing and trimming of the phase position are set automatically.
- Basically, only one source is allowed to synchronise the internal time base. If "AxisBusIO" is selected, this is the IO axis bus master.

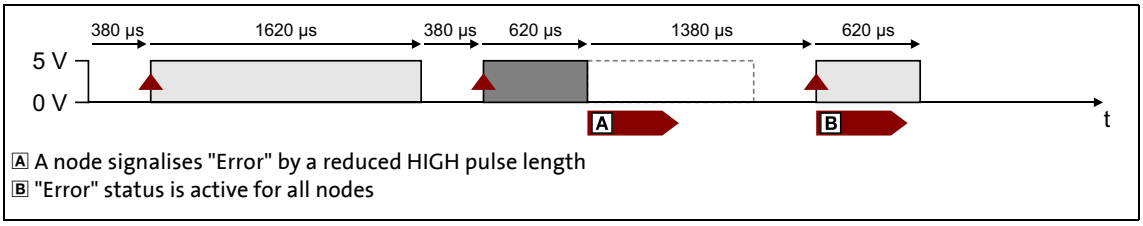
More information on this subject can be found in the following main chapter "Synchronisation of the internal time base". (📖 880)

#### Error-free status

In the error-free status, all nodes are error-active, i.e. each node can set the IO axis bus into the "Error" status.

### Error status

By setting the LS_AxisBusIO.*bSetFail_DigOut* input to TRUE, each node can set the IO axis bus into the "Error" status. The node signalises this by reducing the HIGH pulse length to 620 µs:




A A node signalises "Error" by a reduced HIGH pulse length
B "Error" status is active for all nodes

[14-6] Synchronisation cycle (error status)

All other nodes detect the "error status" due to the changed pulse length which has the following effect:

- The LS_AxisBusIO.*bFail_DigIn* output is set to TRUE. This digital signal can be used for any application within the function block interconnection.
- From version 12.00.00:
  - The error response set in C00591/2 is activated (Lenze setting: "Fault").
  - The error message "Ab02: Axis bus IO error" is entered into the logbook.
- The nodes are now error-passive, i.e. they cannot signalise any further errors in order that the HIGH edge for synchronisation can be detected after 2 ms.

### Reset "error" status

**Note!**

The "error" status can only be reset by the master!

A FALSE/TRUE edge at the LS_AxisBusIO.*bResetFail* input (for the master) serves to reset the "error" status again. The master signalises this by providing a LOW pulse for 3 ms.

All other nodes detect the error reset due to this signal, which has the following effect:

- The LS_AxisBusIO.*bFail_DigIn* output is reset to FALSE.
- After a delay time of 2 ms, the LS_AxisBusIO.*bResetFailIn* output is set for 3 ms to TRUE. This digital signal can be used within the function block interconnection to e.g. reset errors.

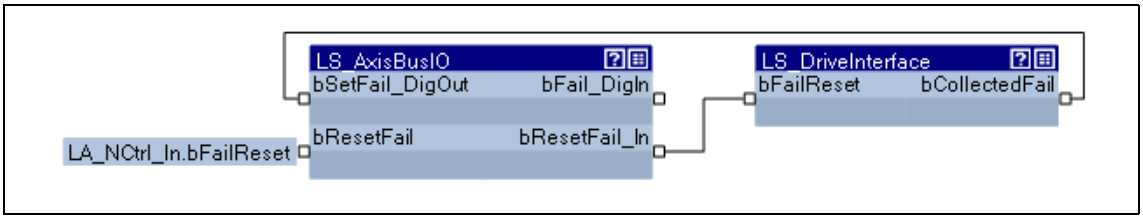
**Application notes**

In the event of an error in a network, a slave drive should never trigger the "Fault error response as this causes the pulse inhibit. This deactivates the coupling to the master and hence the network.

Possible solution:

1. Reconfigure the "Fault" error response to "Warning" in the slave axes.
2. Use the "master/slave" function ("release cord") to report errors and warnings to the master. The master can then initiate the respective response for the entire network.

In the following interconnection example, the slave drive uses the *bCollectedFail* group error output of the SB LS_DriveInterface for "pulling the release cord". Non-relevant inputs/outputs of the SB LS_DriveInterface are hidden for an easier presentation.




[14-7] Configuration for "master/slave" function ("release cord")

In order that this mechanism works, the following parameter setting is additionally required for the slave drive:

1. Go to C00148 and set bit 6 to "1" in order that warnings are also provided via the *bCollectedFail* group error output of the SB LS_DriveInterface.
2. Go to C02444/1 and set bit 0 to "1" in order that the *bSetFail_DigOut* input is edge-sensitive instead of level-sensitive. Otherwise, the error set input is "latched" and the slave drive does not respond anymore to an error reset by the master.

### 14.3.2 "IO" function

With this setting, the IO axis bus is used as a pure IO transmission medium, just like it the case for the 9300 and 9400 device series.

The "IO" function is activated in C02440/1 by selecting "3: IO".

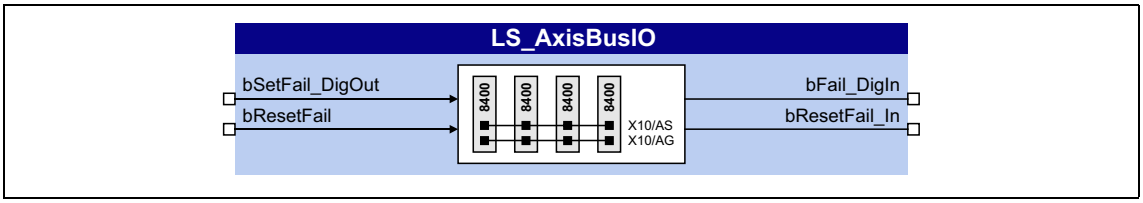
**Note!**

- "IO" function and Data transfer axis bus are mutually exclusive since with activated "IO" function, a synchronisation of the internal time base is not possible (which is required for a "safe" data transfer).
- The "IO" function must be set for all IO axis bus nodes.
- Triggering an error message with this function is only possible by function block interconnection.
- Resetting an error message is not possible with this function.

- By setting the LS_AxisBusIO.*bSetFail_DigOut* input to TRUE, the IO axis bus is set to the dominant level and for all nodes, the LS_AxisBusIO.*bFail_DigIn* output is set to TRUE.
- The LS_AxisBusIO.*bResetFail* input has no function.
- The LS_AxisBusIO.*bResetFail_In* output is always FALSE.

### 14.3.3 Internal interfaces | System block "LS_AxisBusIO"

The **LS_AxisBusIO** system block maps the IO axis bus in the FB Editor:




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bSetFail_DigOut<br>BOOL | Set IO axis bus into the "error" status / IO data exchange | |
| | TRUE<br>or<br>FALSE↗TRUE | For all IO axis bus nodes, the *bFail_DigIn* output is set to TRUE.<br>• From version 12.00.00 onwards, bit 0 in C02444/1 can be used to reconfigure the *bSetFail_DigOut* input from level-sensitive to edge-sensitive. |
| bResetFail<br>BOOL | Reset "error" status<br>• Only possible with "Master/slave" function via the IO axis bus master. | |
| | FALSE↗TRUE | For all IO axis bus nodes, the *bFail_DigIn* output is reset to FALSE.<br>After a delay time of 2 ms, the *bResetFail_In* output is set to TRUE for 3 ms for error reset. |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bFail_DigIn<br>BOOL | With "master/slave" function: "Error" status<br>With "IO" function: Data exchange | |
| | TRUE | An IO axis bus nodes has set the *bSetFail_Digout* input to TRUE. |
| bResetFail_In<br>BOOL | "Error" status has been reset<br>• With "IO" function, this output is always FALSE. | |
| | TRUE | The IO axis bus master has reset the error message.<br>This status is only pending for 3 ms. |

# 15 Synchronisation of the internal time base

In a drive system, synchronising the internal time bases of all inverters involved makes sense because cyclic process data should be processed synchronously in all drives.

- One of the following signal sources can be used for automatic synchronisation of the internal time base of the inverter:
  - CAN bus ("CAN on board") ➞ sync telegram
  - Axis bus ("AxisBusIO") ➞ master/slave function
  - MCI ➞ sync signal of a plugged-in communication module (EtherCAT, PROFINET or Powerlink)

**Short overview of the parameters for the synchronisation of the internal time base:**

| Parameters | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00370/1 | CAN Sync instant of transmission | - | µs |
| C00370/2 | Sync instant of reception | - | µs |
| C01120 | Sync signal source | Off | |
| C01121 | Sync cycle time setpoint | 1000 | µs |
| C01122 | Sync phase position | 0 | µs |
| C01123 | Sync window | 100 | µs |
| C01124 | Sync correction width | 320 | ns |
| Greyed out = display parameter | | | |

### Sync signal source

The synchronisation signal source can be selected in C01120. As a general rule, only one source can be used to synchronise the internal time base.

### Sync cycle time setpoint

Time after which the internal phase-locking loop (PLL) anticipates the synchronisation signals. The time must be set in C01121 according to the cycle of the synchronisation source selected in C01120.

**Note!**

- Only integer multiples of 1000 µs can be set in C01121.
- Intelligent communication modules usually define the cycle time setpoint derived from the bus cycle. In this case, a manual change is not possible.
- Even in case of a synchronisation via axis bus, a setting of the cycle time and the phase position is not required/possible and is executed automatically.

Example: For the CAN bus, 2 ms has been selected as interval between two synchronisation signals. If the CAN bus is to be used as synchronisation source, a cycle time setpoint of 2000 µs must be selected in C01121.

### Sync phase position

The phase position determines the zero-time of the internal system cycle with regard to the synchronisation signal (bus cycle). Since PDO processing is an inherent part of the system part of the application, the instant of acceptance of the PDOs is postponed as well by a changed phase position.

- If "0" is set, the internal system cycle starts at the same time as the synchronisation signal.
- If a value > 0 is set, the internal system cycle starts by the set time earlier (the phase position has a negative effect) than the synchronisation signal.
- Intelligent communication modules define the optimal time with activated synchronisation by themselves. In this case, a manual change is not possible.
- For determining C01122, the point in time where all bus nodes have valid PDOs is decisive.

Example: If the phase position is set to 550 µs, the system part of the application starts 550 µs before the arrival of the synchronisation signal.

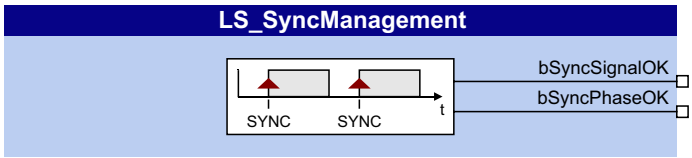
### Sync correction width

If the cycle times of the synchronisation signal and the phase-locking loop (PLL) are different, the setting in C01124 defines the correction increments for the phase-locking loop.

- The recommended reset time for the CAN bus as synchronisation source in case of occurring deviations is 320 ns (Lenze setting).
- If synchronisation is not reached, select a higher correction width.
- The optimum setting depends on quartz precision and must be determined empirically if required.

## 15.1 Internal interfaces | System block "LS_SyncManagement"

The SB **LS_SyncManagement** provides status information for synchronising the internal time base:




### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bSyncSignalOK<br>BOOL | TRUE | Sync signal OK |
| bSyncPhaseOK<br>BOOL | TRUE | Sync phase position OK |

# 16 Parameter change-over

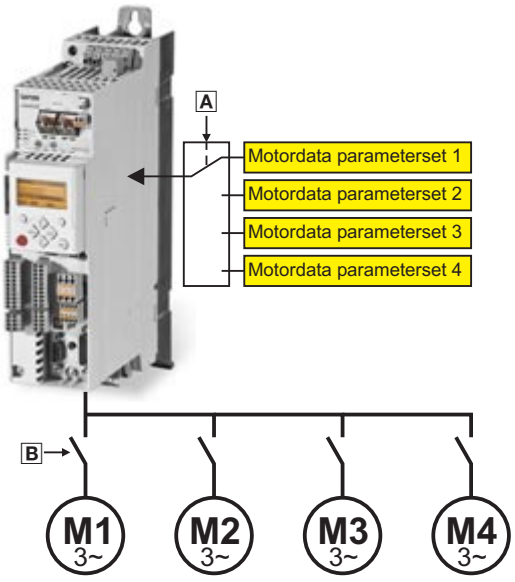
For up to 32 freely selectable parameters, this basic function provides a change-over between four sets with different parameter values.

The parameter list is created in the same way as the user menu is composed, namely by means of parameterisation. In the »Engineer«, a user-friendly parameterisation dialog with import and export functions is available for this purpose.

### Motor data changeover

From version 12.00.00, an optional changeover between four parameter sets with different motor and control settings is supported in addition.

The optional motor data changeover is provided for applications/machines which have multiple axes controlled successively but which do not require the simultaneous operation of multiple motors. In this case, the very same inverter can control the motors successively. Advantages of this solution: Fewer components ( inverters) and hence less energy consumption.




A Motor data changeover
B Motor changeover (e.g. via motor contactors)

[16-1] Principle of the selective control of multiple motors using one inverter

Principle:

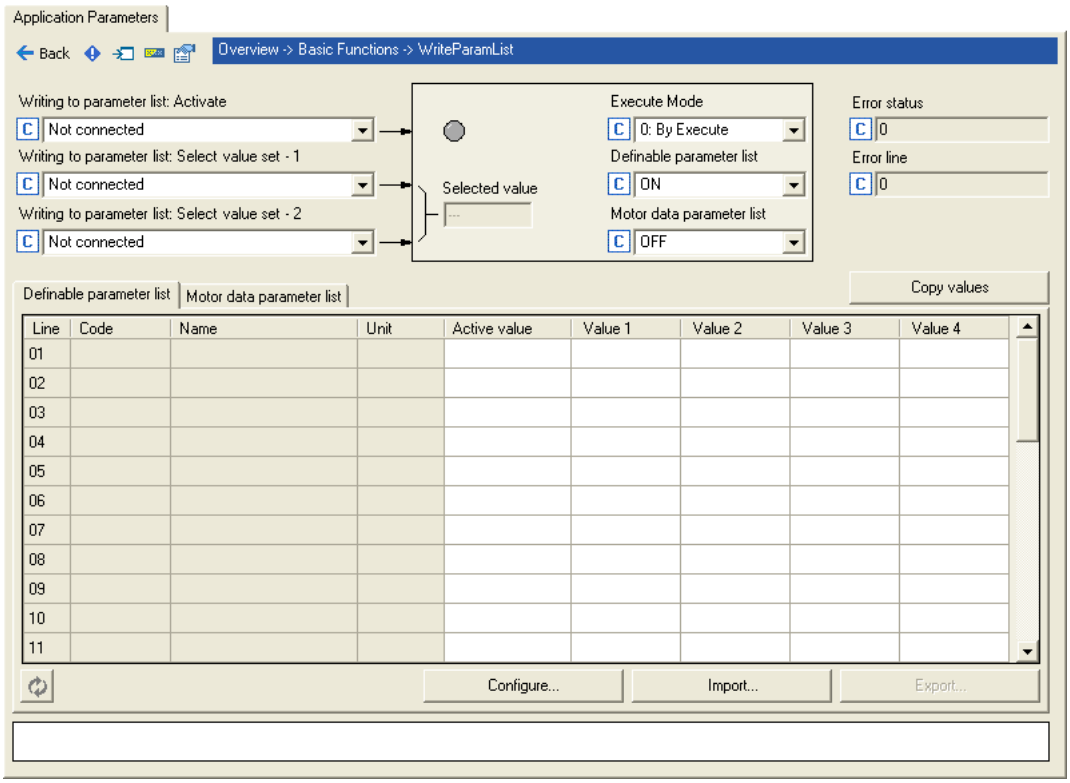
- The motor which is being controlled at the moment is connected to the inverter via motor contactors. (The contactor system can e.g. be controlled via the digital outputs of the inverter.)
- At the same time, the motor data changeover serves to activate the motor and control settings suitable for the motor in the inverter.

Note:
The motor data changeover is carried out within 2 ms and hence within the opening and closing times of most motor contactors.

## 16.1 Configuring parameter change-over via the »Engineer« parameterisation dialog

**Proceed as follows to open the dialog for parameterising the parameter change-over:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine inverter.
2. Select the **Application parameters** tab from the *Workspace*.
3. Go to the *Overview* dialog level and click the "**Basic functions**" button.
4. Go to the *Overview* → *Basic functions* dialog box and click the **Parameter change-over** button.




### Note!

The "parameter change-over" basic function is always processed, even if the corresponding LS_WriteParamList system block has been removed from the interconnection by the function block editor.

If you do not require this basic function anymore, delete the composed parameter list in order that no unwanted parameter write operations take place.

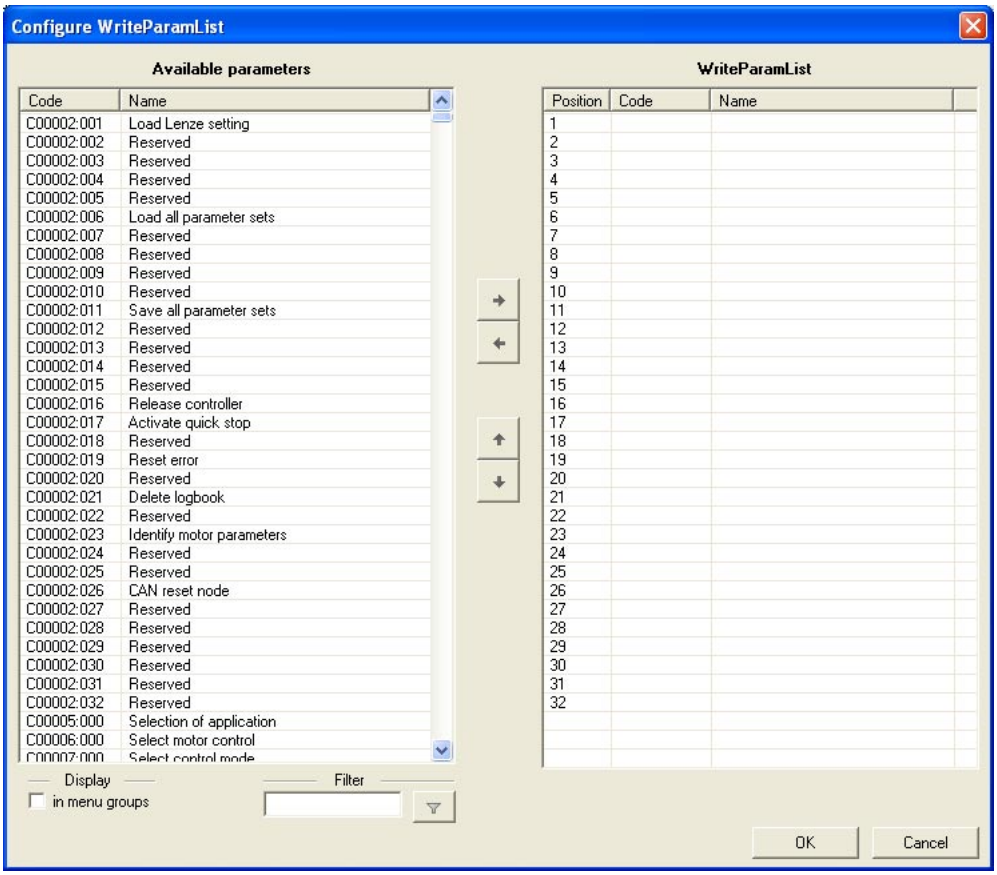
### 16.1.1 Configuring the parameter list(s)

#### Configuring the definable parameter list

In the Lenze setting, the definable parameter list does not yet contain any parameters.

**How to configure the definable parameter list:**

1. Click the **Adapt...** button.
   - The dialog box entitled *Configure WriteParamList* is shown:




   - On the left-hand side, all the parameters of the inverter with write and read access are shown in the list entitled **Available parameters**.
   - If the option **In menu groups** is activated, all parameters are shown assigned to their functions.
   - By clicking on the button in the **Filter** area, you can shorten the list of available parameters. If, for example, you enter the text "ain1" and then click on the button, only those parameters whose designation contains this text are shown for selection.
2. Highlight the parameter/parameters in the **Available parameters** list that is/are to be added to the *WriteParamList*.
   - Here, you can use the <**Ctrl**> key and the<**Shift**> key for multiple selection, as in the case of general Windows functions.

3. Click on the → button in order to add the highlighted parameters to the *WriteParamList* on the right-hand side.
   - With the ↑ and ↓ buttons, you can alter the sequence of parameters in the *WriteParamList*.

   To remove parameters from the *WriteParamList*, proceed as follows:
   - Highlight the parameter/parameters in the **WriteParamList** that is/are to be removed from the *WriteParamList*.
   - Click on the ← button to remove the highlighted parameters from the *WriteParamList*.
4. Click on the **OK** button to accept the configuration and close the dialog box.
   - You can call the configuration dialog again at any time in order to change or expand the *WriteParamList* retrospectively.

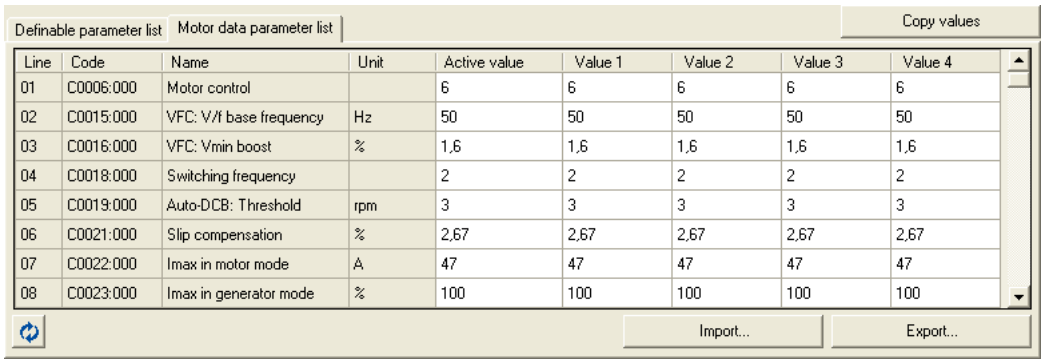
### Changing the values of the definable parameter list

After the compilation of the definable parameter list, the values in the columns **1st value** ... **4th value** first correspond to the Lenze setting of the respective parameter.

- Just click one of the input fields in these columns to change the displayed value.
- If you place the cursor in an input field, the permitted value range for the corresponding parameter is shown under the table.

### Changing the values of the motor data parameter list

Click the **Motor data parameter list** register to put it in the foreground:




| Line | Code | Name | Unit | Active value | Value 1 | Value 2 | Value 3 | Value 4 |
|---|---|---|---|---|---|---|---|---|
| 01 | C0006:000 | Motor control | | 6 | 6 | 6 | 6 | 6 |
| 02 | C0015:000 | VFC: V/f base frequency | Hz | 50 | 50 | 50 | 50 | 50 |
| 03 | C0016:000 | VFC: Vmin boost | % | 1,6 | 1,6 | 1,6 | 1,6 | 1,6 |
| 04 | C0018:000 | Switching frequency | | 2 | 2 | 2 | 2 | 2 |
| 05 | C0019:000 | Auto-DCB: Threshold | rpm | 3 | 3 | 3 | 3 | 3 |
| 06 | C0021:000 | Slip compensation | % | 2,67 | 2,67 | 2,67 | 2,67 | 2,67 |
| 07 | C0022:000 | Imax in motor mode | A | 47 | 47 | 47 | 47 | 47 |
| 08 | C0023:000 | Imax in generator mode | % | 100 | 100 | 100 | 100 | 100 |

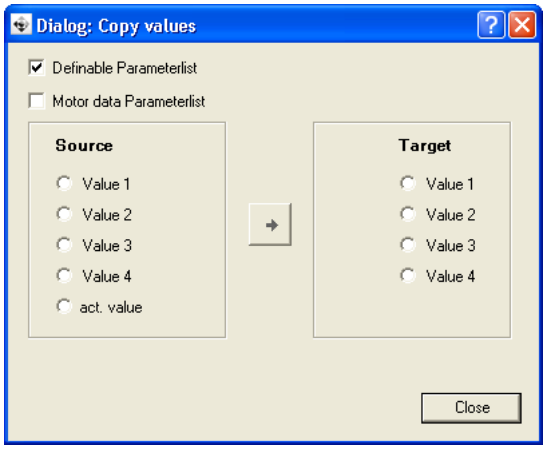
- Unlike the "definable" parameter list, the motor data parameter list has a fixed assignment to the motor and control parameters.
  - A list of the switchable motor and control parameters is provided in chapter "Configuring the motor data parameter list by means of parameterisation (📖 890)".
- The preset values are changed in the same way as the definable parameter list.

### Copying values

All the settings of a value set can be copied to another value set.

**To copy values, proceed as follows:**

1. Click on the **Copy values** button.
   - The *Copy values* dialog box is displayed:



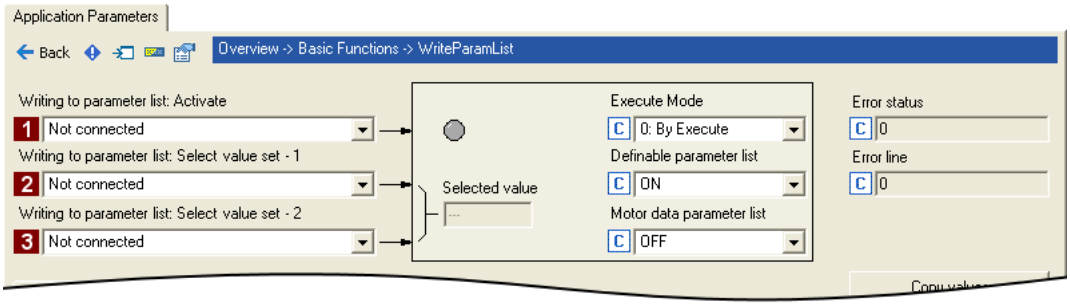

2. Define the parameter list to be copied via the two upper checkboxes.
3. Select **Source** and **Target**.
4. Click on → button in order to copy the values from **Source** to **target**.

### Importing/exporting the list

For cross-device reuse of the configured *WriteParamList*, you can click on the **Export** and **Import** buttons to save the parameter selection as an *.epc file and then to re-import the saved *.epc file into another 8400 inverter.

### 16.1.2 Configuring control inputs

The three control inputs for parameter change-over can be configured via the following parameters:

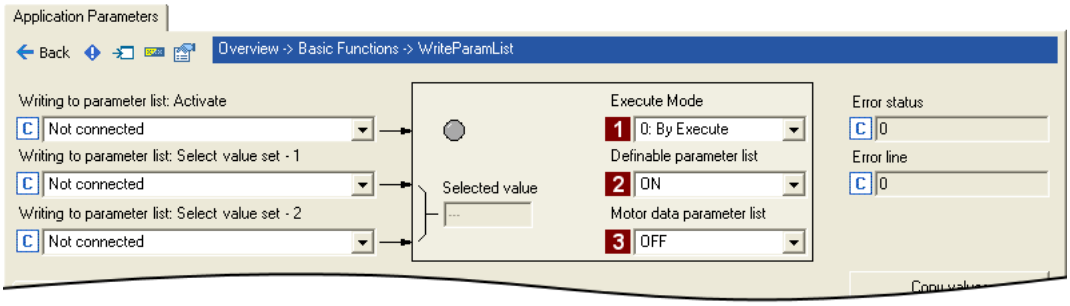



| Parameters | | Lenze setting | Info |
|---|---|---|---|
| 1 | **Activate writing** (C00621/123) | 0: Not connected | Selection of the signal source that activates writing to the parameter list (for Execute Mode = "0: by Execute"). |
| 2 | **Selection value set - 1** (C00621/124) | 0: Not connected | Selection of the two signal sources for the binary-coded selection of the value set 1 ... 4 to be used (see following truth table). |
| 3 | **Selection value set - 2** (C00621/125) | 0: Not connected | |

Truth table for the selection of the value set to be used:

| Selection value set - 1 | Selection value set - 2 | Value set used |
|---|---|---|
| FALSE | FALSE | Value set 1 |
| TRUE | FALSE | Value set 2 |
| FALSE | TRUE | Value set 3 |
| TRUE | TRUE | Value set 4 |

### 16.1.3 Functional settings




| Parameters | | Lenze setting | Info |
|---|---|---|---|
| 1 | **Execute Mode** (C01082) | 0: by Execute | For writing the parameter list, two modes are available:<br>• 0: by Execute (Lenze setting)<br>The writing of the parameter list is activated by a FALSE/TRUE edge at the *bExecute* control input.<br>• 1: by Input Select<br>The parameter list is written when the selection inputs *bSelectWriteValue_1* and *bSelectWriteValue_2* are changed and once during the initialisation of the inverter.<br>**Note:** If the execute mode changes from "0: by Execute" to "1: by Input Select", the parameter list selected by the selection inputs is written once. In the execute mode "1: by Input Select", the parameter list then is only written again when a change occurs on the selection inputs . |
| 2 | **Definable parameter list** (C02200/1) | 1: On | Switch on/off parameter changeover for the definable parameter list. |
| 3 | **Motor data parameter list** (C02200/2) | 0: Off | Switch on/off parameter changeover for the motor data parameter list. |

### 16.1.4 Error message

With every run through the main program, one parameter of the definable parameter list is written until the parameter list has been executed completely. If an error occurs, C01083 displays an error status and C01084 displays the number of the list entry causing the error (in connection with the selected value set).

- If several errors occur at the same time, only the first incorrect list entry will be displayed. Hence, after elimination of the displayed error and another activation, more errors may be displayed.
- The parameter list will always be processed from beginning to end, even if errors occur in the meantime.

## 16.2 Configuring the definable parameter list by means of parameterisation

The following application example shows the necessary procedure for configuring the list without using the »Engineer« parameterisation dialog.

Task:

The **LS_WriteParamList** SB is to be used to write to parameters C00012, C00026/1 and C00027/1.

### Compiling the parameter list

In C01085/1 ... n, specify the above-named parameters in the <Code>,<Subcode> format:

- C01085/1 = 12.000
- C01085/2 = 26.001
- C01085/3 = 27.001
- C01085/4 ... n = 0.000 (no parameter)

**Note!**

Gaps in the parameter list (setting = 0.000) are permissible and are skipped in the process.

Invalid parameter entries are not accepted when being entered.

### Entering values for the parameters (value set 1)

In C01086/1 ... n, specify the values to be used to describe the selected parameters. The values are entered according to the scaling format/scaling factor of the respective parameter.

- C01086/1 = <value> for list entry 1 (in our example: for parameter C00012)
- C01086/2 = <value> for list entry 2 (in our example: for parameter C00026/1)
- C01086/3 = <value> for list entry 3 (in our example: for parameter C00027/1)

These values are used in the writing process if the two *bSelectWriteValue_1* and *bSelectWriteValue_2* inputs are not assigned or both set to FALSE.

### Entering other values for the parameters (value sets 2 ... 4)

If required, up to three other sets can be set in the same way in C01087/1 ... n to C01089/1 ... n which can optionally be written to the parameters. The decision as to which value set is finally used is dependent upon the assignment of the two bSelectWriteValue_1 and bSelectWriteValue_2 inputs:

## 16.3 Configuring the motor data parameter list by means of parameterisation

The setting of the values for value sets 1 ... 4 of the motor data parameter list can also be carried out directly via the codes from code C02210 (see following table; columns 1st value ... 4th value).

- Unlike the "definable" parameter list, the motor data parameter list has a fixed assignment to the motor and control parameters.
- The preset values for value sets 1 ... 4 correspond to the Lenze setting of the respective motor or control parameter.
- The values are entered according to the scaling format / scaling factor of the respective motor or control parameter.

| Code | Name | Lenze setting | | 1st value | 2nd value | 3rd value | 4th value |
|---|---|---|---|---|---|---|---|
| C00006 | Motor control | 6 | | C02210/1 | C02210/2 | C02210/3 | C02210/4 |
| C00015 | VFC: V/f base frequency | 50.0 | Hz | C02212/1 | C02212/2 | C02212/3 | C02212/4 |
| C00016 | VFC: Vmin boost | 1.60 | % | C02213/1 | C02213/2 | C02213/3 | C02213/4 |
| C00018 | Switching frequency | 2 | | C02214/1 | C02214/2 | C02214/3 | C02214/4 |
| C00019 | Auto-DCB: Threshold | 3 | rpm | C02215/1 | C02215/2 | C02215/3 | C02215/4 |
| C00021 | Slip comp. | 2.67 | % | C02216/1 | C02216/2 | C02216/3 | C02216/4 |
| C00022 | Imax in motor mode | 47.00 | A | C02217/1 | C02217/2 | C02217/3 | C02217/4 |
| C00023 | Imax in generator mode | 100.00 | % | C02218/1 | C02218/2 | C02218/3 | C02218/4 |
| C00036 | DC braking: Current | 50.00 | % | C02219/1 | C02219/2 | C02219/3 | C02219/4 |
| C00070/1 | SLVC: Vp speed controller | 15.00 | | C02220/1 | C02220/2 | C02220/3 | C02220/4 |
| C00070/2 | SC: Vp speed controller | 6.00 | | C02220/5 | C02220/6 | C02220/7 | C02220/8 |
| C00070/3 | SLPSM: Vp speed controller | 3.00 | | C02220/9 | C02220/10 | C02220/11 | C02220/12 |
| C00071/1 | SLVC: Ti speed controller | 100.0 | ms | C02221/1 | C02221/2 | C02221/3 | C02221/4 |
| C00071/2 | SC: Ti speed controller | 50.0 | ms | C02221/5 | C02221/6 | C02221/7 | C02221/8 |
| C00071/3 | SLPSM: Ti speed controller | 100.0 | ms | C02221/9 | C02221/10 | C02221/11 | C02221/12 |
| C00072 | SC: Tdn speed controller | 0.00 | ms | C02222/1 | C02222/2 | C02222/3 | C02222/4 |
| C00073/1 | VFC: Vp Imax controller | 0.25 | | C02223/1 | C02223/2 | C02223/3 | C02223/4 |
| C00073/2 | SLVC: Vp torque controller | 1.25 | | C02223/5 | C02223/6 | C02223/7 | C02223/8 |
| C00074/1 | VFC: Ti Imax controller | 65 | ms | C02224/1 | C02224/2 | C02224/3 | C02224/4 |
| C00074/2 | SLVC: Ti torque controller | 30 | ms | C02224/5 | C02224/6 | C02224/7 | C02224/8 |
| C00075 | Vp current controller | 7.00 | V/A | C02225/1 | C02225/2 | C02225/3 | C02225/4 |
| C00076 | Ti current controller | 10.61 | ms | C02226/1 | C02226/2 | C02226/3 | C02226/4 |
| C00077 | SC: Vp field controller | 12.80 | | C02227/1 | C02227/2 | C02227/3 | C02227/4 |
| C00078 | SC: Tn field controller | 256.0 | ms | C02228/1 | C02228/2 | C02228/3 | C02228/4 |
| C00079/1 | SC: Current controller - feedforward control | 0 | | C02229/1 | C02229/2 | C02229/3 | C02229/4 |
| C00079/2 | SC: adapt. field weak. controller | 1 | | C02229/5 | C02229/6 | C02229/7 | C02229/8 |
| C00079/3 | SC: n-Ctrl Anti-Wind-Up | 0 | | C02229/9 | C02229/10 | C02229/11 | C02229/12 |
| C00080 | Override point of field weakening | 0 | Hz | C02230/1 | C02230/2 | C02230/3 | C02230/4 |
| C00081 | Rated motor power | 11.00 | kW | C02231/1 | C02231/2 | C02231/3 | C02231/4 |
| C00082 | Motor rotor resistance | 276 | mOhm | C02232/1 | C02232/2 | C02232/3 | C02232/4 |
| C00084 | Motor stator resistance | 330 | mOhm | C02233/1 | C02233/2 | C02233/3 | C02233/4 |
| C00085 | Motor stator leakage inductance | 3.50 | mH | C02234/1 | C02234/2 | C02234/3 | C02234/4 |
| C00087 | Rated motor speed | 1460 | rpm | C02236/1 | C02236/2 | C02236/3 | C02236/4 |
| C00088 | Rated motor current | 21.00 | A | C02237/1 | C02237/2 | C02237/3 | C02237/4 |
| C00089 | Rated motor frequency | 50 | Hz | C02238/1 | C02238/2 | C02238/3 | C02238/4 |
| C00090 | Rated motor voltage | 400 | V | C02239/1 | C02239/2 | C02239/3 | C02239/4 |
| C00091 | Motor cosine phi | 0.85 | | C02240/1 | C02240/2 | C02240/3 | C02240/4 |
| C00092 | Motor magnetising inductance | 81.0 | mH | C02241/1 | C02241/2 | C02241/3 | C02241/4 |
| C00095 | Motor magnetising current | 8.50 | A | C02242/1 | C02242/2 | C02242/3 | C02242/4 |
| C00106 | Auto-DCB: Hold time | 0.500 | s | C02244/1 | C02244/2 | C02244/3 | C02244/4 |
| C00107 | DC braking: Hold time | 999.000 | s | C02245/1 | C02245/2 | C02245/3 | C02245/4 |

| Code | Name | Lenze setting | | 1st value | 2nd value | 3rd value | 4th value |
|---|---|---|---|---|---|---|---|
| C00120 | Setting of motor overload (I²xt) | 100.00 | % | C02246/1 | C02246/2 | C02246/3 | C02246/4 |
| C00234 | Oscillation damping influence | 5.00 | % | C02249/1 | C02249/2 | C02249/3 | C02249/4 |
| C00235 | Oscillation damping filter time | 32 | ms | C02250/1 | C02250/2 | C02250/3 | C02250/4 |
| C00236 | Oscillation damping field weakening | 14 | | C02251/1 | C02251/2 | C02251/3 | C02251/4 |
| C00254 | Kp position controller | 5.00 | 1/s | C02252/1 | C02252/2 | C02252/3 | C02252/4 |
| C00273 | Motor moment of inertia | 0.00 | kg cm^2 | C02256/1 | C02256/2 | C02256/3 | C02256/4 |
| C00495 | Speed sensor selection | 0 | | C02260/1 | C02260/2 | C02260/3 | C02260/4 |
| C00576 | SC: Field feedforward control | 200 | % | C02261/1 | C02261/2 | C02261/3 | C02261/4 |
| C00577 | SC: Vp field weakening controller | 0.0010 | | C02262/1 | C02262/2 | C02262/3 | C02262/4 |
| C00578 | SC: Tn field weakening controller | 20.0 | ms | C02263/1 | C02263/2 | C02263/3 | C02263/4 |
| C00653/1 | Sensibility - Setpoint feedforward control | 0 | | C02264/1 | C02264/2 | C02264/3 | C02264/4 |
| C00905 | Motor phase direction of rotation | 0 | | C02272/1 | C02272/2 | C02272/3 | C02272/4 |
| C00909/1 | Max. pos. speed | 120.00 | % | C02273/1 | C02273/2 | C02273/3 | C02273/4 |
| C00909/2 | Max. neg. speed | 120.00 | % | C02273/5 | C02273/6 | C02273/7 | C02273/8 |
| C00910/1 | Max. pos. output frequency | 1000 | Hz | C02274/1 | C02274/2 | C02274/3 | C02274/4 |
| C00910/2 | Max. neg. output frequency | 1000 | Hz | C02274/5 | C02274/6 | C02274/7 | C02274/8 |
| C00915 | Motor cable length | 5.0 | m | C02275/1 | C02275/2 | C02275/3 | C02275/4 |
| C00916 | Motor cable cross-section | 6.00 | mm^2 | C02276/1 | C02276/2 | C02276/3 | C02276/4 |
| C00938 | PSM: Maximum motor current field weakening | 30.00 | % | C02278/1 | C02278/2 | C02278/3 | C02278/4 |
| C00939 | Ultimate motor current | 3000.0 | A | C02279/1 | C02279/2 | C02279/3 | C02279/4 |
| C00965 | Max. motor speed | 60000 | rpm | C02280/1 | C02280/2 | C02280/3 | C02280/4 |
| C00966 | VFC: Time const. slip comp. | 100 | ms | C02281/1 | C02281/2 | C02281/3 | C02281/4 |
| C00971/1 | VFC: Controller limitation V/f +encoder | 10.00 | Hz | C02284/1 | C02284/2 | C02284/3 | C02284/4 |
| C00971/2 | VFC: Slip limitation V/f +encoder | 100.00 | Hz | C02284/5 | C02284/6 | C02284/7 | C02284/8 |
| C00972 | VFC: Vp V/f +encoder | 0.100 | Hz/Hz | C02285/1 | C02285/2 | C02285/3 | C02285/4 |
| C00973 | VFC: Ti V/f +encoder | 100.0 | ms | C02286/1 | C02286/2 | C02286/3 | C02286/4 |
| C00975 | VFC-ECO: Vp CosPhi controller | 0.500 | Hz/Hz | C02287/1 | C02287/2 | C02287/3 | C02287/4 |
| C00976 | VFC-ECO: Ti CosPhi controller | 200.0 | ms | C02288/1 | C02288/2 | C02288/3 | C02288/4 |
| C00977 | VFC-ECO: Minimum voltage V/f | 20.00 | % | C02289/1 | C02289/2 | C02289/3 | C02289/4 |
| C00982 | VFC-ECO: Voltage reduction ramp | 0.8 | s | C02290/1 | C02290/2 | C02290/3 | C02290/4 |
| C00985 | SLVC: Gain of field current controller | 0.50 | % | C02291/1 | C02291/2 | C02291/3 | C02291/4 |
| C00986 | SLVC: Gain of cross current controller | 0.00 | % | C02292/1 | C02292/2 | C02292/3 | C02292/4 |
| C00987 | Inverter motor brake: nAdd | 80 | rpm | C02293/1 | C02293/2 | C02293/3 | C02293/4 |
| C00988 | Inverter motor brake: PT1 filter time | 0.0 | ms | C02294/1 | C02294/2 | C02294/3 | C02294/4 |
| C00990 | Flying restart fct.: Activate | 0 | | C02295/1 | C02295/2 | C02295/3 | C02295/4 |
| C00991 | Flying restart fct.: Process | 2 | | C02296/1 | C02296/2 | C02296/3 | C02296/4 |
| C00992 | Flying restart: Start frequency | 10 | Hz | C02297/1 | C02297/2 | C02297/3 | C02297/4 |
| C00993 | Flying restart: Integration time | 300.0 | ms | C02298/1 | C02298/2 | C02298/3 | C02298/4 |
| C00994 | Flying restart: Current | 25.00 | % | C02299/1 | C02299/2 | C02299/3 | C02299/4 |
| C00995/1 | SLPSM: Controlled accelerating current | 100.00 | % | C02300/1 | C02300/2 | C02300/3 | C02300/4 |
| C00995/2 | SLPSM: Controlled standstill current | 20.00 | % | C02300/5 | C02300/6 | C02300/7 | C02300/8 |
| C00996/1 | SLPSM: Switching speed, closed-loop control | 13.00 | % | C02301/1 | C02301/2 | C02301/3 | C02301/4 |
| C00996/2 | SLPSM: Switching speed, open-loop control | 8.00 | % | C02301/5 | C02301/6 | C02301/7 | C02301/8 |
| C00997 | SLPSM: Filter cutoff frequency | 5.00 | % | C02302/1 | C02302/2 | C02302/3 | C02302/4 |
| C00998/1 | SLPSM: Filter time rotor position | 3.0 | ms | C02303/1 | C02303/2 | C02303/3 | C02303/4 |
| C00998/2 | SLPSM: Filter time actual speed value | 5.0 | ms | C02303/5 | C02303/6 | C02303/7 | C02303/8 |
| C00999 | SLPSM: PLL gain | 400 | % | C02304/1 | C02304/2 | C02304/3 | C02304/4 |
| C01001/1 | Manual entry of motor type | 0 | | C02315/1 | C02315/2 | C02315/3 | C02315/4 |
| C02853/1 | PSM: Lss saturation characteristic | 100 | % | C02305/1 | C02305/2 | C02305/3 | C02305/4 |
| C02853/2 | PSM: Lss saturation characteristic | 100 | % | C02305/5 | C02305/6 | C02305/7 | C02305/8 |
| C02853/3 | PSM: Lss saturation characteristic | 100 | % | C02305/9 | C02305/10 | C02305/11 | C02305/12 |
| C02853/4 | PSM: Lss saturation characteristic | 100 | % | C02305/13 | C02305/14 | C02305/15 | C02305/16 |

| Code | Name | Lenze setting | | 1st value | 2nd value | 3rd value | 4th value |
|---|---|---|---|---|---|---|---|
| C02853/5 | PSM: Lss saturation characteristic | 100 | % | C02305/17 | C02305/18 | C02305/19 | C02305/20 |
| C02853/6 | PSM: Lss saturation characteristic | 100 | % | C02305/21 | C02305/22 | C02305/23 | C02305/24 |
| C02853/7 | PSM: Lss saturation characteristic | 100 | % | C02305/25 | C02305/26 | C02305/27 | C02305/28 |
| C02853/8 | PSM: Lss saturation characteristic | 100 | % | C02305/29 | C02305/30 | C02305/31 | C02305/32 |
| C02853/9 | PSM: Lss saturation characteristic | 100 | % | C02305/33 | C02305/34 | C02305/35 | C02305/36 |
| C02853/10 | PSM: Lss saturation characteristic | 100 | % | C02305/37 | C02305/38 | C02305/39 | C02305/40 |
| C02853/11 | PSM: Lss saturation characteristic | 100 | % | C02305/41 | C02305/42 | C02305/43 | C02305/44 |
| C02853/12 | PSM: Lss saturation characteristic | 100 | % | C02305/45 | C02305/46 | C02305/47 | C02305/48 |
| C02853/13 | PSM: Lss saturation characteristic | 100 | % | C02305/49 | C02305/50 | C02305/51 | C02305/52 |
| C02853/14 | PSM: Lss saturation characteristic | 100 | % | C02305/53 | C02305/54 | C02305/55 | C02305/56 |
| C02853/15 | PSM: Lss saturation characteristic | 100 | % | C02305/57 | C02305/58 | C02305/59 | C02305/60 |
| C02853/16 | PSM: Lss saturation characteristic | 100 | % | C02305/61 | C02305/62 | C02305/63 | C02305/64 |
| C02853/17 | PSM: Lss saturation characteristic | 100 | % | C02305/65 | C02305/66 | C02305/67 | C02305/68 |
| C02855 | PSM: Imax Lss saturation characteristic | 3000.0 | A | C02306/1 | C02306/2 | C02306/3 | C02306/4 |
| C02859 | PSM: Activate Ppp saturation char. | 0 | | C02307/1 | C02307/2 | C02307/3 | C02307/4 |
| C02872/1 | PLI without motion: Adaptation of time duration | 0 | | C02311/1 | C02311/2 | C02311/3 | C02311/4 |
| C02874/1 | PLI without motion | 0x0001 | | C02312/1 | C02312/2 | C02312/3 | C02312/4 |
| C02875/1 | PLI without motion: Adaptation of ident angle | 0 | ° | C02313/1 | C02313/2 | C02313/3 | C02313/4 |

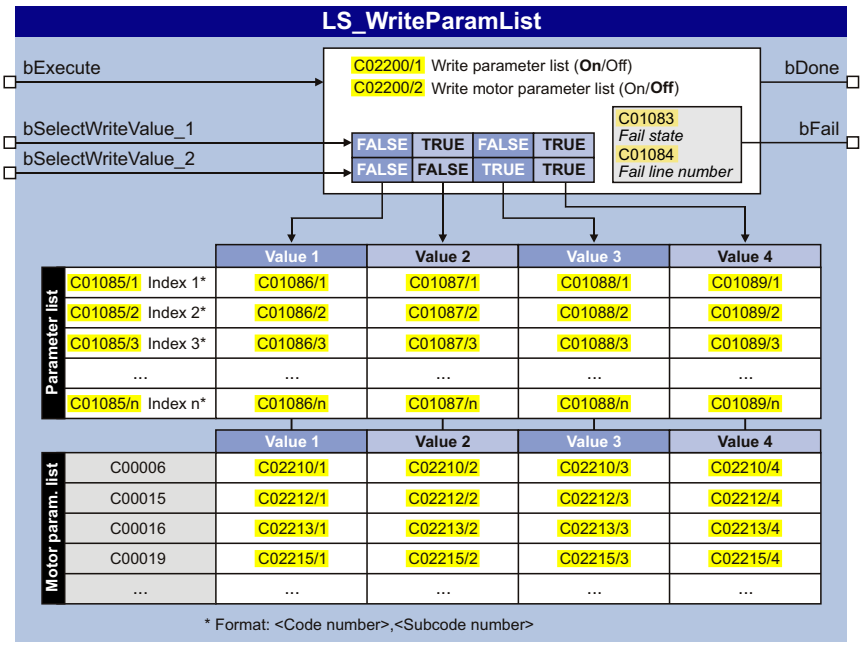
## 16.4 Internal interfaces | System block "LS_WriteParamList"

The **LS_WriteParamList** system block provides the internal interfaces for the basic "Parameter change-over" function.

### Note!

The "parameter change-over" basic function is always processed, even if the **LS_WriteParamList** system block has been removed from the interconnection by the FB Editor.

If you do not require this basic function anymore, delete the composed parameter list in order that no unwanted parameter write operations take place.




### Inputs

| Designator<br>Data type | Information/possible settings | | |
|---|---|---|---|
| bExecute<br>BOOL | FALSE↗TRUE | For Execute Mode (C01082) = "0: by Execute":<br>Activate writing of the parameter list | |
| bSelectWriteValue_1<br>bSelectWriteValue_2<br>BOOL | Binary coded selection of the value set 1 ... 4 to be used. | | |
| | *bSelectWrite Value_1* | *bSelectWrite Value_2* | |
| | FALSE | FALSE | Value set 1 |
| | TRUE | FALSE | Value set 2 |
| | FALSE | TRUE | Value set 3 |
| | TRUE | TRUE | Value set 4 |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bDone<br>BOOL | "Writing of the parameter list completed" status signal<br>• The output is automatically reset to FALSE if writing via *bExecute* is activated again. | |
| | TRUE | Writing of the parameter list successfully completed. |
| | FALSE | The FALSE status can have the following meanings:<br>1. There is no active writing of the parameter list.<br>2. Writing of the parameter list has not been completed yet.<br>3. An error has occurred (if *bFail* = TRUE). |
| bFail<br>BOOL | "Error" status | |
| | TRUE | An error has occurred (group signal).<br>• See display parameter (C01083) for details. |

# 17 Parameter reference

This chapter describes all parameters which can be used for parameterising and monitoring the inverter.

Parameters which are only available in the inverter from a certain software version onwards are marked with a corresponding note in the parameter description ("from version xx.xx.xx").

The parameter descriptions are based on software version V15.00.00

**Tip!**

For quick reference of a parameter with a certain name simply use the **index** of the online documentation. The index always contains the corresponding code in parentheses behind the name.

General information on parameter setting can be found in the chapter "Introduction: Parameterising the inverter (📖 31)".

For general information on how to read and change parameters, please see the online documentation for the »Engineer«.

## 17.1 Structure of the parameter descriptions

Each parameter is described in the Parameter list in the form of a table which consists of the following three areas:

### Table header

The table header contains the following general information:

- Parameter number (Cxxxxx)
- Parameter name (display text in the »Engineer» and keypad)
- Data type
- Parameter index in decimal and hexadecimal notation for access via a fieldbus (e.g. CAN system bus).

**Tip!**

The parameter index is calculated as follows:

- Index [dec] = 24575 - code
- Index [hex] = 0x5FFF - code

Example for code C00005:

- Index [dec] = 24575 - 5 = 24570
- Index [hex] = 0x5FFF - 0x{5} = 0x5FFA

### Table contents

The table contains further general explanations & notes on the parameter and the possible settings, which are represented in different ways depending on the parameter type:

- Parameters with read-only access
- Parameters with write access

### Table footer

The table footer contains the Parameter attributes.

### 17.1.1 Data type

The following data types are available for parameters:

| Data type | Meaning |
|---|---|
| INTEGER_16 | 16-bit value with sign |
| INTEGER_32 | 32-bit value with sign |
| UNSIGNED_8 | 8-bit value without sign |
| UNSIGNED_16 | 16-bit value without sign |
| UNSIGNED_32 | 32-bit value without sign |
| VISIBLE_STRING | String of characters of printable characters |

### 17.1.2 Parameters with read-only access

Parameters for which the "write access" attribute has not been set can only be read and not be changed by the user.

#### Description structure

| Parameter \| Name:<br>**Cxxxxx \| _____** | Data type: _____<br>Index: _____ |
|---|---|
| Description | |
| **Display area** (min. value \| unit \| max. value) | |
| | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

#### Representation in the »Engineer«

The »Engineer« displays these parameters with a grey background or, with an online connection, with a pale-yellow background:

| | C... | S | Name | Value | Unit |
|---|---|---|---|---|---|
| | 3 | 0 | Status of last device command | Successful | |

### 17.1.3 Parameters with write access

Only parameters with a check mark (☑) in front of the "write access" attribute can be changed by the user. The Lenze setting for these parameters is **printed in bold**.

- The settings can either be selected from a selection list or the values can be entered directly.
- Values outside the valid setting range are represented in red in the »Engineer«.

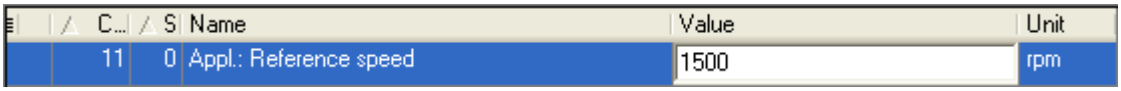
#### 17.1.3.1 Parameters with setting range

**Description structure**

| Parameter \| Name: **Cxxxxx \| _____** | | | Data type: _____ Index: _____ |
|---|---|---|---|
| Description | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**Parameter setting in the »Engineer«**

In the »Engineer«, parameters are set by entering the desired value into the input field:



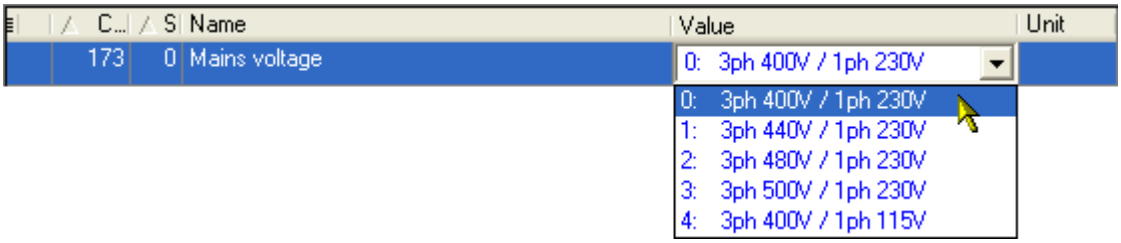

#### 17.1.3.2 Parameters with selection list

**Description structure**

| Parameter \| Name: **Cxxxxx \| _____** | | Data type: _____ Index: _____ |
|---|---|---|
| Description | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **1** | | |
| 2 | | |
| 3 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**Parameter setting in the »Engineer«**

In the »Engineer«, a list field is used for parameter setting:



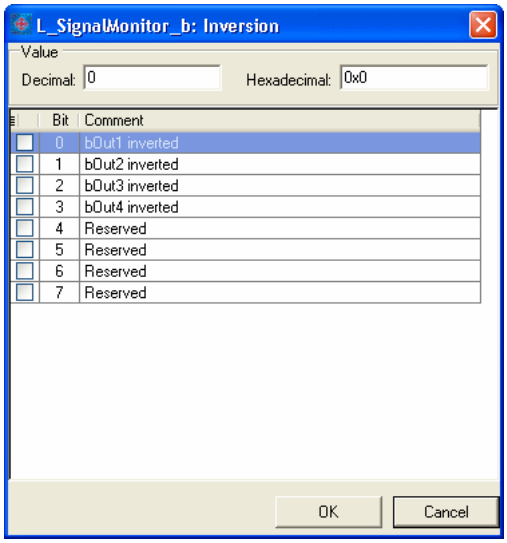

### 17.1.3.3 Parameters with bit-coded setting

**Description structure**

| Parameter \| Name: **Cxxxxx \| _____** | | Data type: _____ Index: _____ |
|---|---|---|
| Description | | |
| **Value is bit-coded:** | | |
| Bit 0 | | |
| ... | | |
| Bit 31 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**Parameter setting in the »Engineer«**

The »Engineer« uses a dialog box for parameter setting in which the individual bits can be set or reset. Alternatively, the value can be entered as a decimal or hexadecimal value:



### 17.1.3.4 Parameters with subcodes

#### Description structure

| Parameter \| Name: **Cxxxxx \| _____** | | | Data type: _____ Index: _____ |
|---|---|---|---|
| Description | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| | | | |
| **Subcodes** | **Lenze setting** | | |
| Cxxxxx/1 | | | |
| Cxxxxx/2 | | | |
| Cxxxxx/3 | | | |
| Cxxxxx/4 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

#### Parameter setting in the »Engineer«

The »Engineer« parameter list displays each subcode individually. The parameters are set as described in the previous chapters.

| | C... | S | Name | Value | Unit |
|---|---|---|---|---|---|
| | 39 | 1 | Fixed setpoint 1 | 40.00 | % |
| | 39 | 2 | Fixed setpoint 2 | 60.00 | % |
| | 39 | 3 | Fixed setpoint 3 | 80.00 | % |
| | 39 | 4 | Fixed setpoint 4 | 0.00 | % |

### 17.1.4 Parameter attributes

The table footers contain the parameter attributes:

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

| Attribute | Meaning | |
|---|---|---|
| ☑ Read access | Read access to parameter possible. | |
| ☑ Write access | Write access to parameter possible.<br>• Please also observe the following attributes: | |
| | ☑ CINH | Parameter value can only be changed when the controller is inhibited. |
| | ☑ PLC STOP | Parameter value can only be changed when the application is stopped. |
| ☑ No transfer | Parameter is **not** transferred to inverter when the command Download parameter set is executed. | |
| ☑ COM | Communication-relevant parameter<br>• This parameter is relevant for parameter data transfer via the (CAN) system bus. | |
| ☑ MOT | Motor control parameters | |

### Scaling factor

The "scaling factor" is important for parameter access via a bus system.

| Signal type | Scaling factor | Resolution | Value range |
|---|---|---|---|
| Analog (scaled) | 100 | 16 bits signed | ± 199.99 % |
| Angular velocity | 1 | 16 bits signed | ± 32767 incr./ms |
| Position in [units] | 10000 | 32 bits signed | ± 214748.3647 [units] |
| Digital (BOOL) | 1 | 8 bits unsigned | 0 ≡ FALSE; 1 ≡ TRUE |
| Time | 1000 | 16 bits unsigned | 0 ... 999.000 s |
| Selection value | 1 | 16 bits unsigned | 0 ... 65535 |

Example 1: The value "654" of the parameter C00028/1 (AIN1: input voltage) read via a bus system must be divided by the corresponding scaling factor "100" to obtain the actual display value "6.54 V".

$$\frac{\text{Read value (via bus system)}}{\text{Scaling factor}} = \text{Indicated value (Engineer)}$$

[17-1] Conversion formula for read access via bus system

Example 2: In order to set the parameter C00012 (acceleration time main setpoint) to the value "123.45 %" via a bus system, the integer value "12345" must be transferred, i.e. the value to be set must be multiplied by the corresponding scaling factor "100".

$$\text{Value to be written (via bus system)} = \text{Value to be set} \cdot \text{Scaling factor}$$

[17-2] Conversion formula for write access via bus system

### Character length

In case of parameters of "VISIBLE_STRING" data type, the character length is given in addition. This is also important for the parameter access via a bus system.

## 17.2 Parameter list

This chapter lists all parameters of the operating system in numerically ascending order.

**Note!**

The parameter descriptions are based on the software version V16.00.00.

**C00001**

| Parameter \| Name: **C00001 \| Keypad Userlevel** | | Data type: UNSIGNED_8 Index: $24574_d = 5FFE_h$ |
|---|---|---|
| From version 12.00.00 onwards, the extent of menus, submenus and codes shown in the keypad can be adapted by selecting the "userlevel".<br>• When the userlevel has been changed, the menus of the keypad will be reconfigured according to the selected user level.<br>• Independently of the set userlevel, the parameters of the attached communication module will always be displayed completely. | | |
| **Selection list** | | **Info** |
| 10 | Standard | Only the most important menus and codes are displayed on the keypad. |
| 20 | Expert | All menus and codes are displayed on the keypad. |
| 30 | Service | Only for servicing purposes (Lenze service). |
| **Subcodes** | **Lenze setting** | **Info** |
| C00001/1 | 10: Standard | Keypad UserLevel |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00002**

| Parameter \| Name: **C00002 \| Device command** | | Data type: UNSIGNED_8 Index: $24573_d = 5FFD_h$ |
|---|---|---|
| **Note:**<br>• Before switching off the supply voltage after a device command has been executed, check the successful execution of the device command via the status display in C00003!<br>• Before activating device commands by a master control, wait for the "Ready" signal of the inverter.<br>• The device will reject a write process to C00002/x if the value is >1 and issue an error message.<br>▸ Drive control (DCTRL): Device commands | | |
| **Selection list** | | |
| 0 | Off / ready | |
| 1 | On / start | |
| 4 | Action cancelled | |
| 5 | No access | |
| 6 | No access controller inhibit | |
| 20 | 20% working | |
| 40 | 40% working | |
| 60 | 60% working | |
| 80 | 80% working | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00002/1 | 0: Off / ready | Load Lenze setting<br>• All parameters are reset to the Lenze setting.<br>• Only possible when the controller is inhibited.<br>▸ Load Lenze setting |
| C00002/2 | 0: Off / ready | Reserved |
| C00002/3 | 0: Off / ready | Reserved |
| C00002/4 | 0: Off / ready | Reserved |
| C00002/5 | 0: Off / ready | Reserved |
| C00002/6 | 0: Off / ready | Load all parameter sets<br>• All parameter sets are loaded by the memory module.<br>• Only possible when the controller is inhibited.<br>▸ Load all parameter sets |
| C00002/7 | 0: Off / ready | Reserved |

| Parameter \| Name:<br>**C00002 \| Device command** | | Data type: UNSIGNED_8<br>Index: $24573_d = 5FFD_h$ |
|---|---|---|
| C00002/8 | 0: Off / ready | Reserved |
| C00002/9 | 0: Off / ready | Reserved |
| C00002/10 | 0: Off / ready | Reserved |
| C00002/11 | 0: Off / ready | Save all parameter sets<br>• All parameter sets are saved to the memory module safe against mains failure.<br>▸ Save all parameter sets |
| C00002/12 | 0: Off / ready | Reserved |
| C00002/13 | 0: Off / ready | Reserved |
| C00002/14 | 0: Off / ready | Reserved |
| C00002/15 | 0: Off / ready | Reserved |
| C00002/16 | 1: On / start | Enable inverter<br>1 ≡ Enable inverter<br>0 ≡ Inhibit inverter<br>▸ Enable/inhibit inverter |
| C00002/17 | 0: Off / ready | Activate quick stop<br>1 ≡ Activate quick stop<br>0 ≡ Deactivate quick stop<br>▸ Activate/deactivate quick stop |
| C00002/18 | 0: Off / ready | Reserved |
| C00002/19 | 0: Off / ready | Reset error<br>• After the reset (acknowledgement) of the current error, further errors may be pending which must also be reset.<br>• The status determining error is displayed in C00168.<br>• The current error is displayed in C00170.<br>▸ Error messages of the operating system |
| C00002/20 | 0: Off / ready | Reserved |
| C00002/21 | 0: Off / ready | Delete logbook<br>• All entries in the logbook of the inverter are deleted.<br>• In the logbook, information on the error history is saved.<br>▸ Logbook |
| C00002/22 | 0: Off / ready | Reserved |
| C00002/23 | 0: Off / ready | Identify motor parameters<br>• This device command serves to carry out automatic motor parameter identification.<br>• The identification method can be selected in C02867.<br>• The device command is only executed when the inverter is in the "SwitchedOn" status.<br>• In order to identify the motor parameters, the inverter must be enabled after this device command.<br>▸ Automatic motor parameter identification |
| C00002/24 | 0: Off / ready | Setting Hiperface position<br>• From version 14.00.00<br>▸ SinCos absolute value encoder with HIPERFACE® protocol |
| C00002/25 | 0: Off / ready | Identify resolver error<br>• The device command is only executed when the inverter is in the "SwitchedOn" status.<br>e ▸ Optimise resolver behaviour |

| Parameter \| Name:<br>**C00002 \| Device command** | | Data type: UNSIGNED_8<br>Index: $24573_d = 5FFD_h$ |
|---|---|---|
| C00002/26 | 0: Off / ready | CAN reset node<br>• Reinitialise "CAN on board" interface.<br>• Required when changing the baud rate, node address, or identifiers.<br>▸ "CAN on board" system bus |
| C00002/27 | 0: Off / ready | Device search function<br>• This device command serves to optically locate an inverter connected online (e.g. for maintenance work).<br>▸ Device search function |
| C00002/28 | 0: Off / ready | Check MasterPin<br>▸ Unlocking the inverter with a MasterPin |
| C00002/29 | 0: Off / ready | Set binding ID<br>▸ Device personalisation |
| C00002/30 | 0: Off / ready | Delete binding ID<br>▸ Device personalisation |
| C00002/31 | 0: Off / ready | Set password<br>▸ Password protection |
| C00002/32 | 0: Off / ready | Check password<br>▸ Password protection |
| C00002/33 | 0: Off / ready | Delete password<br>▸ Password protection |
| C00002/34 | 0: Off / ready | Pole position identification (360°)<br>• This device command is serves to carry out an automatic identification of the pole position.<br>• The device command is only executed when the inverter is in the "SwitchedOn" status.<br>• In order to identify the pole position, the inverter must be enabled after this device command.<br>▸ Pole position identification |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00003**

| Parameter \| Name: **C00003 \| Status of last device command** | | Data type: UNSIGNED_8 Index: $24572_d = 5FFC_h$ |
|---|---|---|
| Status of the device command executed last (C00002).<br><br>**Note:**<br>Before switching off the supply voltage after carrying out a device command, check whether the device command has been carried out successfully via the status display!<br>▸ Drive control (DCTRL): Device commands | | |
| **Selection list** (read only) | | **Info** |
| 0 | Successful | Device command has been executed successfully. |
| 1 | Command unknown | Device command implausible or unknown to the system. |
| 2 | Password protection | Unauthorised access for requested device command.<br>▸ Password protection |
| 3 | Time-out | Device command could not be processed in the defined time (timeout). |
| 4 | System fault | |
| 5 | Command server assigned | |
| 6 | Controller inhibit required | |
| 7 | unexpected controller inhibit | |
| 10 | Memory module binding error | ▸ Device personalisation |
| 11 | Password too short | ▸ Password protection |
| 12 | Wrong password | |
| 13 | Password already set | |
| 14 | Password not assigned | |
| 15 | Incorrect checksum | |
| 20 | Setting the hiperface position failed | |
| 30 | Incorrect pole position identification | |
| 31 | Pole position identification started without encoder | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00005**

| Parameter \| Name: **C00005 \| Application** | | Data type: UNSIGNED_16 Index: $24570_d = 5FFA_h$ |
|---|---|---|
| Selection of the technology application | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | Wiring has changed | This display appears if the FB interconnection has been changed in the application level using the FB Editor. |
| **1000** | **Actuating drive speed** | This technology application is used to solve speed-controlled drive tasks, e.g. conveying belts.<br>▸ TA "Speed actuating drive" |
| 1100 | Actuating drive speed (AC Drive Profile) | From version 13.00.00<br>This technology application is used to solve speed-controlled drive tasks, e.g. conveying belts.<br>The process data word received by the master control is interpreted as "AC Drive Profile" control word.<br>▸ TA "Actuating drive speed (AC Drive Profile)" |
| 2000 | Table positioning | This technology application is used to solve position-controlled drive tasks which are usually controlled by a higher-level control system using a fieldbus.<br>▸ TA "Table positioning" |
| 3000 | Switch-off positioning | This technology application is used to solve speed-controlled drive tasks which require a pre-switch off or stopping at certain positions, e.g. roller conveyors and conveying belts. This is implemented by connecting switch-off sensors.<br>▸ TA "Switch-off positioning" |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00006**

| Parameter \| Name: **C00006 \| Motor control** | | Data type: UNSIGNED_8 Index: $24569_d = 5FF9_h$ |
|---|---|---|
| Selection of the motor control mode<br>▸ Motor control (MCTRL): Select control mode | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 1 | SC: SM servo control | This control type is used for the servo control of a synchronous motor.<br>• The control type requires a speed feedback via an encoder mounted to the motor!<br>▸ Servo control |
| 2 | SC: Servo control ASM | This control type is used for the servo control of an asynchronous motor.<br>• The control type requires a speed feedback via an encoder mounted to the motor!<br>▸ Servo control |
| 3 | SLPSM: Sensorless PSM | This control type is used for the sensorless control of a synchronous motor.<br>▸ Sensorless control for synchronous motors |
| 4 | SLVC: Vector control | This control type is used for sensorless vector control of an asynchronous motor.<br>• The control type requires motor parameters to be set as exactly as possible!<br>▸ Sensorless vector control |

| Parameter \| Name:<br>**C00006 \| Motor control** | | Data type: UNSIGNED_8<br>Index: $24569_d = 5FF9_h$ |
|---|---|---|
| **6** | **VFCplus: V/f linear** | This control type is used for the speed control of an asynchronous motor via a linear V/f characteristic and is the simplest control type.<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>▸ V/f characteristic control |
| 7 | VFCplus: V/f linear + encoder | This control type is used for speed control of an asynchronous motor via a linear V/f characteristic.<br>• The control type requires a speed feedback via an encoder mounted to the motor!<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>▸ V/f control |
| 8 | VFCplus: V/f quadr | This control type is used for speed control of an asynchronous motor via a square-law V/f characteristic.<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>▸ V/f characteristic control |
| 9 | VFCplus: V/f quadr + encoder | This control type is used for speed control of an asynchronous motor via a square-law V/f characteristic.<br>• The control type requires a speed feedback via an encoder mounted to the motor!<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>▸ V/f control |
| 10 | VFCplus: V/f definable | This type of control is used for the speed control of an asynchronous motor via a user-definable characteristic with several interpolation points.<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>▸ V/f characteristic control |
| 11 | VFCplusEco: V/f energy-saving | This control type is used for energy-saving speed control of an asynchronous motor via a linear V/f characteristic.<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>• Predestinated application areas of this control type are materials handling technology and pump and fan systems.<br>▸ V/f characteristic control, energy-saving |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00007**

| Parameter \| Name: **C00007 \| Control mode** | | Data type: UNSIGNED_16 Index: $24568_d = 5FF8_h$ |
|---|---|---|
| With this parameter the control mode for the technology application selected under C00005 is defined, i.e. how the inputs and outputs of the technology application are connected to the I/Os of the inverter.<br>• How the inputs and outputs are connected in the individual control modes is described in the corresponding technology application:<br>• "Actuating drive speed" TA<br>• TA "actuating drive speed (AC Drive Profile)"<br>• TA "Table positioning"<br>• "Switch-off positioning" TA | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | Wiring has changed | This is displayed when the FB interconnection has been changed in the I/O level via the FB Editor. |
| **10** | **Terminals 0** | The technology application is controlled via the digital and analog input terminals of the inverter.<br>• For a short overview of the preconfigured terminal assignment see the following section "Terminal assignment of the control modes 10 ... 16". |
| 12 | Terminals 2 | |
| 14 | Terminals 11 | |
| 16 | Terminal 16 | |
| 20 | Keypad | The technology application is controlled via the keypad. |
| 21 | PC | The technology application is controlled via the "Free parameters" of the inverter (PC control). |
| 30 | CAN | The technology application is controlled by means of CAN-PDOs via the system bus "CAN on board".<br>▸ "CAN on board" system bus |
| 40 | MCI | The technology application is controlled by means of MCI-PDOs via the MCI-interface of an attached communication module (e.g. PROFIBUS). |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

### Terminal assignment of the control modes 10 ... 16

Note: The following tables only list the connected inputs/outputs.

| "Actuating drive speed" TA | | | | |
|---|---|---|---|---|
| **Input/output** | **10: Terminals 0** | **12: Terminals 2** | **14: Terminals 11** | **16: Terminals 16** |
| **RFR** | Controller enable / Reset of error message | | | |
| **DI1** | Fixed setpoint 1/3 | | Change of direction of rotation | Fixed setpoint 1/3 |
| **DI2** | Fixed setpoint 2/3 | | Activate manual DC-injection braking (DCB) | Fixed setpoint 2/3 |
| **DI3** | Activate manual DC-injection braking (DCB) | Quick stop | Motor potentiometer: Increase speed | CW rotation quick stop |
| **DI4** | Change of direction of rotation | | Motor potentiometer: Decrease speed | CCW rotation quick stop |
| **A1U, A1I** | Main speed setpoint (10 V ≡ 100 % reference speed) | | | |
| **O1U, O1I** | Actual speed value (10 V ≡ 100 % reference speed) | | | |
| **DO1** | Status "Drive is ready" | | | |
| **Relay output** | Status "Error is pending" | | | |

| TA "actuating drive speed (AC Drive Profile)" | | | | |
|---|---|---|---|---|
| **Input/output** | **10: Terminals 0** | **12: Terminals 2** | **14: Terminals 11** | **16: Terminals 16** |
| **RFR** | Controller enable / Reset of error message | | | |
| **DI1** | Fixed setpoint 1/3 | | Change of direction of rotation | Fixed setpoint 1/3 |
| **DI2** | Fixed setpoint 2/3 | | Activate manual DC-injection braking (DCB) | Fixed setpoint 2/3 |
| **DI3** | Activate manual DC-injection braking (DCB) | Quick stop | Motor potentiometer: Increase speed | CW rotation quick stop |
| **DI4** | Change of direction of rotation | | Motor potentiometer: Decrease speed | CCW rotation quick stop |
| **A1U, A1I** | Local speed/torque setpoint (speed mode: 10 V ≡ 100 % reference speed; torque mode: 10 V ≡ 100 % $M_{max}$) | | | |
| **O1U, O1I** | Actual speed value (10 V ≡ 100 % reference speed) | | | |
| **DO1** | Status "Drive is ready" | | | |
| **Relay output** | Status "Error is pending" | | | |

| TA "Table positioning" | | | | |
|---|---|---|---|---|
| **Input/output** | **10: Terminals 0** | **12: Terminals 2** | **14: Terminals 11** | **16: Terminals 16** |
| **RFR** | Controller enable / Reset of error message | | | |
| **DI1** | - | - | Positive limit switch | - |
| **DI2** | - | - | Negative limit switch | - |
| **DI3** | Positive limit switch | - | - | Manual jog in positive direction |
| **DI4** | Negative limit switch | Positioning profile, array bit (valency 4) | | Manual jog in negative direction |
| **DI5** | Positioning profile, array bit (valency 2) | | | |
| **DI6** | Positioning profile, array bit (valency 1) | | | |
| **DI7** | Start of profile generation | | | |
| **A1U, A1I** | Main speed setpoint (10 V ≡ 100 % reference speed) | | | |
| **O1U, O1I** | Actual speed value (10 V ≡ 100 % reference speed) | | | |
| **O2U, O2I** | Current motor current (10 V ≡ 100 % $I_{max_mot}$) | | | |
| **DO1** | Status "Drive is ready" | | | |
| **DO2** | Status "Target position (actual value) is in the target window" | | | |
| **DO3** | Status "Home position is known" | | | |
| **BD1, BD2** | Control of the holding brake | | | |
| **Relay output** | Status "Error is pending" | | | |

| "Switch-off positioning" TA | | | | |
|---|---|---|---|---|
| **Input/output** | **10: Terminals 0** | **12: Terminals 2** | **14: Terminals 11** | **16: Terminals 16** |
| **RFR** | Controller enable | Controller enable / Reset of error message | | |
| **DI1** | Fixed setpoint 1/3 | Stop function 1 | | Fixed setpoint 1/3 |
| **DI2** | Fixed setpoint 2/3 | Stop function 2 | Selection: Pre-switch off 1 | Fixed setpoint 2/3 |
| **DI3** | Reset error message | CW rotation quick stop<br>Selection: Switch-off position 1 | | CW rotation quick stop |
| **DI4** | Change of direction of rotation | CCW rotation quick stop<br>Selection: Switch-off position 2 | | CCW rotation quick stop |
| **DI5** | - | - | Stop function 2 | - |
| **DI6** | - | - | Selection: Pre-switch off 2 | - |
| **A1U** | Main speed setpoint (10 V ≡ 100 % reference speed) | | | |
| **BD1, BD2** | Control of the holding brake | | | |

**C00008**

| Parameter \| Name: **C00008 \| Original application\|control source** | | Data type: UNSIGNED_16 Index: $24567_d = 5FF7_h$ |
|---|---|---|
| Display of the originally selected technology application and the originally selected control mode.<br>• This parameter shows the selection that was set with C00005 and C00007 before a change in the I/O level or the application level was carried out.<br>• For diagnostic purposes, this display serves to determine whether there is a standard interconnection in the inverter or a change carried out by the user. | | |
| **Selection list** (read only) | | **Info** |
| 0 | Free\|Free | Application: Interconnection has been changed.<br>I/O level: Interconnection has been changed. |
| 10 | Free\|Terminal0 | Application: Interconnection has been changed.<br>I/O level: "Terminals 0" control mode |
| 12 | Free\|Terminal2 | Application: Interconnection has been changed.<br>I/O level: "Terminal 2" control mode |
| 14 | Free\|Terminal11 | Application: Interconnection has been changed.<br>I/O level: "Terminal 11" control mode |
| 16 | Free\|Terminal 16 | Application: Interconnection has been changed.<br>I/O level: "Terminal 16" control mode |
| 20 | Free\|Keypad | Application: Interconnection has been changed.<br>I/O level: "Keypad" control mode |
| 21 | Free\|PC | Application: Interconnection has been changed.<br>I/O level: "PC" control mode |
| 30 | Free\|CAN | Application: Interconnection has been changed.<br>I/O level: "CAN" control mode |
| 40 | Free\|MCI | Application: Interconnection has been changed.<br>I/O level: "MCI" control mode |
| 1000 | Speed\|Free | Application: Actuating drive speed<br>I/O level: Interconnection has been changed. |
| 1010 | Speed\|Terminal0 | Application: Actuating drive speed<br>I/O level: "Terminals 0" control mode |
| 1012 | Speed\|Terminal2 | Application: Actuating drive speed<br>I/O level: "Terminal 2" control mode |
| 1014 | Speed\|Terminal11 | Application: Actuating drive speed<br>I/O level: "Terminal 11" control mode |
| 1016 | Speed\|Terminal16 | Application: Actuating drive speed<br>I/O level: "Terminal 16" control mode |
| 1020 | Speed\|Keypad | Application: Actuating drive speed<br>I/O level: "Keypad" control mode |
| 1021 | Speed\|PC | Application: Actuating drive speed<br>I/O level: "PC" control mode |
| 1030 | Speed\|CAN | Application: Actuating drive speed<br>I/O level: "CAN" control mode |
| 1040 | Speed\|MCI | Application: Actuating drive speed<br>I/O level: "MCI" control mode |
| 1100 | SpeedACdrive\|Free | Application: Actuating drive speed (AC Drive Profile)<br>I/O level: Interconnection has been changed. |
| 1110 | SpeedACdrive\|Terminal0 | Application: Actuating drive speed (AC Drive Profile)<br>I/O level: "Terminals 0" control mode |
| 1112 | SpeedACdrive\|Klemme2 | Application: Actuating drive speed (AC Drive Profile)<br>I/O level: "Terminal 2" control mode |
| 1114 | SpeedACdrive\|Klemme11 | Application: Actuating drive speed (AC Drive Profile)<br>I/O level: "Terminal 11" control mode |

| Parameter \| Name:<br>**C00008 \| Original application\|control source** | | Data type: UNSIGNED_16<br>Index: $24567_d = 5FF7_h$ |
|---|---|---|
| 1116 | SpeedACdrive\|Klemme16 | Application: Actuating drive speed (AC Drive Profile)<br>I/O level: "Terminal 16" control mode |
| 1120 | SpeedACdrive\|Keypad | Application: Actuating drive speed (AC Drive Profile)<br>I/O level: "Keypad" control mode |
| 1121 | SpeedACdrive\|PC | Application: Actuating drive speed (AC Drive Profile)<br>I/O level: "PC" control mode |
| 1130 | SpeedACdrive\|CAN | Application: Actuating drive speed (AC Drive Profile)<br>I/O level: "CAN" control mode |
| 1140 | SpeedACdrive\|MCI | Application: Actuating drive speed (AC Drive Profile)<br>I/O level: "MCI" control mode |
| 2000 | TabPos\|Free | Application: Table positioning<br>I/O level: Interconnection has been changed. |
| 2010 | TabPos\|Terminal0 | Application: Table positioning<br>I/O level: "Terminals 0" control mode |
| 2012 | TabPos\|Terminal2 | Application: Table positioning<br>I/O level: "Terminal 2" control mode |
| 2014 | TabPos\|Terminal11 | Application: Table positioning<br>I/O level: "Terminal 11" control mode |
| 2016 | TabPos\|Terminal16 | Application: Table positioning<br>I/O level: "Terminal 16" control mode |
| 2020 | TabPos\|Keypad | Application: Table positioning<br>I/O level: "Keypad" control mode |
| 2021 | TabPos\|PC | Application: Table positioning<br>I/O level: "PC" control mode |
| 2030 | TabPos\|CAN | Application: Table positioning<br>I/O level: "CAN" control mode |
| 2040 | TabPos\|MCI | Application: Table positioning<br>I/O level: "MCI" control mode |
| 3000 | SwitchPos\|Free | Application: Switch-off positioning<br>I/O level: Interconnection has been changed. |
| 3010 | SwitchPos\|Terminal0 | Application: Switch-off positioning<br>I/O level: "Terminals 0" control mode |
| 3012 | SwitchPos\|Terminal 2 | Application: Switch-off positioning<br>I/O level: "Terminal 2" control mode |
| 3014 | SwitchPos\|Terminal 11 | Application: Switch-off positioning<br>I/O level: "Terminal 11" control mode |
| 3016 | SwitchPos\|Terminal 16 | Application: Switch-off positioning<br>I/O level: "Terminal 16" control mode |
| 3020 | SwitchPos\|Keypad | Application: Switch-off positioning<br>I/O level: "Keypad" control mode |
| 3021 | SwitchPos\|PC | Application: Switch-off positioning<br>I/O level: "PC" control mode |
| 3030 | SwitchPos\|CAN | Application: Switch-off positioning<br>I/O level: "CAN" control mode |
| 3040 | SwitchPos\|MCI | Application: Switch-off positioning<br>I/O level: "MCI" control mode |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00010**

| Parameter \| Name: **C00010 \| AIN1: Characteristic** | | | Data type: INTEGER_16 Index: $24565_d = 5FF5_h$ |
|---|---|---|---|
| ▸ Analog terminals: Signal adaptation via characteristic | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00010/1 | 0.00 % | | AIN1: (+y0) = min |
| C00010/2 | 0.00 % | | AIN1: (+x0) = Dead band |
| C00010/3 | 0.00 % | | AIN1: (-y0) = (-min) |
| C00010/4 | 0.00 % | | AIN1: (-x0) = (-Dead band) |
| C00010/5 | 100.00 % | | AIN1: (+ymax) |
| C00010/6 | 100.00 % | | AIN1: (+xmax) |
| C00010/7 | 100.00 % | | AIN1: (-ymax) |
| C00010/8 | 100.00 % | | AIN1: (-xmax) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00011**

| Parameter \| Name: **C00011 \| Appl.: Reference speed** | | | Data type: UNSIGNED_16 Index: $24564_d = 5FF4_h$ |
|---|---|---|---|
| Setting the reference speed<br>• In the inverter, all speed-related signals are processed to one reference variable in percent.<br>• Set a reference speed here that corresponds to 100 %.<br>• The frequency that corresponds to the set reference speed is displayed in C00059.<br><br>**Note:**<br>This is not a maximum limitation!<br>All values in percent in the inverter may be in a range of 0 ... 199.99 %. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 50 | rpm | 60000 | **1500 rpm** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00012**

| Parameter \| Name: **C00012 \| Accel. time - main setpoint** | | | Data type: UNSIGNED_32 Index: $24563_d = 5FF3_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Acceleration time of the ramp generator for the main speed setpoint<br>• Generally, this ramp generator is used for all speed-controlled technology applications. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **2.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00013**

| Parameter \| Name: **C00013 \| Decel. time - main setpoint** | | | Data type: UNSIGNED_32 Index: $24562_d = 5FF2_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Deceleration time of the ramp generator for the main speed setpoint<br>• Generally, this ramp generator is used for all speed-controlled technology applications. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **2.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00015**

| Parameter \| Name: **C00015 \| VFC: V/f base frequency** | | | Data type: UNSIGNED_16 Index: $24560_d = 5FF0_h$ |
|---|---|---|---|
| V/f base frequency for V/f characteristic control (VFCplus) and V/f control (VFCplus+encoder)<br>• The motor voltage increases linearly with the frequency until the base frequency is reached. From this value on, the motor voltage remains constant, the speed increases and the maximum torque decreases.<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 7.5 | Hz | 2600.0 | **50.0 Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00016**

| Parameter \| Name: **C00016 \| VFC: Vmin boost** | | | Data type: UNSIGNED_16 Index: $24559_d = 5FEF_h$ |
|---|---|---|---|
| Boost of the V/f voltage characteristic in the range of small speeds or frequencies with V/f characteristic control (VFCplus) and V/f control (VFCplus+encoder)<br>• This may increase the starting torque.<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well.<br>▸ Motor control (MCTRL): Setting the Vmin boost | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00018**

| Parameter \| Name: **C00018 \| Switching frequency** | Data type: UNSIGNED_8 Index: $24557_d = 5FED_h$ |
|---|---|

Selection of the pulse width modulated switching frequency transferred from the inverter to the motor
- Select between an ideal setting for the drive which provides smooth running, and an optimal setting with regard to the inverter which keeps its losses to a minimum (min. Pv).
- Both possibilities offer fixed and variable switching frequencies.
- When a variable switching frequency is selected, the switching frequency may change as a function of the load and rotational frequency.

▸ Selection of switching frequency

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| 1 | 4 kHz var./drive-optimised |
| **2** | **8 kHz var./drive-optimised** |
| 3 | 16 kHz var./drive-optimised |
| 5 | 2 kHz constant/drive-optimised |
| 6 | 4 kHz constant/drive-optimised |
| 7 | 8 kHz constant/drive-optimised |
| 8 | 16 kHz constant/drive-optimised |
| 11 | 4 kHz var./min. Pv |
| 12 | 8 kHz var./min. Pv |
| 13 | 16 kHz var./min. Pv |
| 15 | 2 kHz constant/min. Pv |
| 16 | 4 kHz constant/min. Pv |
| 17 | 8 kHz constant/min. Pv |
| 18 | 16 kHz constant/min. Pv |
| 21 | 8 kHz var./drive-opt./4 kHz min |
| 22 | 16 kHz var./drive-opt./4 kHz min |
| 23 | 16 kHz var./drive-opt./8 kHz min |
| 31 | 8 kHz var./min. Pv/4 kHz min |
| 32 | 16 kHz var./min. Pv/4 kHz min |
| 33 | 16 kHz var./min. Pv/8 kHz min |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00019**

| Parameter \| Name: **C00019 \| Auto DCB: Threshold** | Data type: UNSIGNED_16 Index: $24556_d = 5FEC_h$ |
|---|---|

Setpoint speed threshold for automatic DC injection braking
- For speed setpoints with values below the thresholds a DC current is injected or the motor is not supplied with current, depending on the setting.

▸ DC-injection braking

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0 | rpm | 60000 | **3 rpm** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00020**

| Parameter \| Name: **C00020 \| AIN2: Characteristic** | | | Data type: INTEGER_16 Index: $24555_d = 5FEB_h$ |
|---|---|---|---|
| ▸ Analog terminals: Signal adaptation via characteristic | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00020/1 | 0.00 % | | AIN2: (+y0) = min |
| C00020/2 | 0.00 % | | AIN2: (+x0) = Dead band |
| C00020/3 | 0.00 % | | AIN2: (-y0) = (-min) |
| C00020/4 | 0.00 % | | AIN2: (-x0) = (-Dead band) |
| C00020/5 | 100.00 % | | AIN2: (+ymax) |
| C00020/6 | 100.00 % | | AIN2: (+xmax) |
| C00020/7 | 100.00 % | | AIN2: (-ymax) |
| C00020/8 | 100.00 % | | AIN2: (-xmax) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00021**

| Parameter \| Name: **C00021 \| Slip comp.** | | | Data type: INTEGER_16 Index: $24554_d = 5FEA_h$ |
|---|---|---|---|
| Slip compensation for V/f characteristic control (VFCplus) and sensorless vector control (SLVC)<br>• A higher slip compensation results in a higher increase in frequency and voltage when the machine is under load.<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well.<br>▸ Motor control (MCTRL): Optimising the operational performance by slip compensation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -100.00 | % | 100.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00022**

| Parameter \| Name: **C00022 \| Imax in motor mode** | | | Data type: UNSIGNED_16 Index: $24553_d = 5FE9_h$ |
|---|---|---|---|
| Maximum current in motor mode for all motor control modes<br>**Note:**<br>The parameter has a value range depending on the device power. The device automatically limits the entered value to the corresponding value range. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | A | 655.35 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00023**

| Parameter \| Name: **C00023 \| Imax in generator mode** | | | Data type: INTEGER_16 Index: $24552_d = 5FE8_h$ |
|---|---|---|---|
| Maximum current in generator mode for all motor control modes<br>• 100 % ≡ Imax in motor mode (C00022) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **100.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00024**

| Parameter \| Name: **C00024 \| LS_DriveInterface: bNActCompare** | | | Data type: INTEGER_16 Index: $24551_d = 5FE7_h$ |
|---|---|---|---|
| Threshold for the actual speed comparison<br>• This parameter serves to set a threshold that is compared with the actual speed value.<br>• If the value falls below this threshold, the *bNactCompare* output sets the LS_DriveInterface system block to TRUE.<br>• Switching hysteresis = +1 %<br>CAUTION:<br>There are errors in the evaluation if the sum of 199.98 % is exceeded in case of the set value combinations. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 199.99 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00025**

| Parameter \| Name: **C00025 \| LS_DriveInterface: bNActHysterese** | | | Data type: INTEGER_16 Index: $24550_d = 5FE6_h$ |
|---|---|---|---|
| From version 16.00.00<br>Selection of the switching hysteresis for the actual speed value comparison in C00024.<br>CAUTION:<br>There are errors in the evaluation if the sum of 199.98 % is exceeded in case of the set value combinations. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00025/1 | 1.00 % | | LS_DriveInterface: bNActHysterese |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00026**

| Parameter \| Name: **C00026 \| AINx: Offset** | | | Data type: INTEGER_16 Index: $24549_d = 5FE5_h$ |
|---|---|---|---|
| Offset for analog inputs<br>▸ Analog terminals | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00026/1 | 0.00 % | | AIN1: Offset |
| C00026/2 | 0.00 % | | AIN2: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00027**

| Parameter \| Name: **C00027 \| AINx: Gain** | | | Data type: INTEGER_32 Index: $24548_d = 5FE4_h$ |
|---|---|---|---|
| Gain for analog inputs<br>▸ Analog terminals | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -100.0000 | | 100.0000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00027/1 | 1.0000 | | AIN1: Gain |
| C00027/2 | 1.0000 | | AIN2: Gain |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C00028**

| Parameter \| Name: **C00028 \| AINx: Input voltage** | | | Data type: INTEGER_16 Index: $24547_d = 5FE3_h$ |
|---|---|---|---|
| Display of the input voltage at the analog inputs<br>▸ Analog terminals | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -10.00 | V | 10.00 | |
| **Subcodes** | | | **Info** |
| C00028/1 | | | AIN1: Input voltage |
| C00028/2 | | | AIN2: Input voltage |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00029**

| Parameter \| Name: **C00029 \| AINx: Input current** | | | Data type: INTEGER_16 Index: $24546_d = 5FE2_h$ |
|---|---|---|---|
| Display of the input current at the analog inputs<br>• When the corresponding analog input is configured for current measurement (C00034/x = 1 or 2).<br>• When C00034/x is set = 2 (4 ... 20 mA), 0 ... 16 mA is displayed.<br>▸ Analog terminals | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | mA | 20.00 | |
| **Subcodes** | | | **Info** |
| C00029/1 | | | AIN1: Input current |
| C00029/2 | | | AIN2: Input current |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00030**

| Parameter \| Name: **C00030 \| LS_DFOut: Const.** | | | Data type: INTEGER_16 Index: $24545_d = 5FE1_h$ |
|---|---|---|---|
| From version 12.00.00<br>Settings for the digital frequency output (multi-encoder interface X8)<br>▸ Digital frequency coupling | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 16 | | 16384 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00030/1 | 2048 | | LS_DFOut: number of increments<br>• The number of increments determines after how many output increments a zero pulse will be generated. Each zero pulse defines a covered "revolution" of the rotary transducer simulated by the digital frequency output. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00033**

| Parameter \| Name: **C00033 \| AINx: Output value** | | | Data type: INTEGER_16 Index: $24542_d = 5FDE_h$ |
|---|---|---|---|
| Display of the output value in percent of the analog input amplifier<br>• 100 % ≡ 16384 ≡ +10 V / +20 mA<br>▸ Analog terminals | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00033/1 | | | AIN1: Output value |
| C00033/2 | | | AIN2: Output value |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00034**

| Parameter \| Name: **C00034 \| AINx: Configuration** | | Data type: UNSIGNED_8 Index: $24541_d = 5FDD_h$ |
|---|---|---|
| Configuration of the analog inputs for current or voltage measurement<br>▸ Analog terminals | | |
| **Selection list** | | **Info** |
| 0 | -10V..+10V | Input signal is the voltage signal -10 V ... +10 V<br>• -10 V ... +10 V ≡ -100 % ... +100 % |
| 1 | 0mA..20mA | Input signal is the current signal 0 mA ... 20 mA<br>• 0 mA ... 20 mA ≡ 0 % ... +100 % |
| 2 | 4mA..20mA | Input signal is the current signal 4 mA ... 20 mA<br>• 4 mA ... 20 mA ≡ 0 % ... +100 %<br>• The current loop is monitored for open circuit (I < 4 mA) by the device. |
| **Subcodes** | **Lenze setting** | **Info** |
| C00034/1 | 0: -10V..+10V | AIN1: Config. |
| C00034/2 | 0: -10V..+10V | AIN2: Config. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00036**

| Parameter \| Name: **C00036 \| DC braking: Current** | | | Data type: INTEGER_16 Index: $24539_d = 5FDB_h$ |
|---|---|---|---|
| Braking current in [%] based on the rated device current (C00098)<br>• 100 % ≡ C00098<br>▸ DC-injection braking | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 200.00 | **50.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00039**

| Parameter \| Name: **C00039 \| Fixed setpoint x (L_NSet_1 n-Fix)** | | | Data type: INTEGER_16 Index: $24536_d = 5FD8_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Fixed speed setpoints (JOG values) for the setpoint generator<br>• 100 % ≡ C00011 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00039/1 | 40.00 % | | Fixed setpoint 1 |
| C00039/2 | 60.00 % | | Fixed setpoint 2 |
| C00039/3 | 80.00 % | | Fixed setpoint 3 |
| C00039/4 | 0.00 % | | Fixed setpoint 4 |
| C00039/5 | 0.00 % | | Fixed setpoint 5 |
| C00039/6 | 0.00 % | | Fixed setpoint 6 |
| C00039/7 | 0.00 % | | Fixed setpoint 7 |
| C00039/8 | 0.00 % | | Fixed setpoint 8 |
| C00039/9 | 0.00 % | | Fixed setpoint 9 |
| C00039/10 | 0.00 % | | Fixed setpoint 10 |
| C00039/11 | 0.00 % | | Fixed setpoint 11 |
| C00039/12 | 0.00 % | | Fixed setpoint 12 |
| C00039/13 | 0.00 % | | Fixed setpoint 13 |
| C00039/14 | 0.00 % | | Fixed setpoint 14 |
| C00039/15 | 0.00 % | | Fixed setpoint 15 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00050**

| Parameter \| Name: **C00050 \| MCTRL: Speed setpoint** | | | Data type: INTEGER_32 Index: $24525_d = 5FCD_h$ |
|---|---|---|---|
| Display of the speed setpoint at the speed setpoint input of the motor control | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -120000 | rpm | 120000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00051**

| Parameter \| Name: **C00051 \| MCTRL: Actual speed value** | | | Data type: INTEGER_32 Index: $24524_d$ = $5FCC_h$ |
|---|---|---|---|
| Display of the actual speed value of the motor shaft<br><br>**Note:**<br>The displayed value only corresponds to the real actual speed value of the motor shaft if an encoder is connected to the motor and the evaluation of the feedback signal has been set correctly ("Closed loop" operation).<br>In case of operation without speed feedback, the signal is calculated from the motor control and thus may not correspond to the real actual speed. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -120000 | rpm | 120000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00052**

| Parameter \| Name: **C00052 \| Motor voltage** | | | Data type: UNSIGNED_16 Index: $24523_d$ = $5FCB_h$ |
|---|---|---|---|
| Display of the current motor voltage/output voltage of the inverter | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | V | 1000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00053**

| Parameter \| Name: **C00053 \| DC-bus voltage** | | | Data type: UNSIGNED_16 Index: $24522_d$ = $5FCA_h$ |
|---|---|---|---|
| Display of the current DC-bus voltage | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | V | 1000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00054**

| Parameter \| Name: **C00054 \| Motor current** | | | Data type: UNSIGNED_16 Index: $24521_d$ = $5FC9_h$ |
|---|---|---|---|
| Display of the current motor current/output current of the inverter | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | A | 300.00 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00055**

| Parameter \| Name: **C00055 \| Actual values** | | | Data type: INTEGER_16 Index: $24520_d = 5FC8_h$ |
|---|---|---|---|
| Actual value of the encoder/feedback system<br>**Note:**<br>When a single-track feedback has been selected for the HTL encoder (C00115 = 1 or 3), the sign of the actual speed value is created from the sign of the speed setpoint. In C00055/1 und C00055/2, a positive speed is always displayed.<br>▸ Encoder/feedback system | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -32767 | rpm | 32767 | |
| **Subcodes** | | | **Info** |
| C00055/1 | | | Actual value - HTL encoder FreqIn12 |
| C00055/2 | | | Actual value - HTL encoder FreqIn67 |
| C00055/3 | | | Actual value - multi-encoder |
| C00055/4 | | | Actual value - resolver |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00056**

| Parameter \| Name: **C00056 \| Torque** | | | Data type: INTEGER_32 Index: $24519_d = 5FC7_h$ |
|---|---|---|---|
| Display of the current torque | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -65000.00 | Nm | 65000.00 | |
| **Subcodes** | | | **Info** |
| C00056/1 | | | Torque demand<br>• Only in case of sensorless vector control (SLVC) and servo control (SC). |
| C00056/2 | | | Actual torque value<br>• Estimated actual torque for all motor control modes. |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00057**

| Parameter \| Name: **C00057 \| Maximum torque** | | | Data type: UNSIGNED_32 Index: $24518_d = 5FC6_h$ |
|---|---|---|---|
| Display of the maximum torque to be generated by the motor<br>• The maximum torque to be generated by the motor depends on various factors, e.g. on Imax in motor mode (C00022) and the motor type used. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | Nm | 65000.00 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00058**

| Parameter \| Name: **C00058 \| Output frequency** | | | Data type: INTEGER_32 Index: $24517_d = 5FC5_h$ |
|---|---|---|---|
| Display of the current output frequency | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -1300.00 | Hz | 1300.00 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00059**

| Parameter \| Name: **C00059 \| Appl.: Reference frequency C11** | | | Data type: UNSIGNED_32 Index: $24516_d = 5FC4_h$ |
|---|---|---|---|
| Display of the field frequency which corresponds to the reference speed set in C00011. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | Hz | 1300.00 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00060**

| Parameter \| Name: **C00060 \| Motor rotor position** | | | Data type: UNSIGNED_16 Index: $24515_d = 5FC3_h$ |
|---|---|---|---|
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 2047 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00061**

| Parameter \| Name: **C00061 \| Heatsink temperature** | | | Data type: INTEGER_16 Index: $24514_d = 5FC2_h$ |
|---|---|---|---|
| Display of the current heatsink temperature | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -50 | °C | 150 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00062**

| Parameter \| Name: **C00062 \| Temperature inside the controller** | | | Data type: INTEGER_16 Index: $24513_d = 5FC1_h$ |
|---|---|---|---|
| From version 02.00.00<br>Display of the current temperature inside the controller | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -200 | °C | 200 | |
| **Subcodes** | | | **Info** |
| C00062/1 | | | Interior temperature of CU |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00063**

| Parameter \| Name: **C00063 \| Motor temperature** | | | Data type: INTEGER_16 Index: $24512_d = 5FC0_h$ |
|---|---|---|---|
| Display of the current motor temperature | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -200 | °C | 250 | |
| **Subcodes** | | | **Info** |
| C00063/1 | | | Motor temperature |
| C00063/2 | | | Motor temperature via resolver |
| C00063/3 | | | Motor temperature via MultiEncoder |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00064**

| Parameter \| Name: **C00064 \| Device utilisation (Ixt)** | | | Data type: INTEGER_16 Index: $24511_d$ = $5FBF_h$ |
|---|---|---|---|
| Display of the device utilisation Ixt in different time resolutions<br>• If the value displayed here exceeds the threshold set in C00123, the fault message "OC5: Device overload (Ixt)" is output and the fault response set in C00604 is executed (default setting: "Warning"). | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 250.00 | |

| Subcodes | Info |
|---|---|
| C00064/1 | Device utilisation (Ixt)<br>• Maximum value of the pulse utilisation (C00064/2) and permanent utilisation (C00064/3). |
| C00064/2 | Device utilisation (Ixt) 15s<br>• Pulse utilisation over the last 15 seconds (only for loads >160 %). |
| C00064/3 | Device utilisation (Ixt) 3 min<br>• Permanent utilisation over the last 3 minutes. |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00065**

| Parameter \| Name: **C00065 \| Supply voltage 24V** | | | Data type: INTEGER_16 Index: $24510_d$ = $5FBE_h$ |
|---|---|---|---|
| Display of the 24V supply voltage for the supply of the control electronics<br><br>**Note:**<br>The 24 V supply for the control electronics is either provided by an external supply or by the invter itself if it is connected to the mains voltage. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.0 | V | 3276.7 | |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10

**C00066**

| Parameter \| Name: **C00066 \| Thermal motor load (I²xt)** | | | Data type: INTEGER_16 Index: $24509_d$ = $5FBD_h$ |
|---|---|---|---|
| Display of the thermal motor load, sensorlessly determined using a motor model<br>• If the value displayed here exceeds "100.00 %", the error message "OC6: Thermal motor overload (I2xt)" is output and the fault response set in C00606 is executed (default setting: "Warning").<br>▸ Motor overload monitoring (I2xt) | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00070**

| Parameter \| Name: **C00070 \| Vp speed controller** | | | Data type: UNSIGNED_16 Index: $24505_d = 5FB9_h$ |
|---|---|---|---|
| Gain factor Vp of the speed controller for different motor control types | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | | 600.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00070/1 | power-related (see table) | | SLVC: Vp speed controller |
| C00070/2 | power-related (see table) | | SC: Vp speed controller |
| C00070/3 | power-related (see table) | | SLPSM: Vp speed controller |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00071**

| Parameter \| Name: **C00071 \| Ti speed controller** | | | Data type: UNSIGNED_16 Index: $24504_d = 5FB8_h$ |
|---|---|---|---|
| Reset time Ti of the speed controller for different motor control types | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | ms | 6000.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00071/1 | 100.0 ms | | SLVC: Ti speed controller |
| C00071/2 | 50.0 ms | | SC: Ti speed controller |
| C00071/3 | 50.0 ms | | SLPSM: Ti speed controller |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00072**

| Parameter \| Name: **C00072 \| SC: Tdn speed controller** | | | Data type: UNSIGNED_16 Index: $24503_d = 5FB7_h$ |
|---|---|---|---|
| Differential time constant Tdn of the speed controller for servo control (SC) and sensorless control for synchronous motors (SLPSM) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | ms | 3.00 | **0.00 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00073**

| Parameter \| Name: **C00073 \| Imax/M controller gain** | | | Data type: UNSIGNED_16 Index: $24502_d = 5FB6_h$ |
|---|---|---|---|
| Amplification factor Vp of certain controllers for different motor control modes | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00073/1 | power-related (see table) | | VFC: Vp Imax controller<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. |
| C00073/2 | power-related (see table) | | SLVC: Vp torque controller |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00074**

| Parameter \| Name: **C00074 \| Reset time Imax/M controller** | | | Data type: UNSIGNED_16 Index: $24501_d = 5FB5_h$ |
|---|---|---|---|
| Reset time Ti of certain controllers for different motor control modes | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 9990 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00074/1 | power-related (see table) | | VFC: Ti Imax controller |
| C00074/2 | power-related (see table) | | SLVC: Ti torque controller |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00075**

| Parameter \| Name: **C00075 \| Vp current controller** | | | Data type: UNSIGNED_16 Index: $24500_d = 5FB4_h$ |
|---|---|---|---|
| Gain factor Vp of the current controller for servo control (SC) and certain inverter functions (parameter identification, flying restart circuit)<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | V/A | 500.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00076**

| Parameter \| Name: **C00076 \| Ti current controller** | | | Data type: UNSIGNED_16 Index: $24499_d = 5FB3_h$ |
|---|---|---|---|
| Reset time Ti of the current controller for servo control (SC) and certain inverter functions (parameter identification, flying restart circuit)<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | ms | 500.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00077**

| Parameter \| Name: **C00077 \| SC: Vp field controller** | | | Data type: UNSIGNED_16 Index: $24498_d = 5FB2_h$ |
|---|---|---|---|
| Gain factor Vp of the field controller for servo control (SC)<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | | 500.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00078**

| Parameter \| Name: **C00078 \| SC: Tn field controller** | | | Data type: UNSIGNED_16 Index: $24497_d = 5FB1_h$ |
|---|---|---|---|
| Reset time Tn of the field controller for servo control (SC)<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | ms | 6000.0 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00079**

| Parameter \| Name: **C00079 \| SC: Settings** | | Data type: UNSIGNED_8 Index: $24496_d = 5FB0_h$ |
|---|---|---|
| Configuration of different options for servo control (SC) and sensorless control for synchronous motors (SLPSM) | | |
| **Selection list** | | |
| 0 | Off | |
| 1 | On | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00079/1 | 0: Off | SC: Current controller - feedforward control<br>• Please observe the notes regarding this function in the chapter "Feedforward control of the current controller"! |
| C00079/2 | 1: On | SC: Adapt. VP in the field weakening range<br>• Speed-dependent adaptive field weakening controller. |
| C00079/3 | 0: Off | SC: n-ctrl anti-wind-up<br>• "Anti-wind-up" effect of the speed controller in case of an output voltage limitation in the field weakening range. |
| C00079/4 | 1: On | Field weakening for synchronous motors<br>• From version 02.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00080**

| Parameter \| Name: **C00080 \| Override point of field weakening** | | | Data type: INTEGER_16 Index: $24495_d = 5FAF_h$ |
|---|---|---|---|
| Offset of the override point for field weakening<br>• In the V/f characteristic control mode (VFCplus), the stall protection function or the max. permissible current in the field weakening range can be adapted. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -500 | Hz | 500 | **0 Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00081**

| Parameter \| Name: **C00081 \| Rated motor power** | | | Data type: UNSIGNED_16 Index: $24494_d$ = $5FAE_h$ |
|---|---|---|---|
| This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.<br><br>**Note:**<br>It is mandatory to give the rated motor power for the sensorless vector control (SLVC) and the servo control (SC). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | kW | 500.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00082**

| Parameter \| Name: **C00082 \| Motor rotor resistance** | | | Data type: UNSIGNED_32 Index: $24493_d$ = $5FAD_h$ |
|---|---|---|---|
| After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | mOhm | 200000 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00083**

| Parameter \| Name: **C00083 \| Motor rotor time constant** | | | Data type: UNSIGNED_16 Index: $24492_d$ = $5FAC_h$ |
|---|---|---|---|
| Display of the rotor time constant of the motor<br>• This value is calculated from the rotor resistance and the rotor inductance (leakage and magnetising inductance). | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | ms | 32767 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00084**

| Parameter \| Name: **C00084 \| Motor stator resistance** | | | Data type: UNSIGNED_32 Index: $24491_d$ = $5FAB_h$ |
|---|---|---|---|
| After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | mOhm | 200000 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00085**

| Parameter \| Name: **C00085 \| Motor stator leakage inductance** | | | Data type: UNSIGNED_16 Index: $24490_d$ = $5FAA_h$ |
|---|---|---|---|
| After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | mH | 650.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00087**

| Parameter \| Name: **C00087 \| Rated motor speed** | | | Data type: UNSIGNED_16 Index: $24488_d = 5FA8_h$ |
|---|---|---|---|
| This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.<br><br>**Note:**<br>It is mandatory to give the rated motor speed for the sensorless vector control (SLVC) and the servo control (SC). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 50 | rpm | 60000 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00088**

| Parameter \| Name: **C00088 \| Rated motor current** | | | Data type: UNSIGNED_16 Index: $24487_d = 5FA7_h$ |
|---|---|---|---|
| This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.20 | A | 320.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00089**

| Parameter \| Name: **C00089 \| Rated motor frequency** | | | Data type: UNSIGNED_16 Index: $24486_d = 5FA6_h$ |
|---|---|---|---|
| This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.<br><br>**Note:**<br>It is mandatory to give the rated motor frequency for the sensorless vector control (SLVC) and the servo control (SC). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | Hz | 1000 | **50 Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00090**

| Parameter \| Name: **C00090 \| Rated motor voltage** | | | Data type: UNSIGNED_16 Index: $24485_d = 5FA5_h$ |
|---|---|---|---|
| This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | V | 5000 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00091**

| Parameter \| Name: **C00091 \| Motor cosine phi** | | | Data type: UNSIGNED_8 Index: $24484_d$ = $5FA4_h$ |
|---|---|---|---|
| This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.20 | | 1.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00092**

| Parameter \| Name: **C00092 \| Motor magnetising inductance** | | | Data type: UNSIGNED_16 Index: $24483_d$ = $5FA3_h$ |
|---|---|---|---|
| After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | mH | 6500.0 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00093**

| Parameter \| Name: **C00093 \| Power section identification** | | | Data type: UNSIGNED_16 Index: $24482_d$ = $5FA2_h$ |
|---|---|---|---|
| Display of the identification of the detected power section of the inverter | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00095**

| Parameter \| Name: **C00095 \| Motor magnetising current** | | | Data type: UNSIGNED_16 Index: $24480_d$ = $5FA0_h$ |
|---|---|---|---|
| After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | A | 320.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00097**

| Parameter \| Name: **C00097 \| Rated motor torque** | | | Data type: UNSIGNED_32 Index: $24478_d$ = $5F9E_h$ |
|---|---|---|---|
| Display of the rated motor torque<br>• The value displayed here is calculated from different parameters, e.g. the maximum current set in C00022. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | Nm | 100000.00 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00098**

| Parameter \| Name: **C00098 \| Device rated current** | | | Data type: UNSIGNED_16 Index: $24477_d$ = $5F9D_h$ |
|---|---|---|---|
| Display of the rated inverter current which is defined by the integrated power section. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.0 | A | 6000.0 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00099**

| Parameter \| Name: **C00099 \| Firmware version** | Data type: VISIBLE_STRING Index: $24476_d$ = $5F9C_h$ |
|---|---|
| Display of the firmware version of the device as string | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 12 | |

**C00100**

| Parameter \| Name: **C00100 \| Firmware version** | | | Data type: UNSIGNED_8 Index: $24475_d$ = $5F9B_h$ |
|---|---|---|---|
| Display of the firmware version of the device, divided into subsections. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 99 | |
| **Subcodes** | | | **Info** |
| C00100/1 | | | Firmware version - main version |
| C00100/2 | | | Firmware version - subversion |
| C00100/3 | | | Firmware version - release |
| C00100/4 | | | Firmware version - build |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00101**

| Parameter \| Name: **C00101 \| Add. accel. time x** | | | Data type: UNSIGNED_32 Index: $24474_d$ = $5F9A_h$ |
|---|---|---|---|
| The L_NSet_1: FB: Additional acceleration times for the main setpoint<br>• The additional acceleration times set here can be selected via the binary inputs *bTI1* ... *bTI8* of the L_NSet_1 FB. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00101/1 | 0.000 s | | Add. accel. time 1 ... 15 |
| C00101/... | | | |
| C00101/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00103**

| Parameter \| Name: **C00103 \| Add. decel. time x** | | | Data type: UNSIGNED_32 Index: $24472_d = 5F98_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Additional deceleration times for the main setpoint<br>• The additional deceleration times set here can be selected via the binary inputs *bTI1* ... *bTI8* of the L_NSet_1 FB. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00103/1 | 0.000 s | | Add. decel. time 1 ... 15 |
| C00103/... | | | |
| C00103/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00104**

| Parameter \| Name: **C00104 \| Quick stop setting** | | Data type: UNSIGNED_16 Index: $24471_d = 5F97_h$ |
|---|---|---|
| From version 02.00.00 | | ▸ Quick stop: Optional settings |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** | | **Info** |
| Bit 0 | Standstill position-controlled | 1 ≡ At standstill the position control is activated.<br>The basic conditions are shown in the flow diagram for the quick stop function. |
| Bit 1 | Ramp position-controlled | 1 ≡ During the ramp-down phase, the position control is activated.<br>The basic conditions are shown in the flow diagram for the quick stop function. |
| Bit 2 | Continue following error | From version 15.00.00<br>1 ≡ The current following error is continued if the quick stop with position control is activated while ramping down. |
| Bit 3 | Starting value is the setpoint speed | From version 15.00.00<br>0 ≡ The starting value at quick stop is the actual speed if it derives more than 5 rpm from the setpoint. This serves to prevent a jump to a strongly differing setpoint.<br>1 ≡ The starting value for quick stop is the setpoint speed . |
| Bit 4 | Use position encoder | From version 15.00.00<br>1 ≡ In case of quick stop with position control, the position encoder is used instead of the speed encoder.<br>The basic conditions are shown in the flow diagram for the quick stop function. |
| Bit 5 | Standstill depending on the actual speed | From version 16.00.00<br>1 ≡ Extended option for quick stop with position control at standstill.<br>When this setting is activated, the standstill of the motor shaft is detected under consideration of the speed setpoint and the actual speed. The standstill of the motor shaft is reached when<br>• Bit 5 = FALSE (Lenze setting)<br>  • Speed setpoint = 0<br>• Bit 5 = TRUE<br>  • Speed setpoint = 0 AND 0 ≤ actual speed value ≤ 1 rpm |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00104/1 | 0x0000 | Quick stop setting |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C00105**

| Parameter \| Name: **C00105 \| Decel. time - quick stop** | | | Data type: UNSIGNED_32 Index: $24470_d = 5F96_h$ |
|---|---|---|---|
| The set deceleration time determines the ramp slope at quick stop<br>▸ Activate/deactivate quick stop | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.900 | **2.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00106**

| Parameter \| Name: **C00106 \| Auto DCB: Hold time** | | | Data type: UNSIGNED_32 Index: $24469_d = 5F95_h$ |
|---|---|---|---|
| Hold time of the automatic DC injection brake<br>• The DC injection brake is applied for the time set here if the value falls below the speed setpoint set in C00019.<br>▸ DC-injection braking | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.000 | **0.500 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00107**

| Parameter \| Name: **C00107 \| DC braking: Hold time** | | | Data type: UNSIGNED_32 Index: $24468_d = 5F94_h$ |
|---|---|---|---|
| Maximum hold time of the manual DC injection brake<br>• A time can be set here after which the DC injection brake is switched off automatically to prevent the motor from thermal overload.<br>• When "999.000 s" is set, the hold time is infinite.<br>▸ DC-injection braking | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.000 | **999.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00114**

| Parameter \| Name: **C00114 \| DigInX: Inversion** | | | Data type: UNSIGNED_16 Index: $24461_d = 5F8D_h$ |
|---|---|---|---|
| The polarity of each digital input of the device can be inverted via this bit field. ▸ Digital input terminals | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | DI1 inverted | | Inversion of digital input 1 |
| Bit 1 ☐ | DI2 inverted | | Inversion of digital input 2 |
| Bit 2 ☐ | DI3 inverted | | Inversion of digital input 3 |
| Bit 3 ☐ | DI4 inverted | | Inversion of digital input 4 |
| Bit 4 ☐ | DI5 inverted | | Inversion of digital input 5 |
| Bit 5 ☐ | DI6 inverted | | Inversion of digital input 6 |
| Bit 6 ☐ | DI7 inverted | | Inversion of digital input 7 |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | CInh inverted | | Inversion of digital input RFR (controller enable) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00115**

| Parameter \| Name: **C00115 \| DI 1/2 & 6/7: Fct.** | | Data type: UNSIGNED_8 Index: $24460_d = 5F8C_h$ |
|---|---|---|
| Function assignment of the digital terminals DI1/2 and DI6/7 ▸ Digital input terminals: Function assignment | | |
| **Selection list** | | **Info** |
| 0 | DI1(6)=In / DI2(7)=In | DI1/6 = digital input<br>DI2/7 = digital input |
| 1 | DI1(6)=FreqIn / DI2(7)=In | DI1/6 = 1-track frequency input<br>DI2/7 = digital input |
| 2 | DI1(6)&DI2(7)=FreqIn (2-track) | DI1/6 and DI2/7 = 2-track frequency input |
| 3 | DI1(6)=FreqIn / DI2(7)=direction | DI1/6 = 1-track frequency input<br>DI2/7 = specification of direction |
| 4 | DI1(6)=CountIn / DI2(7)=In | DI1/6 = counter input<br>DI2/7 = digital input |
| **Subcodes** | **Lenze setting** | **Info** |
| C00115/1 | 0: DI1(6)=In / DI2(7)=In | Function assignment of DI1 and DI2 |
| C00115/2 | 0: DI1(6)=In / DI2(7)=In | Function assignment of DI6 and DI7 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00117**

| Parameter \| Name: **C00117 \| Status of brake output BD** | | | Data type: UNSIGNED_8 Index: $24458_d = 5F8A_h$ |
|---|---|---|---|
| Status message of brake output | | | |
| **Selection list** (read only) | | | **Info** |
| 0 | Inactive | | Brake output is inactive |
| 1 | Active | | Brake output is active |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00118**

| Parameter \| Name: **C00118 \| DigOutX: Inversion** | | | Data type: UNSIGNED_8 Index: $24457_d = 5F89_h$ |
|---|---|---|---|
| The polarity of each digital output of the device can be inverted via this bit field.<br>▸ Digital output terminals | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x00** (decimal: 0) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | Relay inverted | | Relay inversion |
| Bit 1 ☐ | DO1 inverted | | Inversion of digital output 1 |
| Bit 2 ☐ | DO2 inverted | | Inversion of digital output 2 |
| Bit 3 ☐ | DO3 inverted | | Inversion of digital output 3 |
| Bit 4 ☐ | High current inverted | | Inversion of high-current output |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00120**

| Parameter \| Name: **C00120 \| Setting of motor overload (I²xt)** | | | Data type: INTEGER_16 Index: $24455_d = 5F87_h$ |
|---|---|---|---|
| Operating threshold for the "OC6: Motor overload (I²xt)" error message<br>• The response for reaching the threshold can be selected in C00606.<br>• The current thermal motor load is displayed in C00066.<br>▸ Motor overload monitoring (I2xt) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 250.00 | **100.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00121**

| Parameter \| Name: **C00121 \| Warning threshold motor temperature** | | | Data type: UNSIGNED_16 Index: $24454_d = 5F86_h$ |
|---|---|---|---|
| Operating threshold for error message "Motor: Temperature at Res./Enc. > C121"<br>• The response for reaching the threshold can be selected in C00583.<br>• The current motor temperature is displayed in C00063.<br>▸ Encoder/feedback system: Motor temperature monitoring (KTY) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | °C | 250 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00121/1 | 120 °C | | Warning threshold motor temperature resolver |
| C00121/2 | 120 °C | | Warning threshold motor temperature MultiEncoder |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00122**

| Parameter \| Name: **C00122 \| Initial value motor overload (I²xt)** | | | Data type: UNSIGNED_16 Index: $24453_d = 5F85_h$ |
|---|---|---|---|
| From version 12.00.00<br>The thermal motor load displayed in C00066 is pre-initialised with the value set here when the device is connected to the mains.<br>• If "100.00 %" is set, the last value at switching off the device is used for the initialisation.<br>• Recommended setting for operation according to UL: 30.00 %<br>▸ Motor overload monitoring (I2xt) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00122/1 | 0.00 % | | Initial value motor overload (I²xt) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00123**

| Parameter \| Name: **C00123 \| Device utilisation threshold (Ixt)** | | | Data type: INTEGER_16 Index: $24452_d = 5F84_h$ |
|---|---|---|---|
| Operating threshold for the "OC5: Device overload (Ixt)" error message<br>• The response for reaching the threshold can be selected in C00604.<br>• The current device utilisation is displayed in C00064. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 200.00 | **100.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00124**

| Parameter \| Name: **C00124 \| Current monitoring: Breaking current** | | | Data type: UNSIGNED_16 Index: $24451_d = 5F83_h$ |
|---|---|---|---|
| From version 16.00.00<br>▸ Current monitoring overload | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 200.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00124/1 | 200.00 % | | Current monitoring: Breaking current overload |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00129**

| Parameter \| Name: **C00129 \| Brake resistance value** | | | Data type: UNSIGNED_16 Index: $24446_d = 5F7E_h$ |
|---|---|---|---|
| Resistance value of the connected brake resistor<br>• The value to be entered can be obtained from the nameplate of the brake resistor. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | Ohm | 500.0 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00130**

| Parameter \| Name: **C00130 \| Rated power - brake resistor** | | | Data type: UNSIGNED_16 Index: $24445_d = 5F7D_h$ |
|---|---|---|---|
| Rated power of the connected brake resistor<br>• The value to be entered can be obtained from the nameplate of the brake resistor. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | W | 65535 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00131**

| Parameter \| Name: **C00131 \| Rated heat amount of brake resist.** | | | Data type: UNSIGNED_16 Index: $24444_d = 5F7C_h$ |
|---|---|---|---|
| Thermal capacity of the connected brake resistor<br>• The value to be entered can be obtained from the nameplate of the brake resistor. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | kWs | 6553.5 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00133**

| Parameter \| Name: **C00133 \| Brake resistor utilisation** | | | Data type: UNSIGNED_16 Index: $24442_d = 5F7A_h$ |
|---|---|---|---|
| Display of the utilisation of the connected brake resistor | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | % | 65535 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00134**

| Parameter \| Name: **C00134 \| L_NSet_1: Ramp smoothing** | | Data type: UNSIGNED_8 Index: $24441_d = 5F79_h$ |
|---|---|---|
| The L_NSet_1 FB: Configuration of the ramp rounding for the main setpoint | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Off** | Ramp rounding deactivated |
| 1 | PT1 behaviour | Ramp rounding with PT1 behaviour<br>• The corresponding S-ramp time must be set in C00182. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00136**

| Parameter \| Name: **C00136 \| Communication control words** | | | Data type: UNSIGNED_16 Index: $24439_d = 5F77_h$ |
|---|---|---|---|
| Control words of the communication interfaces | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | SwitchOn | | |
| Bit 1 | DisableVoltage | | |
| Bit 2 | SetQuickStop | | |
| Bit 3 | EnableOperation | | |
| Bit 4 | ModeSpecific_1 | | |
| Bit 5 | ModeSpecific_2 | | |
| Bit 6 | ModeSpecific_3 | | |
| Bit 7 | ResetFault | | |
| Bit 8 | SetHalt | | |
| Bit 9 | Reserved_1 | | |
| Bit 10 | Reserved_2 | | |
| Bit 11 | LenzeSpecific_1 | | |
| Bit 12 | LenzeSpecific_2 | | |
| Bit 13 | LenzeSpecific_3 | | |
| Bit 14 | SetFail | | |
| Bit 15 | LenzeSpecific_4 | | |
| **Subcodes** | | | **Info** |
| C00136/1 | | | MCI control word<br>• Control word of the MCI communication interface (communication module) |
| C00136/2 | | | CAN control word<br>• Control word of the CAN communication interface (CAN on board) |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00137**

| Parameter \| Name: **C00137 \| Device status** | | Data type: UNSIGNED_16 Index: $24438_d = 5F76_h$ |
|---|---|---|
| Display of the current device status | | |
| **Selection list** (read only) | | |
| 0 | FirmwareUpdate | |
| 1 | Init | |
| 2 | Ident | |
| 3 | ReadyToSwitchON | |
| 4 | SwitchedON | |
| 5 | OperationEnable | |
| 6 | Warning | |
| 7 | Trouble | |
| 8 | Fault | |
| 9 | TroubleQSP | |
| 10 | SafeTorqueOff | |
| 11 | SystemFail | |
| 12 | Reserved_1 | |
| 13 | Reserved_2 | |
| 14 | Reserved_3 | |
| 15 | Reserved_4 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00138**

| Parameter \| Name: **C00138 \| Internal control signals** | | | Data type: UNSIGNED_16 Index: $24437_d = 5F75_h$ |
|---|---|---|---|
| Bit coded display of internal control signals of different sources | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Reserved | | |
| Bit 1 | DisableVoltage | | |
| Bit 2 | SetQuickStop | | |
| Bit 3 | EnableOperation | | |
| Bit 4 | InitFinishedOK | | |
| Bit 5 | ModeSpecific_2 | | |
| Bit 6 | ModeSpecific_3 | | |
| Bit 7 | ResetFault | | |
| Bit 8 | SetHalt | | |
| Bit 9 | FirmwareUpdate | | |
| Bit 10 | MotorIdent | | |
| Bit 11 | SetMessage | | |
| Bit 12 | SetIMP | | |
| Bit 13 | SetSystemFail | | |
| Bit 14 | SetFail | | |
| Bit 15 | SetFailQSP | | |
| **Subcodes** | | | **Info** |
| C00138/1 | | | SYS control signals |
| C00138/2 | | | MCK control signals |
| C00138/3 | | | FWM control signals |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00142**

| Parameter \| Name: **C00142 \| Auto-start option** | | | Data type: UNSIGNED_8 Index: $24433_d = 5F71_h$ |
|---|---|---|---|
| Starting performance of the inverter after mains connection, undervoltage, loading of the Lenze setting as well as a reset of "Trouble" or "Fault" can be parameterised individually. ▸ Automatic restart after mains connection/fault... | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x19** (decimal: 25) |
| **Value is bit-coded: (☑ = bit set)** | | | |
| Bit 0 ☑ | Inhibit at power-on | | |
| Bit 1 ☐ | Inhibit at trouble | | |
| Bit 2 ☐ | Inhibit at fault | | |
| Bit 3 ☑ | Inhibit at undervoltage | | |
| Bit 4 ☑ | Inhibit at Lenze setting | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00144**

| Parameter \| Name: **C00144 \| Switching freq. reduct. (Temp.)** | | Data type: UNSIGNED_8 Index: $24431_d = 5F6F_h$ |
|---|---|---|
| Activation of the automatic switching frequency reduction if the temperature is too high | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | Off | Automatic switching frequency reduction deactivated |
| **1** | **On** | Automatic switching frequency reduction activated |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00148**

| Parameter \| Name: **C00148 \| LS_DriveInterface: Error message config.** | | | Data type: UNSIGNED_16 Index: $24427_d = 5F6B_h$ |
|---|---|---|---|
| Selection of the device statuses for which the *bCollectedFail* group error output of SB LS_DriveInterface is to be set to TRUE. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0030** (decimal: 48) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | SafeTorqueOff | | |
| Bit 1 ☐ | ReadyToSwitchOn | | |
| Bit 2 ☐ | SwitchedOn | | |
| Bit 3 ☐ | TroubleQSP | | |
| Bit 4 ☑ | Trouble | | |
| Bit 5 ☑ | Fault | | |
| Bit 6 ☐ | Warning | | |
| Bit 7 ☐ | ImpIsActive | | |
| Bit 8 ☐ | CInhIsActive | | |
| Bit 9 ☐ | Fail CAN_Management | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Lock bFail at TroubleQSP | | From version 02.00.00<br>1 ≡ The *bFail* output of the SB LS_DriveInterface is also set in the "TroubleQSP" status.<br>• Advantage: Even in the "TroubleQSP" status, an error occurred before can still be recognised. |
| Bit 15 ☐ | Use 16BitFailNo. | | 1 ≡ The short 16-bit error number *wStateDetermFailNoShort* is provided at the *wStateDetermFailNoLow* output of the SB LS_DriveInterface.<br>• In this case, the *wStateDetermFailNoHigh* output is "0".<br>• Advantage: The bus transfer of the error numbers is possible via a data word without changing the interconnection of the technology application. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00150**

| Parameter \| Name: **C00150 \| Status word** | | | Data type: UNSIGNED_16 Index: $24425_d = 5F69_h$ |
|---|---|---|---|
| Bit coded device status word | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | FreeStatus | | Free status bit 0 |
| Bit 1 | PowerDisabled | | Power switched off |
| Bit 2 | FreeStatus | | Free status bit 2 |
| Bit 3 | FreeStatus | | Free status bit 3 |
| Bit 4 | FreeStatus | | Free status bit 4 |
| Bit 5 | FreeStatus | | Free status bit 5 |
| Bit 6 | ActSpeedIsZero | | Current speed is 0 |
| Bit 7 | ControllerInhibit | | Controller is inhibited |
| Bit 8 | StatusCodeBit0 | | Status code bit 0 |
| Bit 9 | StatusCodeBit1 | | Status code bit 1 |
| Bit 10 | StatusCodeBit2 | | Status code bit 2 |
| Bit 11 | StatusCodeBit3 | | Status code bit 3 |
| Bit 12 | Warning | | Warning |
| Bit 13 | Trouble | | Fault |
| Bit 14 | FreeStatus | | Free status bit 14 |
| Bit 15 | FreeStatus | | Free status bit 15 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00155**

| Parameter \| Name: **C00155 \| Extended status word** | | | Data type: UNSIGNED_16 Index: $24420_d = 5F64_h$ |
|---|---|---|---|
| Bit coded device status word 2 | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Fail | | Error |
| Bit 1 | M_max | | Maximum torque |
| Bit 2 | I_max | | Maximum current |
| Bit 3 | PowerDisabled | | Power switched off |
| Bit 4 | Ready | | Controller is ready for operation |
| Bit 5 | ControllerInhibit | | Controller is inhibited |
| Bit 6 | Trouble | | Fault |
| Bit 7 | InitState | | Initialisation |
| Bit 8 | CwCcw | | CW/CCW rotation |
| Bit 9 | TroubleQSP | | Quick stop due to fault is active |
| Bit 10 | SafeTorqueOff | | Safe torque off |
| Bit 11 | AplicationRunning | | Application is running |
| Bit 12 | AplParSetBit0 | | Application parameter set - bit 0 |
| Bit 13 | AplParSetBit1 | | Application parameter set - bit 1 |
| Bit 14 | quick stop | | Quick stop active |
| Bit 15 | Motor parameter identification | | Motor parameter identification is active |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00158**

| Parameter \| Name: **C00158 \| Cause of controller inhibit** | | | Data type: UNSIGNED_16 Index: $24417_d = 5F61_h$ |
|---|---|---|---|
| Bit coded display of the cause/source of the controller inhibit | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Terminal controller enable | | |
| Bit 1 | CAN control word | | |
| Bit 2 | MCI control word | | |
| Bit 3 | SwitchOn | | |
| Bit 4 | Application | | |
| Bit 5 | Device command | | |
| Bit 6 | Error response | | |
| Bit 7 | Internal signal | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | AutoStartLock | | |
| Bit 11 | Motor parameter identification | | |
| Bit 12 | Automatic brake operation | | |
| Bit 13 | DCB-IMP | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00159**

| Parameter \| Name: **C00159 \| Cause of quick stop QSP** | | | Data type: UNSIGNED_16 Index: $24416_d = 5F60_h$ |
|---|---|---|---|
| Bit coded display of the cause/source of the quick stop | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Reserved | | |
| Bit 1 | CAN control word | | |
| Bit 2 | MCI control word | | |
| Bit 3 | Reserved | | |
| Bit 4 | Application | | |
| Bit 5 | Device command | | |
| Bit 6 | Error response | | |
| Bit 7 | Internal signal | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Operating system of | | |
| Bit 11 | Reserved | | |
| Bit 12 | MCK | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00160**

| Parameter \| Name: **C00160 \| Status determining error (16bit)** | | | Data type: UNSIGNED_16 Index: $24415_d = 5F5F_h$ |
|---|---|---|---|
| Display of the short 16-bit error number of the status determining error<br>▸ Structure of the 16-bit error number (bit coding) | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | | | **Info** |
| C00160/1 | | | Status determining error (16-bit) |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00161**

| Parameter \| Name: **C00161 \| LS_SetError_x: Error number** | | | Data type: UNSIGNED_16 Index: $24414_d = 5F5E_h$ |
|---|---|---|---|
| Setting of the error number for user error messages | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00161/1 | 1 | | LS_SetError_1: Error no.1 |
| C00161/2 | 2 | | LS_SetError_1: Error no.2 |
| C00161/3 | 3 | | LS_SetError_1: Error no.3 |
| C00161/4 | 4 | | LS_SetError_1: Error no.4 |
| C00161/5 | 1 | | LS_SetError_2: Error no.1 |
| C00161/6 | 2 | | LS_SetError_2: Error no.2 |
| C00161/7 | 3 | | LS_SetError_2: Error no.3 |
| C00161/8 | 4 | | LS_SetError_2: Error no.4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00162**

| Parameter \| Name: **C00162 \| Error number masked** | | | Data type: UNSIGNED_32 Index: $24413_d = 5F5D_h$ |
|---|---|---|---|
| From version 13.00.00<br>Display of the 32-bit error number of the status determining error without error type<br>• The error number displayed here only contains the error subject area and error ID (lower 26 bits of the 32-bit error number)<br>▸ Structure of the 32-bit error number (bit coding) | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 4294967295 | |
| **Subcodes** | | | **Info** |
| C00162/1 | | | Subject area + Id statuserror |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00163**

| Parameter \| Name: **C00163 \| Logbook - binary elements** | | Data type: UNSIGNED_16 Index: $24412_d = 5F5C_h$ |
|---|---|---|
| Selection of two binary signals to be logged in the logbook | | |
| **Selection list** | | **Info** |
| 0 | No signal | |
| 1 | DI1: Input signal | |
| 2 | DI2: Input signal | |
| 3 | DI3: Input signal | |
| 4 | DI4: Input signal | |
| 5 | Controller inhibit signal | |
| 6 | Digital counter: Comparison bit | |
| 7 | CAN1 input bit 0 | |
| 8 | CAN1 input bit 1 | |
| 9 | CAN1 input bit 2 | |
| 10 | CAN1 input bit 3 | |

| Parameter \| Name: **C00163 \| Logbook - binary elements** | | Data type: UNSIGNED_16 Index: $24412_d = 5F5C_h$ |
|---|---|---|
| 11 | CAN1 input bit 4 | |
| 12 | CAN1 input bit 5 | |
| 13 | CAN1 input bit 6 | |
| 14 | CAN1 input bit 7 | |
| 15 | CAN1 input bit 8 | |
| 16 | CAN1 input bit 9 | |
| 17 | CAN1 input bit 10 | |
| 18 | CAN1 input bit 11 | |
| 19 | CAN1 input bit 12 | |
| 20 | CAN1 input bit 13 | |
| 21 | CAN1 input bit 14 | |
| 22 | CAN1 input bit 15 | |
| 23 | CAN2 input bit 0 | |
| 24 | CAN2 input bit 1 | |
| 25 | CAN2 input bit 2 | |
| 26 | CAN2 input bit 3 | |
| 27 | CAN2 input bit 4 | |
| 28 | CAN2 input bit 5 | |
| 29 | CAN2 input bit 6 | |
| 30 | CAN2 input bit 7 | |
| 31 | CAN2 input bit 8 | |
| 32 | CAN2 input bit 9 | |
| 33 | CAN2 input bit 10 | |
| 34 | CAN2 input bit 11 | |
| 35 | CAN2 input bit 12 | |
| 36 | CAN2 input bit 13 | |
| 37 | CAN2 input bit 14 | |
| 38 | CAN2 input bit 15 | |
| 39 | CAN3 input bit 0 | |
| 40 | CAN3 input bit 1 | |
| 41 | CAN3 input bit 2 | |
| 42 | CAN3 input bit 3 | |
| 43 | CAN3 input bit 4 | |
| 44 | CAN3 input bit 5 | |
| 45 | CAN3 input bit 6 | |
| 46 | CAN3 input bit 7 | |
| 47 | CAN3 input bit 8 | |
| 48 | CAN3 input bit 9 | |
| 49 | CAN3 input bit 10 | |
| 50 | CAN3 input bit 11 | |
| 51 | CAN3 input bit 12 | |
| 52 | CAN3 input bit 13 | |
| 53 | CAN3 input bit 14 | |
| 54 | CAN3 input bit 15 | |

| Parameter \| Name: **C00163 \| Logbook - binary elements** | | Data type: UNSIGNED_16 Index: $24412_d = 5F5C_h$ |
|---|---|---|
| 55 | MCI word1 input bit0 | |
| 56 | MCI word1 input bit1 | |
| 57 | MCI word1 input bit2 | |
| 58 | MCI word1 input bit3 | |
| 59 | MCI Word 1 Input bit 4 | |
| 60 | MCI word1 input bit5 | |
| 61 | MCI word1 input bit6 | |
| 62 | MCI word1 input bit7 | |
| 63 | MCI word1 input bit8 | |
| 64 | MCI word1 input bit9 | |
| 65 | MCI word1 input bit10 | |
| 66 | MCI word1 input bit11 | |
| 67 | MCI word1 input bit12 | |
| 68 | MCI word1 input bit13 | |
| 69 | MCI word1 input bit14 | |
| 70 | MCI word1 input bit15 | |
| 71 | MCI word2 input bit0 | |
| 72 | MCI word2 input bit1 | |
| 73 | MCI Word 2 Input bit 2 | |
| 74 | MCI word2 input bit3 | |
| 75 | MCI word2 input bit4 | |
| 76 | MCI word2 input bit5 | |
| 77 | MCI word2 input bit6 | |
| 78 | MCI word 2 input bit 7 | |
| 79 | MCI word2 input bit8 | |
| 80 | MCI word2 input bit9 | |
| 81 | MCI word2 input bit10 | |
| 82 | MCI word2 input bit11 | |
| 83 | MCI Word 2 Input bit 12 | |
| 84 | MCI word2 input bit13 | |
| 85 | MCI word2 input bit14 | |
| 86 | MCI word2 input bit15 | |
| 87 | Position controller: Limit | |
| 88 | Speed controller: Limit | |
| 89 | Speed setpoint: Limit | |
| 90 | Torque setpoint: Limit | |
| 91 | Current setpoint: Limit | |
| 92 | DC injection brake active | |
| 93 | Quick stop active | |
| 94 | Pulse inhibit active | |
| 95 | Controller inhibit active | |
| 96 | Safe status active | |
| 97 | Direction of rotation ccw | |
| 98 | Actual speed = 0 | |

| Parameter \| Name: **C00163 \| Logbook - binary elements** | | Data type: UNSIGNED_16 Index: $24412_d = 5F5C_h$ |
|---|---|---|
| 99 | L_Or_1: Out | |
| 100 | L_DFlipFlop_1: Out | |
| 101 | L_DigitalDelay_1: Out | |
| 102 | L_Compare_1: Out | |
| 103 | L_Compare_2: Out | |
| 104 | L_NSet_1: Setpoint reached | |
| 105 | L_DigitalLogic_1: Out | |
| 106 | L_SignalMonitor_b: Out1 | |
| 107 | L_SignalMonitor_b: Out2 | |
| 108 | L_SignalMonitor_b: Out3 | |
| 109 | L_SignalMonitor_b: Out4 | |
| 110 | L_PCTRL_1: act=set | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00163/1 | 0: No signal | Logbook - binary element 1 |
| C00163/2 | 0: No signal | Logbook - binary element 2 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00164**

| Parameter \| Name: **C00164 \| Logbook - analog elements** | | Data type: UNSIGNED_16 Index: $24411_d = 5F5B_h$ |
|---|---|---|
| Selection of an analog signal to be logged in the logbook | | |
| **Selection list** | | **Info** |
| 0 | No signal | |
| 1 | AIN1 | |
| 2 | CAN1 control word | |
| 3 | CAN1 input word 2 | |
| 4 | CAN1 input word 3 | |
| 5 | CAN1 input word 4 | |
| 6 | CAN2 input word 1 | |
| 7 | CAN2 input word 2 | |
| 8 | CAN2 input word 3 | |
| 9 | CAN2 input word 4 | |
| 10 | CAN3 input word 1 | |
| 11 | CAN3 input word 2 | |
| 12 | CAN3 input word 3 | |
| 13 | CAN3 input word 4 | |
| 14 | Digital counter LowWord | |
| 15 | Digital counter HighWord | |
| 16 | MCI word 1 | |
| 17 | MCI word 2 | |
| 18 | MCI word 3 | |
| 19 | MCI word 4 | |
| 20 | MCI word 5 | |
| 21 | MCI word 6 | |

| Parameter \| Name: **C00164 \| Logbook - analog elements** | | Data type: UNSIGNED_16 Index: $24411_d = 5F5B_h$ |
|---|---|---|
| 22 | MCI word 7 | |
| 23 | MCI word 8 | |
| 24 | MCI word 9 | |
| 25 | MCI word 10 | |
| 26 | MCI word 11 | |
| 27 | MCI word 12 | |
| 28 | MCI word 13 | |
| 29 | MCI word 14 | |
| 30 | MCI word 15 | |
| 31 | MCI word 16 | |
| 32 | Current motor speed | |
| 33 | Current motor torque | |
| 34 | DC-bus voltage | |
| 35 | Current motor current | |
| 36 | Current motor voltage | |
| 37 | Current motor frequency | |
| 38 | Effective speed setpoint | |
| 39 | Device utilisation | |
| 40 | Motor utilisation | |
| 41 | L_OffsetGainPar_1: Out | |
| 42 | L_OffsetGainPar_2: Out | |
| 43 | L_OffsetGainPar_3: Out | |
| 44 | L_Aritmethik_1: Out | |
| 45 | L_AnalogSwitch_1: Out | |
| 46 | L_NSet_1: Out | |
| 47 | L_MotorPoti_1: Out | |
| 48 | L_PCTRL_1: Out | |
| 49 | L_SignalMonitor_a: Out1 | |
| 50 | L_SignalMonitor_a: Out2 | |
| 51 | L_SignalMonitor_a: Out3 | |
| 52 | L_SignalMonitor_a: Out4 | |
| 53 | L_MulDiv_1: Out | |
| 54 | L_NSet_1: Target setpoint | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00164/1 | 0: No signal | Logbook - analog element 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00165**

| Parameter \| Name: **C00165 \| Error information** | Data type: VISIBLE_STRING Index: $24410_d = 5F5A_h$ |
|---|---|
| Display of the error number divided into sectors in the event of an error | |
| **Subcodes** | **Info** |
| C00165/1 | Status determining error |
| C00165/2 | Current error |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 14 | |

**C00166**

| Parameter \| Name: **C00166 \| Error information text** | Data type: VISIBLE_STRING Index: $24409_d = 5F59_h$ |
|---|---|
| Display of details on the status determining error and on the currently pending error | |
| **Subcodes** | **Info** |
| C00166/1 | Resp. to status det. error<br>• Response to the status determining error |
| C00166/2 | Subj. - status det. error<br>• Subject area of the status determining error |
| C00166/3 | Mess. - status det. error<br>• Textual message of the status determining error |
| C00166/4 | Resp. to curr. error<br>• Response of the currently pending error |
| C00166/5 | Subj. - curr. error<br>• Subject area of the currently pending error |
| C00166/6 | Mess. - curr. error<br>• Textual message of the currently pending error |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 31 | |

**C00167**

| Parameter \| Name: **C00167 \| Logbook data** | Data type: OCTET_STRING Index: $24408_d = 5F58_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00168**

| Parameter \| Name: **C00168 \| Status determining error** | | | Data type: UNSIGNED_32 Index: $24407_d = 5F57_h$ |
|---|---|---|---|
| Display of the 32-bit error number of the status determining error<br>▸ Structure of the 32-bit error number (bit coding) | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 4294967295 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00169**

| Parameter \| Name: **C00169 \| Logbook setting** | | | Data type: UNSIGNED_16 Index: $24406_d = 5F56_h$ |
|---|---|---|---|
| Configuration which message types are to be logged in the logbook. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x067E** (decimal: 1662) |
| **Value is bit-coded: (☑ = bit set)** | | | |
| Bit 0 ☐ | Reserved | | |
| Bit 1 ☑ | Log entry: Fault | | |
| Bit 2 ☑ | Log entry: Trouble | | |
| Bit 3 ☑ | Log entry: TroubleQuickstop | | |
| Bit 4 ☑ | Log entry: WarningLocked | | |
| Bit 5 ☑ | Log entry: Warning | | |
| Bit 6 ☑ | Log entry: Information | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☑ | Activation: Error counter | | |
| Bit 10 ☑ | Activation: Log line refresh | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00170**

| Parameter \| Name: **C00170 \| Current error** | | | Data type: UNSIGNED_32 Index: $24405_d = 5F55_h$ |
|---|---|---|---|
| Display of the internal error number of the currently pending error | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 4294967295 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00171**

| Parameter \| Name: **C00171 \| Logbook access index** | Data type: UNSIGNED_8 Index: $24404_d = 5F54_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00173**

| Parameter \| Name: **C00173 \| Mains voltage** | | Data type: UNSIGNED_8 Index: $24402_d = 5F52_h$ |
|---|---|---|
| If the rated mains voltage differs from 230 V or 400 V, set the mains voltage the drive is operated with.<br>The set mains voltage influences the brake chopper threshold, the monitoring of the device utilisation (Ixt) and the switch-off threshold in case of undervoltage in the DC bus. | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **3ph 400V / 1ph 230V** | 3-phase 400 V or 1-phase 230 V |
| 1 | 3ph 440V / 1ph 230V | 3-phase 440 V or 1-phase 230 V |
| 2 | 3ph 480V / 1ph 230V | 3-phase 480 V or 1-phase 230 V |
| 3 | 3ph 500V / 1ph 230V | 3-phase 500 V or 1-phase 230 V |
| 4 | Reserved / reserved | |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00174**

| Parameter \| Name: **C00174 \| Reduc. brake chopper threshold** | | | Data type: UNSIGNED_8 Index: $24401_d = 5F51_h$ |
|---|---|---|---|
| The threshold from which on the brake chopper is controlled is reduced by the voltage value set here. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | V | 150 | **0 V** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00175**

| Parameter \| Name: **C00175 \| Brake energy management** | | Data type: UNSIGNED_8 Index: $24400_d = 5F50_h$ |
|---|---|---|
| Selection of the braking procedure<br>▸ Select response if the brake resistor is controlled | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **R_Brems** | The brake resistor is used. When the threshold voltage (C00174) is exceeded, the brake resistor is energised. |
| 1 | RfgStop | The "Ramp function generator stop" signal (*MCTRL_bRfgStop*) is used. When the threshold voltage is exceeded (C00174), the ramp function generator is stopped. |
| 2 | R_Brems + HlgStop | The brake resistor and the "Ramp function generator stop" signal are used. When the threshold voltage is exceeded (C00174), the brake resistor is energised and the ramp function generator is stopped. |
| 3 | FI_MotBrk + RfgStop | Braking is performed by a superimposed speed setpoint vibration in conjunction with "Ramp function generator stop". |
| 4 | R_Brems + FU_MotBrk + HlgStop | Braking is performed by combining all three braking procedures. |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00176**

| Parameter \| Name: **C00176 \| Undervoltage threshold for mains OFF** | Data type: UNSIGNED_16 Index: $24399_d = 5F4F_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00177**

| Parameter \| Name: **C00177 \| Switching cycles** | | | Data type: UNSIGNED_32 Index: $24398_d = 5F4E_h$ |
|---|---|---|---|
| Counter of different switching cycles and stressful situations | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00177/1 | | | Number of mains switching cycles |
| C00177/2 | | | Number of switching cycles of the output relay |
| C00177/3 | | | Short circuit counter |
| C00177/4 | | | Earth fault counter |
| C00177/5 | | | "Clamp" counter |
| C00177/6 | | | Counter for "safe torque off" (STO) after power-on<br>• From version 12.00.00 |
| C00177/7 | | | Counter for controller inhibit via terminal after power-on<br>• From version 12.00.00 |
| C00177/8 | | | Counter for pulse inhibit (IMP) after power-on<br>• From version 12.00.00 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00178**

| Parameter \| Name: **C00178 \| Elapsed-hour meter** | | | Data type: UNSIGNED_32 Index: $24397_d = 5F4D_h$ |
|---|---|---|---|
| Display of operating hours in seconds | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | s | 2147483647 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00179**

| Parameter \| Name: **C00179 \| Power-on time meter** | | | Data type: UNSIGNED_32 Index: $24396_d = 5F4C_h$ |
|---|---|---|---|
| Display of the power-on time in seconds | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | s | 2147483647 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00180**

| Parameter \| Name: **C00180 \| Running time** | | | Data type: UNSIGNED_32 Index: $24395_d = 5F4B_h$ |
|---|---|---|---|
| Display of various running times in seconds | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | s | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00180/1 | | | Running time - control card |
| C00180/2 | | | Running time - heatsink fan |
| C00180/3 | | | Running time - internal fan |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00181**

| Parameter \| Name: **C00181 \| Time settings** | | | Data type: UNSIGNED_16 Index: $24394_d = 5F4A_h$ |
|---|---|---|---|
| Time for device search function (optical location)<br>▸ Device search function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | s | 6000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00181/1 | 5 s | | Time - device search function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00182**

| Parameter \| Name: **C00182 \| L_NSet_1: S-ramp time PT1** | | | Data type: INTEGER_16 Index: $24393_d = 5F49_h$ |
|---|---|---|---|
| FB L_NSet_1: PT1 S-ramp time for the main setpoint ramp function generator<br>• Only effective with activated ramp rounding (C00134 = "1"). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.01 | s | 50.00 | **20.00 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00184**

| Parameter \| Name: **C00184 \| AutoFailReset repetition time** | | | Data type: UNSIGNED_16 Index: $24391_d = 5F47_h$ |
|---|---|---|---|
| After the time set here has expired, an error message of an error that has occurred will be reset automatically if "AutoFailReset" had been configured correspondingly in C00188.<br>▸ AutoFailReset function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | s | 600 | **3 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00185**

| Parameter \| Name: **C00185 \| AutoFailReset remaining time** | | | Data type: UNSIGNED_16 Index: $24390_d = 5F46_h$ |
|---|---|---|---|
| Display of the residual runtime of the "AutoFailReset" function<br>▸ AutoFailReset function | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | s | 600 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00186**

| Parameter \| Name: **C00186 \| Max. number of AutoFailReset processes** | | | Data type: UNSIGNED_8 Index: $24389_d = 5F45_h$ |
|---|---|---|---|
| Maximum number of "AutoFailReset" procedures<br>▸ AutoFailReset function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | | 16 | **4** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00187**

| Parameter \| Name: **C00187 \| Current AutoFailReset processes** | | | Data type: UNSIGNED_8 Index: $24388_d = 5F44_h$ |
|---|---|---|---|
| Data of the current number of "AutoFailReset" procedures<br>▸ AutoFailReset function | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 16 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00188**

| Parameter \| Name: **C00188 \| AutoFailReset configuration** | | Data type: UNSIGNED_8 Index: $24387_d = 5F43_h$ |
|---|---|---|
| Setting which error messages are to be reset automatically.<br>▸ AutoFailReset function | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Off** | No automatic error message reset |
| 1 | Fault + TroubleQSP | Error messages with the response "Fault" and "TroubleQSP" are reset automatically |
| 2 | WarningLocked | Error messages with the response "WarningLocked" are reset automatically |
| 3 | All locking | All "locking" error messages are reset automatically |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00189**

| Parameter \| Name: **C00189 \| Resp. to too frequent AutoFailReset** | | Data type: UNSIGNED_8 Index: $24386_d = 5F42_h$ |
|---|---|---|
| Response to exceeding the maximum number of "AutoFailReset" processes set in C00186.<br>▸ AutoFailReset function | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| **1** | **Fault** | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00190**

| Parameter \| Name: **C00190 \| L_NSet_1: Setpoint arithmetic** | | | Data type: UNSIGNED_8 Index: $24385_d = 5F41_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Selection of arithmetics<br>• To be able to influence the main setpoint (NSet) by an additional setpoint (NAdd). | | | |
| **Selection list** (Lenze setting printed in bold) | | | |
| **0** | **Out = Set** | | |
| 1 | Out = Set + Add | | |
| 2 | NOut = NSet - NAdd | | |
| 3 | NOut = (NSet * NAdd) / 100% | | |
| 4 | NOut = (NSet * 1%) / \|NAdd\| | | |
| 5 | Out = (Set*100%)/(100%-Add) | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00191**

| Parameter \| Name: **C00191 \| Logbook access index User** | | | Data type: UNSIGNED_8 Index: $24384_d = 5F40_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ Logbook read interface |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 255 | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C00191/1 | 255 | Logbook access index User | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00192**

| Parameter \| Name: **C00192 \| Logbook data User** | | | Data type: OCTET_STRING Index: $24383_d = 5F3F_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ Logbook read interface |
| **Display area** (min. value \| unit \| max. value) | | | |
| | | | |
| **Subcodes** | | **Info** | |
| C00192/1 | | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00193**

| Parameter \| Name: **C00193 \| Logbook element User** | | | Data type: UNSIGNED_32 Index: $24382_d = 5F3E_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ Logbook read interface |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00193/1 | | | Response index |
| C00193/2 | | | Activity |
| C00193/3 | | | Counter |
| C00193/4 | | | Error type |
| C00193/5 | | | Error number |
| C00193/6 | | | Time stamp |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00199**

| Parameter \| Name: **C00199 \| Description data** | | Data type: VISIBLE_STRING Index: $24376_d = 5F38_h$ |
|---|---|---|
| Parameters for storing decription data for the inverter | | ▸ Device identification |
| **Subcodes** | **Lenze setting** | **Info** |
| C00199/1 | | Device name |
| C00199/2 | | User text<br>• From version 12.00.00 |
| C00199/3 | | User text<br>• From version 12.00.00 |
| C00199/4 | | User text<br>• From version 12.00.00 |
| C00199/5 | | User text<br>• From version 12.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ COM ☐ MOT Character length: 32 | | |

**C00200**

| Parameter \| Name: **C00200 \| Firmware product type** | Data type: VISIBLE_STRING Index: $24375_d = 5F37_h$ |
|---|---|
| Display of the firmware product type | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 19 | |

**C00201**

| Parameter \| Name: **C00201 \| Firmware** | Data type: VISIBLE_STRING Index: $24374_d = 5F36_h$ |
|---|---|
| Display of the firmware data of the control card and the power section | |
| **Subcodes** | **Info** |
| C00201/1 | Firmware type - ctrl card |
| C00201/2 | Firmware version - ctrl card |
| C00201/3 | Firmware comp. file - ctrl card |
| C00201/4 | Firmware type - power section |
| C00201/5 | Firmware version - power sect. |
| C00201/6 | Firmw. comp. file - power sect. |
| C00201/7 | Firmware type CU2 |
| C00201/8 | Firmware version CU2 |
| C00201/9 | Firmware comp. dat. CU2 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 22 | |

**C00203**

| Parameter \| Name: **C00203 \| Product type code** | Data type: VISIBLE_STRING Index: $24372_d = 5F34_h$ |
|---|---|
| Display of the types of the individual device components | |
| **Subcodes** | **Info** |
| C00203/1 | Type: Control card |
| C00203/2 | Type: Power section |
| C00203/3 | Type: MCI module |
| C00203/4 | Reserved |
| C00203/5 | Type: Memory module |
| C00203/6 | Type: Safety card |
| C00203/7 | Type: Standard device |
| C00203/8 | Type: Complete device |
| C00203/9 | Reserved |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 24 | |

**C00204**

| Parameter \| Name: **C00204 \| Serial number** | Data type: VISIBLE_STRING Index: $24371_d = 5F33_h$ |
|---|---|
| Display of the serial numbers of the individual device components | |
| **Subcodes** | **Info** |
| C00204/1 | Serial no.: Control card |
| C00204/2 | Serial no.: Power section |
| C00204/3 | Serial no.: MCI module |
| C00204/4 | Reserved |
| C00204/5 | Serial no.: Memory module |
| C00204/6 | Serial no.: Safety card |
| C00204/7 | Serial no.: Standard device |
| C00204/8 | Serial no.: Complete device |
| C00204/9 | Reserved |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 24 | |

**C00205**

| Parameter \| Name: **C00205 \| Info** | Data type: VISIBLE_STRING Index: $24370_d = 5F32_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00206**

| Parameter \| Name: **C00206 \| Production date** | Data type: VISIBLE_STRING Index: $24369_d = 5F31_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00210**

| Parameter \| Name: **C00210 \| HW version** | Data type: VISIBLE_STRING Index: $24365_d = 5F2D_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00219**

| Parameter \| Name: **C00219 \| Identity** | | | Data type: UNSIGNED_32 Index: $24356_d = 5F24_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 4294967295 | |
| **Subcodes** | | | **Info** |
| C00219/1 | | | CAN manufacturer no. |
| C00219/2 | | | CAN device type |
| C00219/3 | | | CAN version |
| C00219/4 | | | CAN count no. |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00220**

| Parameter \| Name: **C00220 \| L_NSet_1: Acceleration time - add. setpoint** | | | Data type: UNSIGNED_32 Index: $24355_d = 5F23_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Acceleration time for the additional setpoint *nNAdd_a* | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **0.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00221**

| Parameter \| Name: **C00221 \| L_NSet_1: Deceleration time - add. setpoint** | | | Data type: UNSIGNED_32 Index: $24354_d = 5F22_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Deceleration time for the additional setpoint *nNAdd_a* | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **0.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00222**

| Parameter \| Name: **C00222 \| L_PCTRL_1: Vp** | | | Data type: INTEGER_16 Index: $24353_d = 5F21_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Gain factor Vp for the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.1 | | 500.0 | **1.0** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C00223**

| Parameter \| Name: **C00223 \| L_PCTRL_1: Tn** | | | Data type: UNSIGNED_16 Index: $24352_d = 5F20_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Reset time Tn for the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 20 | ms | 6000 | **400 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00224**

| Parameter \| Name: **C00224 \| L_PCTRL_1: Kd** | | | Data type: UNSIGNED_16 Index: $24351_d = 5F1F_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Derivative-action coefficient Kd for the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | | 5.0 | **0.0** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C00225**

| Parameter \| Name: **C00225 \| L_PCTRL_1: MaxLimit** | | | Data type: INTEGER_16 Index: $24350_d = 5F1E_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Maximum output value of the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **199.99 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00226**

| Parameter \| Name: **C00226 \| L_PCTRL_1: MinLimit** | | | Data type: INTEGER_16 Index: $24349_d = 5F1D_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Minimum output value of the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **-199.99 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00227**

| Parameter \| Name: **C00227 \| L_PCTRL_1: Acceleration time** | | | Data type: UNSIGNED_32 Index: $24348_d = 5F1C_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Acceleration time for the output value of the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **0.010 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00228**

| Parameter \| Name: **C00228 \| L_PCTRL_1: Deceleration time** | | | Data type: UNSIGNED_32 Index: $24347_d = 5F1B_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Deceleration time for the output value of the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **0.010 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00231**

| Parameter \| Name: **C00231 \| L_PCTRL_1: Operating range** | | | Data type: INTEGER_16 Index: $24344_d = 5F18_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Operating range for the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00231/1 | 199.99 % | | L_PCTRL_1: Pos. maximum |
| C00231/2 | 0.00 % | | L_PCTRL_1: Pos. minimum |
| C00231/3 | 0.00 % | | L_PCTRL_1: Neg. minimum |
| C00231/4 | 199.99 % | | L_PCTRL_1: Neg. maximum |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00233**

| Parameter \| Name: **C00233 \| L_PCTRL_1: Root function** | | | Data type: UNSIGNED_8 Index: $24342_d = 5F16_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Use of the root function at the actual value input | | | |
| **Selection list** (Lenze setting printed in bold) | | | **Info** |
| **0** | **Off** | | Root function inactive<br>• The actual value *nAct_a* remains unchanged for further processing |
| 1 | On | | Root function active<br>• The root is extracted of the actual value *nAct_a* for further processing |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00234**

| Parameter \| Name: **C00234 \| Oscillation damping influence** | | | Data type: UNSIGNED_16 Index: $24341_d = 5F15_h$ |
|---|---|---|---|
| | | | ▸ Oscillation damping |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 250.00 | **5.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00235**

| Parameter \| Name: **C00235 \| Oscillation damping filter time** | | | Data type: UNSIGNED_8 Index: $24340_d = 5F14_h$ |
|---|---|---|---|
| | | | ▸ Oscillation damping |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 2 | ms | 250 | **32 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00236**

| Parameter \| Name: **C00236 \| Field weakening oscillation damping** | | | Data type: UNSIGNED_8 Index: $24339_d = 5F13_h$ |
|---|---|---|---|
| Oscillation damping for idling machines | | | ▸ Oscillation damping |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 40 | **14** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00241**

| Parameter \| Name: **C00241 \| L_NSet_1: Hyst. NSet reached** | | | Data type: INTEGER_16 Index: $24334_d = 5F0E_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Hysteresis window for the zero detection of the speed output setpoint<br>• The speed threshold for the zero detection is 1 % | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **0.50 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00242**

| Parameter \| Name: **C00242 \| L_PCTRL_1: Operating mode** | | | Data type: UNSIGNED_8 Index: $24333_d = 5F0D_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Selection of the operating mode<br>• Depending on the selection, the blue switches in the displayed signal flow are set accordingly in the Engineer on the **Application parameters** tab for the L_PCTRL_1 FB. | | | |
| **Selection list** (Lenze setting printed in bold) | | | **Info** |
| **0** | **Off** | | The input setpoint *nNSet_a* is output without any changes at the output *nOut_a*. |
| 1 | nNSet + nNSet_PID | | *nNSet_a* and *nAct_a* are used as PID input values. The arriving *nNSet_a* is additively linked to the value output by the PID element. |
| 2 | nSet_PID | | *nSet_a* and *nAct_a* are used as PID input values. The input *nNSet_a* is not considered. |
| 3 | nNSet_PID | | *nNSet_a* and *nAct_a* are used as PID input values. The input *nSet_a* is not considered. |
| 4 | nNSet + nSet_PID | | *nSet_a* and *nAct_a* are used as PID input values. The arriving *nNSet_a* setpoint is additively linked to the value output by the PID element. |
| 5 | nNSet \|\| nSet_PID | | *nSet_a* and *nAct_a* are used as PID input values. The setpoint *nNSet_a* is output at the output *nOut_a* . The PID output value is output at the output *nPIDOut_a*. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00243**

| Parameter \| Name: **C00243 \| L_PCTRL_1: Acceleration time influence** | | | Data type: UNSIGNED_32 Index: $24332_d = 5F0C_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Acceleration time for showing the PID output value | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **5.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00244**

| Parameter \| Name: **C00244 \| L_PCTRL_1: Deceleration time influence** | | | Data type: UNSIGNED_32 Index: $24331_d = 5F0B_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Deceleration time for masking out the PID output value | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **5.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00245**

| Parameter \| Name: **C00245 \| L_PCTRL_1: PID output value** | | | Data type: INTEGER_16 Index: $24330_d = 5F0A_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Display of the output value of the PID process controller | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00246**

| Parameter \| Name: **C00246 \| L_PCTRL_1: nAct_a internal** | | | Data type: INTEGER_16 Index: $24329_d = 5F09_h$ |
|---|---|---|---|
| FB L_PCTRL_1: Display of the internal actual value | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00247**

| Parameter \| Name: **C00247 \| L_PCTRL_1: Window setpoint reached** | | | Data type: INTEGER_16 Index: $24328_d = 5F08_h$ |
|---|---|---|---|
| FB L_PCTRL_1: Window for comparison operation "actual value = setpoint" | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **2.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00249**

| Parameter \| Name: **C00249 \| L_PT1_1: Time constant** | | | Data type: UNSIGNED_16 Index: $24326_d = 5F06_h$ |
|---|---|---|---|
| FB L_PT1_1: Time constant Tn | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | ms | 5000 | **2000 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00250**

| Parameter \| Name: **C00250 \| L_PT1 2-3: Time constant** | | | Data type: INTEGER_16 Index: $24325_d = 5F05_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 5000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00250/1 | 2000 ms | | L_PT1_2: Time constant |
| C00250/2 | 2000 ms | | L_PT1_3: Time constant |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00251**

| Parameter \| Name: **C00251 \| L_DT1_1: Time constant** | | | Data type: INTEGER_16 Index: $24324_d = 5F04_h$ |
|---|---|---|---|
| FB L_DT1_1: Time constant Tn | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 10 | ms | 5000 | **1000 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00252**

| Parameter \| Name: **C00252 \| L_DT1_1: Gain** | | | Data type: INTEGER_16 Index: $24323_d = 5F03_h$ |
|---|---|---|---|
| FB L_DT1_1: Gain factor Vp | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -320.00 | | 320.00 | **1.00** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00253**

| Parameter \| Name: **C00253 \| L_DT1_1: Sensitivity** | | Data type: UNSIGNED_8 Index: $24322_d = 5F02_h$ |
|---|---|---|
| FB L_DT1_1: Selection of sensitivity<br>• Depending on the selection, the number of indicated higher-order bits is evaluated.<br><br>**Note:**<br>The most significant bit determines the sign of the value, the remaining bits determine the numerical value. | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **1** | **15 bits** | Bit 0 ... bit 14 are evaluated |
| 2 | 14 bits | Bit 0 ... bit 13 are evaluated |
| 3 | 13 bits | Bit 0 ... bit 12 are evaluated |
| 4 | 12 bits | Bit 0 ... bit 11 are evaluated |
| 5 | 11 Bit | Bit 0 ... bit 10 are evaluated |
| 6 | 10 Bit | Bit 0 ... bit 9 are evaluated |
| 7 | 9 Bit | Bit 0 ... bit 8 are evaluated |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00254**

| Parameter \| Name: **C00254 \| Kp position controller** | | | Data type: UNSIGNED_16 Index: $24321_d = 5F01_h$ |
|---|---|---|---|
| Gain for following error compensation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | 1/s | 500.00 | **5.00 1/s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00265**

| Parameter \| Name: **C00265 \| SLVC: Filtering of setpoint and actual currents** | Data type: UNSIGNED_8 Index: $24310_d = 5EF6_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00270**

| Parameter \| Name: **C00270 \| SC: Freq. current setpoint filter** | | | Data type: UNSIGNED_16 Index: $24305_d = 5EF1_h$ |
|---|---|---|---|
| Frequency to be inhibited by the current setpoint filter at servo control (SC) and sensorless control for synchronous motors (SLPSM). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 40.0 | Hz | 1000.0 | **200.0 Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00271**

| Parameter \| Name: **C00271 \| SC: Width of current setpoint filter** | | | Data type: UNSIGNED_16 Index: $24304_d = 5EF0_h$ |
|---|---|---|---|
| Frequency width of the current setpoint filter at servo control (SC) and sensorless control for synchronous motors (SLPSM)<br>• Width around the frequency to be inhibited (C00270). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | Hz | 500.0 | **0.0 Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00272**

| Parameter \| Name: **C00272 \| SC: Depth of current setpoint filter** | | | Data type: UNSIGNED_16 Index: $24303_d = 5EEF_h$ |
|---|---|---|---|
| Damping of the current setpoint filter at servo control (SC) and sensorless control for synchronous motors (SLPSM) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | dB | 100 | **0 db** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00273**

| Parameter \| Name: **C00273 \| Moment of inertia motor** | | | Data type: UNSIGNED_32 Index: $24302_d = 5EEE_h$ |
|---|---|---|---|
| Moment of inertia for setpoint feedforward control at servo control (SC) and sensorless vector control (SLVC) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | kg cm^2 | 6000000.00 | **0.00 kg cm^2** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00274**

| Parameter \| Name: **C00274 \| SC: Max. change in acceleration** | | | Data type: UNSIGNED_16 Index: $24301_d = 5EED_h$ |
|---|---|---|---|
| Limitation of the acceleration change at servo control (SC) and sensorless control for synchronous motors (SLPSM)<br>• Setting in % of M_Nenn per ms. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | %/ms | 400.0 | **400.0 %/ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00275**

| Parameter \| Name: **C00275 \| Filter setpoint feedforward control** | | | Data type: UNSIGNED_16 Index: $24300_d = 5EEC_h$ |
|---|---|---|---|
| Filter time of setpoint feedforward control at servo control (SC) and sensorless vector control (SLVC)<br>• The setpoint feedforward control requires the entry of the moment of inertia in C00273. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | ms | 1000.0 | **1.0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00276**

| Parameter \| Name: **C00276 \| SC: Max. output voltage** | | | Data type: UNSIGNED_8 Index: $24299_d$ = $5EEB_h$ |
|---|---|---|---|
| Maximum output voltage at servo control (SC)<br>• Regarding the current DC-bus voltage. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 80 | % | 99 | **95 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00280**

| Parameter \| Name: **C00280 \| SC: Filter time const. DC detection** | | | Data type: UNSIGNED_16 Index: $24295_d$ = $5EE7_h$ |
|---|---|---|---|
| Filter time constant for DC-bus voltage filtering<br>• The filter time constant is e.g. used for field weakening control at servo control (SC). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | ms | 1000 | **25 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00290**

| Parameter \| Name: **C00290 \| RCOM error counter** | Data type: UNSIGNED_16 Index: $24285_d$ = $5EDD_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00291**

| Parameter \| Name: **C00291 \| Error type RCOM** | Data type: UNSIGNED_8 Index: $24284_d$ = $5EDC_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00295**

| Parameter \| Name: **C00295 \| savezycle memory modul** | Data type: INTEGER_32 Index: $24280_d$ = $5ED8_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00296**

| Parameter \| Name: **C00296 \| ICOM error number** | Data type: UNSIGNED_16 Index: $24279_d$ = $5ED7_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00297**

| Parameter \| Name: **C00297 \| Counter Receive Error Isr** | Data type: UNSIGNED_8 Index: $24278_d$ = $5ED6_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00301**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00301 \| DebugAccess** | Index: $24274_d = 5ED2_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00302**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C00302 \| Internal Commands** | Index: $24273_d = 5ED1_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00304**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00304 \| Password1** | Index: $24271_d = 5ECF_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00305**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00305 \| Password2** | Index: $24270_d = 5ECE_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00306**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00306 \| Debug address** | Index: $24269_d = 5ECD_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00307**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00307 \| Debug value** | Index: $24268_d = 5ECC_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00308**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00308 \| PartitionOffset** | Index: $24267_d = 5ECB_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00309**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C00309 \| PartitionSel** | Index: $24266_d = 5ECA_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00310**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00310 \| PartitionValue** | Index: $24265_d = 5EC9_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00311**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00311 \| Runtime measurement** | Index: $24264_d = 5EC8_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00312**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00312 \| System runtimes** | Index: $24263_d = 5EC7_h$ |

From version 02.00.00

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0.000 | µs | 1310.700 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00312/1 | 0.000 µs | | System runtime reserve |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C00313**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C00313 \| LS_DataAccess: Activation** | Index: $24262_d = 5EC6_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00314**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00314 \| LS_DataAccess: Address access** | Index: $24261_d = 5EC5_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00315**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00315 \| SystemFail-Adr** | Index: $24260_d = 5EC4_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00316**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00316 \| SystemFail-Info** | Index: $24259_d = 5EC3_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00317**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00317 \| WatchdogTimeMax** | Index: $24258_d = 5EC2_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00320**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00320 \| Debug information** | Index: $24255_d = 5EBF_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00321**

| Parameter \| Name: **C00321 \| Main program runtime** | | | Data type: UNSIGNED_16 Index: $24254_d = 5EBE_h$ |
|---|---|---|---|
| Display of the current and the maximum runtime of the main program in the inverter | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00321/1 | 0 ms | | Curr. runtime of main program |
| C00321/2 | 0 ms | | Max. runtime of main program |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00322**

| Parameter \| Name: **C00322 \| Transmission mode CAN TxPDOs** | | | Data type: UNSIGNED_8 Index: $24253_d = 5EBD_h$ |
|---|---|---|---|
| TPDO transmission type according to DS301 V4.02<br>• The following transmission modes are supported:<br>• 0: Synchronous and acyclic<br>• 1 ... 240: Synchronous and cyclic<br>• 252: Synchronous - RTR only<br>• 253: Asynchronous - RTR only<br>• 254: Asynchronous - manufacturer-specific<br>• 255: Asynchronous - device-profile specific<br>• The basic setting for all PDOs is "Asynchronous - manufacturer-specific" (254).<br>• Illustration of the CANopen objects I-1800/2 ... I-1803/2 (see DS301 V4.02).<br>▸ "CAN on board" system bus | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 255 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00322/1 | 254 | | Transmission mode CAN1 OUT |
| C00322/2 | 254 | | Transmission mode CAN2 OUT |
| C00322/3 | 254 | | Transmission mode CAN3 OUT |
| C00322/4 | 254 | | Transmission mode CAN4 OUT<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00323**

| Parameter \| Name: **C00323 \| Transmission mode CAN Rx PDOs** | | | Data type: UNSIGNED_8 Index: $24252_d = 5EBC_h$ |
|---|---|---|---|
| RPDO transmission type according to DS301 V4.02<br>• In the case of the RPDO serves as monitoring setting in the case of sync-controlled PDOs.<br>• The following transmission modes are supported:<br>• 0: Synchronous and acyclic<br>• 1 ... 240: Synchronous and cyclic<br>• 252: Synchronous - RTR only<br>• 253: Asynchronous - RTR only<br>• 254: Asynchronous - manufacturer-specific<br>• 255: Asynchronous - device-profile specific<br>• The basic setting for all PDOs is "Asynchronous - manufacturer-specific" (254).<br>• Illustration of the CANopen objects I-1400/2 ... I-1403/2 (see DS301 V4.02).<br>▸ "CAN on board" system bus | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 255 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00323/1 | 254 | | Transmission mode CAN1 IN |
| C00323/2 | 254 | | Transmission mode CAN2 IN |
| C00323/3 | 254 | | Transmission mode CAN3 IN |
| C00323/4 | 254 | | Transmission mode CAN4 IN<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00324**

| Parameter \| Name: **C00324 \| CAN transmission blocking time** | | | Data type: UNSIGNED_16 Index: $24251_d = 5EBB_h$ |
|---|---|---|---|
| Blocking time for the transmission of the emergency telegram and the process data<br><br>**Note:**<br>If the "Asynchronous - manufacturer-specific/device profile-specific" transmisson type is set, the transmission cycle timer is reset to 0 if event-controlled transmission has been triggered.<br>Example: Cycle time (C00356/x) = 500 ms, blocking time = 100 ms, data change sporadically:<br>• In the case of a sporadic data change < 500 ms, due to the blocking time set, transmission takes place every 100 ms (event-controlled transmission) as quickly as possible.<br>• In the case of a sporadic data change > 500 ms, due to the cycle time set, transmission takes place every 500 ms (cyclic transmission).<br>▸ "CAN on board" system bus | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 6500 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00324/1 | 0 ms | | CAN emergency blocking time |
| C00324/2 | 0 ms | | CAN1_OUT blocking time |
| C00324/3 | 0 ms | | CAN2_OUT blocking time |
| C00324/4 | 0 ms | | CAN3_OUT blocking time |
| C00324/5 | 0 ms | | CAN4_OUT blocking time<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00338**

| Parameter \| Name: **C00338 \| L_Arithmetik_1: Function** | Data type: UNSIGNED_8 Index: $24237_d = 5EAD_h$ |
|---|---|
| The L_Arithmetik_1 FB: Selection of internal arithmetics | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **Out = In1** |
| 1 | Out = In1 + In2 |
| 2 | nOut_a = nIn1_a - nIn2_a |
| 3 | Out = (In1 * In2) / 100% |
| 4 | nOut_a = (nIn1_a * 1%) / \|nIn2_a\| |
| 5 | nOut_a = (nIn1_a * 100%) / (100% - nIn2_a) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00339**

| Parameter \| Name: **C00339 \| L_Arithmetik_2: Function** | Data type: UNSIGNED_8 Index: $24236_d = 5EAC_h$ |
|---|---|
| The L_Arithmetik_2 FB: Selection of internal arithmetics | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **nOut_a = nIn1_a** |
| 1 | nOut_a = nIn1_a + nIn2_a |
| 2 | nOut_a = nIn1_a - nIn2_a |
| 3 | nOut_a = (nIn1_a * nIn2_a) / 100% |
| 4 | nOut_a = (nIn1_a * 1%) / \|nIn2_a\| |
| 5 | nOut_a = (nIn1_a * 100%) / (100% - nIn2_a) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00341**

| Parameter \| Name: **C00341 \| CAN management - error configuration** | | | Data type: UNSIGNED_16 Index: $24234_d = 5EAA_h$ |
|---|---|---|---|
| Selection of the events for which the *bFail* error output of the LS_CANMAnagement SB must be set to TRUE. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | BusOff_MsgErr | | |
| Bit 1 ☐ | Warning | | |
| Bit 2 ☐ | NodeStopped | | |
| Bit 3 ☐ | HeartBeatEvent | | |
| Bit 4 ☐ | CAN1_In_Überw. | | |
| Bit 5 ☐ | CAN2_In_Überw. | | |
| Bit 6 ☐ | CAN3_In_Überw. | | |
| Bit 7 ☐ | CAN4_In_Überw. | | From version 15.00.00 |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00342**

| Parameter \| Name: **C00342 \| CAN decoupling PDOInOut** | | Data type: UNSIGNED_16 Index: $24233_d = 5EA9_h$ |
|---|---|---|
| Configuration defining the events that lead to a decoupling of the process data words. ▸ Configuring exception handling of the CAN PDOs | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |

| **Value is bit-coded:** | | **Info** |
|---|---|---|
| Bit 0 | BusOff_MsgErr | |
| Bit 1 | Warning | |
| Bit 2 | NodeStopped | |
| Bit 3 | HeartBeatEvent | |
| Bit 4 | CAN1_In_Überw. | |
| Bit 5 | CAN2_In_Überw. | |
| Bit 6 | CAN3_In_Überw. | |
| Bit 7 | CAN4_In_Überw. | From version 15.00.00 |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Trouble | |
| Bit 15 | Fault | |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C00342/1 | 0x0000 | CAN decoupling PDO_In from the bus |
| C00342/2 | 0x0000 | CAN decoupling PDO_Out from the appl. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

**C00343**

| Parameter \| Name: **C00343 \| LP_CanIn decoupling value** | | | Data type: UNSIGNED_16 Index: $24232_d = 5EA8_h$ |
|---|---|---|---|
| Definition of the value the process data words are to have in the decoupled state. ▸ Configuring exception handling of the CAN PDOs | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00343/1 | 0 | | LP_CanIn1:wCtrl DiscVal |
| C00343/2 | 0 | | LP_CanIn1:wIn2 DiscVal |
| C00343/3 | 0 | | LP_CanIn1:wIn3 DiscVal |
| C00343/4 | 0 | | LP_CanIn1:wIn4 DiscVal |
| C00343/5 | 0 | | LP_CanIn2:wIn1 DiscVal |
| C00343/6 | 0 | | LP_CanIn2:wIn2 DiscVal |
| C00343/7 | 0 | | LP_CanIn2:wIn3 DiscVal |
| C00343/8 | 0 | | LP_CanIn2:wIn4 DiscVal |
| C00343/9 | 0 | | LP_CanIn3:wIn1 DiscVal |
| C00343/10 | 0 | | LP_CanIn3:wIn2 DiscVal |
| C00343/11 | 0 | | LP_CanIn3:wIn3 DiscVal |
| C00343/12 | 0 | | LP_CanIn3:wIn4 DiscVal |
| C00343/13 | 0 | | LP_CanIn4:wIn1 DiscVal<br>• From version 15.00.00 |
| C00343/14 | 0 | | LP_CanIn4:wIn2 DiscVal<br>• From version 15.00.00 |
| C00343/15 | 0 | | LP_CanIn4:wIn3 DiscVal<br>• From version 15.00.00 |
| C00343/16 | 0 | | LP_CanIn4:wIn4 DiscVal<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00344**

| Parameter \| Name: **C00344 \| LP_CanOut decoupling value** | | | Data type: UNSIGNED_16 Index: $24231_d = 5EA7_h$ |
|---|---|---|---|
| Definition of the value the process data words are to have in the decoupled state. ▸ Configuring exception handling of the CAN PDOs | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00344/1 | 0 | | LP_CanOut1:wState DiscVal |
| C00344/2 | 0 | | LP_CanOut1:wOut2 DiscVal |
| C00344/3 | 0 | | LP_CanOut1:wOut3 DiscVal |
| C00344/4 | 0 | | LP_CanOut1:wOut4 DiscVal |
| C00344/5 | 0 | | LP_CanOut2:wOut1 DiscVal |
| C00344/6 | 0 | | LP_CanOut2:wOut2 DiscVal |
| C00344/7 | 0 | | LP_CanOut2:wOut3 DiscVal |
| C00344/8 | 0 | | LP_CanOut2:wOut4 DiscVal |
| C00344/9 | 0 | | LP_CanOut3:wOut1 DiscVal |
| C00344/10 | 0 | | LP_CanOut3:wOut2 DiscVal |
| C00344/11 | 0 | | LP_CanOut3:wOut3 DiscVal |
| C00344/12 | 0 | | LP_CanOut3:wOut4 DiscVal |
| C00344/13 | 0 | | LP_CanOut4:wOut1 DiscVal<br>• From version 15.00.00 |
| C00344/14 | 0 | | LP_CanOut4:wOut2 DiscVal<br>• From version 15.00.00 |
| C00344/15 | 0 | | LP_CanOut4:wOut3 DiscVal<br>• From version 15.00.00 |
| C00344/16 | 0 | | LP_CanOut4:wOut4 DiscVal<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00345**

| Parameter \| Name: **C00345 \| CAN error status** | | Data type: UNSIGNED_8 Index: $24230_d = 5EA6_h$ |
|---|---|---|
| ▸ "CAN on board" system bus | | |
| **Selection list** (read only) | | |
| 0 | No Error | |
| 1 | Warning ErrActive | |
| 2 | Warning ErrPassive | |
| 3 | Bus off | |
| 4 | Reserved | |
| 5 | Reserved | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00347**

| Parameter \| Name: **C00347 \| CAN status HeartBeat producer** | | Data type: UNSIGNED_8 Index: $24228_d = 5EA4_h$ |
|---|---|---|
| | | ▸ Heartbeat protocol |
| **Selection list** | | |
| 0 | Boot-up | |
| 4 | Stopped | |
| 5 | Operational | |
| 127 | Pre-Operat. | |
| 250 | Failed | |
| 255 | NoResponse | |
| **Subcodes** | | **Info** |
| C00347/1 | | Status node 1 ... 15 |
| C00347/... | | |
| C00347/15 | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00349**

| Parameter \| Name: **C00349 \| CAN setting - DIP switch** | | Data type: UNSIGNED_16 Index: $24226_d = 5EA2_h$ |
|---|---|---|
| DIP switch setting during last mains power-on | | ▸ "CAN on board" system bus |
| **Display area** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** | | |
| Bit 0 | Node address 1 | |
| Bit 1 | Node address 2 | |
| Bit 2 | Node address 4 | |
| Bit 3 | Node address 8 | |
| Bit 4 | Node address 16 | |
| Bit 5 | Node address 32 | |
| Bit 6 | Node address 64 | |
| Bit 7 | Baud rate 1 | |
| Bit 8 | Baud rate 2 | |
| Bit 9 | Baud rate 4 | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | DIP switch at 24V-ON accepted | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | |

**C00350**

| Parameter \| Name: **C00350 \| CAN node address** | | | Data type: UNSIGNED_8 Index: $24225_d = 5EA1_h$ |
|---|---|---|---|
| Setting of the node address via parameters<br>• The node address can only be parameterised if the node address "0" is set via the DIP switches.<br>• A change in the node address will not be effective until a CAN Reset Node is performed.<br>▸ "CAN on board" system bus | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | | 127 | **1** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00351**

| Parameter \| Name: **C00351 \| CAN baud rate** | Data type: UNSIGNED_8 Index: $24224_d = 5EA0_h$ |
|---|---|
| Setting of the baud rate via parameters<br>• The baud rate can only be parameterised if the baud rate "0" is set via the DIP switches.<br>• A change in the baud rate will not be effective until a CAN Reset Node is performed.<br>▸ "CAN on board" system bus | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **500 kbps** |
| 1 | 250 kbps |
| 2 | 125 kbps |
| 3 | 50 kbps |
| 4 | 1000 kbps |
| 5 | 20 kbps |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | |

**C00352**

| Parameter \| Name: **C00352 \| CAN slave/master** | Data type: UNSIGNED_8 Index: $24223_d = 5E9F_h$ |
|---|---|
| The drive starts as CAN master after mains switching if a value of "1" has been entered and saved here.<br>▸ "CAN on board" system bus | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **slave** |
| 1 | Master |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | |

**C00353**

| Parameter \| Name: **C00353 \| CAN IN/OUT COBID source** | | Data type: UNSIGNED_8 Index: $24222_d = 5E9E_h$ |
|---|---|---|
| Identifier assignment procedure for the CANx In/Out process data<br>▸ System bus "CAN on board": Identifier of the process data objects | | |
| **Selection list** | | **Info** |
| 0 | COBID = C0350 + LenzeBaseID | COBID = node address + LenzeBaseID |
| 1 | COBID = C0350 + CANBaseID | COBID = node address + CANBaseID (C00354/x) |
| 2 | COBID = C0354/x | COBID = direct setting from C00354/x |
| **Subcodes** | **Lenze setting** | **Info** |
| C00353/1 | 1: COBID = C0350 + CANBaseID | COBID source CAN1_IN/OUT |
| C00353/2 | 1: COBID = C0350 + CANBaseID | COBID source CAN2_IN/OUT |
| C00353/3 | 1: COBID = C0350 + CANBaseID | COBID source CAN3_IN/OUT |
| C00353/4 | 1: COBID = C0350 + CANBaseID | COBID source CAN4_IN/OUT<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C00354**

| Parameter \| Name: **C00354 \| COBID** | | | Data type: UNSIGNED_32 Index: $24221_d = 5E9D_h$ |
|---|---|---|---|
| Setting of the default COBID according to CANopen<br>• A change in the COBID will not be effective until a CAN reset node is performed.<br>▸ System bus "CAN on board": Identifier of the process data objects | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x00000000 | | 0xFFFFFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | COBID Bit0 | | • Bit 0 ... 10: COB-ID<br>• Bit 11 ... 30: Reserved<br>• Bit 31: PDO invalid (is not transmitted) |
| ... | ... | | |
| Bit 31 | PDO invalid | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00354/1 | 0x00000201 | | COBID CAN1_IN |
| C00354/2 | 0x00000181 | | COBID CAN1_OUT |
| C00354/3 | 0x00000301 | | COBID CAN2_IN |
| C00354/4 | 0x00000281 | | COBID CAN2_OUT |
| C00354/5 | 0x00000401 | | COBID CAN3_IN |
| C00354/6 | 0x00000381 | | COBID CAN3_OUT |
| C00354/7 | 0x00000501 | | COBID CAN4_IN<br>• From version 15.00.00 |
| C00354/8 | 0x00000481 | | COBID CAN4_OUT<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT | | | |

**C00355**

| Parameter \| Name: **C00355 \| Active COBID** | | | Data type: UNSIGNED_16 Index: $24220_d = 5E9C_h$ |
|---|---|---|---|
| Display of the COBID of the PDOs that is active in the CAN stack<br>▸ System bus "CAN on board": Identifier of the process data objects | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 2047 | |
| **Subcodes** | | | **Info** |
| C00355/1 | | | Active COBID CAN1_IN |
| C00355/2 | | | Active COBID CAN1_OUT |
| C00355/3 | | | Active COBID CAN2_IN |
| C00355/4 | | | Active COBID CAN2_OUT |
| C00355/5 | | | Active COBID CAN3_IN |
| C00355/6 | | | Active COBID CAN3_OUT |
| C00355/7 | | | Active COBID CAN4_IN<br>• From version 15.00.00 |
| C00355/8 | | | Active COBID CAN4_OUT<br>• From version 15.00.00 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00356**

| Parameter \| Name: **C00356 \| CAN time settings** | | | Data type: UNSIGNED_16 Index: $24219_d = 5E9B_h$ |
|---|---|---|---|
| Different time settings for the CAN interface<br>▸ "CAN on board" system bus | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 65000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00356/1 | 3000 ms | | CAN delay boot-up - operat.<br>• Delay time for NMT state change from "Boot-up" to "Operational". |
| C00356/2 | 0 ms | | CAN2_OUT cycle time |
| C00356/3 | 0 ms | | CAN3_OUT cycle time |
| C00356/4 | 0 ms | | CANx_OUT operat. - 1.transmit<br>• When the "Operational" NMT status is reached, the delay time set here is started. After the delay time has been elapsed, the PDOs CAN2_OUT and CAN3_OUT are sent for the first time. |
| C00356/5 | 0 ms | | CAN1_OUT cycle time |
| C00356/6 | 0 ms | | CAN4_OUT cycle time<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00357**

| Parameter \| Name: C00357 \| CAN monitoring times | | | Data type: UNSIGNED_16 Index: $24218_d$ = $5E9A_h$ |
|---|---|---|---|
| Mapping of the RPDO event time (see DS301 V4.02)<br>• If a value unequal to "0" is entered, the RPDO is not expected before the set time has expired.<br>• If the RPDO is not received within the expected time, the response set in C00593/1...4 will be triggered.<br>▸ "CAN on board" system bus | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 65000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00357/1 | 3000 ms | | CAN1_IN monitoring time |
| C00357/2 | 3000 ms | | CAN2_IN monitoring time |
| C00357/3 | 3000 ms | | CAN3_IN monitoring time |
| C00357/4 | 3000 ms | | CAN4_IN monitoring time<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00358**

| Parameter \| Name: C00358 \| CANx_OUT data length | | | Data type: UNSIGNED_8 Index: $24217_d$ = $5E99_h$ |
|---|---|---|---|
| Setting of the data length for TX PDOs<br>▸ "CAN on board" system bus | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | | 8 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00358/1 | 8 | | CAN1_OUT data length |
| C00358/2 | 8 | | CAN2_OUT data length |
| C00358/3 | 8 | | CAN3_OUT data length |
| C00358/4 | 8 | | CAN4_OUT data length<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00359**

| Parameter \| Name: C00359 \| CAN status | | Data type: UNSIGNED_8 Index: $24216_d$ = $5E98_h$ |
|---|---|---|
| ▸ "CAN on board" system bus | | |
| **Selection list** (read only) | | |
| 0 | Operational | |
| 1 | Pre-Operat. | |
| 2 | Reserved | |
| 3 | Reserved | |
| 4 | BootUp | |
| 5 | Stopped | |
| 6 | Reserved | |
| 7 | Reset | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C00360**

| Parameter \| Name: **C00360 \| CAN telegram counter** | | | Data type: UNSIGNED_16 Index: $24215_d = 5E97_h$ |
|---|---|---|---|
| | | | ▸ "CAN on board" system bus |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | | | **Info** |
| C00360/1 | | | All PDO/SDO sent |
| C00360/2 | | | All PDO/SDO received |
| C00360/3 | | | Telegram counter CAN1_OUT |
| C00360/4 | | | Telegram counter CAN2_OUT |
| C00360/5 | | | Telegram counter CAN3_OUT |
| C00360/6 | | | Telegram counter SDO1 OUT |
| C00360/7 | | | Telegram counter SDO2 OUT |
| C00360/8 | | | Telegram counter CAN1_IN |
| C00360/9 | | | Telegram counter CAN2_IN |
| C00360/10 | | | Telegram counter CAN3_IN |
| C00360/11 | | | Telegram counter SDO1 IN |
| C00360/12 | | | Telegram counter SDO2 IN |
| C00360/13 | | | Telegram counter CAN4_OUT<br>• From version 15.00.00 |
| C00360/14 | | | Telegram counter CAN4_IN<br>• From version 15.00.00 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00364**

| Parameter \| Name: **C00364 \| CAN MessageError** | | Data type: UNSIGNED_8 Index: $24211_d = 5E93_h$ |
|---|---|---|
| | | ▸ "CAN on board" system bus |
| **Selection list** (read only) | | |
| 0 | No Error | |
| 1 | StuffError | |
| 2 | FormError | |
| 3 | AckError | |
| 4 | Bit1Error | |
| 5 | Bit0Error | |
| 6 | CRCError | |
| 7 | Reserved | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C00366**

| Parameter \| Name: **C00366 \| Number of CAN SDO channels** | Data type: UNSIGNED_8 Index: $24209_d = 5E91_h$ |
|---|---|

Selection of the number of active parameter data channels
- In the Lenze setting, only the parameter data channel 1 is activated according to CANopen.
- In order to activate both parameter data channels, set the selection "2 SDO Lenze".

▸ "CAN on board" system bus

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **1 SDO CANOpen** |
| 1 | 2 SDO Lenze |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C00367**

| Parameter \| Name: **C00367 \| CAN SYNC Rx identifier** | Data type: UNSIGNED_16 Index: $24208_d = 5E90_h$ |
|---|---|

Identifier by means of which the sync slave is to receive sync telegrams.
- Mapping of the CANopen object I-1005 (see DS301 V4.02).

▸ "CAN on board" system bus

| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
|---|---|---|---|
| 0x0000 | | 0xFFFF | **0x0080** (decimal: 128) |

| **Value is bit-coded: (☑ = bit set)** | |
|---|---|
| Bit 0 ☐ | COBID Bit0 |
| Bit 1 ☐ | COBID Bit1 |
| Bit 2 ☐ | COBID Bit2 |
| Bit 3 ☐ | COBID Bit3 |
| Bit 4 ☐ | COBID Bit4 |
| Bit 5 ☐ | COBID Bit5 |
| Bit 6 ☐ | COBID Bit6 |
| Bit 7 ☑ | COBID Bit7 |
| Bit 8 ☐ | COBID Bit8 |
| Bit 9 ☐ | COBID Bit9 |
| Bit 10 ☐ | COBID Bit10 |
| Bit 11 ☐ | Reserved |
| Bit 12 ☐ | Reserved |
| Bit 13 ☐ | Reserved |
| Bit 14 ☐ | Reserved |
| Bit 15 ☐ | Reserved |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT

**C00368**

| Parameter \| Name: **C00368 \| CAN SYNC Tx identifier** | | | Data type: UNSIGNED_16 Index: $24207_d = 5E8F_h$ |
|---|---|---|---|
| Identifier by means of which the sync master is to transmit sync telegrams.<br>• Mapping of the CANopen object I-1005 (see DS301 V4.02).<br>▸ "CAN on board" system bus | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0080** (decimal: 128) |
| **Value is bit-coded: (☑ = bit set)** | | | |
| Bit 0 ☐ | COBID Bit0 | | |
| Bit 1 ☐ | COBID Bit1 | | |
| Bit 2 ☐ | COBID Bit2 | | |
| Bit 3 ☐ | COBID Bit3 | | |
| Bit 4 ☐ | COBID Bit4 | | |
| Bit 5 ☐ | COBID Bit5 | | |
| Bit 6 ☐ | COBID Bit6 | | |
| Bit 7 ☑ | COBID Bit7 | | |
| Bit 8 ☐ | COBID Bit8 | | |
| Bit 9 ☐ | COBID Bit9 | | |
| Bit 10 ☐ | COBID Bit10 | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Sync-transmit off | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT | | | |

**C00369**

| Parameter \| Name: **C00369 \| CAN sync transmission cycle time** | | | Data type: UNSIGNED_16 Index: $24206_d = 5E8E_h$ |
|---|---|---|---|
| Cycle during which the sync master is to transmit sync telegrams.<br>• If "0 ms" is set (Lenze setting), no sync telegrams are generated.<br>• Mapping of the CANopen object I-1006 (see DS301 V4.02).<br>▸ "CAN on board" system bus | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | ms | 65000 | **0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00370**

| Parameter \| Name: **C00370 \| SyncTxRxTimes** | | | Data type: INTEGER_16 Index: $24205_d$ = $5E8D_h$ |
|---|---|---|---|
| | | | "CAN on board" system bus |
| **Display area** (min. value \| unit \| max. value) | | | |
| -1310 | µs | 1310 | |
| **Subcodes** | | | **Info** |
| C00370/1 | | | CAN Sync instant of transmission |
| C00370/2 | | | Sync instant of reception |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00371**

| Parameter \| Name: **C00371 \| CAN ErrorCode** | | | Data type: UNSIGNED_16 Index: $24204_d$ = $5E8C_h$ |
|---|---|---|---|
| From version 13.00.00 | | | |
| | | | "CAN on board" system bus |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | | | **Info** |
| C00371/1 | | | CAN ErrorCode |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00372**

| Parameter \| Name: **C00372 \| CAN_Tx_Rx_Error** | | | Data type: UNSIGNED_8 Index: $24203_d$ = $5E8B_h$ |
|---|---|---|---|
| | | | "CAN on board" system bus |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 255 | |
| **Subcodes** | | | **Info** |
| C00372/1 | | | CAN Tx_Error |
| C00372/2 | | | CAN Rx_Error |
| C00372/3 | | | CAN Tx_Overflow |
| C00372/4 | | | CAN Rx_Overflow |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00381**

| Parameter \| Name: **C00381 \| CAN Heartbeat producer time** | | | Data type: UNSIGNED_16 Index: $24194_d$ = $5E82_h$ |
|---|---|---|---|
| Time interval for the transmission of the heartbeat telegram to the consumer(s).<br>• The heartbeat telegram is sent automatically as soon as a time > 0 ms is set.<br>• Mapping of the CANopen object I-1017 (see DS301 V4.02). | | | |
| | | | Heartbeat protocol |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | ms | 65535 | **0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00385**

| Parameter \| Name: **C00385 \| CAN node addr. HeartBeat producer** | | | Data type: UNSIGNED_8 Index: $24190_d = 5E7E_h$ |
|---|---|---|---|
| The subcodes represent the nodes to be monitored by heartbeat. ▸ Heartbeat protocol | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 127 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00385/1 | 0 | | CAN node address HeartBeat producer 1 ... 15 |
| C00385/... | | | |
| C00385/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00386**

| Parameter \| Name: **C00386 \| CAN HeartBeat ConsumerTime** | | | Data type: UNSIGNED_16 Index: $24189_d = 5E7D_h$ |
|---|---|---|---|
| The subcodes represent the nodes to be monitored by heartbeat. ▸ Heartbeat protocol | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00386/1 | 0 ms | | ConsumerTime HeartBeat producer 1 ... 15 |
| C00386/... | | | |
| C00386/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00387**

| Parameter \| Name: **C00387 \| CAN-GatewayAddr** | | | Data type: UNSIGNED_8 Index: $24188_d = 5E7C_h$ |
|---|---|---|---|
| From version 12.00.00 ▸ CAN gateway | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 127 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00387/1 | 0 | | CAN_Gateway: Addr. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00400**

| Parameter \| Name: **C00400 \| LS_PulseGenerator** | | | Data type: UNSIGNED_16 Index: $24175_d = 5E6F_h$ |
|---|---|---|---|
| Time setting of the pulse to be output by the SB LS_PulseGenerator<br><br>**Note:**<br>The real length of the low or high level at the *bSquareWave* output is always 1 ms higher than set here in subcode 1 or 2. If "0 ms" is set, a level with a length of 1 ms is created length, for instance, and if "1000 ms" is set, a level with a length of 1001 ms is created. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00400/1 | 1000 ms | | Length of the low level (break) + 1 ms |
| C00400/2 | 1000 ms | | Length of the high level + 1 ms |
| C00400/3 | 100 ms | | Delay time for status signal *bFirstCycleDone*<br>• The *bFirstCycleDone* status signal is set to TRUE when the first task cycle is complete and the time set here has expired. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00401**

| Parameter \| Name: **C00401 \| CANxInOut: Inversion** | | | Data type: UNSIGNED_16 Index: $24174_d = 5E6E_h$ |
|---|---|---|---|
| This parameter serves to invert the control/status bits of the CAN port blocks.<br>▸ CAN port block | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Active | | Bit set = bit is inverted |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00401/1 | 0x0000 | | Inversion of LP_CanIn1.bCtrl1_B0...15 |
| C00401/2 | 0x0000 | | Inversion of LP_CanOut1.bState1_B0...15 |
| C00401/3 | 0x0000 | | Inversion of LP_CanIn2.bIn1_B0...15 |
| C00401/4 | 0x0000 | | Inversion of LP_CanOut2.bOut1_B0...15 |
| C00401/5 | 0x0000 | | Inversion of LP_CanIn3.bIn1_B0...15 |
| C00401/6 | 0x0000 | | Inversion of LP_CanOut3.bOut1_B0...15 |
| C00401/7 | 0x0000 | | Inversion of LP_CanIn4.bIn1_B0...15<br>• From version 15.00.00 |
| C00401/8 | 0x0000 | | Inversion of LP_CanOut4.bOut1_B0...15<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00407**

| Parameter \| Name: **C00407 \| LP_CanIn Mapping** | | | Data type: UNSIGNED_32 Index: $24168_d = 5E68_h$ |
|---|---|---|---|
| From version 12.00.00<br>Mapping for port blocks LP_CanIn1...4<br>• These settings are ORed with the mapping settings for the single words in C00409/x.<br>▸ CAN port block | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 4294967295 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00407/1 | 0 | | LP_CanIn1:dwIn12 MapVal |
| C00407/2 | 0 | | LP_CanIn1:dwIn34 MapVal |
| C00407/3 | 0 | | LP_CanIn2:dwIn12 MapVal |
| C00407/4 | 0 | | LP_CanIn2:dwIn34 MapVal |
| C00407/5 | 0 | | LP_CanIn3:dwIn12 MapVal |
| C00407/6 | 0 | | LP_CanIn3:dwIn34 MapVal |
| C00407/7 | 0 | | LP_CanIn4:dwIn12 MapVal<br>• From version 15.00.00 |
| C00407/8 | 0 | | LP_CanIn4:dwIn34 MapVal<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00408**

| Parameter \| Name: **C00408 \| LP_CanIn mapping selection** | | Data type: UNSIGNED_8 Index: $24167_d = 5E67_h$ |
|---|---|---|
| Selection of the mapping source for port blocks LP_CanIn1...4<br>▸ CAN port block | | |
| **Selection list** | | **Info** |
| 0 | CanIn | CanIn |
| 1 | Par.C409\|C407 | Mapping configured in C00409 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00408/1 | 0: CanIn | Mapping selection LP_CanIn1 |
| C00408/2 | 0: CanIn | Mapping selection LP_CanIn2 |
| C00408/3 | 0: CanIn | Mapping selection LP_CanIn3 |
| C00408/4 | 0: CanIn | Mapping selection LP_CanIn4<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00409**

| Parameter \| Name: **C00409 \| LP_CanIn Mapping** | | Data type: UNSIGNED_16 Index: $24166_d = 5E66_h$ |
|---|---|---|
| Mapping for port blocks LP_CanIn1...4<br>• These settings are ORed with the mapping settings for the double words in C00407/x.<br>▸ CAN port block | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | | 65535 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00409/1 | 0 | LP_CanIn1:wCtrl MapVal |
| C00409/2 | 0 | LP_CanIn1:wIn2 MapVal |
| C00409/3 | 0 | LP_CanIn1:wIn3 MapVal |
| C00409/4 | 0 | LP_CanIn1:wIn4 MapVal |
| C00409/5 | 0 | LP_CanIn2:wIn1 MapVal |
| C00409/6 | 0 | LP_CanIn2:wIn2 MapVal |
| C00409/7 | 0 | LP_CanIn2:wIn3 MapVal |
| C00409/8 | 0 | LP_CanIn2:wIn4 MapVal |
| C00409/9 | 0 | LP_CanIn3:wIn1MapVal |
| C00409/10 | 0 | LP_CanIn3:wIn2 MapVal |
| C00409/11 | 0 | LP_CanIn3:wIn3 MapVal |
| C00409/12 | 0 | LP_CanIn3:wIn4 MapVal |
| C00409/13 | 0 | LP_CanIn4:wIn1 MapVal<br>• From version 15.00.00 |
| C00409/14 | 0 | LP_CanIn4:wIn2 MapVal<br>• From version 15.00.00 |
| C00409/15 | 0 | LP_CanIn4:wIn3 MapVal<br>• From version 15.00.00 |
| C00409/16 | 0 | LP_CanIn4:wIn4 MapVal<br>• From version 15.00.00 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00410**

| Parameter \| Name: **C00410 \| L_SignalMonitor_a: Signal sources** | | Data type: UNSIGNED_16 Index: $24165_d = 5E65_h$ |
|---|---|---|
| The L_SignalMonitor_a FB: Selection of the signal sources | | |
| **Selection list** | | |
| See selection list - analog signals | | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00410/1 | 0: Not connected | Signal source for output *nOut1_a* |
| C00410/2 | 0: Not connected | Signal source for output *nOut2_a* |
| C00410/3 | 0: Not connected | Signal source for output *nOut3_a* |
| C00410/4 | 0: Not connected | Signal source for output *nOut4_a* |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00411**

| Parameter \| Name: **C00411 \| L_SignalMonitor_b: Signal sources** | | Data type: UNSIGNED_16 Index: $24164_d = 5E64_h$ |
|---|---|---|
| The L_SignalMonitor_b FB: Selection of the signal sources | | |
| **Selection list** | | |
| See selection list - digital signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00411/1 | 0: Not connected | Signal source for output *bOut1* |
| C00411/2 | 0: Not connected | Signal source for output *bOut2* |
| C00411/3 | 0: Not connected | Signal source for output *bOut3* |
| C00411/4 | 0: Not connected | Signal source for output *bOut4* |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00412**

| Parameter \| Name: **C00412 \| L_SignalMonitor_b: Inversion** | | | Data type: UNSIGNED_8 Index: $24163_d = 5E63_h$ |
|---|---|---|---|
| The L_SignalMonitor_b FB: Inversion of the binary outputs | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x00** (decimal: 0) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | bOut1 inverted | | Bit set = inversion active |
| Bit 1 ☐ | bOut2 inverted | | |
| Bit 2 ☐ | bOut3 inverted | | |
| Bit 3 ☐ | bOut4 inverted | | |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00413**

| Parameter \| Name: **C00413 \| L_SignalMonitor_a: Offs./gain** | | | Data type: INTEGER_16 Index: $24162_d = 5E62_h$ |
|---|---|---|---|
| The L_SignalMonitor_a FB: Gain and offset of the analog signals | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00413/1 | 0.00 % | | Offset for output *nOut1_a* |
| C00413/2 | 100.00 % | | Gain for output *nOut1_a* |
| C00413/3 | 0.00 % | | Offset for output *nOut2_a* |
| C00413/4 | 100.00 % | | Gain for output *nOut2_a* |
| C00413/5 | 0.00 % | | Offset for output *nOut3_a* |
| C00413/6 | 100.00 % | | Gain for output *nOut3_a* |
| C00413/7 | 0.00 % | | Offset for output *nOut4_a* |
| C00413/8 | 100.00 % | | Gain for output *nOut4_a* |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00417**

| Parameter \| Name: **C00417 \| Deactivate resolver error comp.** | | | Data type: UNSIGNED_8 Index: $24158_d = 5E5E_h$ |
|---|---|---|---|
| ▸ Encoder/feedback system: Resolver | | | |
| **Selection list** (Lenze setting printed in bold) | | | |
| **0** | **Resolver error comp. act.** | | |
| 1 | Resolver error comp. deact. | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00420**

| Parameter \| Name: **C00420 \| Number of encoder increments** | | | Data type: UNSIGNED_16 Index: $24155_d = 5E5B_h$ |
|---|---|---|---|
| Indication of the encoder constant<br>▸ Encoder/feedback system: Multi-Encoder | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | Incr./rev. | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00420/1 | 128 incr./rev. | | FreqIn12: Encoder increment |
| C00420/2 | 128 incr./rev. | | FreqIn67: Encoder increment |
| C00420/3 | 512 incr./rev. | | LS_MultiEncoder: Encoder increment |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00421**

| Parameter \| Name: **C00421 \| LS_MultiEncoder: Supply voltage** | | | Data type: UNSIGNED_16 Index: $24154_d = 5E5A_h$ |
|---|---|---|---|
| **Stop!**<br>Before connecting an encoder, make sure that the encoder supply voltage is set correctly!<br>If the set supply voltage exceeds the permissible supply voltage of the connected encoder, the encoder may be destroyed!<br>▸ Encoder/feedback system: Multi-Encoder | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 5.0 | V | 12.0 | **5.0 V** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00422**

| Parameter \| Name: **C00422 \| LS_MultiEncoder: Encoder type** | | Data type: UNSIGNED_8 Index: $24153_d = 5E59_h$ |
|---|---|---|
| | | ▸ Encoder/feedback system: Multi-Encoder |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Incremental encoder (TTL)** | |
| 1 | Sin/cos encoder | |
| 2 | Absolute value encoder (Hiperface) | |
| 3 | Absolute value encoder (external) | |
| 4 | Absolute value encoder (SSI) | |
| 5 | LF In/Out | From version 12.00.00<br>▸ Digital frequency coupling |
| 6 | Position encoder (Hiperface) | |
| 7 | Position encoder (external) | |
| 8 | Position encoder (SSI) | |
| 10 | Incremental encoder (TTL) with LF Out | |
| 11 | Sine/cos encoder with LF Out | |
| 12 | Absolute value encoder Hiperface with LF Out | |
| 16 | Position encoder Hiperface with LF Out | |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00423**

| Parameter \| Name: **C00423 \| DOx: Delay times** | | | Data type: UNSIGNED_16 Index: $24152_d = 5E58_h$ |
|---|---|---|---|
| Delay times for the digital output terminals | | | ▸ Digital output terminals |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 65.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00423/1 | 0.000 s | | Relay ON delay |
| C00423/2 | 0.000 s | | Relay OFF delay |
| C00423/3 | 0.000 s | | DO1 ON delay |
| C00423/4 | 0.000 s | | DO1 OFF delay |
| C00423/5 | 0.000 s | | DO2 ON delay |
| C00423/6 | 0.000 s | | DO2 OFF delay |
| C00423/7 | 0.000 s | | DO3 ON delay |
| C00423/8 | 0.000 s | | DO3 OFF delay |
| C00423/9 | 0.000 s | | DO "High Current" ON delay |
| C00423/10 | 0.000 s | | DO "High Current" OFF delay |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00424**

| Parameter \| Name: **C00424 \| Pulse form TTL encoder** | | Data type: UNSIGNED_8 Index: $24151_d = 5E57_h$ |
|---|---|---|
| | | ▸ Encoder/feedback system: Multi-Encoder |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **4x evaluation A/B** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00425**

| Parameter \| Name: **C00425 \| Encoder scanning time** | | Data type: UNSIGNED_8 Index: $24150_d = 5E56_h$ |
|---|---|---|
| Encoder sample time for the digital input terminals when configured as frequency inputs | | ▸ Using DI1(6) and DI2(7) as frequency inputs |
| **Selection list** | | |
| 0 | 1 ms | |
| 1 | 2 ms | |
| 2 | 5 ms | |
| 3 | 10 ms | |
| 4 | 20 ms | |
| 5 | 50 ms | |
| 6 | 100 ms | |
| 7 | 200 ms | |
| 8 | 500 ms | |
| 9 | 1000 ms | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00425/1 | 3: 10 ms | FreqIn12: Encoder scanning time<br>• Only active with edge-counting procedure (C00496 = 3). |
| C00425/2 | 3: 10 ms | FreqIn67: Encoder scanning time |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00426**

| Parameter \| Name: **C00426 \| SSI encoder: Data bits** | | | Data type: UNSIGNED_8 Index: $24149_d = 5E55_h$ |
|---|---|---|---|
| Example: Setting for an SSI encode with the coding "8192x4096":<br>• 8192 = $2^{13}$ = 13 bits Multiturn ➞ C00426/4 = 13<br>• 4096 = $2^{12}$ = 12 bits Singleturn ➞ C00426/2 = 12<br>• This results in a data word length of 25 bits ➞ C00426/1 = 25<br>• Start bit Singleturn = bit 0 ➞ C00426/3 = 0<br>• Start bit Multiturn = bit 12 ➞ C00426/5 = 12<br>▸ Encoder/feedback system: SSI encoder | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 32 | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00426/1 | 25 | SSI encoder: Data word length |
| C00426/2 | 13 | SSI encoder: Bits SingleTurn |
| C00426/3 | 0 | SSI encoder: Start bit Singleturn |
| C00426/4 | 12 | SSI encoder: Bits Multiturn |
| C00426/5 | 13 | SSI encoder: Start bit Multiturn |
| C00426/6 | 0 | SSI encoder: Status bit 1<br>• From version 02.00.00 |
| C00426/7 | 0 | SSI encoder: Shift of raw value<br>• From version 02.00.00 |
| C00426/8 | 0 | SSI encoder: Status bit 2<br>• From version 14.00.00 |
| C00426/9 | 0 | SSI encoder: Status bit 3<br>• From version 14.00.00 |

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00427**

| Parameter \| Name: **C00427 \| SSI encoder: Bit rate** | | Data type: UNSIGNED_8 Index: $24148_d = 5E54_h$ |
|---|---|---|
| ▸ Encoder/feedback system: SSI encoder | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **1** | **100 kbps** | |
| 2 | 200 kbps | |
| 3 | 300 kbps | |
| 4 | 400 kbps | |
| 5 | 500 kbps | |
| 6 | 750 kbps | |
| 7 | 1000 kbps | |

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00428**

| Parameter \| Name: **C00428 \| SSI encoder: Coding** | | Data type: UNSIGNED_8 Index: $24147_d = 5E53_h$ |
|---|---|---|
| | | ▸ Encoder/feedback system: SSI encoder |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Binary code** | |
| 1 | Gray code | |
| 2 | Position Gray | From version 14.00.00, SSI encoders are supported which transmit the position data with Gray coding and the status bits in binary format. |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00430**

| Parameter \| Name: **C00430 \| LS_MultiEncoder: Max. initialization time** | | | Data type: UNSIGNED_16 Index: $24145_d = 5E51_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ Encoder/feedback system: sin/cos absolute value encoder with HIPERFACE® protocol |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 500 | ms | 20000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00430/1 | 1500 ms | | Maximum time required by the encoder to respond to the read request of the inverter. |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00431**

| Parameter \| Name: **C00431 \| LS_MultiEncoder: bFail selection** | | | Data type: UNSIGNED_16 Index: $24144_d = 5E50_h$ |
|---|---|---|---|
| From version 13.00.00<br>Bit coded setting which events cause the *bFail* output to be set. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Encoder open circuit | | |
| Bit 1 | KTY open circuit | | |
| Bit 2 | Position beyond C1112/2..3 | | |
| Bit 3 | Angular drift error | | |
| Bit 4 | Communication error | | |
| Bit 5 | Hiperface position invalid | | From version 14.00.00 |
| Bit 6 | Linear data overflow SSI | | From version 14.00.00 |
| Bit 7 | Max. encoder speed | | From version 14.00.00 |
| Bit 8 | Hiperface init active | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00431/1 | 0x001D | | LS_MultiEncoder: bFail selection |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT | | | |

**C00432**

| Parameter \| Name: **C00432 \| LS_Resolver: bFail selection** | | | Data type: UNSIGNED_16 Index: $24143_d = 5E4F_h$ |
|---|---|---|---|
| From version 13.00.00<br>Bit coded setting which events cause the *bFail* output to be set. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Resolver wire breakage | | |
| Bit 1 | KTY open circuit | | |
| Bit 2 | Reserved | | |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00432/1 | 0x0001 | | LS_Resolver: bFail selection |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT | | | |

**C00434**

| Parameter \| Name: **C00434 \| OxU/I: Gain** | | | Data type: INTEGER_16 Index: $24141_d = 5E4D_h$ |
|---|---|---|---|
| Gain of the analog outputs<br>▸ Analog terminals | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00434/1 | 100.00 % | | O1U: Gain |
| C00434/2 | 100.00 % | | O2U: Gain |
| C00434/3 | 100.00 % | | O1I: Gain |
| C00434/4 | 100.00 % | | O2I: Gain |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00435**

| Parameter \| Name: **C00435 \| OxU/I: Offset** | | | Data type: INTEGER_16 Index: $24140_d = 5E4C_h$ |
|---|---|---|---|
| Offset of the analog outputs<br>▸ Analog terminals | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00435/1 | 0.00 % | | O1U: Offset |
| C00435/2 | 0.00 % | | O2U: Offset |
| C00435/3 | 0.00 % | | O1I: Offset |
| C00435/4 | 0.00 % | | O2I: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00436**

| Parameter \| Name: **C00436 \| OxU: Voltage** | | | Data type: INTEGER_16 Index: $24139_d = 5E4B_h$ |
|---|---|---|---|
| Display of the voltage at the analog outputs<br>▸ Analog terminals | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | V | 10.00 | |
| **Subcodes** | | | **Info** |
| C00436/1 | | | O1U: Voltage |
| C00436/2 | | | O2U: Voltage |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00437**

| Parameter \| Name: **C00437 \| OxI: Current** | | | Data type: INTEGER_32 Index: $24138_d = 5E4A_h$ |
|---|---|---|---|
| Display of the current at the analog outputs<br>▸ Analog terminals | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.000 | mA | 20.000 | |
| **Subcodes** | | | **Info** |
| C00437/1 | | | O1I: Current |
| C00437/2 | | | O2I: Current |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00439**

| Parameter \| Name: **C00439 \| OxU/I: Input value** | | | Data type: INTEGER_16 Index: $24136_d = 5E48_h$ |
|---|---|---|---|
| Display of the input values for the analog outputs<br>▸ Analog terminals | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00439/1 | | | O1U: Input value<br>• 100 % ≡ 10 V |
| C00439/2 | | | O2U: Input value<br>• 100 % ≡ 10 V |
| C00439/3 | | | O1I: Input value<br>• 100 % ≡ 20 mA |
| C00439/4 | | | O2I: Input value<br>• 100 % ≡ 20 mA |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00440**

| Parameter \| Name: **C00440 \| LS_AnalogIn1: PT1 time constant** | | | Data type: UNSIGNED_16 Index: $24135_d = 5E47_h$ |
|---|---|---|---|
| PT1 time constant (S-ramp time) for the analog inputs<br>▸ Analog terminals | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 1000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00440/1 | 10 ms | | PT1 rounding AnalogIn1 |
| C00440/2 | 10 ms | | PT1 rounding AnalogIn2 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00441**

| Parameter \| Name: **C00441 \| Decoupling AnalogOut** | | | Data type: UNSIGNED_16 Index: $24134_d = 5E46_h$ |
|---|---|---|---|
| Configuration defining the events that lead to a decoupling of the analog output terminals. ▸ Configuring exception handling of the output terminals | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded: (☑ = bit set)** | | | |
| Bit 0 ☐ | SafeTorqueOff | | |
| Bit 1 ☐ | ReadyToSwitchOn | | |
| Bit 2 ☐ | SwitchedOn | | |
| Bit 3 ☐ | Reserved | | |
| Bit 4 ☐ | Trouble | | |
| Bit 5 ☐ | Fault | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Fail CAN_Management | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00442**

| Parameter \| Name: **C00442 \| AOutx: Decoupling value** | | | Data type: INTEGER_16 Index: $24133_d = 5E45_h$ |
|---|---|---|---|
| Definition of the value the analog output terminals are to have in the decoupled state. ▸ Configuring exception handling of the output terminals | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00442/1 | 0.00 % | | AOut1_U: Decoupling value |
| C00442/2 | 0.00 % | | AOut2_U: Decoupling value |
| C00442/3 | 0.00 % | | AOut1_I: Decoupling value |
| C00442/4 | 0.00 % | | AOut2_I: Decoupling value |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00443**

| Parameter \| Name: **C00443 \| DIx: Level** | | | Data type: UNSIGNED_16 Index: $24132_d = 5E44_h$ |
|---|---|---|---|
| Bit coded display of the level of the digital inputs ▸ Digital input terminals | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | DI1 | | Bit set = HIGH level |
| Bit 1 | DI2 | | |
| Bit 2 | DI3 | | |
| Bit 3 | DI4 | | |
| Bit 4 | DI5 | | |
| Bit 5 | DI6 | | |
| Bit 6 | DI7 | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | CINH | | |
| **Subcodes** | | | **Info** |
| C00443/1 | | | DIx: Terminal level |
| C00443/2 | | | DIx: Output level |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00444**

| Parameter \| Name: **C00444 \| DOx: Level** | | | Data type: UNSIGNED_16 Index: $24131_d = 5E43_h$ |
|---|---|---|---|
| Bit coded display of the level of the digital outputs<br>▸ Digital output terminals | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Relay | | Bit set = HIGH level |
| Bit 1 | DO1 | | |
| Bit 2 | DO2 | | |
| Bit 3 | DO3 | | |
| Bit 4 | High current | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | | | **Info** |
| C00444/1 | | | DOx: Input level |
| C00444/2 | | | DOx: Terminal level |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00445**

| Parameter \| Name: **C00445 \| FreqInxx_nOut_v** | | | Data type: INTEGER_16 Index: $24130_d = 5E42_h$ |
|---|---|---|---|
| Display of the frequency input signals which are fed into the application.<br>▸ Using DI1(6) and DI2(7) as frequency inputs | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -32767 | Incr/ms | 32767 | |
| **Subcodes** | | | **Info** |
| C00445/1 | | | FreqIn12_nOut_v |
| C00445/2 | | | FreqIn67_nOut_v |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00446**

| Parameter \| Name: **C00446 \| FreqInxx_nOut_a** | | | Data type: INTEGER_16 Index: $24129_d = 5E41_h$ |
|---|---|---|---|
| Display of the frequency input signals which are fed into the application. ▸ Using DI1(6) and DI2(7) as frequency inputs | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00446/1 | | | FreqIn12_nOut_a |
| C00446/2 | | | FreqIn67_nOut_a |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00447**

| Parameter \| Name: **C00447 \| DigOut decoupling** | | | Data type: UNSIGNED_16 Index: $24128_d = 5E40_h$ |
|---|---|---|---|
| Configuration defining the events that lead to a decoupling of the digital output terminals. ▸ Configuring exception handling of the output terminals | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded: (☑ = bit set)** | | | |
| Bit 0 ☐ | SafeTorqueOff | | |
| Bit 1 ☐ | ReadyToSwitchOn | | |
| Bit 2 ☐ | SwitchedOn | | |
| Bit 3 ☐ | Reserved | | |
| Bit 4 ☐ | Trouble | | |
| Bit 5 ☐ | Fault | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Fail CAN_Management | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00448**

| Parameter \| Name: **C00448 \| DigOut decoupling value** | | | Data type: UNSIGNED_16 Index: $24127_d = 5E3F_h$ |
|---|---|---|---|
| Definition of the value the digital output terminals are to have in the decoupled state.<br>• Bit set = HIGH level<br>▸ Configuring exception handling of the output terminals | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded: (☑ = bit set)** | | | |
| Bit 0 ☐ | Relay_ON | | |
| Bit 1 ☐ | DigOut1_ON | | |
| Bit 2 ☐ | DigOut2_ON | | |
| Bit 3 ☐ | DigOut3_ON | | |
| Bit 4 ☐ | HighCurrent_ON | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00449**

| Parameter \| Name: **C00449 \| FreqInxx_dnOut_p** | | | Data type: INTEGER_32 Index: $24126_d = 5E3E_h$ |
|---|---|---|---|
| ▸ Output of the encoder position of the DI1/DI2 frequency input | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00449/1 | | | FreqIn12_dnOut_p |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00451**

| Parameter \| Name: **C00451 \| LS_MultiEncoder: bFail information** | | | Data type: UNSIGNED_16 Index: $24124_d = 5E3C_h$ |
|---|---|---|---|
| From version 14.00.00 | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Encoder open circuit | | |
| Bit 1 | KTY open circuit | | |
| Bit 2 | Position beyond C1112/2..3 | | |
| Bit 3 | Angular drift error | | |
| Bit 4 | Communication error | | |
| Bit 5 | Hiperface position invalid | | |
| Bit 6 | Linear data overflow SSI | | |
| Bit 7 | Max. encoder speed | | |
| Bit 8 | Hiperface init active | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | | | **Info** |
| C00451/1 | | | LS_MultiEncoder: bFail information |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☑ MOT | | | |

**C00452**

| Parameter \| Name: **C00452 \| LS_Resolver: bFail information** | | | Data type: UNSIGNED_16 Index: $24123_d = 5E3B_h$ |
|---|---|---|---|
| From version 14.00.00 | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Resolver wire breakage | | |
| Bit 1 | KTY open circuit | | |
| Bit 2 | Reserved | | |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | | | **Info** |
| C00452/1 | | | LS_Resolver: bFail information |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☑ MOT | | | |

**C00453**

| Parameter \| Name: **C00453 \| Keypad: Default manual jog** | | | Data type: UNSIGNED_32 Index: $24122_d = 5E3A_h$ |
|---|---|---|---|
| From version 14.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | | 16000.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00453/1 | 729.001 | | Keypad: Speed setpoint |
| C00453/2 | 56.002 | | Keypad: Display bar |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00455**

| Parameter \| Name: **C00455 \| FB_call table** | Data type: UNSIGNED_16 Index: $24120_d = 5E38_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00456**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C00456 \| Editor level** | Index: $24119_d = 5E37_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00458**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00458 \| SYS_call table** | Index: $24117_d = 5E35_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00459**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00459 \| SYS_Output table** | Index: $24116_d = 5E34_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00460**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C00460 \| Reserved** | Index: $24115_d = 5E33_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00461**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00461 \| Remote: Acceleration/deceleration time** | Index: $24114_d = 5E32_h$ |

▸ PC manual control

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0.000 | s | 999.999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00461/1 | 2.000 s | | Remote: Acceleration/deceleration time |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C00462**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00462 \| Remote: Control** | Index: $24113_d = 5E31_h$ |

▸ PC manual control

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00462/1 | 0 | | Remote: Control mode |
| C00462/2 | 0 | | Remote: Monitoring counter |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00463**

| Parameter \| Name: **C00463 \| Remote: MCK control** | | | Data type: UNSIGNED_32 Index: $24112_d = 5E30_h$ |
|---|---|---|---|
| This parameter serves to control the functions of the Motion Control Kernel for PC manual control. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x00000000 | | 0xFFFFFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | OpMode_Bit0 | | |
| Bit 1 | OpMode_Bit1 | | |
| Bit 2 | OpMode_Bit2 | | |
| Bit 3 | OpMode_Bit3 | | |
| Bit 4 | ManJogPos | | |
| Bit 5 | ManJogNeg | | |
| Bit 6 | ManExecute2ndSpeed | | |
| Bit 7 | ReleaseLimitSwitch | | |
| Bit 8 | HomStartStop | | |
| Bit 9 | HomSetPos | | |
| Bit 10 | HomResetPos | | |
| Bit 11 | EnableSpeedOverride | | |
| Bit 12 | EnableAccOverride | | |
| Bit 13 | EnableSRampOverride | | |
| Bit 14 | PosTeachSetPos | | |
| Bit 15 | PosTeachActPos | | |
| Bit 16 | PosExecute | | |
| Bit 17 | PosFinishTarget | | |
| Bit 18 | PosDisableFollowProfile | | |
| Bit 19 | PosStop | | |
| Bit 20 | PosModeBit0 | | |
| Bit 21 | PosModeBit1 | | |
| Bit 22 | PosModeBit2 | | |
| Bit 23 | PosModeBit3 | | |
| Bit 24 | ProfileNo_Bit0 | | |
| Bit 25 | ProfileNo_Bit1 | | |
| Bit 26 | ProfileNo_Bit2 | | |
| Bit 27 | ProfileNo_Bit3 | | |
| Bit 28 | ProfileNo_Bit4 | | |
| Bit 29 | ProfileNo_Bit5 | | |
| Bit 30 | ProfileNo_Bit6 | | |
| Bit 31 | ProfileNo_Bit7 | | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C00463/1 | 0x00000000 | Remote: MCK control | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00464**

| Parameter \| Name: **C00464 \| Remote: Monitoring timeout** | | | Data type: UNSIGNED_16 Index: $24111_d = 5E2F_h$ |
|---|---|---|---|
| | | | ▸ PC manual control |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 200 | ms | 5000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00464/1 | 2000 ms | | Remote: Monitoring timeout |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00465**

| Parameter \| Name: **C00465 \| Keypad: Timeout welcome screen** | Data type: INTEGER_32 Index: $24110_d = 5E2E_h$ |
|---|---|
| Time setting for the automatic change of the keypad display to the welcome screen | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **Never show welcome screen** |
| 5 | 5 min |
| 15 | 15 min |
| 30 | 30 min |
| 60 | 60 min |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00466**

| Parameter \| Name: **C00466 \| Keypad: Default parameter** | | | Data type: INTEGER_32 Index: $24109_d = 5E2D_h$ |
|---|---|---|---|
| Setting of the default parameter for the keypad | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 65535 | **51** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00467**

| Parameter \| Name: **C00467 \| Keypad: Default welcome screen** | Data type: INTEGER_32 Index: $24108_d = 5E2C_h$ |
|---|---|
| Selection of the welcome screen for the keypad | |
| **Selection list** (Lenze setting printed in bold) | |
| 0 | Main menu |
| **1** | **Parameter list** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00468**

| Parameter \| Name: **C00468 \| Service code** | Data type: INTEGER_32 Index: $24107_d = 5E2B_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00469**

| Parameter \| Name: **C00469 \| Keypad: Fct. STOP key** | | Data type: INTEGER_32 Index: $24106_d = 5E2A_h$ |
|---|---|---|
| Selection of the function for the STOP key on the keypad | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | No function | STOP key does not have any function |
| **1** | **Inhibit inverter** | STOP key sets controller inhibit in the drive |
| 2 | Activate quick stop | STOP key sets quick stop in the drive |
| 4 | Inhibit inverter and reset error | From version 14.00.00<br>STOP key sets controller inhibit in the drive. An error reset is carried out at the same time. |
| 5 | Activate quick stop and reset error | From version 14.00.00<br>STOP key sets quick stop in the drive. An error reset is carried out at the same time. |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00470**

| Parameter \| Name: **C00470 \| LS_ParFree_b** | | Data type: UNSIGNED_8 Index: $24105_d = 5E29_h$ |
|---|---|---|
| The LS_ParFree_b SB: Setting of the signal level to be output | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00470/1 | 0: FALSE | Signal level for output *bPar1* ... *bPar32* |
| C00470/... | | |
| C00470/32 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00471**

| Parameter \| Name: **C00471 \| LS_ParFree** | | | Data type: UNSIGNED_16 Index: $24104_d = 5E28_h$ |
|---|---|---|---|
| The LS_ParFree SB: Setting of the words to be output | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Bit0 | | |
| ... | ... | | |
| Bit 15 | Bit15 | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00471/1 | 0x0000 | | Value for output *wPar1* ... *wPar32* |
| C00471/... | | | |
| C00471/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00472**

| Parameter \| Name: **C00472 \| LS_ParFree_a** | | | Data type: INTEGER_16 Index: $24103_d = 5E27_h$ |
|---|---|---|---|
| The LS_ParFree_a SB: Setting of the analog signals to be output | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00472/1 | 0.00 % | | Value for output *nPar1_a* |
| C00472/2 | 0.00 % | | Value for output *nPar2_a* |
| C00472/3 | 100.00 % | | Value for output *nPar3_a* |
| C00472/4 | 100.00 % | | Value for output *nPar4_a* |
| C00472/5 | 0.00 % | | Value for output *nPar5_a* |
| C00472/6 | 0.00 % | | Value for output *nPar6_a* |
| C00472/7 | 0.00 % | | Value for output *nPar7_a* |
| C00472/8 | 0.00 % | | Value for output *nPar8_a* |
| C00472/9 | 0.00 % | | Value for output *nPar9_a* |
| C00472/10 | 0.00 % | | Value for output *nPar10_a* |
| C00472/11 | 0.00 % | | Value for output *nPar11_a* |
| C00472/12 | 0.00 % | | Value for output *nPar12_a* |
| C00472/13 | 0.00 % | | Value for output *nPar13_a* |
| C00472/14 | 0.00 % | | Value for output *nPar14_a* |
| C00472/15 | 0.00 % | | Value for output *nPar15_a* |
| C00472/16 | 0.00 % | | Value for output *nPar16_a* |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00473**

| Parameter \| Name: **C00473 \| LS_ParFree_v** | | | Data type: INTEGER_16 Index: $24102_d = 5E26_h$ |
|---|---|---|---|
| The LS_ParFree_v SB: Setting of the speed signals to be output | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | Incr./ms | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00473/1 | 0 incr./ms | | Values for output *nPar1_v* ... *nPar8_v* |
| C00473/... | | | |
| C00473/8 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00474**

| Parameter \| Name: **C00474 \| LS_ParFree_p** | | | Data type: INTEGER_32 Index: $24101_d = 5E25_h$ |
|---|---|---|---|
| SB LS_ParFree_p: Setting of the position signals to be output | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00474/1 | 0 incr. | | Values for output *dnPar1_p* ... *dnPar8_p* |
| C00474/... | | | |
| C00474/8 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00475**

| Parameter \| Name: **C00475 \| LS_ParFreeUnit_1_2** | | | Data type: INTEGER_32 Index: $24100_d = 5E24_h$ |
|---|---|---|---|
| From version 02.00.00<br>SB LS_ParFreeUnit / LS_ParFreeUnit_2: Setting of the position signals to be output | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00475/1 | 0.0000 units | | Values for output *dnC475_1* ... *dnC475_32* |
| C00475/... | | | |
| C00475/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C00476**

| Parameter \| Name: **C00476 \| LS_ParFree_a_2** | | | Data type: INTEGER_16 Index: $24099_d = 5E23_h$ |
|---|---|---|---|
| From version 02.00.00<br>SB LS_ParFree_a_2: Setting of the analog signals to be output | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00476/1 | 0.00 % | | Value for output *nC476_1_a* |
| C00476/2 | 0.00 % | | Value for output *nC476_2_a* |
| C00476/3 | 0.00 % | | Value for output *nC476_3_a* |
| C00476/4 | 0.00 % | | Value for output *nC476_4_a* |
| C00476/5 | 0.00 % | | Value for output *nC476_5_a* |
| C00476/6 | 0.00 % | | Value for output *nC476_6_a* |
| C00476/7 | 0.00 % | | Value for output *nC476_7_a* |
| C00476/8 | 0.00 % | | Value for output *nC476_8_a* |
| C00476/9 | 0.00 % | | Value for output *nC476_9_a* |
| C00476/10 | 0.00 % | | Value for output *nC476_10_a* |
| C00476/11 | 0.00 % | | Value for output *nC476_11_a* |
| C00476/12 | 0.00 % | | Value for output *nC476_12_a* |
| C00476/13 | 0.00 % | | Value for output *nC476_13_a* |
| C00476/14 | 0.00 % | | Value for output *nC476_14_a* |
| C00476/15 | 0.00 % | | Value for output *nC476_15_a* |
| C00476/16 | 0.00 % | | Value for output *nC476_16_a* |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00477**

| Parameter \| Name: **C00477 \| LS_ParFree_2** | | | Data type: UNSIGNED_16 Index: $24098_d = 5E22_h$ |
|---|---|---|---|
| From version 02.00.00<br>SB LS_ParFree_2: Setting of the words to be output | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Bit0 | | |
| ... | ... | | |
| Bit 15 | Bit15 | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00477/1 | 0x0000 | | Value for output *wC477_1* ... *wC477_32* |
| C00477/... | | | |
| C00477/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00478**

| Parameter \| Name: **C00478 \| LS_ParFree_v_2** | | | Data type: INTEGER_16 Index: $24097_d = 5E21_h$ |
|---|---|---|---|
| From version 02.00.00<br>SB LS_ParFree_v_2: Setting of the speed signals to be output | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | Incr./ms | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00478/1 | 0 incr./ms | | Values for output *nC478_1_v* ... *nC478_8_v* |
| C00478/... | | | |
| C00478/8 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00479**

| Parameter \| Name: **C00479 \| LS_ParFree32** | | | Data type: INTEGER_32 Index: $24096_d = 5E20_h$ |
|---|---|---|---|
| From version 02.00.00<br>SB LS_ParFree32: Setting of the 32-bit values to be output | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00479/1 | 0 | | Values for output *dnC479_1* ... *dnC479_8* |
| C00479/... | | | |
| C00479/8 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00480**

| Parameter \| Name: **C00480 \| LS_DisFree_b** | | | Data type: UNSIGNED_16 Index: $24095_d = 5E1F_h$ |
|---|---|---|---|
| The LS_DisFree_b SB: Display of the input values | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | bDis1 | | Signal level input *bDis1* ... *bDis16* |
| ... | ... | | |
| Bit 15 | bDis16 | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00481**

| Parameter \| Name: **C00481 \| LS_DisFree** | | | Data type: UNSIGNED_16 Index: $24094_d = 5E1E_h$ |
|---|---|---|---|
| The LS_DisFree SB: Display of the input values | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00481/1 | | | Input values *wDis1* ... *wDis8* |
| C00481/... | | | |
| C00481/8 | | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00482**

| Parameter \| Name: **C00482 \| LS_DisFree_a** | | | Data type: INTEGER_16 Index: $24093_d = 5E1D_h$ |
|---|---|---|---|
| The LS_DisFree_a SB: Display of the input values | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00482/1 | | | Input values *nDis1_a* ... *nDis8_a* |
| C00482/... | | | |
| C00482/8 | | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00483**

| Parameter \| Name: **C00483 \| LS_DisFree_p** | | | Data type: INTEGER_32 Index: $24092_d = 5E1C_h$ |
|---|---|---|---|
| SB LS_DisFree_p: Display of the input values | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00483/1 | | | Input values *dnDis1_p* ... *dnDis8_p* |
| C00483/... | | | |
| C00483/8 | | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00484**

| Parameter \| Name: **C00484 \| Application units: Offset** | | | Data type: INTEGER_16 Index: $24091_d = 5E1B_h$ |
|---|---|---|---|
| The LS_DisFree_a SB: Offset for display of the input variables in application unit ▸ Display of internal process factors in application units | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00484/1 | 0.00 % | | Application unit 1: Offset |
| C00484/2 | 0.00 % | | Application unit 2: Offset |
| C00484/3 | 0.00 % | | Application unit 3: Offset |
| C00484/4 | 0.00 % | | Application unit 4: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00485**

| Parameter \| Name: **C00485 \| Application units: Display factor** | | | Data type: INTEGER_32 Index: $24090_d = 5E1A_h$ |
|---|---|---|---|
| The LS_DisFree_a SB: Display factor for display of the input variables in application unit ▸ Display of internal process factors in application units | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -65536.0000 | | 65536.0000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00485/1 | 1.0000 | | Application unit 1: Display factor |
| C00485/2 | 1.0000 | | Application unit 2: Display factor |
| C00485/3 | 1.0000 | | Application unit 3: Display factor |
| C00485/4 | 1.0000 | | Application unit 4: Display factor |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C00486**

| Parameter \| Name: **C00486 \| Application units: Text** | | Data type: VISIBLE_STRING Index: $24089_d = 5E19_h$ |
|---|---|---|
| The LS_DisFree_a SB: Text for the display of the input variables in application unit ▸ Display of internal process factors in application units | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00486/1 | | Application unit 1: Text |
| C00486/2 | | Application unit 2: Text |
| C00486/3 | | Application unit 3: Text |
| C00486/4 | | Application unit 4: Text |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ COM ☐ MOT Character length: 7 | | |

**C00487**

| Parameter \| Name: **C00487 - Application units** | | | Data type: INTEGER_32 Index: $24088_d = 5E18_h$ |
|---|---|---|---|
| SB LS_DisFree_a: Display of the input values in a configurable application unit<br>▸ Display of internal process factors in application units | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -21474836.47 | units | 21474836.47 | |
| **Subcodes** | | | **Info** |
| C00487/1 | | | Application units 1 |
| C00487/2 | | | Application units 2 |
| C00487/3 | | | Application units 3 |
| C00487/4 | | | Application units 4 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00488**

| Parameter \| Name: **C00488 \| L_JogCtrlEdgeDetect** | | | Data type: UNSIGNED_8 Index: $24087_d = 5E17_h$ |
|---|---|---|---|
| The L_JogCtrlExtension_1 FB: Signal methodology<br>• Selection whether the corresponding function is to be activated by edge or level. | | | |
| **Selection list** | | | |
| 0 | Level | | |
| 1 | Edge | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00488/1 | 0: Level | | InputSens.SlowDown1<br>• Selection of edge or level for starting slow-down function 1 |
| C00488/2 | 0: Level | | InputSens.Stop1<br>• Selection of edge or level for stop function 1 |
| C00488/3 | 0: Level | | InputSens.SlowDown2<br>• Selection of edge or level for starting slow-down function 2 |
| C00488/4 | 0: Level | | InputSens.Stop2<br>• Selection of edge or level for stop function 2 |
| C00488/5 | 0: Level | | InputSens.SlowDown3<br>• Selection of edge or level for starting slow-down function 3 |
| C00488/6 | 0: Level | | InputSens.Stop3<br>• Selection of edge or level for stop function 3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00490**

| Parameter \| Name: **C00490 \| Position encoder selection** | | | Data type: UNSIGNED_8 Index: $24085_d = 5E15_h$ |
|---|---|---|---|
| Selection of the feedback system for the generation of the actual position on the load<br>▸ Encoder/feedback system | | | |
| **Selection list** (Lenze setting printed in bold) | | | **Info** |
| **0** | **No sensor: nSpeedSetValue_a** | | No encoder available on the load.<br>The position signal *dnMotorPosAct_p* is derived from the speed setpoint *nSpeedSetValue_a*. |
| 1 | Sensor signal FreqIn12 | | Position encoder signal is fed via digital inputs DI1 and DI2 |
| 2 | Encoder signal FreqIn67 | | Position encoder signal is fed via digital inputs DI6 and DI7 |
| 3 | Multi encoder | | Position encoder signal is supplied via multi-encoder interface |
| 4 | Resolver | | Position encoder signal is supplied via resolver interface |
| 5 | Encoder signal FreqIn1267 | | From version 15.00.00<br>Position encoder signal is fed via digital inputs DI1/DI2 and DI6/DI7 (connection of 4-track HTL encoders) |
| 10 | Encoder selection C495 or nVoltageAngleAct_a | | From version 12.00.00<br>The position signal *dnMotorPosAct_p* is either calculated from the set speed feedback (for C00495 > 0) or from the speed signal *nMotorSpeedAct_v* (for C00495 = 0). |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00491**

| Parameter \| Name: **C00491 \| Hiperface messages** | | | Data type: UNSIGNED_8 Index: $24084_d = 5E14_h$ |
|---|---|---|---|
| From version 14.00.00 | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 255 | |
| **Subcodes** | | | **Info** |
| C00491/1 | | | Hiperface: Status 1 |
| C00491/2 | | | Hiperface: Status 2 |
| C00491/3 | | | Hiperface: Status 3 |
| C00491/4 | | | Hiperface: Status 4 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00492**

| Parameter \| Name: **C00492 \| Hiperface: Detected TypCode** | | | Data type: UNSIGNED_8 Index: $24083_d = 5E13_h$ |
|---|---|---|---|
| From version 02.00.00<br>▸ Encoder/feedback system: sin/cos absolute value encoder with HIPERFACE® protocol | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 255 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00493**

| Parameter \| Name: C00493 \| Hiperface: TypCode | | | Data type: UNSIGNED_8 Index: $24082_d = 5E12_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| ▸ Encoder/feedback system: sin/cos absolute value encoder with HIPERFACE® protocol | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 255 | **0** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00494**

| Parameter \| Name: C00494 \| Hiperface: Resolutions | | | Data type: UNSIGNED_32 Index: $24081_d = 5E11_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| ▸ Encoder/feedback system: sin/cos absolute value encoder with HIPERFACE® protocol | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65536 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00494/1 | 0 | | Hiperface: Number of revolutions |
| C00494/2 | 0 | | Hiperface: Steps per revolution |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00495**

| Parameter \| Name: C00495 \| Speed sensor selection | | Data type: UNSIGNED_8 Index: $24080_d = 5E10_h$ |
|---|---|---|
| Selection of the feedback system for the actual speed for motor control and display ▸ Encoder/feedback system | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **No sensor** | No sensor available for the actual speed detection |
| 1 | Sensor signal FreqIn12 | Speed sensor signal is fed via the digital DI1 and DI2 inputs |
| 2 | Encoder signal FreqIn67 | Speed encoder signal is fed via digital inputs DI6 and DI7 |
| 3 | Multi encoder | Speed encoder signal is supplied via multi-encoder interface |
| 4 | Resolver | Speed encoder signal is supplied via resolver interface |
| 5 | Encoder signal FreqIn1267 | From version 15.00.00<br>Speed encoder signal is fed via digital inputs DI1/DI2 and DI6/DI7 (connection of 4-track HTL encoders) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00496**

| Parameter \| Name: **C00496 \| Encoder evaluation method DigIn12** | | Data type: UNSIGNED_8 Index: $24079_d = 5E0F_h$ |
|---|---|---|
| ▸ Encoder/feedback system | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | High-resolution encoders | High-precision procedure for high-resolution encoders (>=512 increments) |
| 1 | Low-resolution encoder (StateLine) | High-precision procedure for low-resolution encoders (<=128 increments) |
| **2** | **Comb. encoder procedure** | Combination of the first two procedures as a function of the speed (recommended procedure) |
| 3 | Edge-counting procedure | Simple edge counting procedure with adjustable scanning time (C00425) |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00497**

| Parameter \| Name: **C00497 \| Filter time constant** | | | Data type: UNSIGNED_16 Index: $24078_d = 5E0E_h$ |
|---|---|---|---|
| ▸ Encoder/feedback system | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | ms | 500.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00497/1 | 1.0 ms | | FreqIn12: Encoder filter time |
| C00497/2 | 1.0 ms | | FreqIn67: Encoder filter time |
| C00497/3 | 1.0 ms | | LS_MultiEncoder: Encoder filter time |
| C00497/4 | 2.0 ms | | LS_Resolver: Encoder filter time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00498**

| Parameter \| Name: **C00498 \| Open-circuit monitoring** | | Data type: UNSIGNED_8 Index: $24077_d = 5E0D_h$ |
|---|---|---|
| Configuration of open-circuit monitoring for encoder/feedback system<br>• We recommend using the Lenze setting.<br>• The settings 1 ... 3 make sense if the respective encoder is not to be used as speed encoder or position encoder but within the scope of the application. Moreover, a directed disconnection of the monitoring is possible.<br>▸ Encoder/feedback system | | |
| **Selection list** | | **Info** |
| 0 | Speed and position encoder | Open-circuit monitoring is active for the speed encoder selected in C00495 and the position encoder selected in C00490.<br>• Depending on whether a speed encoder or position encoder has been selected, the monitoring mode for the resolver and/or encoder is activated.<br>• If no speed encoder and position encoder have been selected, open-circuit monitoring is deactivated. |
| 1 | Resolver only | Open-circuit monitoring is only active for the resolver, independent on the selection of the speed encoder and position encoder.<br>• Open-circuit monitoring for the encoder is deactivated. |
| 2 | Encoder only | Open-circuit monitoring is only active for the encoder, independent on the selection of the speed encoder and position encoder.<br>• Open-circuit monitoring for the resolver is deactivated. |
| 3 | Resolver and encoder | Open-circuit monitoring is active for the resolver and encoder, independent on the selection of the speed encoder and position encoder. |
| **Subcodes** | **Lenze setting** | **Info** |
| C00498/1 | 0: Speed encoder and position encoder | Open-circuit monitoring |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00499**

| Parameter \| Name: **C00499 \| Hiperface: Settings** | | | Data type: UNSIGNED_16 Index: $24076_d = 5E0C_h$ |
|---|---|---|---|
| From version 14.00.00 | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Manual entry of encoder data | | |
| Bit 1 | write defined position | | |
| Bit 2 | Also write value directly in case of PSM | | |
| Bit 3 | PLI compensation | | |
| Bit 4 | Delay open circuit monit. | | From version 15.00.00 |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00499/1 | 0x0000 | | Hiperface: Advanced settings |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT | | | |

**C00505**

| Parameter \| Name: **C00505 \| Password data** | | | Data type: VISIBLE_STRING Index: $24070_d = 5E06_h$ |
|---|---|---|---|
| | | | ▸ Device access protection |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00505/1 | | | MasterPin |
| C00505/2 | | | Binding ID |
| C00505/3 | | | Password |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 16 | | | |

**C00506**

| Parameter \| Name: **C00506 \| PW protection internal config** | | Data type: UNSIGNED_16 Index: $24069_d = 5E05_h$ |
|---|---|---|
| From version 15.00.00<br>Configuration of the password protection for device-internal parameter access<br>▸ Individual password protection for single communication channels | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** | | **Info** |
| Bit 0 | Only access to user menu | The menus in the keypad are reduced. Only the user menu can be used to change parameters. |
| Bit 1 | Parameter write protection | Via system blocks (LS_ParReadWrite1-6 and LS_WriteParamList), write/read parameters cannot be written. |
| Bit 2 | Parameter read protection | Via system blocks (LS_ParReadWrite1-6 and LS_WriteParamList), write/read parameters cannot be read. |
| Bit 3 | Reserved | |
| Bit 4 | Reserved | |
| Bit 5 | Reserved | |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00506/1 | 0x0007 | PW protection internal config |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C00507**

| Parameter \| Name: **C00507 \| Current password protection** | | | Data type: UNSIGNED_16 Index: $24068_d = 5E04_h$ |
|---|---|---|---|
| Display of the currently active device access protection (password protection and device personalisation)<br>**Note:**<br>As the password protection can be configured individually for each single communication channel from version 15.00.00 bit 1 and bit 2 indicate the active protection always with regard to the communication channel used.<br>▸ Device access protection | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Only access to user menu | | 1 ≡The menus in the keypad are reduced. Only the user menu can be used to change parameters.<br>▸ Password protection |
| Bit 1 | Parameter write protection | | 1 ≡ The currently used communication channel cannot be used to write into write/read parameters.<br>▸ Password protection |
| Bit 2 | Parameter read protection | | 1 ≡ The currently used communication channel cannot be used to read write/read parameters.<br>▸ Password protection |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Memory module binding on | | 1 ≡ Device personalisation is active. All write/read actions between inverter and memory module are only executed if both components have the same binding ID.<br>▸ Device personalisation |
| **Subcodes** | | | **Info** |
| C00507/1 | | | Password protection - all communication channels |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00508**

| Parameter \| Name: **C00508 \| PW protection config X6(DIAG)** | | Data type: UNSIGNED_16 Index: $24067_d = 5E03_h$ |
|---|---|---|
| From version 15.00.00<br>Configuration of the password protection for parameter access via diagnostic interface X6 ("DIAG")<br>▸ Individual password protection for single communication channels | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |

| **Value is bit-coded:** | | **Info** |
|---|---|---|
| Bit 0 | Reserved | |
| Bit 1 | Parameter write protection | The diagnostics interface X6 cannot be used to write into write/read parameters. |
| Bit 2 | Parameter read protection | The diagnostics interface X6 cannot be used to read write/read parameters. |
| Bit 3 | Reserved | |
| Bit 4 | Reserved | |
| Bit 5 | Reserved | |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C00508/1 | 0x0006 | PW protection config X6(DIAG) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

**C00509**

| Parameter \| Name: **C00509 \| PW protection config X1(CAN)** | | Data type: UNSIGNED_16 Index: $24066_d = 5E02_h$ |
|---|---|---|
| From version 15.00.00<br>Configuration of the password protection for parameter access via CANopen interface X1 ("CAN on board")<br>▸ Individual password protection for single communication channels | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |

| **Value is bit-coded:** | | **Info** |
|---|---|---|
| Bit 0 | Reserved | |
| Bit 1 | Parameter write protection | The CANopen interface X1 cannot be used to write into write/read parameters. |
| Bit 2 | Parameter read protection | The CANopen interface X1 cannot be used to read write/read parameters. |
| Bit 3 | Reserved | |
| Bit 4 | Reserved | |
| Bit 5 | Reserved | |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C00509/1 | 0x0006 | PW protection config X1 (CAN) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

**C00510**

| Parameter \| Name: **C00510 \| PW protection config MCI** | | Data type: UNSIGNED_16 Index: $24065_d = 5E01_h$ |
|---|---|---|
| From version 15.00.00<br>Configuration of the password protection for parameter access via fieldbus interface ("MCI")<br>▸ Individual password protection for single communication channels | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |

| **Value is bit-coded:** | | **Info** |
|---|---|---|
| Bit 0 | Reserved | |
| Bit 1 | Parameter write protection | The fieldbus interface (MCI) cannot be used to write into write/read parameters. |
| Bit 2 | Parameter read protection | The fieldbus interface (MCI) cannot be used to read write/read parameters. |
| Bit 3 | Reserved | |
| Bit 4 | Reserved | |
| Bit 5 | Reserved | |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C00510/1 | 0x0006 | PW protection config. MCI |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

**C00516**

| Parameter \| Name: **C00516 \| Checksums** | Data type: UNSIGNED_32 Index: $24059_d = 5DFB_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00517**

| Parameter \| Name: **C00517 \| User menu** | | | Data type: INTEGER_32 Index: $24058_d = 5DFA_h$ |
|---|---|---|---|
| When a system is installed, parameters must be changed time and again until the system runs satisfactorily. The user menu of a device serves to create a selection of frequently used parameters to be able to access and change these parameters quickly.<br>• Format: <code number>,<subcode number><br>• If "0.000" is set, no entry will be displayed in the user menu. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | | 16000.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00517/1 | 51.000 | | C00051: Display of actual speed value |
| C00517/2 | 53.000 | | C00053: Display of DC-bus voltage |
| C00517/3 | 54.000 | | C00054: Display of motor current |
| C00517/4 | 61.000 | | C00061: Display of heatsink temperature |
| C00517/5 | 137.000 | | C00137: Display of device status |
| C00517/6 | 166.003 | | C00166/3: Display of current error message |
| C00517/7 | 0.000 | | User menu: Entry 7 |
| C00517/8 | 11.000 | | C00011: Reference speed |
| C00517/9 | 39.001 | | C00039/1: Fixed setpoint 1 |
| C00517/10 | 39.002 | | C00039/2: Fixed setpoint 2 |
| C00517/11 | 12.000 | | C00012: Accel. time - main setpoint |
| C00517/12 | 13.000 | | C00013: Decel. time - main setpoint |
| C00517/13 | 15.000 | | C00015: V/f base frequency |
| C00517/14 | 16.000 | | C00016: Vmin boost |
| C00517/15 | 22.000 | | C00022: Imax in motor mode |
| C00517/16 | 120.000 | | C00120: Setting of motor overload ($I^2xt$) |
| C00517/17 | 87.000 | | C00087: Rated motor speed |
| C00517/18 | 99.000 | | C00099: Display of firmware version |
| C00517/19 | 200.000 | | C00200: Display of firmware product type |
| C00517/20 | 0.000 | | User menu: Entry 20 |
| C00517/21 | 0.000 | | User menu: Entry 21 |
| C00517/22 | 0.000 | | User menu: Entry 22 |
| C00517/23 | 0.000 | | User menu: Entry 23 |
| C00517/24 | 105.000 | | C00105: Decel. time - quick stop |
| C00517/25 | 173.000 | | C00173: Mains voltage |
| C00517/26 | 0.000 | | User menu: Entry 26 |
| C00517/27 | 0.000 | | User menu: Entry 27 |
| C00517/28 | 0.000 | | User menu: Entry 28 |
| C00517/29 | 0.000 | | User menu: Entry 29 |
| C00517/30 | 0.000 | | User menu: Entry 30 |
| C00517/31 | 0.000 | | User menu: Entry 31 |
| C00517/32 | 0.000 | | User menu: Entry 32 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00540**

| Parameter \| Name: **C00540 \| LS_DFOut: Function** | Data type: UNSIGNED_8 Index: $24035_d = 5DE3_h$ |
|---|---|
| From version 12.00.00<br>Selection of the signal source for the digital frequency output (multi-encoder interface X8)<br>▸ Digital frequency coupling | |

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| 1 | Off |
| **2** | **DFIn** |
| 3 | MotorSpeed |
| 4 | LoadSpeed |
| 5 | Resolver |
| 6 | DigIn 1/2 |
| 7 | FB input |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00545**

| Parameter \| Name: **C00545 \| LS_DFOut: Angular offset** | Data type: INTEGER_32 Index: $24030_d = 5DDE_h$ |
|---|---|
| From version 12.00.00<br>▸ Digital frequency coupling | |

| **Setting range** (min. value \| unit \| max. value) | | |
|---|---|---|
| 0 | Incr. | 65535 |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C00545/1 | 0 incr. | LS_DFOut: Angular offset |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00560**

| Parameter \| Name: **C00560 \| Fan switching status** | Data type: UNSIGNED_8 Index: $24015_d = 5DCF_h$ |
|---|---|
| Display of the function status of the device fans | |

| **Selection list** | |
|---|---|
| 0 | Off |
| 1 | On |
| 2 | No fan |

| **Subcodes** | **Info** |
|---|---|
| C00560/1 | Switching status - internal fan |
| C00560/2 | Switching status - heatsink fan |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00561**

| Parameter \| Name: **C00561 \| Failure indication** | | Data type: UNSIGNED_8 Index: $24014_d$ = $5DCE_h$ |
|---|---|---|
| Failure display of device fans and motor phases | | |
| **Selection list** | | |
| 0 | No error | |
| 1 | Error | |

| Subcodes | Info |
|---|---|
| C00561/1 | Internal fan |
| C00561/2 | Heatsink fan |
| C00561/3 | Motor phase U<br>• From version 02.00.00 |
| C00561/4 | Motor phase V<br>• From version 02.00.00 |
| C00561/5 | Motor phase W<br>• From version 02.00.00 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00563**

| Parameter \| Name: **C00563 \| Current monitoring: Delay time** | | | Data type: UNSIGNED_32 Index: $24012_d$ = $5DCC_h$ |
|---|---|---|---|
| From version 16.00.00 | | | ▸ Current monitoring overload |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.900 | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00563/1 | 3.000 s | Current monitoring: Delay time overload |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C00565**

| Parameter \| Name: **C00565 \| Resp. to mains phase failure** | | Data type: UNSIGNED_8 Index: $24010_d$ = $5DCA_h$ |
|---|---|---|
| Response to the failure of mains phases | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| 1 | Fault | |
| **5** | **Warning** | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00566**

| Parameter \| Name: **C00566 \| Resp. to fan failure** | | Data type: UNSIGNED_8 Index: $24009_d = 5DC9_h$ |
|---|---|---|
| Response to the detection of a fan failure | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| 1 | Fault | |
| **5** | **Warning** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00567**

| Parameter \| Name: **C00567 \| Resp. to speed controller limited** | | Data type: UNSIGNED_8 Index: $24008_d = 5DC8_h$ |
|---|---|---|
| Response if speed controller output is limited (*bLimSpeedCtrlOut* = TRUE) | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **No Reaction** | |
| 1 | Fault | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00569**

| Parameter \| Name: **C00569 \| Resp. to peak current** | | Data type: UNSIGNED_8 Index: $24006_d = 5DC6_h$ |
|---|---|---|
| Configuration of monitoring of the motor control (group 1) | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 5 | Warning | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00569/1 | 0: No Reaction | Response for overcurrent detection and clamp operation |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00570**

| Parameter \| Name: **C00570 \| Resp. to controller limitations** | | | Data type: UNSIGNED_8 Index: $24005_d$ = $5DC5_h$ |
|---|---|---|---|
| Configuration of monitoring of the motor control (group 2) | | | |
| **Selection list** | | | |
| 0 | No Reaction | | |
| 1 | Fault | | |
| 5 | Warning | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00570/1 | 0: No Reaction | | Response if direct-axis current controller is limited<br>• e.g. at servo control (SC) |
| C00570/2 | 0: No Reaction | | Response if cross current controller is limited<br>• e.g. at servo control (SC) |
| C00570/3 | 0: No Reaction | | Response if torque setpoint is limited<br>• Limitation of the speed controller output, the differential setpoint precontrol, and of the additive torque at (SC) servo control and (SLVC) sensorless vector control. |
| C00570/4 | 0: No Reaction | | Response if field controller is limited |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00571**

| Parameter \| Name: **C00571 \| Resp. to wrong controller setting** | | | Data type: UNSIGNED_8 Index: $24004_d$ = $5DC4_h$ |
|---|---|---|---|
| From version 14.00.00 | | | |
| **Selection list** | | | |
| 0 | No Reaction | | |
| 1 | Fault | | |
| 2 | Trouble | | |
| 3 | TroubleQuickStop | | |
| 4 | WarningLocked | | |
| 5 | Warning | | |
| 6 | Information | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00571/1 | 6: Information | | Resp. to: Motor control does not match motor data |
| C00571/2 | 1: Fault | | Resp. to a non-set speed controller |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00572**

| Parameter \| Name: **C00572 \| Brake resistor overload threshold** | | | Data type: UNSIGNED_8 Index: $24003_d$ = $5DC3_h$ |
|---|---|---|---|
| Adjustable threshold for monitoring the brake resistor utilisation<br>• The response for reaching the threshold can be selected in C00574. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | % | 100 | **100 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00574**

| Parameter \| Name: **C00574 \| Resp. to brake resist. overtemp.** | Data type: UNSIGNED_8 Index: $24001_d$ = $5DC1_h$ |
|---|---|

Response which is triggered if the threshold set in C00572 for monitoring brake resistor utilisation is reached.

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **No Reaction** |
| 1 | Fault |
| 5 | Warning |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00575**

| Parameter \| Name: **C00575 \| Resp. to logbook information** | Data type: UNSIGNED_8 Index: $24000_d$ = $5DC0_h$ |
|---|---|

**This code is for device-internal use only and must not be written to by the user!**

**C00576**

| Parameter \| Name: **C00576 \| SC: Optimisation of field feedforward control** | Data type: UNSIGNED_16 Index: $23999_d$ = $5DBF_h$ |
|---|---|

Point of action of the direct-axis current setpoint precontrol for an early reduction of the field current. In this way, the acceleration behaviour can be improved in the field weakening range.
- The entry [%] refers to the slip of the asynchronous motor.

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0 | % | 600 | **200 %** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00577**

| Parameter \| Name: **C00577 \| SC: Vp field weakening controller** | Data type: UNSIGNED_16 Index: $23998_d$ = $5DBE_h$ |
|---|---|

Proportional gain of the field weakening controller
- When "0" is set, the P component of the controller is deactivated.
- The recommended setting is in the range between 0 and 0.0020

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0.0000 | | 2.0000 | **0.0010** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C00578**

| Parameter \| Name: **C00578 \| SC: Tn field weakening controller** | Data type: UNSIGNED_16 Index: $23997_d$ = $5DBD_h$ |
|---|---|

Time constant of the field weakening controller
- The recommended setting is in the range between 10 and 30 ms

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0.1 | ms | 6200.0 | **20.0 ms** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10

**C00579**

| Parameter \| Name: **C00579 \| Resp. to max. speed/output freq. reached** | Data type: UNSIGNED_8 Index: $23996_d$ = $5DBC_h$ |
|---|---|
| Response when the max. speed limit (C00909) or output frequency limit (C00910) has been reached. | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **No Reaction** |
| 1 | Fault |
| 5 | Warning |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | |

**C00580**

| Parameter \| Name: **C00580 \| Resp. to operating system error** | Data type: UNSIGNED_8 Index: $23995_d$ = $5DBB_h$ |
|---|---|
| From version 02.00.00<br>Response if the required computing time of the application exceeds the available computing time. | |
| **Selection list** | |
| 0 | No Reaction |
| 1 | Fault |
| 5 | Warning |
| **Subcodes** | **Lenze setting** | **Info** |
| C00580/1 | 0: No Reaction | Resp. to runtime exceedance |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00581**

| Parameter \| Name: **C00581 \| Resp. to LS_SetError_x** | Data type: UNSIGNED_8 Index: $23994_d$ = $5DBA_h$ |
|---|---|
| Selection of the error responses for application error messages<br>• An application error message is tripped by a FALSE/TRUE edge at the binary inputs *bSetError1...4*. | |
| **Selection list** | |
| 0 | No Reaction |
| 1 | Fault |
| 2 | Trouble |
| 3 | TroubleQuickStop |
| 4 | WarningLocked |
| 5 | Warning |
| 6 | Information |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00581/1 | 0: No Reaction | LS_SetError_1: Resp. to bSetError1 |
| C00581/2 | 0: No Reaction | LS_SetError_1: Resp. to bSetError2 |
| C00581/3 | 0: No Reaction | LS_SetError_1: Resp. to bSetError3 |
| C00581/4 | 0: No Reaction | LS_SetError_1: Resp. to bSetError4 |
| C00581/5 | 0: No Reaction | LS_SetError_2: Resp. to bSetError1 |
| C00581/6 | 0: No Reaction | LS_SetError_2: Resp. to bSetError2 |
| C00581/7 | 0: No Reaction | LS_SetError_2: Resp. to bSetError3 |
| C00581/8 | 0: No Reaction | LS_SetError_2: Resp. to bSetError4 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00582**

| Parameter \| Name: **C00582 \| Resp. to heatsink temp. > shutdown temp. -5°C** | Data type: UNSIGNED_8 Index: $23993_d = 5DB9_h$ |
|---|---|
| Response if the heatsink temperature has reached the switch-off temperature threshold. | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** **No Reaction** | |
| 1 Fault | |
| 5 Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | |

**C00583**

| Parameter \| Name: **C00583 \| Resp. to motor temperature KTY** | Data type: UNSIGNED_8 Index: $23992_d = 5DB8_h$ |
|---|---|
| Response to motor overtemperature<br>• The motor temperature is measured via the resolver or encoder cable.<br>▸ Encoder/feedback system: Motor temperature monitoring (KTY) | |
| **Selection list** | |
| 0 No Reaction | |
| 1 Fault | |
| 3 TroubleQuickStop | |
| 4 WarningLocked | |
| 5 Warning | |
| 6 Information | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00583/1 | 1: Fault | Resp. to motor overtemp. KTY resolver |
| C00583/2 | 1: Fault | Resp. to motor overtemp. KTY MultiEncoder |
| C00583/3 | 5: Warning | Resp. to motor temp. > C00121 resolver |
| C00583/4 | 5: Warning | Resp. to motor temp. > C00121 encoder |
| C00583/5 | 1: Fault | Resp. to temp. sensor error KTY resolver |
| C00583/6 | 1: Fault | Resp. to temp. sensor error KTY MultiEncoder |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00584**

| Parameter \| Name: **C00584 \| Resp. to current monitoring** | | Data type: UNSIGNED_8 Index: $23991_d = 5DB7_h$ |
|---|---|---|
| From version 16.00.00<br>Response in the event of current overload<br>▸ Current monitoring overload | | |
| **Selection list** | | **Info** |
| 0 | No Reaction | |
| 1 | Fault | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00584/1 | 0: No Reaction | Resp. to current monitoring overload |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00585**

| Parameter \| Name: **C00585 \| Resp. to motor overtemp. PTC** | | Data type: UNSIGNED_8 Index: $23990_d = 5DB6_h$ |
|---|---|---|
| Response to motor overtemperature<br>• The motor temperature is measured by means of a PTC thermistor at terminal X106. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| **1** | **Fault** | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00586**

| Parameter \| Name: **C00586 \| Resp. to open circuit HTL encoder** | | Data type: UNSIGNED_8 Index: $23989_d = 5DB5_h$ |
|---|---|---|
| Response to encoder feedback system failure or encoder feedback system track failure due to open circuit | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| **1** | **Fault** | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00588**

| Parameter \| Name: **C00588 \| Resp. to max. speed at switching freq.** | | Data type: UNSIGNED_8 Index: $23987_d = 5DB3_h$ |
|---|---|---|
| Response if the maximum speed for the set inverter switching frequency is reached (C00018) | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **No Reaction** | |
| 1 | Fault | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00590**

| Parameter \| Name: **C00590 \| Resp. to switching frequency red.** | | Data type: UNSIGNED_8 Index: $23985_d = 5DB1_h$ |
|---|---|---|
| Response to reduction of the inverter switching frequency (C00018) | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **No Reaction** | |
| 1 | Fault | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00591**

| Parameter \| Name: **C00591 \| Resp. to axis bus error** | | Data type: UNSIGNED_8 Index: $23984_d = 5DB0_h$ |
|---|---|---|
| From version 12.00.00<br>Configuration of axis bus monitoring<br>▸ Axis bus | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00591/1 | 0: No Reaction | Resp. to axis bus data error<br>▸ Data transfer axis bus |
| C00591/2 | 0: No Reaction | Resp. to axis bus IO error<br>▸ IO axis bus |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00592**

| Parameter \| Name: C00592 \| Resp. to CAN bus connection | | Data type: UNSIGNED_8 Index: $23983_d = 5DAF_h$ |
|---|---|---|
| Configuration of monitoring of the CAN interface (group 1) ▸ "CAN on board" system bus | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00592/1 | 0: No Reaction | Response to incorrect telegram for CAN communication |
| C00592/2 | 0: No Reaction | Response to "BusOff" (bus system switched off) |
| C00592/3 | 0: No Reaction | Response to warnings of the CAN controller |
| C00592/4 | 0: No Reaction | Response to communication stop of a CAN bus node |
| C00592/5 | 0: No Reaction | Response to an event in the case of monitoring via heartbeat protocol |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C00593**

| Parameter \| Name: C00593 \| Resp. to CANx_IN monitoring | | Data type: UNSIGNED_8 Index: $23982_d = 5DAE_h$ |
|---|---|---|
| Configuration of monitoring of the CAN interface (group 2) ▸ "CAN on board" system bus | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00593/1 | 0: No Reaction | Response if the monitoring time set in C00357/1 for the reception of the PDO CAN1_IN is exceeded. |
| C00593/2 | 0: No Reaction | Response if the monitoring time set in C00357/2 for the reception of the PDO CAN2_IN is exceeded. |
| C00593/3 | 0: No Reaction | Response if the monitoring time set in C00357/3 for the reception of the PDO CAN3_IN is exceeded. |
| C00593/4 | 0: No Reaction | Response if the monitoring time set in C00357/4 for the reception of the PDO CAN4_IN is exceeded. • From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C00594**

| Parameter \| Name: **C00594 \| Resp. to control word error** | | Data type: UNSIGNED_8 Index: $23981_d = 5DAD_h$ |
|---|---|---|
| Configuration of device control monitoring | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 5 | Warning | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00594/1 | 1: Fault | Response if error bit 14 in the CAN control word is set. |
| C00594/2 | 1: Fault | Response if error bit 14 in the MCI control word is set. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00595**

| Parameter \| Name: **C00595 \| MCK: Resp. to MCK error** | | Data type: UNSIGNED_8 Index: $23980_d = 5DAC_h$ |
|---|---|---|
| Configuration of monitoring of the Motion Control Kernel<br>▸ Basic drive functions | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00595/1 | 3: TroubleQuickStop | Response if the input *bLimitSwitchPos* for travel range monitoring is set to FALSE (fail-safe) by the positive hardware limit switch . |
| C00595/2 | 3: TroubleQuickStop | Response if the input *bLimitSwitchNeg* for travel range monitoring is set to FALSE (fail-safe) by the negative hardware limit switch . |
| C00595/3 | 3: TroubleQuickStop | Response for detection that the position is beyond the positive software limit position (C01229/1). |
| C00595/4 | 3: TroubleQuickStop | Response for detection that the position is beyond the negative software limit position (C01229/2). |
| C00595/5 | 5: Warning | Response if following error limit 1 is exceeded (C01215/1) |
| C00595/6 | 5: Warning | Response if following error limit 2 is exceeded (C01215/2) |
| C00595/7 | 3: TroubleQuickStop | Response if the maximum travel distance (display in C01213/1) is exceeded |
| C00595/8 | 4: WarningLocked | Response to start procedures with reference condition when the reference is not set |
| C00595/9 | 4: WarningLocked | Response to a non-supported positioning mode |
| C00595/10 | 4: WarningLocked | Response to implausible profile set data |
| C00595/11 | 5: Warning | Response to the selection of an invalid operating mode of the MCK |
| C00595/12 | 4: WarningLocked | Response to indicating an invalid profile data set |
| C00595/13 | 5: Warning | Response to an error of the FB L_MckCtrlInterface_1 |
| C00595/14 | 4: WarningLocked | Response to a profile start with a target position outside the software limit positions (C01229/1 and C01229/2). |
| C00595/15 | 1: Fault | Response to activated connection monitoring in case of PC manual control |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00597**

| Parameter \| Name: **C00597 \| Resp. to motor phase failure** | | | Data type: UNSIGNED_8 Index: $23978_d$ = $5DAA_h$ |
|---|---|---|---|
| Response to motor phase failure<br>• If a phase current does not exceed the threshold set in C00599 for more than one period, the response set here will be triggered. | | | |
| **Selection list** (Lenze setting printed in bold) | | | |
| **0** | **No Reaction** | | |
| 1 | Fault | | |
| 5 | Warning | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00598**

| Parameter \| Name: **C00598 \| Resp. to open circuit AINx** | | | Data type: UNSIGNED_8 Index: $23977_d$ = $5DA9_h$ |
|---|---|---|---|
| Configuration of monitoring of the analog inputs<br>▸ Analog terminals | | | |
| **Selection list** | | | |
| 0 | No Reaction | | |
| 1 | Fault | | |
| 2 | Trouble | | |
| 3 | TroubleQuickStop | | |
| 5 | Warning | | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C00598/1 | 3: TroubleQuickStop | Response to open circuit at AIN1 if configured as 4 ... 20 mA current loop | |
| C00598/2 | 3: TroubleQuickStop | Response to open circuit at AIN2 when being configured as 4 ... 20 mA-current loop | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00599**

| Parameter \| Name: **C00599 \| Motor phase failure threshold** | | | Data type: INTEGER_16 Index: $23976_d$ = $5DA8_h$ |
|---|---|---|---|
| Threshold for motor phase failure monitoring<br>• 100 % ≡ rated inverter current (C00098)<br>• If a phase current does not exceed the threshold set here for more than one period, the response to motor phase failure set in C00597 will be triggered. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **5.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00600**

| Parameter \| Name: **C00600 \| Resp. to DC bus voltage** | | | Data type: UNSIGNED_8 Index: $23975_d = 5DA7_h$ |
|---|---|---|---|
| Configuration of monitoring of the motor control (group 3) | | | |
| **Selection list** | | | |
| 1 | Fault | | |
| 2 | Trouble | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00600/1 | 2: Trouble | | Response to DC bus undervoltage |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00601**

| Parameter \| Name: **C00601 \| Del. resp. to fault: DC bus overvoltage** | | | Data type: UNSIGNED_16 Index: $23974_d = 5DA6_h$ |
|---|---|---|---|
| Error response delay times | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 65.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00601/1 | 2.000 s | | Delay time for triggering the "DC-bus overvoltage" error<br>• If a DC-bus overvoltage occurs, an error will not be triggered until the set delay time has elapsed. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1000 | | | |

**C00602**

| Parameter \| Name: **C00602 \| Resp. to earth fault** | | | Data type: UNSIGNED_8 Index: $23973_d = 5DA5_h$ |
|---|---|---|---|
| Response to earth fault in the motor phase(s) | | | |
| **Selection list** (Lenze setting printed in bold) | | | |
| 0 | No Reaction | | |
| **1** | **Fault** | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00603**

| Parameter \| Name: **C00603 \| Resp. to feedback** | | Data type: UNSIGNED_8 Index: $23972_d = 5DA4_h$ |
|---|---|---|
| Response of different monitoring modes for the encoder/feedback system<br>▸ Encoder/feedback system | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00603/1 | 1: Fault | Resp. to MultiEncoder open circuit |
| C00603/2 | 1: Fault | Resp. to resolver open circuit |
| C00603/3 | 0: No Reaction | Up to and including version 02.xx.xx:<br>Resp. to encoder communication error<br>From version 12.00.00:<br>Resp. to encoder angular drift monitoring |
| C00603/4 | 1: Fault | Up to and including version 02.xx.xx:<br>Resp. to encoder angular drift monitoring<br>From version 12.00.00:<br>Resp. to encoder communication error |
| C00603/5 | 6: Information | Resp. to Hiperface status<br>• From version 14.00.00 |
| C00603/6 | 6: Information | Resp. to invalid Hiperface position<br>• From version 14.00.00 |
| C00603/7 | 6: Information | Resp. to SinCos inaccuracy<br>• From version 14.00.00 |
| C00603/8 | 1: Fault | Resp. to Hiperface sin/cos deviation<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00604**

| Parameter \| Name: **C00604 \| Resp. to device overload (Ixt)** | | Data type: UNSIGNED_8 Index: $23971_d = 5DA3_h$ |
|---|---|---|
| Response if the adjustable device utilisation threshold (C00123) is reached.<br>• The current device utilisation is displayed in C00064. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| 1 | Fault | |
| **5** | **Warning** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00605**

| Parameter \| Name: **C00605 \| Resp. to feedback** | | Data type: UNSIGNED_8 Index: $23970_d = 5DA2_h$ |
|---|---|---|
| From version 15.00.00 | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00605/1 | 1: Fault | Resp. to open circuit HTL 4-fold |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00606**

| Parameter \| Name: **C00606 \| Resp. to motor overload (I²xt)** | | Data type: UNSIGNED_8 Index: $23969_d = 5DA1_h$ |
|---|---|---|
| Response when the motor load displayed in C00066 reaches the value "100.00 %". ▸ Motor overload monitoring (I2xt) | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| 1 | Fault | |
| **5** | **Warning** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00607**

| Parameter \| Name: **C00607 \| Resp. to max freq. feedb. DIG12/67** | | Data type: UNSIGNED_8 Index: $23968_d = 5DA0_h$ |
|---|---|---|
| Response when the maximum input frequency has been reached via the digital inputs. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| **1** | **Fault** | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00608**

| Parameter \| Name: **C00608 \| Resp. to maximum torque** | | Data type: UNSIGNED_8 Index: $23967_d = 5D9F_h$ |
|---|---|---|
| Response if the maximum torque (C00057) is reached. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **No Reaction** | |
| 1 | Fault | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00609**

| Parameter \| Name: **C00609 \| Resp. to maximum current** | Data type: UNSIGNED_8 Index: $23966_d$ = $5D9E_h$ |
|---|---|
| Response if the maximum current (C00022, C00023) is reached. | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** **No Reaction** | |
| 1 Fault | |
| 5 Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | |

**C00610**

| Parameter \| Name: **C00610 \| 16-bit connection table** | Data type: UNSIGNED_16 Index: $23965_d$ = $5D9D_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00611**

| Parameter \| Name: **C00611 \| Bool connection table** | Data type: UNSIGNED_16 Index: $23964_d$ = $5D9C_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00612**

| Parameter \| Name: **C00612 \| 32-bit connection table** | Data type: UNSIGNED_16 Index: $23963_d$ = $5D9B_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00613**

| Parameter \| Name: **C00613 \| 16-bit connection table AdditionalFBsHL** | Data type: UNSIGNED_16 Index: $23962_d$ = $5D9A_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00615**

| Parameter \| Name: **C00615 \| Bool connection table AdditionalFBsHL** | Data type: UNSIGNED_16 Index: $23960_d$ = $5D98_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00617**

| Parameter \| Name: **C00617 \| 32-bit connection table AdditionalFBsHL** | Data type: UNSIGNED_16 Index: $23958_d$ = $5D96_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00620**

| Parameter \| Name: **C00620 \| System connection list: 16-bit** | | Data type: UNSIGNED_16 Index: $23955_d = 5D93_h$ |
|---|---|---|
| Connection parameters: 16-bit inputs<br>• Selection of the 16 bit output signals to be connected to the 16 bit input signals<br>• The selection list contains all 16 bit output signals which can be assigned to the 16 bit inputs displayed by the subcodes.<br>• Non-listed subcodes are "reserved". | | |
| **Selection list** | | |
| See selection list - analog signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00620/1 | 1003: LA_NCtrl: nMotorSpeedAct_a | LS_AnalogOutput: nOut1_a (V)<br>Analog output 1: Voltage |
| C00620/2 | 0: Not connected | LP_CanOut1: wState<br>CAN1 output: Status word |
| C00620/3 | 0: Not connected | LP_CanOut1: wOut2<br>CAN1 output: Data word 2 |
| C00620/4 | 0: Not connected | LP_CanOut1: wOut3<br>CAN1 output: Data word 3 |
| C00620/5 | 0: Not connected | LP_CanOut1: wOut4<br>CAN1 output: Data word 4 |
| C00620/6 | 0: Not connected | LP_CanOut2: wOut1<br>CAN2 output: Data word 1 |
| C00620/7 | 0: Not connected | LP_CanOut2: wOut2<br>CAN2 output: Data word 2 |
| C00620/8 | 0: Not connected | LP_CanOut2: wOut3<br>CAN2 output: Data word 3 |
| C00620/9 | 0: Not connected | LP_CanOut2: wOut4<br>CAN2 output: Data word 4 |
| C00620/10 | 0: Not connected | LP_CanOut3: wOut1<br>CAN3 output: Data word 1 |
| C00620/11 | 0: Not connected | LP_CanOut3: wOut2<br>CAN3 output: Data word 2 |
| C00620/12 | 0: Not connected | LP_CanOut3: wOut3<br>CAN3 output: Data word 3 |
| C00620/13 | 0: Not connected | LP_CanOut3: wOut4<br>CAN3 output: Data word 4 |
| C00620/14 | 0: Not connected | LS_DisFree_a: nDis1_a<br>Display of analog signal 1 |
| C00620/15 | 0: Not connected | LS_DisFree_a: nDis2_a<br>Display of analog signal 2 |
| C00620/16 | 0: Not connected | LS_DisFree_a: nDis3_a<br>Display of analog signal 3 |
| C00620/17 | 0: Not connected | LS_DisFree_a: nDis4_a<br>Display of analog signal 4 |
| C00620/18 | 0: Not connected | LS_DisFree: wDis1<br>Display of 16-bit signal 1 |
| C00620/19 | 0: Not connected | LS_DisFree: wDis2<br>Display of 16-bit signal 2 |
| C00620/20 | 0: Not connected | LS_DisFree: wDis3<br>Display of 16-bit signal 3 |

| Parameter \| Name:<br>**C00620 \| System connection list: 16-bit** | | Data type: UNSIGNED_16<br>Index: $23955_d = 5D93_h$ |
|---|---|---|
| C00620/21 | 0: Not connected | LS_DisFree: wDis4<br>Display of 16-bit signal 4 |
| C00620/22 | 0: Not connected | LP_MciOut: wState<br>MCI output: Status word |
| C00620/23 | 0: Not connected | LP_MciOut: wOut2<br>MCI output: Data word 2 |
| C00620/24 | 0: Not connected | LP_MciOut: wOut3<br>MCI output: Data word 3 |
| C00620/25 | 0: Not connected | LP_MciOut: wOut4<br>MCI output: Data word 4 |
| C00620/26 | 0: Not connected | LP_MciOut: wOut5<br>MCI output: Data word 5 |
| C00620/27 | 0: Not connected | LP_MciOut: wOut6<br>MCI output: Data word 6 |
| C00620/28 | 0: Not connected | LP_MciOut: wOut7<br>MCI output: Data word 7 |
| C00620/29 | 0: Not connected | LP_MciOut: wOut8<br>MCI output: Data word 8 |
| C00620/30 | 0: Not connected | LP_MciOut: wOut9<br>MCI output: Data word 9 |
| C00620/31 | 0: Not connected | LP_MciOut: wOut10<br>MCI output: Data word 10 |
| C00620/32 | 0: Not connected | LP_MciOut: wOut11<br>MCI output: Data word 11 |
| C00620/33 | 0: Not connected | LP_MciOut: wOut12<br>MCI output: Data word 12 |
| C00620/34 | 0: Not connected | LP_MciOut: wOut13<br>MCI output: Data word 13 |
| C00620/35 | 0: Not connected | LP_MciOut: wOut14<br>MCI output: Data word 14 |
| C00620/36 | 0: Not connected | LP_MciOut: wOut15<br>MCI output: Data word 15 |
| C00620/37 | 0: Not connected | LP_MciOut: wOut16<br>MCI output: Data word 16 |
| C00620/38 | 0: Not connected | LS_AnalogOutput: nOut2_a (V)<br>Analog output 2: Voltage |
| C00620/39 | 0: Not connected | LS_AnalogOutput: nOut1_a (I)<br>Analog output 1: Current |
| C00620/40 | 0: Not connected | LS_AnalogOutput: nOut2_a (I)<br>Analog output 2: Current |
| C00620/41 | 0: Not connected | LS_DisFree_a: nDis5_a<br>Display of analog signal 5 |
| C00620/42 | 0: Not connected | LS_DisFree_a: nDis6_a<br>Display of analog signal 6 |
| C00620/43 | 0: Not connected | LS_DisFree_a: nDis7_a<br>Display of analog signal 7 |
| C00620/44 | 0: Not connected | LS_DisFree_a: nDis8_a<br>Display of analog signal 8 |
| C00620/45 | 0: Not connected | LS_DisFree: wDis5<br>Display of 16-bit signal 5 |

| Parameter \| Name:<br>**C00620 \| System connection list: 16-bit** | | Data type: UNSIGNED_16<br>Index: $23955_d = 5D93_h$ |
|---|---|---|
| C00620/46 | 0: Not connected | LS_DisFree: wDis6<br>Display of 16-bit signal 6 |
| C00620/47 | 0: Not connected | LS_DisFree: wDis7<br>Display of 16-bit signal 7 |
| C00620/48 | 0: Not connected | LS_DisFree: wDis8<br>Display of 16-bit signal 8 |
| C00620/49 | 0: Not connected | LS_ParReadWrite_1: wParIndex<br>Read/write request 1: Code |
| C00620/50 | 0: Not connected | LS_ParReadWrite_1: wParSubindex<br>Read/write request 1: Subcode |
| C00620/51 | 0: Not connected | LS_ParReadWrite_1: wInHWord<br>Read/write request 1: Value (high word) |
| C00620/52 | 0: Not connected | LS_ParReadWrite_1: wInLWord<br>Read/write request 1: Value (low word) |
| C00620/53 | 0: Not connected | LS_ParReadWrite_2: wParIndex<br>Read/write request 2: Code |
| C00620/54 | 0: Not connected | LS_ParReadWrite_2: wParSubindex<br>Read/write request 2: Subcode |
| C00620/55 | 0: Not connected | LS_ParReadWrite_2: wInHWord<br>Read/write request 2: Value (high word) |
| C00620/56 | 0: Not connected | LS_ParReadWrite_2: wInLWord<br>Read/write request 2: Value (low word) |
| C00620/57 | 0: Not connected | LS_ParReadWrite_3: wParIndex<br>Read/write request 3: Code |
| C00620/58 | 0: Not connected | LS_ParReadWrite_3: wParSubindex<br>Read/write request 3: Subcode |
| C00620/59 | 0: Not connected | LS_ParReadWrite_3: wInHWord<br>Read/write request 3: Value (high word) |
| C00620/60 | 0: Not connected | LS_ParReadWrite_3: wInLWord<br>Read/write request 3: Value (low word) |
| C00620/61 | 0: Not connected | LS_ParReadWrite_4: wParIndex<br>Read/write request 4: Code |
| C00620/62 | 0: Not connected | LS_ParReadWrite_4: wParSubindex<br>Read/write request 4: Subcode |
| C00620/63 | 0: Not connected | LS_ParReadWrite_4: wInHWord<br>Read/write request 4: Value (high word) |
| C00620/64 | 0: Not connected | LS_ParReadWrite_4: wInLWord<br>Read/write request 4: Value (low word) |
| C00620/65 | 0: Not connected | LS_ParReadWrite_5: wParIndex<br>Read/write request 5: Code |
| C00620/66 | 0: Not connected | LS_ParReadWrite_5: wParSubindex<br>Read/write request 5: Subcode |
| C00620/67 | 0: Not connected | LS_ParReadWrite_5: wInHWord<br>Read/write request 5: Value (high word) |
| C00620/68 | 0: Not connected | LS_ParReadWrite_5: wInLWord<br>Read/write request 5: Value (low word) |
| C00620/69 | 0: Not connected | LS_ParReadWrite_6: wParIndex<br>Read/write request 6: Code |
| C00620/70 | 0: Not connected | LS_ParReadWrite_6: wParSubindex<br>Read/write request 6: Subcode |

| Parameter \| Name:<br>**C00620 \| System connection list: 16-bit** | | Data type: UNSIGNED_16<br>Index: $23955_d = 5D93_h$ |
|---|---|---|
| C00620/71 | 0: Not connected | LS_ParReadWrite_6: wInHWord<br>Read/write request 6: Value (high word) |
| C00620/72 | 0: Not connected | LS_ParReadWrite_6: wInLWord<br>Read/write request 6: Value (low word) |
| C00620/89 | 0: Not connected | LS_RetainData: wIn1<br>Retain data: 16-bit input value 1 |
| C00620/90 | 0: Not connected | LS_RetainData: wIn2<br>Retain data: 16-bit input value 2 |
| C00620/91 | 0: Not connected | LS_RetainData: wIn3<br>Retain data: 16-bit input value 3 |
| C00620/92 | 0: Not connected | LS_RetainData: wIn4<br>Retain data: 16-bit input value 4 |
| C00620/93 | 0: Not connected | LS_AxisBusOut: wLine1<br>Axis bus output: Line data (word 1) |
| C00620/94 | 0: Not connected | LS_AxisBusOut: wLine2<br>Axis bus output: Line data (word 2) |
| C00620/95 | 0: Not connected | LS_AxisBusOut: wLine3<br>Axis bus output: Line data (word 3) |
| C00620/96 | 0: Not connected | LS_AxisBusOut: wCas1<br>Axis bus output: Cascaded data (word 1) |
| C00620/97 | 0: Not connected | LS_AxisBusOut: wCas2<br>Axis bus output: Cascaded data (word 2) |
| C00620/98 | 0: Not connected | LS_AxisBusOut: wCas3<br>Axis bus output: Cascaded data (word 3) |
| C00620/99 | 0: Not connected | LS_AxisBusOut: wCas4<br>Axis bus output: Cascaded data (word 4) |
| C00620/100 | 0: Not connected | LS_AxisBusAux: wAuxOut1<br>Axis bus output: @sel data (word 1) |
| C00620/101 | 0: Not connected | LS_AxisBusAux: wAuxOut2<br>Axis bus output: @sel data (word 2) |
| C00620/102 | 0: Not connected | LS_AxisBusAux: wAuxOut3<br>Axis bus output: @sel data (word 3) |
| C00620/103 | 0: Not connected | LS_AxisBusAux: wAuxOut4<br>Axis bus output: @sel data (word 4) |
| C00620/104 | 0: Not connected | LS_AxisBusAux: wSelectSlave<br>Axis bus address of slave |
| C00620/105 | 0: Not connected | LS_DFOut: nOut_v<br>Digital frequency output X8: Speed in [increments/ms] |
| C00620/106 | 0: Not connected | LS_MultiEncoder: wActPosExternalHW |
| C00620/107 | 0: Not connected | LS_MultiEncoder: wActPosExternalLW |
| C00620/108 | 0: Not connected | LP_CanOut4: wOut1 |
| C00620/109 | 0: Not connected | LP_CanOut4: wOut2 |
| C00620/110 | 0: Not connected | LP_CanOut4: wOut3 |
| C00620/111 | 0: Not connected | LP_CanOut4: wOut4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00621**

| Parameter \| Name:<br>**C00621 \| System connection list: Bool** | | Data type: UNSIGNED_16<br>Index: $23954_d = 5D92_h$ |
|---|---|---|
| Connection parameters: Binary inputs<br>• Selection of the binary output signals to be connected to the binary input signals<br>• The selection list contains all binary output signals which can be assigned to the binary inputs mapped by the subcodes.<br>• Non-listed subcodes are "reserved". | | |
| **Selection list** | | |
| See selection list - digital signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00621/1 | 1001: LA_NCtrl: bDriveFail | LS_DigitalOutput: bRelay<br>Digital relay output: Input signal |
| C00621/2 | 1000: LA_NCtrl: bDriveReady | LS_DigitalOutput: bOut1<br>Digital output 1: Input signal |
| C00621/3 | 0: Not connected | LS_DigitalInput: bCountIn1_Reset<br>Digital input 1: Reset counter |
| C00621/4 | 0: Not connected | LS_DigitalInput: bCountIn1_LoadStartValue<br>Digital input 1: Load counter starting value |
| C00621/5 | 0: Not connected | LP_CanOut1: bState_B0<br>CAN1 output: Status word bit 0 |
| C00621/6 | 0: Not connected | LP_CanOut1: bState_B1<br>CAN1 output: Status word bit 1 |
| C00621/7 | 0: Not connected | LP_CanOut1: bState_B2<br>CAN1 output: Status word bit 2 |
| C00621/8 | 0: Not connected | LP_CanOut1: bState_B3<br>CAN1 output: Status word bit 3 |
| C00621/9 | 0: Not connected | LP_CanOut1: bState_B4<br>CAN1 output: Status word bit 4 |
| C00621/10 | 0: Not connected | LP_CanOut1: bState_B5<br>CAN1 output: Status word bit 5 |
| C00621/11 | 0: Not connected | LP_CanOut1: bState_B6<br>CAN1 output: Status word bit 6 |
| C00621/12 | 0: Not connected | LP_CanOut1: bState_B7<br>CAN1 output: Status word bit 7 |
| C00621/13 | 0: Not connected | LP_CanOut1: bState_B8<br>CAN1 output: Status word bit 8 |
| C00621/14 | 0: Not connected | LP_CanOut1: bState_B9<br>CAN1 output: Status word bit 9 |
| C00621/15 | 0: Not connected | LP_CanOut1: bState_B10<br>CAN1 output: Status word bit 10 |
| C00621/16 | 0: Not connected | LP_CanOut1: bState_B11<br>CAN1 output: Status word bit 11 |
| C00621/17 | 0: Not connected | LP_CanOut1: bState_B12<br>CAN1 output: Status word bit 12 |
| C00621/18 | 0: Not connected | LP_CanOut1: bState_B13<br>CAN1 output: Status word bit 13 |
| C00621/19 | 0: Not connected | LP_CanOut1: bState_B14<br>CAN1 output: Status word bit 14 |
| C00621/20 | 0: Not connected | LP_CanOut1: bState_B15<br>CAN1 output: Status word bit 15 |

| Parameter \| Name:<br>**C00621 \| System connection list: Bool** | | Data type: UNSIGNED_16<br>Index: $23954_d = 5D92_h$ |
|---|---|---|
| C00621/21 | 0: Not connected | LS_DisFree_b: bDis1<br>Display of digital signal 1 |
| C00621/22 | 0: Not connected | LS_DisFree_b: bDis2<br>Display of digital signal 2 |
| C00621/23 | 0: Not connected | LS_DisFree_b: bDis3<br>Display of digital signal 3 |
| C00621/24 | 0: Not connected | LS_DisFree_b: bDis4<br>Display of digital signal 4 |
| C00621/25 | 0: Not connected | LS_DisFree_b: bDis5<br>Display of digital signal 5 |
| C00621/26 | 0: Not connected | LS_DisFree_b: bDis6<br>Display of digital signal 6 |
| C00621/27 | 0: Not connected | LS_DisFree_b: bDis7<br>Display of digital signal 7 |
| C00621/28 | 0: Not connected | LS_DisFree_b: bDis8<br>Display of digital signal 8 |
| C00621/29 | 0: Not connected | LP_CanOut2: bOut1_B0<br>CAN2 output: Data word 1- bit 0 |
| C00621/30 | 0: Not connected | LP_CanOut2: bOut1_B1<br>CAN2 output: Data word 1- bit 1 |
| C00621/31 | 0: Not connected | LP_CanOut2: bOut1_B2<br>CAN2 output: Data word 1- bit 2 |
| C00621/32 | 0: Not connected | LP_CanOut2: bOut1_B3<br>CAN2 output: Data word 1- bit 3 |
| C00621/33 | 0: Not connected | LP_CanOut2: bOut1_B4<br>CAN2 output: Data word 1- bit 4 |
| C00621/34 | 0: Not connected | LP_CanOut2: bOut1_B5<br>CAN2 output: Data word 1- bit 5 |
| C00621/35 | 0: Not connected | LP_CanOut2: bOut1_B6<br>CAN2 output: Data word 1- bit 6 |
| C00621/36 | 0: Not connected | LP_CanOut2: bOut1_B7<br>CAN2 output: Data word 1- bit 7 |
| C00621/37 | 0: Not connected | LP_CanOut2: bOut1_B8<br>CAN2 output: Data word 1- bit 8 |
| C00621/38 | 0: Not connected | LP_CanOut2: bOut1_B9<br>CAN2 output: Data word 1- bit 9 |
| C00621/39 | 0: Not connected | LP_CanOut2: bOut1_B10<br>CAN2 output: Data word 1- bit 10 |
| C00621/40 | 0: Not connected | LP_CanOut2: bOut1_B11<br>CAN2 output: Data word 1- bit 11 |
| C00621/41 | 0: Not connected | LP_CanOut2: bOut1_B12<br>CAN2 output: Data word 1- bit 12 |
| C00621/42 | 0: Not connected | LP_CanOut2: bOut1_B13<br>CAN2 output: Data word 1- bit 13 |
| C00621/43 | 0: Not connected | LP_CanOut2: bOut1_B14<br>CAN2 output: Data word 1- bit 14 |
| C00621/44 | 0: Not connected | LP_CanOut2: bOut1_B15<br>CAN2 output: Data word 1- bit 15 |
| C00621/45 | 0: Not connected | LP_CanOut3: bOut1_B0<br>CAN3 output: Data word 1- bit 0 |

| Parameter \| Name:<br>**C00621 \| System connection list: Bool** | | Data type: UNSIGNED_16<br>Index: $23954_d = 5D92_h$ |
|---|---|---|
| C00621/46 | 0: Not connected | LP_CanOut3: bOut1_B1<br>CAN3 output: Data word 1- bit 1 |
| C00621/47 | 0: Not connected | LP_CanOut3: bOut1_B2<br>CAN3 output: Data word 1- bit 2 |
| C00621/48 | 0: Not connected | LP_CanOut3: bOut1_B3<br>CAN3 output: Data word 1- bit 3 |
| C00621/49 | 0: Not connected | LP_CanOut3: bOut1_B4<br>CAN3 output: Data word 1- bit 4 |
| C00621/50 | 0: Not connected | LP_CanOut3: bOut1_B5<br>CAN3 output: Data word 1- bit 5 |
| C00621/51 | 0: Not connected | LP_CanOut3: bOut1_B6<br>CAN3 output: Data word 1- bit 6 |
| C00621/52 | 0: Not connected | LP_CanOut3: bOut1_B7<br>CAN3 output: Data word 1- bit 7 |
| C00621/53 | 0: Not connected | LP_CanOut3: bOut1_B8<br>CAN3 output: Data word 1- bit 8 |
| C00621/54 | 0: Not connected | LP_CanOut3: bOut1_B9<br>CAN3 output: Data word 1- bit 9 |
| C00621/55 | 0: Not connected | LP_CanOut3: bOut1_B10<br>CAN3 output: Data word 1- bit 10 |
| C00621/56 | 0: Not connected | LP_CanOut3: bOut1_B11<br>CAN3 output: Data word 1- bit 11 |
| C00621/57 | 0: Not connected | LP_CanOut3: bOut1_B12<br>CAN3 output: Data word 1- bit 12 |
| C00621/58 | 0: Not connected | LP_CanOut3: bOut1_B13<br>CAN3 output: Data word 1- bit 13 |
| C00621/59 | 0: Not connected | LP_CanOut3: bOut1_B14<br>CAN3 output: Data word 1- bit 14 |
| C00621/60 | 0: Not connected | LP_CanOut3: bOut1_B15<br>CAN3 output: Data word 1- bit 15 |
| C00621/61 | 0: Not connected | LP_MciOut: bState_B0<br>MCI output: Status word - bit 0 |
| C00621/62 | 0: Not connected | LP_MciOut: bState_B1<br>MCI output: Status word - bit 1 |
| C00621/63 | 0: Not connected | LP_MciOut: bState_B2<br>MCI output: Status word - bit 2 |
| C00621/64 | 0: Not connected | LP_MciOut: bState_B3<br>MCI output: Status word - bit 3 |
| C00621/65 | 0: Not connected | LP_MciOut: bState_B4<br>MCI output: Status word - bit 4 |
| C00621/66 | 0: Not connected | LP_MciOut: bState_B5<br>MCI output: Status word - bit 5 |
| C00621/67 | 0: Not connected | LP_MciOut: bState_B6<br>MCI output: Status word - bit 6 |
| C00621/68 | 0: Not connected | LP_MciOut: bState_B7<br>MCI output: Status word - bit 7 |
| C00621/69 | 0: Not connected | LP_MciOut: bState_B8<br>MCI output: Status word - bit 8 |
| C00621/70 | 0: Not connected | LP_MciOut: bState_B9<br>MCI output: Status word - bit 9 |

| Parameter \| Name:<br>**C00621 \| System connection list: Bool** | | Data type: UNSIGNED_16<br>Index: $23954_d = 5D92_h$ |
|---|---|---|
| C00621/71 | 0: Not connected | LP_MciOut: bState_B10<br>MCI output: Status word - bit 10 |
| C00621/72 | 0: Not connected | LP_MciOut: bState_B11<br>MCI output: Status word - bit 11 |
| C00621/73 | 0: Not connected | LP_MciOut: bState_B12<br>MCI output: Status word - bit 12 |
| C00621/74 | 0: Not connected | LP_MciOut: bState_B13<br>MCI output: Status word - bit 13 |
| C00621/75 | 0: Not connected | LP_MciOut: bState_B14<br>MCI output: Status word - bit 14 |
| C00621/76 | 0: Not connected | LP_MciOut: bState_B15<br>MCI output: Status word - bit 15 |
| C00621/77 | 0: Not connected | LP_MciOut: bOut2_B0<br>MCI output: Data word 2 - bit 0 |
| C00621/78 | 0: Not connected | LP_MciOut: bOut2_B1<br>MCI output: Data word 2 - bit 1 |
| C00621/79 | 0: Not connected | LP_MciOut: bOut2_B2<br>MCI output: Data word 2 - bit 2 |
| C00621/80 | 0: Not connected | LP_MciOut: bOut2_B3<br>MCI output: Data word 2 - bit 3 |
| C00621/81 | 0: Not connected | LP_MciOut: bOut2_B4<br>MCI output: Data word 2 - bit 4 |
| C00621/82 | 0: Not connected | LP_MciOut: bOut2_B5<br>MCI output: Data word 2 - bit 5 |
| C00621/83 | 0: Not connected | LP_MciOut: bOut2_B6<br>MCI output: Data word 2 - bit 6 |
| C00621/84 | 0: Not connected | LP_MciOut: bOut2_B7<br>MCI output: Data word 2 - bit 7 |
| C00621/85 | 0: Not connected | LP_MciOut: bOut2_B8<br>MCI output: Data word 2 - bit 8 |
| C00621/86 | 0: Not connected | LP_MciOut: bOut2_B9<br>MCI output: Data word 2 - bit 9 |
| C00621/87 | 0: Not connected | LP_MciOut: bOut2_B10<br>MCI output: Data word 2 - bit 10 |
| C00621/88 | 0: Not connected | LP_MciOut: bOut2_B11<br>MCI output: Data word 2 - bit 11 |
| C00621/89 | 0: Not connected | LP_MciOut: bOut2_B12<br>MCI output: Data word 2 - bit 12 |
| C00621/90 | 0: Not connected | LP_MciOut: bOut2_B13<br>MCI output: Data word 2 - bit 13 |
| C00621/91 | 0: Not connected | LP_MciOut: bOut2_B14<br>MCI output: Data word 2 - bit 14 |
| C00621/92 | 0: Not connected | LP_MciOut: bOut2_B15<br>MCI output: Data word 2 - bit 15 |
| C00621/93 | 0: Not connected | LS_SetError_1: bSetError1<br>Input for tripping "US01: User error 1" |
| C00621/94 | 0: Not connected | LS_SetError_1: bSetError2<br>Input for tripping "US02: User error 2" |
| C00621/95 | 0: Not connected | LS_SetError_1: bSetError3<br>Input for tripping "US03: User error 3" |

| Parameter \| Name:<br>**C00621 \| System connection list: Bool** | | Data type: UNSIGNED_16<br>Index: $23954_d = 5D92_h$ |
|---|---|---|
| C00621/96 | 0: Not connected | LS_SetError_1: bSetError4<br>Input for tripping "US04: User error 4" |
| C00621/97 | 0: Not connected | LS_DigitalInput: bCountIn6_Reset<br>Digital input 6: Reset counter |
| C00621/98 | 0: Not connected | LS_DigitalInput: bCountIn6_LoadStartValue<br>Digital input 6: Load counter starting value |
| C00621/99 | 0: Not connected | LS_DigitalOutput: bOut2<br>Digital output 2: Input signal |
| C00621/100 | 0: Not connected | LS_DigitalOutput: bOut3<br>Digital output 3: Input signal |
| C00621/101 | 0: Not connected | LS_DigitalOutput: bOut HighCurrent<br>Digital output (HC) for brake control: Input signal |
| C00621/102 | 0: Not connected | LS_DisFree_b: bDis9<br>Display of digital signal 9 |
| C00621/103 | 0: Not connected | LS_DisFree_b: bDis10<br>Display of digital signal 10 |
| C00621/104 | 0: Not connected | LS_DisFree_b: bDis11<br>Display of digital signal 11 |
| C00621/105 | 0: Not connected | LS_DisFree_b: bDis12<br>Display of digital signal 12 |
| C00621/106 | 0: Not connected | LS_DisFree_b: bDis13<br>Display of digital signal 13 |
| C00621/107 | 0: Not connected | LS_DisFree_b: bDis14<br>Display of digital signal 14 |
| C00621/108 | 0: Not connected | LS_DisFree_b: bDis15<br>Display of digital signal 15 |
| C00621/109 | 0: Not connected | LS_DisFree_b: bDis16<br>Display of digital signal 16 |
| C00621/111 | 0: Not connected | LS_ParReadWrite_1: bExecute<br>Read/write request 1: Trigger |
| C00621/112 | 0: Not connected | LS_ParReadWrite_1: bReadWrite<br>Read/write request 1: Selection of reading/writing |
| C00621/113 | 0: Not connected | LS_ParReadWrite_2: bExecute<br>Read/write request 2: Trigger |
| C00621/114 | 0: Not connected | LS_ParReadWrite_2: bReadWrite<br>Read/write request 2: Selection of reading/writing |
| C00621/115 | 0: Not connected | LS_ParReadWrite_3: bExecute<br>Read/write request 3: Trigger |
| C00621/116 | 0: Not connected | LS_ParReadWrite_3: bReadWrite<br>Read/write request 3: Selection of reading/writing |
| C00621/117 | 0: Not connected | LS_ParReadWrite_4: bExecute<br>Read/write request 4: Trigger |
| C00621/118 | 0: Not connected | LS_ParReadWrite_4: bReadWrite<br>Read/write request 4: Selection of reading/writing |
| C00621/119 | 0: Not connected | LS_ParReadWrite_5: bExecute<br>Read/write request 5: Trigger |
| C00621/120 | 0: Not connected | LS_ParReadWrite_5: bReadWrite<br>Read/write request 5: Selection of reading/writing |
| C00621/121 | 0: Not connected | LS_ParReadWrite_6: bExecute<br>Read/write request 6: Trigger |

| Parameter \| Name:<br>**C00621 \| System connection list: Bool** | | Data type: UNSIGNED_16<br>Index: $23954_d = 5D92_h$ |
|---|---|---|
| C00621/122 | 0: Not connected | LS_ParReadWrite_6: bReadWrite<br>Read/write request 6: Selection of reading/writing |
| C00621/123 | 0: Not connected | LS_WriteParamList: bExecute<br>Writing the parameter list: Activate |
| C00621/124 | 0: Not connected | LS_WriteParamList: bSelectWriteValue_1<br>Writing to parameter list: Selection of value set - 1 |
| C00621/125 | 0: Not connected | LS_WriteParamList: bSelectWriteValue_2<br>Writing to parameter list: Selection of value set - 2 |
| C00621/126 | 0: Not connected | LS_CANManagement: bResetNode<br>Reset CAN node |
| C00621/127 | 0: Not connected | LS_CANManagement: bReInitCAN<br>Reinitialise CAN interface |
| C00621/128 | 0: Not connected | LS_DigitalInput: bPosIn12_Load<br>Frequency input DI1/DI2: Load angle integrator with starting value and reset status signal |
| C00621/147 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn3<br>TP input DI3: Disable TP function dynamically |
| C00621/148 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn4<br>TP input DI4: Disable TP function dynamically |
| C00621/149 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn5<br>TP input DI5: Disable TP function dynamically |
| C00621/150 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn6<br>TP input DI6: Disable TP function dynamically |
| C00621/151 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn7<br>TP input DI7: Disable TP function dynamically |
| C00621/152 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn3_Rising<br>TP input DI3: Disable recognition of rising edges dynamically |
| C00621/153 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn3_Falling<br>TP input DI3: Disable recognition of falling edges dynamically |
| C00621/154 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn4_Rising<br>TP input DI4: Disable recognition of rising edges dynamically |
| C00621/155 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn4_Falling<br>TP input DI4: Disable recognition of falling edges dynamically |
| C00621/156 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn5_Rising<br>TP input DI5: Disable recognition of rising edges dynamically |
| C00621/157 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn5_Falling<br>TP input DI5: Disable recognition of falling edges dynamically |
| C00621/158 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn3Window<br>TP input DI3: Enable acceptance window |
| C00621/159 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn4Window<br>TP input DI4: Enable acceptance window |
| C00621/160 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn5Window<br>TP input DI5: Enable acceptance window |
| C00621/161 | 0: Not connected | LS_AxisBusIO: bSetFail<br>Set IO axis bus into the "error" status / IO data exchange |
| C00621/162 | 0: Not connected | LS_AxisBusIO: bResetFail<br>IO axis bus: Reset "error" status |

| Parameter \| Name:<br>**C00621 \| System connection list: Bool** | | Data type: UNSIGNED_16<br>Index: $23954_d = 5D92_h$ |
|---|---|---|
| C00621/164 | 0: Not connected | LS_MultiEncoder: bSetRef |
| C00621/165 | 0: Not connected | LS_RetainData: bSetRetain_1<br>Retain data: Save input values (selection 1) in retain memory |
| C00621/166 | 0: Not connected | LS_RetainData: bSetRetain_2<br>Retain data: Save input values (selection 2) in retain memory |
| C00621/167 | 0: Not connected | LS_RetainData: bSetRetain_3<br>Retain data: Save input values (selection 3) in retain memory |
| C00621/168 | 0: Not connected | LS_RetainData: bLoadParams<br>Retain data: Set selected outputs & retain values to parameter values |
| C00621/169 | 0: Not connected | LS_RetainData: bIn1<br>Retain data: Binary input value 1 |
| C00621/170 | 0: Not connected | LS_RetainData: bIn2<br>Retain data: Binary input value 2 |
| C00621/171 | 0: Not connected | LS_RetainData: bIn3<br>Retain data: Binary input value 3 |
| C00621/172 | 0: Not connected | LS_RetainData: bIn4<br>Retain data: Binary input value 4 |
| C00621/173 | 0: Not connected | LS_AxisBusAux: bReadWrite<br>Axis bus: Selection of reading/writing |
| C00621/174 | 0: Not connected | LS_AxisBusAux: bExecute<br>Axis bus: Data acceptance |
| C00621/175 | 0: Not connected | LS_AxisBusAux: bStop<br>Axis bus: Stop data transfer |
| C00621/176 | 0: Not connected | LS_DigitalOutput: bUserLED<br>Switch on USER-LED on the inverter front panel |
| C00621/177 | 0: Not connected | LS_TouchProbe: bDisableTPEncoderWindow |
| C00621/178 | 0: Not connected | LS_TouchProbe: bDisableTPResolverWindow |
| C00621/179 | 0: Not connected | LS_TouchProbe: bDisableTPEncoder |
| C00621/180 | 0: Not connected | LS_TouchProbe: bDisableTPResolver |
| C00621/181 | 0: Not connected | LF_DFOut: bSynRdy |
| C00621/184 | 0: Not connected | LP_CanOut4: bOut1_B0 |
| C00621/185 | 0: Not connected | LP_CanOut4: bOut1_B1 |
| C00621/186 | 0: Not connected | LP_CanOut4: bOut1_B2 |
| C00621/187 | 0: Not connected | LP_CanOut4: bOut1_B3 |
| C00621/188 | 0: Not connected | LP_CanOut4: bOut1_B4 |
| C00621/189 | 0: Not connected | LP_CanOut4: bOut1_B5 |
| C00621/190 | 0: Not connected | LP_CanOut4: bOut1_B6 |
| C00621/191 | 0: Not connected | LP_CanOut4: bOut1_B7 |
| C00621/192 | 0: Not connected | LP_CanOut4: bOut1_B8 |
| C00621/193 | 0: Not connected | LP_CanOut4: bOut1_B9 |
| C00621/194 | 0: Not connected | LP_CanOut4: bOut1_B10 |
| C00621/195 | 0: Not connected | LP_CanOut4: bOut1_B11 |
| C00621/196 | 0: Not connected | LP_CanOut4: bOut1_B12 |
| C00621/197 | 0: Not connected | LP_CanOut4: bOut1_B13 |
| C00621/198 | 0: Not connected | LP_CanOut4: bOut1_B14 |

| Parameter \| Name: C00621 \| System connection list: Bool | | Data type: UNSIGNED_16 Index: $23954_d = 5D92_h$ |
|---|---|---|
| C00621/199 | 0: Not connected | LP_CanOut4: bOut1_B15 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00622**

| Parameter \| Name: C00622 \| System connection list: Angle | | Data type: UNSIGNED_16 Index: $23953_d = 5D91_h$ |
|---|---|---|
| Connection parameters: 32-bit inputs<br>• Selection of the 32-bit output signals for connection with the 32-bit input signals.<br>• The selection list contains all 32-bit output signals which can be assigned to the 32-bit inputs mapped by the subcodes.<br>• Non-listed subcodes are "reserved". | | |
| **Selection list** | | |
| See selection list - angle signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00622/1 | 0: Not connected | LS_DisFree_p: dnDis1_p<br>Display of 32-bit signal 1 |
| C00622/2 | 0: Not connected | LS_DisFree_p: dnDis2_p<br>Display of 32-bit signal 2 |
| C00622/3 | 0: Not connected | LS_DisFree_p: dnDis3_p<br>Display of 32-bit signal 3 |
| C00622/4 | 0: Not connected | LS_DisFree_p: dnDis4_p<br>Display of 32-bit signal 4 |
| C00622/5 | 0: Not connected | LS_DisFree_p: dnDis5_p<br>Display of 32-bit signal 5 |
| C00622/6 | 0: Not connected | LS_DisFree_p: dnDis6_p<br>Display of 32-bit signal 6 |
| C00622/7 | 0: Not connected | LS_DisFree_p: dnDis7_p<br>Display of 32-bit signal 7 |
| C00622/8 | 0: Not connected | LS_DisFree_p: dnDis8_p<br>Display of 32-bit signal 8 |
| C00622/9 | 0: Not connected | LP_CanOut1: dnOut34_p<br>CAN1 output: Data words 3 + 4 |
| C00622/10 | 0: Not connected | LP_CanOut2: dnOut34_p<br>CAN2 output: Data words 3 + 4 |
| C00622/11 | 0: Not connected | LP_CanOut3: dnOut34_p<br>CAN3 output: Data words 3 + 4 |
| C00622/12 | 0: Not connected | LP_MciOut: dnOut34_p<br>MCI output: Data words 3 + 4 |
| C00622/13 | 0: Not connected | LS_DigitalInput: dnPosIn12_Set_p<br>Frequency input DI1/DI2: Starting value for angle integrator |
| C00622/16 | 0: Not connected | LS_RetainData: dnIn1<br>Retain data: 32-bit input value 1 |
| C00622/17 | 0: Not connected | LS_RetainData: dnIn2<br>Retain data: 32-bit input value 2 |
| C00622/18 | 0: Not connected | LS_RetainData: dnIn3<br>Retain data: 32-bit input value 3 |
| C00622/19 | 0: Not connected | LS_RetainData: dnIn4<br>Retain data: 32-bit input value 4 |

| Parameter \| Name: **C00622 \| System connection list: Angle** | | Data type: UNSIGNED_16 Index: $23953_d = 5D91_h$ |
|---|---|---|
| C00622/20 | 0: Not connected | LS_AxisBusOut: dnLine12<br>Axis bus output: Line data (word 1 + 2) |
| C00622/21 | 0: Not connected | LS_AxisBusOut: dnCas12<br>Axis bus output: Cascaded data (word 1 + 2) |
| C00622/22 | 0: Not connected | LS_AxisBusOut: dnCas34<br>Axis bus output: Cascaded data (word 3 + 4) |
| C00622/23 | 0: Not connected | LS_AxisBusAux: dnAuxOut12<br>Axis bus output: @sel data (word 1 + 2) |
| C00622/24 | 0: Not connected | LS_AxisBusAux: dnAuxOut34<br>Axis bus output: @sel data (word 3 + 4) |
| C00622/25 | 0: Not connected | LP_CanOut4: dnOut34_p<br>CAN4 output: Data words 3 + 4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00630**

| Parameter \| Name: **C00630 \| L_Limit 1-2: Min/Max** | | | Data type: INTEGER_16 Index: $23945_d = 5D89_h$ |
|---|---|---|---|
| Setting the limits | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00630/1 | -199.99 % | | L_Limit_1: Min.Limit |
| C00630/2 | 199.99 % | | L_Limit_1: Max.Limit |
| C00630/3 | -199.99 % | | L_Limit_2: Min.Limit |
| C00630/4 | 199.99 % | | L_Limit_2: Max.Limit |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00631**

| Parameter \| Name: **C00631 \| L_LimitPhi 1-3: Min/Max** | | | Data type: INTEGER_32 Index: $23944_d = 5D88_h$ |
|---|---|---|---|
| Setting the limits | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00631/1 | -2147483647 incr. | | L_LimitPhi_1: Min.Limit |
| C00631/2 | 2147483647 incr. | | L_LimitPhi_1: Max.Limit |
| C00631/3 | -2147483647 incr. | | L_LimitPhi_2: Min.Limit |
| C00631/4 | 2147483647 incr. | | L_LimitPhi_2: Max.Limit |
| C00631/5 | -2147483647 incr. | | L_LimitPhi_3: Min.Limit |
| C00631/6 | 2147483647 incr. | | L_LimitPhi_3: Max.Limit |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00632**

| Parameter \| Name: **C00632 \| L_NSet_1: Max.SkipFrq.** | | | Data type: INTEGER_16 Index: $23943_d = 5D87_h$ |
|---|---|---|---|
| Maximum limit values for the speed blocking zones<br>• Selection of the maximum limit values for the blocking zones in which the speed must not be constant. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00632/1 | 0.00 % | | L_NSet_1: Blocking speed1 max |
| C00632/2 | 0.00 % | | L_NSet_1: Blocking speed2 max |
| C00632/3 | 0.00 % | | L_NSet_1: Blocking speed3 max |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00633**

| Parameter \| Name: **C00633 \| L_NSet_1: Min.SkipFrq.** | | | Data type: INTEGER_16 Index: $23942_d = 5D86_h$ |
|---|---|---|---|
| Minimum limit values for the speed blocking zones<br>• Selection of the minimum limit values for the blocking zones in which the speed must not be constant. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00633/1 | 0.00 % | | L_NSet_1: Blocking speed1 min |
| C00633/2 | 0.00 % | | L_NSet_1: Blocking speed2 min |
| C00633/3 | 0.00 % | | L_NSet_1: Blocking speed3 min |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00634**

| Parameter \| Name: **C00634 \| L_NSet_1: wState** | | | Data type: UNSIGNED_16 Index: $23941_d = 5D85_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Bit coded status display | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | No blocking zone active | | 1 ≡ No blocking zone set for constant speeds |
| Bit 1 | Blocking zone 1 active | | 1 ≡ Suppression of constant speed characteristics within the limits of blocking zone 1 |
| Bit 2 | Blocking zone 2 active | | 1 ≡ Suppression of constant speed characteristics within the limits of blocking zone 2 |
| Bit 3 | Blocking zone 3 active | | 1 ≡ Suppression of constant speed characteristics within the limits of blocking zone 3 |
| Bit 4 | Jog in blocking zone | | 1 ≡ A ramp is used to keep the speed setpoint within a speed blocking zone |
| Bit 5 | MaxLimit active | | 1 ≡ Speed setpoint is at the maximum speed limit |
| Bit 6 | MinLimit active | | 1 ≡ Speed setpoint is at the minimum speed limit |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00635**

| Parameter \| Name: **C00635 \| L_NSet_1: nMaxLimit** | | | Data type: INTEGER_16 Index: $23940_d = 5D84_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Maximum speed setpoint for speed setpoint limitation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **199.99 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00636**

| Parameter \| Name: **C00636 \| L_NSet_1: nMinLimit** | | | Data type: INTEGER_16 Index: $23939_d = 5D83_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Minimum speed setpoint for speed setpoint limitation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **-199.99 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00637**

| Parameter \| Name: **C00637 \| L_NSet_1: Output blocking zones** | | | Data type: INTEGER_16 Index: $23938_d = 5D82_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Speed setpoint is displayed after being processed by blocking zone function | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00638**

| Parameter \| Name: **C00638 \| L_NSet_1: Output ramp rounding** | | | Data type: INTEGER_16 Index: $23937_d = 5D81_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Speed setpoint is displayed after being processed by PT1 filter function | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00639**

| Parameter \| Name: **C00639 \| L_NSet_1: Output additional value** | | | Data type: INTEGER_16 Index: $23936_d = 5D80_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Additional speed setpoint is displayed after being processed by the ramp generator | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00640**

| Parameter \| Name: **C00640 \| L_NSet_1: nNOut_a** | | | Data type: INTEGER_16 Index: $23935_d = 5D7F_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Display of the generated main speed setpoint at the output *nNOut_a* | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00643**

| Parameter \| Name: **C00643 \| Resp. to PLI monitoring** | | Data type: UNSIGNED_8 Index: $23932_d = 5D7C_h$ |
|---|---|---|
| ▸ Pole position identification | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00643/1 | 1: Fault | Response to PLI monitoring |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00644**

| Parameter \| Name: C00644 \| PLI traversing direction | | | Data type: UNSIGNED_8 Index: $23931_d = 5D7B_h$ |
|---|---|---|---|
| | | | ▸ Pole position identification |
| **Selection list** | | | |
| 0 | right rotating field | | |
| 1 | left rotating field | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00644/1 | 0: right rotating field | | PLI 360° traversing direction |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00645**

| Parameter \| Name: C00645 \| PLI max. permissible deflection | | | Data type: INTEGER_16 Index: $23930_d = 5D7A_h$ |
|---|---|---|---|
| | | | ▸ Pole position identification |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -6.0 | ° | 90.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00645/1 | 0.0 ° | | Reserved |
| C00645/2 | 0.0 ° | | PLI 360° max. error tolerance |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00646**

| Parameter \| Name: C00646 \| PLI current amplitude | | | Data type: UNSIGNED_16 Index: $23929_d = 5D79_h$ |
|---|---|---|---|
| | | | ▸ Pole position identification |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | % | 1000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00646/1 | 1 % | | Reserved |
| C00646/2 | 100 % | | PLI 360° current amplitude |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00647**

| Parameter \| Name: C00647 \| PLI ramp time | | | Data type: UNSIGNED_16 Index: $23928_d = 5D78_h$ |
|---|---|---|---|
| | | | ▸ Pole position identification |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 5 | | 1000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00647/1 | 5 | | Reserved |
| C00647/2 | 100 | | PLI 360° ramp time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00650**

| Parameter \| Name: **C00650 \| L_Arithmetik 3-5: Function** | | Data type: UNSIGNED_8 Index: $23925_d = 5D75_h$ |
|---|---|---|
| Selection of the internal arithmetics | | |
| **Selection list** | | |
| 0 | nOut_a = nIn1_a | |
| 1 | nOut_a = nIn1_a + nIn2_a | |
| 2 | nOut_a = nIn1_a - nIn2_a | |
| 3 | nOut_a = (nIn1_a * nIn2_a) / 100% | |
| 4 | nOut_a = (nIn1_a * 1%) / \|nIn2_a\| | |
| 5 | nOut_a = (nIn1_a * 100%) / (100% - nIn2_a) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00650/1 | 0: nOut_a = nIn1_a | L_Arithmetik_3: Function |
| C00650/2 | 0: nOut_a = nIn1_a | L_Arithmetik_4: Function |
| C00650/3 | 0: nOut_a = nIn1_a | L_Arithmetik_5: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00653**

| Parameter \| Name: **C00653 \| Sensibility - Setpoint feedforward control** | | Data type: UNSIGNED_8 Index: $23922_d = 5D72_h$ |
|---|---|---|
| From version 12.00.00<br>Selection of the sensitivity of the differential setpoint feedforward control<br>• Depending on the selection, the number of indicated higher-order bits is evaluated.<br><br>**Note:**<br>The most significant bit determines the sign of the value, the remaining bits determine the numerical value. | | |
| **Selection list** | | **Info** |
| 0 | Inactive | |
| 1 | 15 bits | Bit 0 ... bit 14 are evaluated |
| 2 | 14 bits | Bit 0 ... bit 13 are evaluated |
| 3 | 13 bits | Bit 0 ... bit 12 are evaluated |
| 4 | 12 bits | Bit 0 ... bit 11 are evaluated |
| 5 | 11 Bit | Bit 0 ... bit 10 are evaluated |
| 6 | 10 Bit | Bit 0 ... bit 9 are evaluated |
| 7 | 9 Bit | Bit 0 ... bit 8 are evaluated |
| **Subcodes** | **Lenze setting** | **Info** |
| C00653/1 | 0: Inactive | Sensibility - Setpoint feedforward control |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00654**

| Parameter \| Name: **C00654 \| Source of diff. setpoint feedforward control** | | | Data type: UNSIGNED_8 Index: $23921_d = 5D71_h$ |
|---|---|---|---|
| From version 12.00.00<br>Selection of the process signal for the setpoint feedforward control | | | |
| **Selection list** | | | **Info** |
| 0 | nSpeedSetValue_a | | |
| 1 | nSpeedSetValueInertia_a | | The new process signal *nSpeedSetValueInertia_a* at SB LS_MotorInterface can be used to define any input value (e.g. position or process controller setpoint) for torque feedforward control. |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00654/1 | 0: nSpeedSetValue_a | | Source of diff. setpoint feedforward control |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00658**

| Parameter \| Name: **C00658 \| PRBS amplitude** | Data type: INTEGER_16 Index: $23917_d = 5D6D_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00659**

| Parameter \| Name: **C00659 \| PRBS duration** | Data type: UNSIGNED_16 Index: $23916_d = 5D6C_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00660**

| Parameter \| Name: **C00660 \| L_FixSet_a_1: Analog values** | | | Data type: INTEGER_16 Index: $23915_d = 5D6B_h$ |
|---|---|---|---|
| FB L_FixSet_a_1: Setting of the fixed values | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00660/1 | 0.00 % | | Fixed value 0 ... 15 |
| C00660/... | | | |
| C00660/16 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00661**

| Parameter \| Name: **C00661 \| L_FixSet_w_1: Fixed values** | | | Data type: UNSIGNED_16 Index: $23914_d = 5D6A_h$ |
|---|---|---|---|
| FB L_FixSet_w_1: Setting of the fixed values | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00661/1 | 0 | | Fixed value 0 ... 15 |
| C00661/... | | | |
| C00661/16 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00662**

| Parameter \| Name: **C00662 \| L_FixSet_w_2: Fixed values** | | | Data type: UNSIGNED_16 Index: $23913_d = 5D69_h$ |
|---|---|---|---|
| FB L_FixSet_w_2: Setting of the fixed values | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00662/1 | 0 | | Fixed value 0 ... 15 |
| C00662/... | | | |
| C00662/16 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00670**

| Parameter \| Name: **C00670 \| L_OffsetGainP_1: Gain** | | | Data type: INTEGER_32 Index: $23905_d = 5D61_h$ |
|---|---|---|---|
| The L_OffsetGainP_1 FB: Gain as multiplier of the input signal + offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -100.0000 | | 100.0000 | **1.0000** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C00671**

| Parameter \| Name: **C00671 \| L_OffsetGainP_2: Gain** | | | Data type: INTEGER_32 Index: $23904_d = 5D60_h$ |
|---|---|---|---|
| The L_OffsetGainP_2 FB: Gain as multiplier of the input signal + offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -100.0000 | | 100.0000 | **1.0000** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C00672**

| Parameter \| Name: **C00672 \| L_OffsetGainP_3: Gain** | | | Data type: INTEGER_32 Index: $23903_d = 5D5F_h$ |
|---|---|---|---|
| The L_OffsetGainP_3 FB: Gain as multiplier of the input signal + offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -100.0000 | | 100.0000 | **1.0000** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C00673**

| Parameter \| Name: **C00673 \| L_OffsetGainPhiP 1-2: Offset** | | | Data type: INTEGER_32 Index: $23902_d = 5D5E_h$ |
|---|---|---|---|
| Angular offset (is added to the angular input signal) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00673/1 | 0 incr. | | L_OffsetGainPhiP_1: Offset |
| C00673/2 | 0 incr. | | L_OffsetGainPhiP_2: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00674**

| Parameter \| Name: **C00674 \| L_OffsetGainPhiP 1-2: Gain** | | | Data type: INTEGER_32 Index: $23901_d = 5D5D_h$ |
|---|---|---|---|
| Angular gain as multiplier of the input signal + angular offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00674/1 | 65536 | | L_OffsetGainPhiP_1: Gain |
| C00674/2 | 65536 | | L_OffsetGainPhiP_2: Gain |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00677**

| Parameter \| Name: **C00677 \| L_GainOffsetP 1-3: Parameter** | | | Data type: INTEGER_16 Index: $23898_d = 5D5A_h$ |
|---|---|---|---|
| Gain and offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00677/1 | 100.00 % | | L_GainOffsetP_1: Gain |
| C00677/2 | 0.00 % | | L_GainOffsetP_1: Offset |
| C00677/3 | 100.00 % | | L_GainOffsetP_2: Gain |
| C00677/4 | 0.00 % | | L_GainOffsetP_2: Offset |
| C00677/5 | 100.00 % | | L_GainOffsetP_3: Gain |
| C00677/6 | 0.00 % | | L_GainOffsetP_3: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00678**

| Parameter \| Name: **C00678 \| L_GainOffsetPhiP 1-2: Parameter** | | | Data type: INTEGER_32 Index: $23897_d = 5D59_h$ |
|---|---|---|---|
| Gain and offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00678/1 | 65536 | | L_GainOffsetPhiP_1: Gain |
| C00678/2 | 0 | | L_GainOffsetPhiP_1: Offset |
| C00678/3 | 65536 | | L_GainOffsetPhiP_2: Gain |
| C00678/4 | 0 | | L_GainOffsetPhiP_2: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00679**

| Parameter \| Name: **C00679 \| L_MulDiv_2: Parameter** | | | Data type: INTEGER_16 Index: $23896_d = 5D58_h$ |
|---|---|---|---|
| The L_MulDiv_2 FB: Numerator and denominator | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00679/1 | 0 | | L_MulDiv_2: Numerator |
| C00679/2 | 10000 | | L_MulDiv_2: Denominator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00680**

| Parameter \| Name: **C00680 \| L_Compare_1: Fct.** | | Data type: UNSIGNED_8 Index: $23895_d = 5D57_h$ |
|---|---|---|
| The L_Compare_1 FB: Comparison operation<br>• If the statement of the selected comparison operation is true, the binary *bOut* output will be set to TRUE. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 1 | In1 = In2 | |
| 2 | In1 > In2 | |
| 3 | In1 < In2 | |
| 4 | \|In1\| = \|In2\| | |
| 5 | \|In1\| > \|In2\| | |
| **6** | **\|In1\| < \|In2\|** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00681**

| Parameter \| Name: **C00681 \| L_Compare_1: Hysteresis** | | | Data type: INTEGER_16 Index: $23894_d = 5D56_h$ |
|---|---|---|---|
| The L_Compare_1 FB: Hysteresis for the comparison function selected in C00680 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **0.50 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00682**

| Parameter \| Name: **C00682 \| L_Compare_1: Window** | | | Data type: INTEGER_16 Index: $23893_d = 5D55_h$ |
|---|---|---|---|
| The L_Compare_1 FB: Window for the comparison function selected in C00680 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **2.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00685**

| Parameter \| Name: **C00685 \| L_Compare_2: Fct.** | Data type: UNSIGNED_8 Index: $23890_d = 5D52_h$ |
|---|---|
| The L_Compare_2 FB: Comparison operation<br>• If the statement of the selected comparison operation is true, the binary *bOut* output will be set to TRUE. | |
| **Selection list** (Lenze setting printed in bold) | |
| 1 | In1 = In2 |
| 2 | In1 > In2 |
| 3 | In1 < In2 |
| **4** | **\|In1\| = \|In2\|** |
| 5 | \|In1\| > \|In2\| |
| 6 | \|In1\| < \|In2\| |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00686**

| Parameter \| Name: **C00686 \| L_Compare_2: Hysteresis** | | | Data type: INTEGER_16 Index: $23889_d = 5D51_h$ |
|---|---|---|---|
| The L_Compare_2 FB: Hysteresis for the comparison function selected in C00685 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **0.50 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00687**

| Parameter \| Name: **C00687 \| L_Compare_2: Window** | | | Data type: INTEGER_16 Index: $23888_d = 5D50_h$ |
|---|---|---|---|
| The L_Compare_2 FB: Window for the comparison function selected in C00685 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **2.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00690**

| Parameter \| Name: **C00690 \| L_Compare_3: Function** | | | Data type: UNSIGNED_8 Index: $23885_d$ = $5D4D_h$ |
|---|---|---|---|
| The L_Compare_3 FB: Comparison operation<br>• If the statement of the selected comparison operation is true, the binary *bOut* output will be set to TRUE. | | | |
| **Selection list** (Lenze setting printed in bold) | | | |
| **1** | **In1 = In2** | | |
| 2 | In1 > In2 | | |
| 3 | In1 < In2 | | |
| 4 | \|In1\| = \|In2\| | | |
| 5 | \|In1\| > \|In2\| | | |
| 6 | \|In1\| < \|In2\| | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00691**

| Parameter \| Name: **C00691 \| L_Compare_3: Hysteresis** | | | Data type: INTEGER_16 Index: $23884_d$ = $5D4C_h$ |
|---|---|---|---|
| FB L_Compare_3: Hysteresis for the comparison operation selected in C00690 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00692**

| Parameter \| Name: **C00692 \| L_Compare_3: Window** | | | Data type: INTEGER_16 Index: $23883_d$ = $5D4B_h$ |
|---|---|---|---|
| FB L_Compare_3: window for the comparison operation selected in C00690 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00693**

| Parameter \| Name: **C00693 \| L_Compare 4-5: Function** | | | Data type: UNSIGNED_8 Index: $23882_d$ = $5D4A_h$ |
|---|---|---|---|
| Comparison operation<br>• If the statement of the selected comparison operation is true, the binary *bOut* output will be set to TRUE. | | | |
| **Selection list** | | | |
| 1 | In1 = In2 | | |
| 2 | In1 > In2 | | |
| 3 | In1 < In2 | | |
| 4 | \|In1\| = \|In2\| | | |
| 5 | \|In1\| > \|In2\| | | |
| 6 | \|In1\| < \|In2\| | | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C00693/1 | 1: In1 = In2 | L_Compare_4: Function | |
| C00693/2 | 1: In1 = In2 | L_Compare_5: Function | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00694**

| Parameter \| Name: **C00694 \| L_Compare 4-5: Hysteresis** | | | Data type: INTEGER_16 Index: $23881_d = 5D49_h$ |
|---|---|---|---|
| Hysteresis for the comparison operation selected in C00693 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00694/1 | 0.00 % | | L_Compare_4: Hysteresis |
| C00694/2 | 0.00 % | | L_Compare_5: Hysteresis |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00695**

| Parameter \| Name: **C00695 \| L_Compare 4-5: Window** | | | Data type: INTEGER_16 Index: $23880_d = 5D48_h$ |
|---|---|---|---|
| Window for the comparison operation selected in C00693 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00695/1 | 0.00 % | | L_Compare_4: Window |
| C00695/2 | 0.00 % | | L_Compare_5: Window |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00696**

| Parameter \| Name: **C00696 \| L_OffsetGainP_1: Offset** | | | Data type: INTEGER_16 Index: $23879_d = 5D47_h$ |
|---|---|---|---|
| The L_OffsetGainP_1 FB: Offset (additive to the input signal) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00697**

| Parameter \| Name: **C00697 \| L_OffsetGainP_2: Offset** | | | Data type: INTEGER_16 Index: $23878_d = 5D46_h$ |
|---|---|---|---|
| The L_OffsetGainP_2 FB: Offset (additive to the input signal) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00698**

| Parameter \| Name: **C00698 \| L_OffsetGainP_3: Offset** | | | Data type: INTEGER_16 Index: $23877_d = 5D45_h$ |
|---|---|---|---|
| The L_OffsetGainP_3 FB: Offset (additive to the input signal) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00699**

| Parameter \| Name: **C00699 \| L_MulDiv_1: Parameter** | | | Data type: INTEGER_16 Index: $23876_d = 5D44_h$ |
|---|---|---|---|
| The L_MulDiv_1 FB: Numerator and denominator | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00699/1 | 0 | | L_MulDiv_1: Numerator |
| C00699/2 | 10000 | | L_MulDiv_1: Denominator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00700**

| Parameter \| Name: **C00700 \| LA_NCtrl: Analog connection list** | | Data type: UNSIGNED_16 Index: $23875_d = 5D43_h$ |
|---|---|---|
| Connection parameters for "Actuating drive speed" application: 16-bit inputs<br>• Selection of the 16 bit output signals to be connected to the 16 bit input signals<br>• The selection list contains all 16 bit output signals which can be assigned to the 16 bit inputs displayed by the subcodes. | | |
| **Selection list** | | |
| See selection list - analog signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00700/1 | 20005: LS_ParFix: wDriveCtrl | LA_NCtrl: wCANDriveControl<br>Input for control word from CAN to device control |
| C00700/2 | 20005: LS_ParFix: wDriveCtrl | LA_NCtrl: wMCIDriveControl<br>Input for control word from communication interface to device control |
| C00700/3 | 20012: LS_ParFree_a: nC472_3_a | LA_NCtrl: nTorqueMotLim_a<br>Input for maximum torque in motor mode |
| C00700/4 | 20013: LS_ParFree_a: nC472_4_a | LA_NCtrl: nTorqueGenLim_a<br>Input for maximum torque in generator mode |
| C00700/5 | 0: Not connected | LA_NCtrl: nPIDVpAdapt_a<br>Input for adapting the PID controller gain |
| C00700/6 | 0: Not connected | LA_NCtrl: nPIDActValue_a<br>Input for actual PID controller value |
| C00700/7 | 16000: LS_AnalogInput: nIn1_a | LA_NCtrl: nMainSetValue_a<br>Input for main speed setpoint |
| C00700/8 | 0: Not connected | LA_NCtrl: nAuxSetValue_a<br>Input for additional speed setpoint |
| C00700/9 | 0: Not connected | LA_NCtrl: nGPAnalogSwitchIn1_a<br>Input for analog switch - analog signal 1 |
| C00700/10 | 0: Not connected | LA_NCtrl: nGPAnalogSwitchIn2_a<br>Input for analog switch - analog signal 2 |
| C00700/11 | 0: Not connected | LA_NCtrl: nGPArithmetikIn1_a<br>Input for arithmetic function - analog signal 1 |
| C00700/12 | 0: Not connected | LA_NCtrl: nGPArithmetikIn2_a<br>Input for arithmetic function - analog signal 2 |
| C00700/13 | 0: Not connected | LA_NCtrl: nGPMulDivIn_a<br>Input for analog signal for multiplication/division |
| C00700/14 | 0: Not connected | LA_NCtrl: nGPCompareIn1_a<br>Input for comparison operation - analog signal 1 |

| Parameter \| Name: **C00700 \| LA_NCtrl: Analog connection list** | | Data type: UNSIGNED_16 Index: $23875_d = 5D43_h$ |
|---|---|---|
| C00700/15 | 0: Not connected | LA_NCtrl: nGPCompareIn2_a<br>Input for comparison operation - analog signal 2 |
| C00700/16 | 0: Not connected | LA_NCtrl: nVoltageAdd_a<br>Input for additive voltage boost |
| C00700/17 | 0: Not connected | LA_NCtrl: nPIDInfluence_a<br>Input for influence signal of PID controller correcting variable |
| C00700/18 | 0: Not connected | LA_NCtrl: nPIDSetValue_a<br>Input for PID controller setpoint |
| C00700/19 | 0: Not connected | LA_NCtrl: nPWMAngleOffset<br>Input for pulse width modulation phase offset |
| C00700/20 | 0: Not connected | LA_NCtrl: nBoost_a<br>Input for additional setpoint for motor voltage at speed = 0 |
| C00700/21 | 0: Not connected | LA_NCtrl: wSMCtrl<br>Interface to the optional safety system |
| C00700/22 | 0: Not connected | Reserved |
| C00700/23 | 0: Not connected | Reserved |
| C00700/24 | 0: Not connected | Reserved |
| C00700/25 | 0: Not connected | Reserved |
| C00700/26 | 0: Not connected | LA_NCtrl: wFreeIn1<br>Input for user signal 1 |
| C00700/27 | 0: Not connected | LA_NCtrl: wFreeIn2<br>Input for user signal 2 |
| C00700/28 | 0: Not connected | LA_NCtrl: wFreeIn3<br>Input for user signal 3 |
| C00700/29 | 0: Not connected | LA_NCtrl: wFreeIn4<br>Input for user signal 4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00701**

| Parameter \| Name: **C00701 \| LA_NCtrl: Digital connection list** | | Data type: UNSIGNED_16 Index: $23874_d = 5D42_h$ |
|---|---|---|
| Connection parameters for "Actuating drive speed" application: Binary inputs<br>• Selection of the binary output signals to be connected to the binary input signals<br>• The selection list contains all binary output signals which can be assigned to the binary inputs mapped by the subcodes. | | |
| **Selection list** | | |
| See selection list - digital signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00701/1 | 0: Not connected | LA_NCtrl: bCInh<br>Control input for setting controller inhibit |
| C00701/2 | 16008: LS_DigitalInput: bCInh | LA_NCtrl: bFailReset<br>Control input for error acknowledgement |
| C00701/3 | 0: Not connected | LA_NCtrl: bSetQuickstop<br>Control input for quick stop request |
| C00701/4 | 16002: LS_DigitalInput: bIn3 | LA_NCtrl: bSetDCBrake<br>Control input for DC-injection braking request |

| Parameter \| Name:<br>**C00701 \| LA_NCtrl: Digital connection list** | | Data type: UNSIGNED_16<br>Index: $23874_d = 5D42_h$ |
|---|---|---|
| C00701/5 | 0: Not connected | LA_NCtrl: bRFG_Stop<br>Control input for stopping the speed ramp function generator |
| C00701/6 | 0: Not connected | LA_NCtrl: bRFG_0<br>Control input for setting the speed ramp function generator to 0 |
| C00701/7 | 0: Not connected | Reserved |
| C00701/8 | 16003: LS_DigitalInput: bIn4 | LA_NCtrl: bSetSpeedCcw<br>Control input for change of direction of rotation |
| C00701/9 | 16000: LS_DigitalInput: bIn1 | LA_NCtrl: bJogSpeed1<br>Selection input for fixed setpoints |
| C00701/10 | 16001: LS_DigitalInput: bIn2 | LA_NCtrl: bJogSpeed2<br>Selection input for fixed setpoints |
| C00701/11 | 0: Not connected | LA_NCtrl: bJogSpeed4<br>Selection input for fixed setpoints |
| C00701/12 | 0: Not connected | LA_NCtrl: bJogSpeed8<br>Selection input for fixed setpoints |
| C00701/13 | 0: Not connected | LA_NCtrl: bJogRamp1<br>Selection input for additional acceleration/deceleration times |
| C00701/14 | 0: Not connected | LA_NCtrl: bJogRamp2<br>Selection input for additional acceleration/deceleration times |
| C00701/15 | 0: Not connected | LA_NCtrl: bJogRamp4<br>Selection input for additional acceleration/deceleration times |
| C00701/16 | 0: Not connected | LA_NCtrl: bJogRamp8<br>Selection input for additional acceleration/deceleration times |
| C00701/17 | 0: Not connected | LA_NCtrl: bMPOTInAct<br>Control input for deactivation of motor potentiometer |
| C00701/18 | 0: Not connected | LA_NCtrl: bMPOTUp<br>Control input for motor potentiometer ramp-up |
| C00701/19 | 0: Not connected | LA_NCtrl: bMPOTDown<br>Control input for motor potentiometer ramp-down |
| C00701/20 | 0: Not connected | LA_NCtrl: bMBRKRelease<br>Control input for manual holding brake release request |
| C00701/21 | 0: Not connected | LA_NCtrl: bGPFree1<br>Input for binary user signal |
| C00701/22 | 0: Not connected | LA_NCtrl: bGPFree2<br>Input for binary user signal |
| C00701/23 | 0: Not connected | LA_NCtrl: bGPAnalogSwitchSet<br>Control input for analog-value selector change-over |
| C00701/24 | 0: Not connected | LA_NCtrl: bGPDigitalDelayIn<br>Input for digital signal with time delay |
| C00701/25 | 0: Not connected | LA_NCtrl: bGPLogicIn1<br>Input signal 1 for digital logic |
| C00701/26 | 0: Not connected | LA_NCtrl: bGPLogicIn2<br>Input signal 2 for digital logic |
| C00701/27 | 0: Not connected | LA_NCtrl: bGPLogicIn3<br>Input signal 3 for digital logic |

| Parameter \| Name: **C00701 \| LA_NCtrl: Digital connection list** | | Data type: UNSIGNED_16 Index: $23874_d = 5D42_h$ |
|---|---|---|
| C00701/28 | 0: Not connected | LA_NCtrl: bGPDFlipFlopInD<br>Control input for DFlipFlop setting signal |
| C00701/29 | 0: Not connected | LA_NCtrl: bGPDFlipFlopInClk<br>Control input for DFlipFlop clock signal |
| C00701/30 | 0: Not connected | LA_NCtrl: bGPDFlipFlopInClr<br>Control input for DFlipFlop reset signal |
| C00701/31 | 0: Not connected | LA_NCtrl: bMPotEnable<br>Control input for activation of motor potentiometer |
| C00701/32 | 0: Not connected | LA_NCtrl: bPIDEnableInfluenceRamp<br>Control input for activation of influence of output correcting variable of PID controller |
| C00701/33 | 0: Not connected | LA_NCtrl: bPIDIOff<br>Control input for deactivation of PID controller I component |
| C00701/34 | 20000: LS_ParFix: bTrue | LA_NCtrl: bRLQCw<br>Control input for activation of CW direction of rotation of speed setpoint |
| C00701/35 | 0: Not connected | LA_NCtrl: bRLQCcw<br>Control input for activation of CCW direction of rotation of speed setpoint |
| C00701/36 | 0: Not connected | Reserved |
| C00701/37 | 0: Not connected | Reserved |
| C00701/38 | 0: Not connected | Reserved |
| C00701/39 | 0: Not connected | Reserved |
| C00701/40 | 0: Not connected | Reserved |
| C00701/41 | 0: Not connected | LA_NCtrl: bFreeIn1<br>Input for binary user signal 1 |
| C00701/42 | 0: Not connected | LA_NCtrl: bFreeIn2<br>Input for binary user signal 2 |
| C00701/43 | 0: Not connected | LA_NCtrl: bFreeIn3<br>Input for binary user signal 3 |
| C00701/44 | 0: Not connected | LA_NCtrl: bFreeIn4<br>Input for binary user signal 4 |
| C00701/45 | 0: Not connected | LA_NCtrl: bFreeIn5<br>Input for binary user signal 5 |
| C00701/46 | 0: Not connected | LA_NCtrl: bFreeIn6<br>Input for binary user signal 6 |
| C00701/47 | 0: Not connected | LA_NCtrl: bFreeIn7<br>Input for binary user signal 7 |
| C00701/48 | 0: Not connected | LA_NCtrl: bFreeIn8<br>Input for binary user signal 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00705**

| Parameter \| Name: **C00705 \| LA_NCtrl_Out: Analog signal list** | Data type: UNSIGNED_16 Index: $23870_d = 5D3E_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00706**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00706 \| LA_NCtrl_Out digital signal list** | Index: $23869_d = 5D3D_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00710**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00710 \| LA_TabPos: Analog connection list** | Index: $23865_d = 5D39_h$ |

Connection parameters for "Table positioning" application: 16-bit inputs
- Selection of the 16 bit output signals to be connected to the 16 bit input signals
- The selection list contains all 16 bit output signals which can be assigned to the 16 bit inputs displayed by the subcodes.

| **Selection list** | | |
|---|---|---|
| See selection list - analog signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00710/1 | 0: Not connected | LA_TabPos: wCanDriveControl<br>Input for control word from CAN to device control |
| C00710/2 | 0: Not connected | LA_TabPos: wMciDriveControl<br>Input for control word from communication interface to device control |
| C00710/3 | 0: Not connected | LA_TabPos: nTorqueMotLim_a<br>Input for maximum torque in motor mode |
| C00710/4 | 0: Not connected | LA_TabPos: nTorqueGenLim_a<br>Input for maximum torque in generator mode |
| C00710/5 | 0: Not connected | LA_TabPos: nMainSetValue_a<br>Input for main speed setpoint |
| C00710/6 | 0: Not connected | LA_TabPos: nAuxSetValue_a<br>Input for additional speed setpoint |
| C00710/7 | 0: Not connected | LA_TabPos: wMckCtrl1<br>Input for MCK control word 1 |
| C00710/8 | 0: Not connected | LA_TabPos: wMckCtrl2<br>Input for MCK control word 2 |
| C00710/9 | 0: Not connected | LA_TabPos: wMckOperationMode<br>Input for selection of MCK operating mode |
| C00710/10 | 0: Not connected | LA_TabPos: wPosProfileMode<br>Input for selection of MCK positioning mode in positioning mode |
| C00710/11 | 0: Not connected | LA_TabPos: wPosProfileNo<br>Input for MCK positioning profile number in positioning mode |
| C00710/12 | 0: Not connected | LA_TabPos: nGPAnalogSwitchIn1_a<br>Input for analog switch - analog signal 1 |
| C00710/13 | 0: Not connected | LA_TabPos: nGPAnalogSwitchIn2_a<br>Input for analog switch - analog signal 2 |
| C00710/14 | 0: Not connected | LA_TabPos: nGPArithmetikIn1_a<br>Input for arithmetic function - analog signal 1 |
| C00710/15 | 0: Not connected | LA_TabPos: nGPArithmetikIn2_a<br>Input for arithmetic function - analog signal 2 |
| C00710/16 | 0: Not connected | LA_TabPos: nGPMulDivIn_a<br>Input for analog signal for multiplication/division |
| C00710/17 | 0: Not connected | LA_TabPos: nGPCompareIn1_a<br>Input for comparison operation - analog signal 1 |

| Parameter \| Name: **C00710 \| LA_TabPos: Analog connection list** | | Data type: UNSIGNED_16 Index: $23865_d = 5D39_h$ |
|---|---|---|
| C00710/18 | 0: Not connected | LA_TabPos: nGPCompareIn2_a<br>Input for comparison operation - analog signal 2 |
| C00710/19 | 0: Not connected | LA_TabPos: wGPCounter1LdVal<br>Input for load value for counter module 1 |
| C00710/20 | 0: Not connected | LA_TabPos: wGPCounter1CmpVal<br>Input for comparison value for counter module 1 |
| C00710/21 | 0: Not connected | LA_TabPos: nSpeedOverride_a<br>Input for speed override |
| C00710/22 | 0: Not connected | LA_TabPos: nAccOverride_a<br>Input for acceleration override |
| C00710/23 | 0: Not connected | LA_TabPos: wFreeIn1<br>Input for user signal 1 |
| C00710/24 | 0: Not connected | LA_TabPos: wFreeIn2<br>Input for user signal 2 |
| C00710/25 | 0: Not connected | LA_TabPos: wFreeIn3<br>Input for user signal 3 |
| C00710/26 | 0: Not connected | LA_TabPos: wFreeIn4<br>Input for user signal 4 |
| C00710/27 | 0: Not connected | LA_TabPos: nPosCtrlOutLimit<br>Input for correcting variable limitation of position controller |
| C00710/28 | 0: Not connected | LA_TabPos: nPosCtrlPAdapt<br>Input for adapting the position controller gain |
| C00710/29 | 0: Not connected | LA_TabPos: wSMCtrl<br>Interface to the optional safety system |
| C00710/30 | 0: Not connected | LA_TabPos: wPosProfileUnitsLW<br>Input for selecting the target position in [units], Low-Word |
| C00710/31 | 0: Not connected | LA_TabPos: wPosProfileUnitsHW<br>Input for selecting the target position in [units], High-Word |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00711**

| Parameter \| Name: **C00711 \| LA_TabPos: Digital connection list** | | Data type: UNSIGNED_16 Index: $23864_d = 5D38_h$ |
|---|---|---|
| Connection parameters for "Table positioning" application: Binary inputs<br>• Selection of the binary output signals to be connected to the binary input signals<br>• The selection list contains all binary output signals which can be assigned to the binary inputs mapped by the subcodes. | | |
| **Selection list** | | |
| See selection list - digital signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00711/1 | 0: Not connected | LA_TabPos: bCInh<br>Control input for setting controller inhibit |
| C00711/2 | 0: Not connected | LA_TabPos: bFailReset<br>Control input for error acknowledgement |
| C00711/3 | 0: Not connected | LA_TabPos: bSetQuickstop<br>Control input for quick stop request |
| C00711/4 | 0: Not connected | LA_TabPos: bSetSpeedCcw<br>Control input for negation of speed direction |

| Parameter \| Name:<br>**C00711 \| LA_TabPos: Digital connection list** | | Data type: UNSIGNED_16<br>Index: $23864_d = 5D38_h$ |
|---|---|---|
| C00711/5 | 0: Not connected | LA_TabPos: bJogSpeed1<br>Control input for fixed speed selection value 1 |
| C00711/6 | 0: Not connected | LA_TabPos: bJogSpeed2<br>Control input for fixed speed selection value 2 |
| C00711/7 | 0: Not connected | LA_TabPos: bMPotEnable<br>Control input for activation of motor potentiometer |
| C00711/8 | 0: Not connected | LA_TabPos: bMPotUp<br>Control input for motor potentiometer ramp-up |
| C00711/9 | 0: Not connected | LA_TabPos: bMPotDown<br>Control input for motor potentiometer ramp-down |
| C00711/10 | 0: Not connected | LA_TabPos: bMBrakeRelease<br>Control input for manual holding brake release request |
| C00711/11 | 0: Not connected | LA_TabPos: bPosCtrlOn<br>Control input for activation of position controller |
| C00711/12 | 0: Not connected | LA_TabPos: bLimitSwitchPos<br>Input for positive hardware limit switch signal |
| C00711/13 | 0: Not connected | LA_TabPos: bLimitSwitchNeg<br>Input for negative hardware limit switch signal |
| C00711/14 | 0: Not connected | LA_TabPos: bReleaseLimitSwitch<br>Control input for hardware limit switch retracting request |
| C00711/15 | 0: Not connected | LA_TabPos: bManJogPos<br>Control input for manual jog in positive direction request |
| C00711/16 | 0: Not connected | LA_TabPos: bManJogNeg<br>Control input for manual jog in negative direction request |
| C00711/17 | 0: Not connected | LA_TabPos: bManEnable2ndSpeed<br>Control input for activation of the second manual speed for manual jog |
| C00711/18 | 0: Not connected | LA_TabPos: bEnableSpeedOverride<br>Control input for activation of speed override |
| C00711/19 | 0: Not connected | LA_TabPos: bEnableAccOverride<br>Control input for activation of acceleration override |
| C00711/20 | 0: Not connected | LA_TabPos: bHomeStartStop<br>Control input for start/stop homing |
| C00711/21 | 0: Not connected | LA_TabPos: bHomeSetPosition<br>Control input for setting the home position |
| C00711/22 | 0: Not connected | LA_TabPos: bHomeResetPosition<br>Control input for reset of "Home position known" |
| C00711/23 | 0: Not connected | LA_TabPos: bHomeMark<br>Input for pre-stop signal for homing |
| C00711/24 | 0: Not connected | LA_TabPos: bPosSetProfilePosition<br>Control input for accepting the profile position in the profile data set |
| C00711/25 | 0: Not connected | LA_TabPos: bPosSetActualPosition<br>Control input for accepting the current position in the profile data set |
| C00711/26 | 0: Not connected | LA_TabPos: bPosExecute<br>Control input for positioning start |
| C00711/27 | 0: Not connected | LA_TabPos: bPosFinishTarget<br>Control input for completion of positioning to target position |

| Parameter \| Name:<br>**C00711 \| LA_TabPos: Digital connection list** | | Data type: UNSIGNED_16<br>Index: $23864_d = 5D38_h$ |
|---|---|---|
| C00711/28 | 0: Not connected | LA_TabPos: bPosDisableFollowProfile<br>Control input for suppression of sequence profile linkage |
| C00711/29 | 0: Not connected | LA_TabPos: bPosStop<br>Control input for interruption of profile generation by ramp-down procedure |
| C00711/30 | 0: Not connected | LA_TabPos: bGPAnalogSwitchSet<br>Control input for analog-value selector change-over |
| C00711/31 | 0: Not connected | LA_TabPos: bGPDigitalDelayIn<br>Input for digital signal with time delay |
| C00711/32 | 0: Not connected | LA_TabPos: bGPLogicIn1<br>Input signal 1 for digital logic |
| C00711/33 | 0: Not connected | LA_TabPos: bGPLogicIn2<br>Input signal 2 for digital logic |
| C00711/34 | 0: Not connected | LA_TabPos: bGPLogicIn3<br>Input signal 3 for digital logic |
| C00711/35 | 0: Not connected | LA_TabPos: bGPDFlipFlop_InD<br>Control input for DFlipFlop setting signal |
| C00711/36 | 0: Not connected | LA_TabPos: bGPDFlipFlop_InClk<br>Control input for DFlipFlop clock signal |
| C00711/37 | 0: Not connected | LA_TabPos: bGPDFlipFlop_InClr<br>Control input for DFlipFlop reset signal |
| C00711/38 | 0: Not connected | LA_TabPos: bGPCounter1ClkUp<br>Control input for up-counting counter module 1 |
| C00711/39 | 0: Not connected | LA_TabPos: bGPCounter1ClkDown<br>Control input for down-counting counter module 1 |
| C00711/40 | 0: Not connected | LA_TabPos: bGPCounter1Load<br>Control input for load value acceptance in counter module 1 |
| C00711/41 | 0: Not connected | LA_TabPos: bMckOperationMode_1<br>Control input for MCK operating mode changeover value 1 |
| C00711/42 | 0: Not connected | LA_TabPos: bMckOperationMode_2<br>Control input for MCK operating mode changeover value 2 |
| C00711/43 | 0: Not connected | LA_TabPos: bMckOperationMode_4<br>Control input for MCK operating mode changeover value 4 |
| C00711/44 | 0: Not connected | LA_TabPos: bMckOperationMode_8<br>Control input for MCK operating mode changeover value 8 |
| C00711/45 | 0: Not connected | LA_TabPos: bPosProfileNo_1<br>Control input for selection of profile number value 1 |
| C00711/46 | 0: Not connected | LA_TabPos: bPosProfileNo_2<br>Control input for selection of profile number value 2 |
| C00711/47 | 0: Not connected | LA_TabPos: bPosProfileNo_4<br>Control input for selection of profile number value 4 |
| C00711/48 | 0: Not connected | LA_TabPos: bPosProfileNo_8<br>Control input for selection of profile number value 8 |
| C00711/49 | 0: Not connected | LA_TabPos: bFreeIn1<br>Input for binary user signal 1 |
| C00711/50 | 0: Not connected | LA_TabPos: bFreeIn2<br>Input for binary user signal 2 |

| Parameter \| Name: C00711 \| LA_TabPos: Digital connection list | | Data type: UNSIGNED_16 Index: 23864$_d$ = 5D38$_h$ |
|---|---|---|
| C00711/51 | 0: Not connected | LA_TabPos: bFreeIn3<br>Input for binary user signal 3 |
| C00711/52 | 0: Not connected | LA_TabPos: bFreeIn4<br>Input for binary user signal 4 |
| C00711/53 | 0: Not connected | LA_TabPos: bFreeIn5<br>Input for binary user signal 5 |
| C00711/54 | 0: Not connected | LA_TabPos: bFreeIn6<br>Input for binary user signal 6 |
| C00711/55 | 0: Not connected | LA_TabPos: bFreeIn7<br>Input for binary user signal 7 |
| C00711/56 | 0: Not connected | LA_TabPos: bFreeIn8<br>Input for binary user signal 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00712**

| Parameter \| Name: C00712 \| LA_TabPos: phi connection list | | Data type: UNSIGNED_16 Index: 23863$_d$ = 5D37$_h$ |
|---|---|---|
| Connection parameters for "Table positioning" application: 32-bit inputs<br>• Selection of the 32-bit output signals for connection with the 32-bit input signals<br>• The selection list contains all 32-bit output signals which can be assigned to the 32-bit inputs mapped by the subcodes. | | |
| **Selection list** | | |
| See selection list - angle signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00712/1 | 0: Not connected | LA_TabPos: dnPosProfilePosition<br>Input for selecting the target position in [increments] |
| C00712/2 | 0: Not connected | LA_TabPos: dnFreeIn1<br>Input for 32-bit user signal 1 |
| C00712/3 | 0: Not connected | LA_TabPos: dnFreeIn2<br>Input for 32-bit user signal 2 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00715**

| Parameter \| Name: C00715 \| LA_TabPos_Out: Analog signal list | Data type: UNSIGNED_16 Index: 23860$_d$ = 5D34$_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00716**

| Parameter \| Name: C00716 \| LA_TabPos_Out: Digital signal list | Data type: UNSIGNED_16 Index: 23859$_d$ = 5D33$_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00717**

| Parameter \| Name: C00717 \| LA_TabPos_Out: phi signal list | Data type: UNSIGNED_16 Index: 23858$_d$ = 5D32$_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00720**

| Parameter \| Name: **C00720 \| L_DigitalDelay_1: Delay** | | | Data type: UNSIGNED_32 Index: $23855_d = 5D2F_h$ |
|---|---|---|---|
| Switch-on/off delay time | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 3600.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00720/1 | 0.000 s | | L_DigitalDelay_1: ON delay |
| C00720/2 | 0.000 s | | L_DigitalDelay_1: OFF delay |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00721**

| Parameter \| Name: **C00721 \| L_DigitalDelay 2..3: Delay** | | | Data type: UNSIGNED_32 Index: $23854_d = 5D2E_h$ |
|---|---|---|---|
| Switch-on/off delay time | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 3600.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00721/1 | 0.000 s | | L_DigitalDelay_2: ON delay |
| C00721/2 | 0.000 s | | L_DigitalDelay_2: OFF delay |
| C00721/3 | 0.000 s | | L_DigitalDelay_3: ON delay |
| C00721/4 | 0.000 s | | L_DigitalDelay_3: OFF delay |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00725**

| Parameter \| Name: **C00725 \| Current switching frequency** | Data type: UNSIGNED_8 Index: $23850_d = 5D2A_h$ |
|---|---|

Display of the current switching frequency
- In C00018 you can choose between a drive-optimised setting for good smooth-running characteristics and an inverter loss-optimised setting (min. Pv).
- Both possibilities offer fixed and variable switching frequencies.
- When a variable switching frequency is selected in C00018, the switching frequency may change as a function of the load and rotational frequency.

| **Selection list** (read only) | |
|---|---|
| 1 | 4 kHz var./drive-optimised |
| 2 | 8 kHz var./drive-optimised |
| 3 | 16 kHz var./drive-optimised |
| 5 | 2 kHz constant/drive-optimised |
| 6 | 4 kHz constant/drive-optimised |
| 7 | 8 kHz constant/drive-optimised |
| 8 | 16 kHz constant/drive-optimised |
| 11 | 4 kHz var./min. Pv |
| 12 | 8 kHz var./min. Pv |
| 13 | 16 kHz var./min. Pv |
| 14 | Reserved |
| 15 | 2 kHz constant/min. Pv |
| 16 | 4 kHz constant/min. Pv |
| 17 | 8 kHz constant/min. Pv |
| 18 | 16 kHz constant/min. Pv |
| 21 | 8 kHz var./drive-opt./4 kHz min |
| 22 | 16 kHz var./drive-opt./4 kHz min |
| 23 | 16 kHz var./drive-opt./8 kHz min |
| 31 | 8 kHz var./min. Pv/4 kHz min |
| 32 | 16 kHz var./min. Pv/4 kHz min |
| 33 | 16 kHz var./min. Pv/8 kHz min |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00726**

| Parameter \| Name: **C00726 \| Current limit values** | Data type: UNSIGNED_8 Index: $23849_d = 5D29_h$ |
|---|---|

**This code is for device-internal use only and must not be written to by the user!**

**C00727**

| Parameter \| Name: C00727 \| LS_Keypad digital values | | | Data type: UNSIGNED_8 Index: $23848_d = 5D28_h$ |
|---|---|---|---|
| Execution of control commands for keypad operation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 1 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00727/1 | 0 | | 1 ≡ request quick stop |
| C00727/2 | 0 | | 1 ≡ request DC-injection braking |
| C00727/3 | 0 | | 1 ≡ request change of direction of rotation |
| C00727/4 | 0 | | 1 ≡ request fixed speed setpoint 1 |
| C00727/5 | 0 | | 1 ≡ request fixed speed setpoint 2 |
| C00727/6 | 0 | | 1 ≡ motor potentiometer: request activation |
| C00727/7 | 0 | | 1 ≡ motor potentiometer: request pos. acceleration |
| C00727/8 | 0 | | 1 ≡ motor potentiometer: request neg. acceleration |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00728**

| Parameter \| Name: C00728 \| Keypad analog values | | | Data type: INTEGER_16 Index: $23847_d = 5D27_h$ |
|---|---|---|---|
| Selection of different setpoints when operating via keypad | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00728/1 | 100.00 % | | Torque limit in motor mode |
| C00728/2 | 100.00 % | | Torque limit in generator mode |
| C00728/3 | 0.00 % | | Setpoint speed<br>• 100 % ≡ C00011 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00729**

| Parameter \| Name: C00729 \| Remote: Setpoint selection | | | Data type: INTEGER_16 Index: $23846_d = 5D26_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00729/1 | 0.00 | | Remote: Setpoint keypad |
| C00729/2 | 0.00 | | Remote: Setpoint PC |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00730**

| Parameter \| Name: C00730 \| Oscilloscope scanning interval | Data type: UNSIGNED_32 Index: $23845_d = 5D25_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00731**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00731 \| Oscilloscope recording length** | Index: $23844_d = 5D24_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00732**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C00732 \| Oscilloscope command** | Index: $23843_d = 5D23_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00734**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00734 \| Oscilloscope trigger channel selection** | Index: $23841_d = 5D21_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00735**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C00735 \| Oscilloscope channel source type** | Index: $23840_d = 5D20_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00736**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00736 \| Oscilloscope data type/data width** | Index: $23839_d = 5D1F_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00740**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00740 \| Oscilloscope offset variables** | Index: $23835_d = 5D1B_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00741**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C00741 \| Oscilloscope trigger mode** | Index: $23834_d = 5D1A_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00742**

| Parameter \| Name: | Data type: INTEGER_16 |
|---|---|
| **C00742 \| Oscilloscope trigger delay** | Index: $23833_d = 5D19_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00743**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00743 \| Oscilloscope trigger level** | Index: $23832_d = 5D18_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00744**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00744 \| Oscilloscope trigger mask** | Index: $23831_d = 5D17_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00746**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00746 \| Oscilloscope trigger counter** | Index: $23829_d = 5D15_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00747**

| Parameter \| Name: | Data type: UNSIGNED_16 |
|---|---|
| **C00747 \| Oscilloscope status word** | Index: $23828_d = 5D14_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00748**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00748 \| Oscilloscope no. of measured values** | Index: $23827_d = 5D13_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00749**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00749 \| Oscilloscope recording** | Index: $23826_d = 5D12_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00750**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C00750 \| Select. of MCTRL oscillos. channels** | Index: $23825_d = 5D11_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00751**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C00751 \| Oscilloscope data memory** | Index: $23824_d = 5D10_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00753**

| Parameter \| Name: | Data type: OCTET_STRING |
|---|---|
| **C00753 \| Oscilloscope data memory octet string** | Index: $23822_d = 5D0E_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00760**

| Parameter \| Name: **C00760 \| LA_SwitchPos: Analog connection list** | | Data type: UNSIGNED_16 Index: $23815_d = 5D07_h$ |
|---|---|---|
| Connection parameters for "Switch-off positioning" application: 16-bi inputs<br>• Selection of the 16 bit output signals to be connected to the 16 bit input signals<br>• The selection list contains all 16 bit output signals which can be assigned to the 16 bit inputs displayed by the subcodes. | | |
| **Selection list** | | |
| See selection list - analog signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00760/1 | 0: Not connected | LA_SwitchPos: wCANDriveControl<br>Input for control word from CAN to device control |
| C00760/2 | 0: Not connected | LA_SwitchPos: wMCIDriveControl<br>Input for control word from communication interface to device control |
| C00760/3 | 0: Not connected | LA_SwitchPos: nVoltageAdd_a<br>Input for additive voltage impression |
| C00760/4 | 0: Not connected | LA_SwitchPos: nBoost_a<br>Input for additional setpoint for motor voltage at speed = 0 |
| C00760/5 | 0: Not connected | LA_SwitchPos: nPWMAngleOffset<br>Input for additional offset for the electrical angle of rotation |
| C00760/6 | 0: Not connected | LA_SwitchPos: nTorqueMotLim_a<br>Input for maximum torque in motor mode |
| C00760/7 | 0: Not connected | LA_SwitchPos: nTorqueGenLim_a<br>Input for maximum torque in generator mode |
| C00760/8 | 0: Not connected | LA_SwitchPos: nMainSetValue_a<br>Input for main speed setpoint |
| C00760/9 | 0: Not connected | LA_SwitchPos: nAuxSetValue_a<br>Input for additional speed setpoint |
| C00760/10 | 0: Not connected | LA_SwitchPos: nGPAnalogSwitchIn1_a<br>Input for analog switch - analog signal 1 |
| C00760/11 | 0: Not connected | LA_SwitchPos: nGPAnalogSwitchIn2_a<br>Input for analog switch - analog signal 2 |
| C00760/12 | 0: Not connected | LA_SwitchPos: nGPArithmetikIn1_a<br>Input for arithmetic function - analog signal 1 |
| C00760/13 | 0: Not connected | LA_SwitchPos: nGPArithmetikIn2_a<br>Input for arithmetic function - analog signal 2 |
| C00760/14 | 0: Not connected | LA_SwitchPos: nGPMulDivIn_a<br>Input for analog signal for multiplication/division |
| C00760/15 | 0: Not connected | LA_SwitchPos: nGPCompareIn1_a<br>Input for comparison operation - analog signal 1 |
| C00760/16 | 0: Not connected | LA_SwitchPos: nGPCompareIn2_a<br>Input for comparison operation - analog signal 2 |
| C00760/17 | 0: Not connected | LA_SwitchPos: wSMCtrl<br>Interface to the optional safety system |
| C00760/18 | 0: Not connected | Reserved |
| C00760/19 | 0: Not connected | Reserved |
| C00760/20 | 0: Not connected | Reserved |
| C00760/21 | 0: Not connected | Reserved |

| Parameter \| Name: **C00760 \| LA_SwitchPos: Analog connection list** | | Data type: UNSIGNED_16 Index: $23815_d = 5D07_h$ |
|---|---|---|
| C00760/22 | 0: Not connected | LA_SwitchPos: wFreeIn1<br>Input for user signal 1 |
| C00760/23 | 0: Not connected | LA_SwitchPos: wFreeIn2<br>Input for user signal 2 |
| C00760/24 | 0: Not connected | LA_SwitchPos: wFreeIn3<br>Input for user signal 3 |
| C00760/25 | 0: Not connected | LA_SwitchPos: wFreeIn4<br>Input for user signal 4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00761**

| Parameter \| Name: **C00761 \| LA_SwitchPos: Digital connection list** | | Data type: UNSIGNED_16 Index: $23814_d = 5D06_h$ |
|---|---|---|
| Connection parameters for "Switch-off positioning" application: Binary inputs<br>• Selection of the binary output signals to be connected to the binary input signals<br>• The selection list contains all binary output signals which can be assigned to the binary inputs mapped by the subcodes. | | |
| **Selection list** | | |
| See selection list - digital signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00761/1 | 0: Not connected | LA_SwitchPos: bCInh<br>Control input for setting controller inhibit |
| C00761/2 | 0: Not connected | LA_SwitchPos: bFailReset<br>Control input for error acknowledgement |
| C00761/3 | 0: Not connected | LA_SwitchPos: bSetQuickstop<br>Control input for quick stop request |
| C00761/4 | 0: Not connected | LA_SwitchPos: bSetDCBrake<br>Control input for DC-injection braking request |
| C00761/5 | 0: Not connected | LA_SwitchPos: bRFG_Stop<br>Control input for stopping the speed ramp function generator |
| C00761/6 | 0: Not connected | LA_SwitchPos: bSetSpeedCcw<br>Control input for change of direction of rotation |
| C00761/7 | 0: Not connected | LA_SwitchPos: bRLQCw<br>Control input for activation of CW rotation (fail-safe) |
| C00761/8 | 0: Not connected | LA_SwitchPos: bRLQCcw<br>Control input for activation of CCW rotation (fail-safe) |
| C00761/9 | 0: Not connected | LA_SwitchPos: bJogCtrlInputSel1<br>Selection input 1 for binary coded selection of the switch-off position 1 ... 3 |
| C00761/10 | 0: Not connected | LA_SwitchPos: bJogCtrlInputSel2<br>Selection input 2 for binary coded selection of the switch-off position 1 ... 3 |
| C00761/11 | 0: Not connected | LA_SwitchPos: bJogCtrlRfgIn<br>Control input for setpoint generator ramp-down |
| C00761/12 | 0: Not connected | LA_SwitchPos: bJogCtrlJog1<br>Selection input 1 for overriding fixed setpoints (JOG setpoints) for the main setpoint |
| C00761/13 | 0: Not connected | LA_SwitchPos: bJogCtrlJog2<br>Selection input 2 for overriding fixed setpoints (JOG setpoints) for the main setpoint |

| Parameter \| Name:<br>**C00761 \| LA_SwitchPos: Digital connection list** | | Data type: UNSIGNED_16<br>Index: $23814_d = 5D06_h$ |
|---|---|---|
| C00761/14 | 0: Not connected | LA_SwitchPos: bJogCtrlSlowDown1<br>Control input for selection of pre-switch off 1 |
| C00761/15 | 0: Not connected | LA_SwitchPos: bJogCtrlStop1<br>Control input for stop function 1 |
| C00761/16 | 0: Not connected | LA_SwitchPos: bJogCtrlSlowDown2<br>Control input for selection of pre-switch off 2 |
| C00761/17 | 0: Not connected | LA_SwitchPos: bJogCtrlStop2<br>Control input for stop function 2 |
| C00761/18 | 0: Not connected | LA_SwitchPos: bJogCtrlSlowDown3<br>Control input for selection of pre-switch off 3 |
| C00761/19 | 0: Not connected | LA_SwitchPos: bJogCtrlStop3<br>Control input for stop function 3 |
| C00761/20 | 0: Not connected | LA_SwitchPos: bJogSpeed4<br>Selection input for fixed setpoints |
| C00761/21 | 0: Not connected | LA_SwitchPos: bJogSpeed8<br>Selection input for fixed setpoints |
| C00761/22 | 0: Not connected | LA_SwitchPos: bJogRamp1<br>Selection input for additional acceleration/deceleration times |
| C00761/23 | 0: Not connected | LA_SwitchPos: bJogRamp2<br>Selection input for additional acceleration/deceleration times |
| C00761/24 | 0: Not connected | LA_SwitchPos: bJogRamp4<br>Selection input for additional acceleration/deceleration times |
| C00761/25 | 0: Not connected | LA_SwitchPos: bJogRamp8<br>Selection input for additional acceleration/deceleration times |
| C00761/26 | 0: Not connected | LA_SwitchPos: bMBrkRelease<br>Control input for manual holding brake release request |
| C00761/27 | 0: Not connected | LA_SwitchPos: bGPAnalogSwitchSet<br>Control input for analog-value selector change-over |
| C00761/28 | 0: Not connected | LA_SwitchPos: bGPDigitalDelayIn<br>Input for digital signal with time delay |
| C00761/29 | 0: Not connected | LA_SwitchPos: bGPLogicIn1<br>Input signal 1 for digital logic |
| C00761/30 | 0: Not connected | LA_SwitchPos: bGPLogicIn2<br>Input signal 2 for digital logic |
| C00761/31 | 0: Not connected | LA_SwitchPos: bGPLogicIn3<br>Input signal 3 for digital logic |
| C00761/32 | 0: Not connected | LA_SwitchPos: bGPDFlipFlop_InD<br>Control input for DFlipFlop setting signal |
| C00761/33 | 0: Not connected | LA_SwitchPos: bGPDFlipFlop_InClk<br>Control input for DFlipFlop clock signal |
| C00761/34 | 0: Not connected | LA_SwitchPos: bGPDFlipFlop_InClr<br>Control input for DFlipFlop reset signal |
| C00761/35 | 0: Not connected | Reserved |
| C00761/36 | 0: Not connected | Reserved |
| C00761/37 | 0: Not connected | Reserved |
| C00761/38 | 0: Not connected | Reserved |
| C00761/39 | 0: Not connected | Reserved |

| Parameter \| Name: **C00761 \| LA_SwitchPos: Digital connection list** | | Data type: UNSIGNED_16 Index: $23814_d = 5D06_h$ |
|---|---|---|
| C00761/40 | 0: Not connected | LA_SwitchPos: bFreeIn1<br>Input for binary user signal 1 |
| C00761/41 | 0: Not connected | LA_SwitchPos: bFreeIn2<br>Input for binary user signal 2 |
| C00761/42 | 0: Not connected | LA_SwitchPos: bFreeIn3<br>Input for binary user signal 3 |
| C00761/43 | 0: Not connected | LA_SwitchPos: bFreeIn4<br>Input for binary user signal 4 |
| C00761/44 | 0: Not connected | LA_SwitchPos: bFreeIn5<br>Input for binary user signal 5 |
| C00761/45 | 0: Not connected | LA_SwitchPos: bFreeIn6<br>Input for binary user signal 6 |
| C00761/46 | 0: Not connected | LA_SwitchPos: bFreeIn7<br>Input for binary user signal 7 |
| C00761/47 | 0: Not connected | LA_SwitchPos: bFreeIn8<br>Input for binary user signal 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00762**

| Parameter \| Name: **C00762 \| LA_SwitchPos: phi connection list** | Data type: UNSIGNED_16 Index: $23813_d = 5D05_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00765**

| Parameter \| Name: **C00765 \| LA_SwitchPos_Out: Analog signal list** | Data type: UNSIGNED_16 Index: $23810_d = 5D02_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00766**

| Parameter \| Name: **C00766 \| LA_SwitchPos_Out: Digital signal list** | Data type: UNSIGNED_16 Index: $23809_d = 5D01_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00767**

| Parameter \| Name: **C00767 \| LA_SwitchPos_Out: phi signal list** | Data type: UNSIGNED_16 Index: $23808_d = 5D00_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00800**

| Parameter \| Name: **C00800 \| L_MPot_1: Upper limit** | | | Data type: INTEGER_16 Index: $23775_d = 5CDF_h$ |
|---|---|---|---|
| The L_MPot_1 FB: Upper limit of the motor potentiometer function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **100.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00801**

| Parameter \| Name: **C00801 \| L_MPot_1: Lower limit** | | | Data type: INTEGER_16 Index: $23774_d = 5CDE_h$ |
|---|---|---|---|
| The L_MPot_1 FB: Lower limit of the motor potentiometer function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **-100.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00802**

| Parameter \| Name: **C00802 \| L_MPot_1: Acceleration time** | | | Data type: UNSIGNED_16 Index: $23773_d = 5CDD_h$ |
|---|---|---|---|
| The L_MPot_1 FB: Acceleration time of the motor potentiometer function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.1 | s | 6000.0 | **10.0 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C00803**

| Parameter \| Name: **C00803 \| L_MPot_1: Deceleration time** | | | Data type: UNSIGNED_16 Index: $23772_d = 5CDC_h$ |
|---|---|---|---|
| The L_MPot_1 FB: Deceleration time of the motor potentiometer function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.1 | s | 6000.0 | **10.0 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C00804**

| Parameter \| Name: **C00804 \| L_MPot_1: Inactive fct.** | | Data type: UNSIGNED_8 Index: $23771_d = 5CDB_h$ |
|---|---|---|
| The L_MPot_1 FB: Selection of the response if the motor potentiometer is deactivated via input *bInAct* | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Retain value** | Keep output value |
| 1 | Deceleration to 0 | Deceleration via ramp to 0 |
| 2 | Deceleration to lower limit | Deceleration via ramp to the lower limit (C00801) |
| 3 | Without ramp to 0 | Step change to 0 |
| 4 | Without ramp to lower limit | Jump to lower limit (C00800) |
| 5 | Acceleration to upper limit | Acceleration via ramp to upper limit (C00800) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00805**

| Parameter \| Name: **C00805 \| L_MPot_1: Init fct.** | | Data type: UNSIGNED_8 Index: $23770_d = 5CDA_h$ |
|---|---|---|
| The L_MPot_1 FB: Selection of the response at device switch-on | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **Load last value** | |
| 1 | Load lower limit | |
| 2 | Load 0 | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00806**

| Parameter \| Name: **C00806 \| L_MPot_1: Use** | | Data type: UNSIGNED_8 Index: $23769_d = 5CD9_h$ |
|---|---|---|
| The L_MPot_1 FB: Use of the motor potentiometer | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **No** | The motor potentiometer is not used.<br>• The analog value applied to the *nIn_a* input is looped through without any changes to the *nOut_a* output. |
| 1 | Yes | The motor potentiometer is used.<br>• The analog value applied at the *nIn_a* input is led via the motor potentiometer and provided at the *nOut_a* output. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00807**

| Parameter \| Name: **C00807 \| L_NLim_1: Max.SkipFrq.** | | | Data type: INTEGER_16 Index: $23768_d = 5CD8_h$ |
|---|---|---|---|
| FB L_NLim_1: Maximum blocking frequencies | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00807/1 | 0.00 % | | Maximum blocking frequency for zone 1 |
| C00807/2 | 0.00 % | | Maximum blocking frequency for zone 2 |
| C00807/3 | 0.00 % | | Maximum blocking frequency for zone 3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00808**

| Parameter \| Name: **C00808 \| L_NLim_1: Min.SkipFrq.** | | | Data type: INTEGER_16 Index: $23767_d = 5CD7_h$ |
|---|---|---|---|
| FB L_NLim_1: Minimum skip frequencies | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00808/1 | 0.00 % | | Minimum blocking frequency for zone 1 |
| C00808/2 | 0.00 % | | Minimum blocking frequency for zone 2 |
| C00808/3 | 0.00 % | | Minimum blocking frequency for zone 2 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00809**

| Parameter \| Name: **C00809 \| L_NLim_2: Max.SkipFrq.** | | | Data type: INTEGER_16 Index: $23766_d = 5CD6_h$ |
|---|---|---|---|
| FB L_NLim_2: Maximum skip frequencies | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00809/1 | 0.00 % | | Maximum blocking frequency for zone 1 |
| C00809/2 | 0.00 % | | Maximum blocking frequency for zone 2 |
| C00809/3 | 0.00 % | | Maximum blocking frequency for zone 3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00810**

| Parameter \| Name: **C00810 \| L_NLim_2: Min.SkipFrq.** | | | Data type: INTEGER_16 Index: $23765_d = 5CD5_h$ |
|---|---|---|---|
| FB L_NLim_2: Minimum skip frequencies | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00810/1 | 0.00 % | | Minimum blocking frequency for zone 1 |
| C00810/2 | 0.00 % | | Minimum blocking frequency for zone 2 |
| C00810/3 | 0.00 % | | Minimum blocking frequency for zone 3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00811**

| Parameter \| Name: **C00811 \| L_NLim_1: Current output value** | | | Data type: INTEGER_16 Index: $23764_d = 5CD4_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00811/1 | | | L_NLim_1: Current output value |
| C00811/2 | | | L_NLim_2: Current output value |
| ☐ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00812**

| Parameter \| Name: **C00812 \| L_NLim: Current status** | | | Data type: UNSIGNED_16 Index: $23763_d = 5CD3_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | No blocking zone active | | |
| Bit 1 | Blocking zone 1 active | | |
| Bit 2 | Blocking zone 2 active | | |
| Bit 3 | Blocking zone 3 active | | |
| Bit 4 | Value in blocking zone | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | | | **Info** |
| C00812/1 | | | L_NLim_1: Current status |
| C00812/2 | | | L_NLim_2: Current status |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00820**

| Parameter \| Name: **C00820 \| L_DigitalLogic_1: Function** | | Data type: UNSIGNED_8 Index: $23755_d = 5CCB_h$ |
|---|---|---|
| The L_DigitalLogic_1 FB: Selection of the internal logic function | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **bOut = 0** | Constant value "FALSE" |
| 1 | bOut = 1 | Constant value "TRUE" |
| 2 | bOut = bIn1 AND bIn2 AND bIn3 | AND operation |
| 3 | bOut = bIn1 OR bIn2 OR bIn3 | OR operation |
| 4 | bOut = f (truth table) | The truth table parameterised in C00821 is used. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00821**

| Parameter \| Name: **C00821 \| L_DigitalLogic_1: Truth table** | | Data type: UNSIGNED_8 Index: $23754_d = 5CCA_h$ |
|---|---|---|
| The L_DigitalLogic_1 FB: Parameterisation of the truth table | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00821/1 | 0: FALSE | L_DigitalLogic_1: bOut: bIn3...bIn1=0 0 0 |
| C00821/2 | 0: FALSE | L_DigitalLogic_1: bOut: bIn3...bIn1=0 0 1 |
| C00821/3 | 0: FALSE | L_DigitalLogic_1: bOut: bIn3...bIn1=0 1 0 |
| C00821/4 | 0: FALSE | L_DigitalLogic_1: bOut: bIn3...bIn1=0 1 1 |
| C00821/5 | 0: FALSE | L_DigitalLogic_1: bOut: bIn3...bIn1=1 0 0 |
| C00821/6 | 0: FALSE | L_DigitalLogic_1: bOut: bIn3...bIn1=1 0 1 |
| C00821/7 | 0: FALSE | L_DigitalLogic_1: bOut: bIn3...bIn1=1 1 0 |
| C00821/8 | 0: FALSE | L_DigitalLogic_1: bOut: bIn3...bIn1=1 1 1 |
| C00821/9 | 0: FALSE | L_DigitalLogic_1: bOut2: bIn3...bIn1=0 0 0 |
| C00821/10 | 0: FALSE | L_DigitalLogic_1: bOut2: bIn3...bIn1=0 0 1 |
| C00821/11 | 0: FALSE | L_DigitalLogic_1: bOut2: bIn3...bIn1=0 1 0 |
| C00821/12 | 0: FALSE | L_DigitalLogic_1: bOut2: bIn3...bIn1=0 1 1 |
| C00821/13 | 0: FALSE | L_DigitalLogic_1: bOut2: bIn3...bIn1=1 0 0 |
| C00821/14 | 0: FALSE | L_DigitalLogic_1: bOut2: bIn3...bIn1=1 0 1 |
| C00821/15 | 0: FALSE | L_DigitalLogic_1: bOut2: bIn3...bIn1=1 1 0 |
| C00821/16 | 0: FALSE | L_DigitalLogic_1: bOut2: bIn3...bIn1=1 1 1 |
| C00821/17 | 0: FALSE | L_DigitalLogic_1: bOut3: bIn3...bIn1=0 0 0 |
| C00821/18 | 0: FALSE | L_DigitalLogic_1: bOut3: bIn3...bIn1=0 0 1 |
| C00821/19 | 0: FALSE | L_DigitalLogic_1: bOut3: bIn3...bIn1=0 1 0 |
| C00821/20 | 0: FALSE | L_DigitalLogic_1: bOut3: bIn3...bIn1=0 1 1 |
| C00821/21 | 0: FALSE | L_DigitalLogic_1: bOut3: bIn3...bIn1=1 0 0 |
| C00821/22 | 0: FALSE | L_DigitalLogic_1: bOut3: bIn3...bIn1=1 0 1 |
| C00821/23 | 0: FALSE | L_DigitalLogic_1: bOut3: bIn3...bIn1=1 1 0 |
| C00821/24 | 0: FALSE | L_DigitalLogic_1: bOut3: bIn3...bIn1=1 1 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00822**

| Parameter \| Name: **C00822 \| L_DigitalLogic_2: Function** | | Data type: UNSIGNED_8 Index: $23753_d = 5CC9_h$ |
|---|---|---|
| The L_DigitalLogic_2 FB: Selection of the internal logic function | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **bOut = 0** | Constant value "FALSE" |
| 1 | bOut = 1 | Constant value "TRUE" |
| 2 | bOut = bIn1 AND ... bIn3 | AND operation |
| 3 | bOut = bIn1 OR ... bIn3 | OR operation |
| 4 | bOut = f (truth table) | The truth table parameterised in C00823 is used. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00823**

| Parameter \| Name: **C00823 \| L_DigitalLogic_2: Truth table** | | Data type: UNSIGNED_8 Index: $23752_d = 5CC8_h$ |
|---|---|---|
| The L_DigitalLogic_2 FB: Parameterisation of the truth table | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00823/1 | 0: FALSE | L_DigitalLogic_2: bOut: bIn3...bIn1=0 0 0 |
| C00823/2 | 0: FALSE | L_DigitalLogic_2: bOut: bIn3...bIn1=0 0 1 |
| C00823/3 | 0: FALSE | L_DigitalLogic_2: bOut: bIn3...bIn1=0 1 0 |
| C00823/4 | 0: FALSE | L_DigitalLogic_2: bOut: bIn3...bIn1=0 1 1 |
| C00823/5 | 0: FALSE | L_DigitalLogic_2: bOut: bIn3...bIn1=1 0 0 |
| C00823/6 | 0: FALSE | L_DigitalLogic_2: bOut: bIn3...bIn1=1 0 1 |
| C00823/7 | 0: FALSE | L_DigitalLogic_2: bOut: bIn3...bIn1=1 1 0 |
| C00823/8 | 0: FALSE | L_DigitalLogic_2: bOut: bIn3...bIn1=1 1 1 |
| C00823/9 | 0: FALSE | L_DigitalLogic_2: bOut2: bIn3...bIn1=0 0 0 |
| C00823/10 | 0: FALSE | L_DigitalLogic_2: bOut2: bIn3...bIn1=0 0 1 |
| C00823/11 | 0: FALSE | L_DigitalLogic_2: bOut2: bIn3...bIn1=0 1 0 |
| C00823/12 | 0: FALSE | L_DigitalLogic_2: bOut2: bIn3...bIn1=0 1 1 |
| C00823/13 | 0: FALSE | L_DigitalLogic_2: bOut2: bIn3...bIn1=1 0 0 |
| C00823/14 | 0: FALSE | L_DigitalLogic_2: bOut2: bIn3...bIn1=1 0 1 |
| C00823/15 | 0: FALSE | L_DigitalLogic_2: bOut2: bIn3...bIn1=1 1 0 |
| C00823/16 | 0: FALSE | L_DigitalLogic_2: bOut2: bIn3...bIn1=1 1 1 |
| C00823/17 | 0: FALSE | L_DigitalLogic_2: bOut3: bIn3...bIn1=0 0 0 |
| C00823/18 | 0: FALSE | L_DigitalLogic_2: bOut3: bIn3...bIn1=0 0 1 |
| C00823/19 | 0: FALSE | L_DigitalLogic_2: bOut3: bIn3...bIn1=0 1 0 |
| C00823/20 | 0: FALSE | L_DigitalLogic_2: bOut3: bIn3...bIn1=0 1 1 |
| C00823/21 | 0: FALSE | L_DigitalLogic_2: bOut3: bIn3...bIn1=1 0 0 |
| C00823/22 | 0: FALSE | L_DigitalLogic_2: bOut3: bIn3...bIn1=1 0 1 |
| C00823/23 | 0: FALSE | L_DigitalLogic_2: bOut3: bIn3...bIn1=1 1 0 |
| C00823/24 | 0: FALSE | L_DigitalLogic_2: bOut3: bIn3...bIn1=1 1 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00824**

| Parameter \| Name: **C00824 \| L_DigitalLogic5_1: Function** | | Data type: UNSIGNED_8 Index: $23751_d = 5CC7_h$ |
|---|---|---|
| FB L_DigitalLogic5_1: Selection of the interna logic operation | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **bOut = 0** | |
| 1 | bOut = 1 | |
| 2 | bOut = f (truth table) | The truth table parameterised in C00825 is used. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00825**

| Parameter \| Name: **C00825 \| L_DigitalLogic5_1: Truth table** | | Data type: UNSIGNED_8 Index: $23750_d = 5CC6_h$ |
|---|---|---|
| FB L_DigitalLogic5_1: Parameter setting of the truth table | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00825/1 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 0 0 0 |
| C00825/2 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 0 0 1 |
| C00825/3 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 0 1 0 |
| C00825/4 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 0 1 1 |
| C00825/5 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 1 0 0 |
| C00825/6 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 1 0 1 |
| C00825/7 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 1 1 0 |
| C00825/8 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 1 1 1 |
| C00825/9 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 0 0 0 |
| C00825/10 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 0 0 1 |
| C00825/11 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 0 1 0 |
| C00825/12 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 0 1 1 |
| C00825/13 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 1 0 0 |
| C00825/14 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 1 0 1 |
| C00825/15 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 1 1 0 |
| C00825/16 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 1 1 1 |
| C00825/17 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 0 0 0 |
| C00825/18 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 0 0 1 |
| C00825/19 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 0 1 0 |
| C00825/20 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 0 1 1 |
| C00825/21 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 1 0 0 |
| C00825/22 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 1 0 1 |
| C00825/23 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 1 1 0 |
| C00825/24 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 1 1 1 |
| C00825/25 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 0 0 0 |
| C00825/26 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 0 0 1 |
| C00825/27 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 0 1 0 |
| C00825/28 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 0 1 1 |
| C00825/29 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 1 0 0 |
| C00825/30 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 1 0 1 |
| C00825/31 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 1 1 0 |
| C00825/32 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 1 1 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00826**

| Parameter \| Name: **C00826 \| L_DigitalLogic5_2: Function** | | Data type: UNSIGNED_8 Index: $23749_d = 5CC5_h$ |
|---|---|---|
| FB L_DigitalLogic5_2: Selection of the interna logic operation | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **bOut = 0** | |
| 1 | bOut = 1 | |
| 2 | bOut = f (truth table) | The truth table parameterised in C00827 is used. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00827**

| Parameter \| Name: **C00827 \| L_DigitalLogic5_2: Truth table** | | Data type: UNSIGNED_8 Index: $23748_d = 5CC4_h$ |
|---|---|---|
| FB L_DigitalLogic5_2: Parameter setting of the truth table | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00827/1 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 0 0 0 |
| C00827/2 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 0 0 1 |
| C00827/3 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 0 1 0 |
| C00827/4 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 0 1 1 |
| C00827/5 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 1 0 0 |
| C00827/6 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 1 0 1 |
| C00827/7 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 1 1 0 |
| C00827/8 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 1 1 1 |
| C00827/9 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 0 0 0 |
| C00827/10 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 0 0 1 |
| C00827/11 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 0 1 0 |
| C00827/12 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 0 1 1 |
| C00827/13 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 1 0 0 |
| C00827/14 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 1 0 1 |
| C00827/15 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 1 1 0 |
| C00827/16 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 1 1 1 |
| C00827/17 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 0 0 0 |
| C00827/18 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 0 0 1 |
| C00827/19 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 0 1 0 |
| C00827/20 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 0 1 1 |
| C00827/21 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 1 0 0 |
| C00827/22 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 1 0 1 |
| C00827/23 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 1 1 0 |
| C00827/24 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 1 1 1 |
| C00827/25 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 0 0 0 |
| C00827/26 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 0 0 1 |
| C00827/27 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 0 1 0 |
| C00827/28 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 0 1 1 |

| Parameter \| Name: **C00827 \| L_DigitalLogic5_2: Truth table** | | Data type: UNSIGNED_8 Index: $23748_d = 5CC4_h$ |
|---|---|---|
| C00827/29 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 1 0 0 |
| C00827/30 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 1 0 1 |
| C00827/31 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 1 1 0 |
| C00827/32 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 1 1 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00828**

| Parameter \| Name: **C00828 \| L_DigitalLogic_3: Function** | | Data type: UNSIGNED_8 Index: $23747_d = 5CC3_h$ |
|---|---|---|
| From version 02.00.00<br>The L_DigitalLogic_3 FB: Selection of the internal logic function | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **bOut = 0** | Constant value "FALSE" |
| 1 | bOut = 1 | Constant value "TRUE" |
| 2 | bOut = bIn1 AND bIn2 AND bIn3 | AND operation |
| 3 | bOut = bIn1 OR bIn2 OR bIn3 | OR operation |
| 4 | bOut = f (truth table) | The truth table parameterised in C00829 is used. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00829**

| Parameter \| Name: **C00829 \| L_DigitalLogic_3: Truth table** | | Data type: UNSIGNED_8 Index: $23746_d = 5CC2_h$ |
|---|---|---|
| From version 02.00.00<br>The L_DigitalLogic_3 FB: Parameterisation of the truth table | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00829/1 | 0: FALSE | L_DigitalLogic_3: bOut: bIn3...bIn1=0 0 0 |
| C00829/2 | 0: FALSE | L_DigitalLogic_3: bOut: bIn3...bIn1=0 0 1 |
| C00829/3 | 0: FALSE | L_DigitalLogic_3: bOut: bIn3...bIn1=0 1 0 |
| C00829/4 | 0: FALSE | L_DigitalLogic_3: bOut: bIn3...bIn1=0 1 1 |
| C00829/5 | 0: FALSE | L_DigitalLogic_3: bOut: bIn3...bIn1=1 0 0 |
| C00829/6 | 0: FALSE | L_DigitalLogic_3: bOut: bIn3...bIn1=1 0 1 |
| C00829/7 | 0: FALSE | L_DigitalLogic_3: bOut: bIn3...bIn1=1 1 0 |
| C00829/8 | 0: FALSE | L_DigitalLogic_3: bOut: bIn3...bIn1=1 1 1 |
| C00829/9 | 0: FALSE | L_DigitalLogic_3: bOut2: bIn3...bIn1=0 0 0 |
| C00829/10 | 0: FALSE | L_DigitalLogic_3: bOut2: bIn3...bIn1=0 0 1 |
| C00829/11 | 0: FALSE | L_DigitalLogic_3: bOut2: bIn3...bIn1=0 1 0 |
| C00829/12 | 0: FALSE | L_DigitalLogic_3: bOut2: bIn3...bIn1=0 1 1 |
| C00829/13 | 0: FALSE | L_DigitalLogic_3: bOut2: bIn3...bIn1=1 0 0 |
| C00829/14 | 0: FALSE | L_DigitalLogic_3: bOut2: bIn3...bIn1=1 0 1 |
| C00829/15 | 0: FALSE | L_DigitalLogic_3: bOut2: bIn3...bIn1=1 1 0 |
| C00829/16 | 0: FALSE | L_DigitalLogic_3: bOut2: bIn3...bIn1=1 1 1 |
| C00829/17 | 0: FALSE | L_DigitalLogic_3: bOut3: bIn3...bIn1=0 0 0 |
| C00829/18 | 0: FALSE | L_DigitalLogic_3: bOut3: bIn3...bIn1=0 0 1 |
| C00829/19 | 0: FALSE | L_DigitalLogic_3: bOut3: bIn3...bIn1=0 1 0 |
| C00829/20 | 0: FALSE | L_DigitalLogic_3: bOut3: bIn3...bIn1=0 1 1 |
| C00829/21 | 0: FALSE | L_DigitalLogic_3: bOut3: bIn3...bIn1=1 0 0 |
| C00829/22 | 0: FALSE | L_DigitalLogic_3: bOut3: bIn3...bIn1=1 0 1 |
| C00829/23 | 0: FALSE | L_DigitalLogic_3: bOut3: bIn3...bIn1=1 1 0 |
| C00829/24 | 0: FALSE | L_DigitalLogic_3: bOut3: bIn3...bIn1=1 1 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00830**

| Parameter \| Name: **C00830 \| 16-bit inputs [%]** | | | Data type: INTEGER_16 Index: $23745_d = 5CC1_h$ |
|---|---|---|---|
| Display in percent of 16-bit input values of different blocks | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00830/1 | | | L_Absolut_1: nIn_a |
| C00830/2 | | | L_AddSub_1: nIn1_a |
| C00830/3 | | | L_AddSub_1: nIn2_a |
| C00830/4 | | | L_AddSub_1: nIn3_a |

| Parameter \| Name:<br>**C00830 \| 16-bit inputs [%]** | Data type: INTEGER_16<br>Index: $23745_d = 5CC1_h$ |
|---|---|
| C00830/5 | L_OffsetGain_1: nIn_a |
| C00830/6 | L_OffsetGain_1: nOffset_a |
| C00830/7 | L_OffsetGain_1: nGain_a |
| C00830/8 | L_Negation_1: nIn_a |
| C00830/9 | L_GainOffset_1: nIn_a |
| C00830/10 | L_GainOffset_1: nGain_a |
| C00830/11 | L_GainOffset_1: nOffset_a |
| C00830/12 | L_Arithmetik_1: nIn1_a |
| C00830/13 | L_Arithmetik_1: nIn2_a |
| C00830/14 | L_AnalogSwitch_1: nIn1_a |
| C00830/15 | L_AnalogSwitch_1: nIn2_a |
| C00830/16 | L_Compare_1: nIn1_a |
| C00830/17 | L_Compare_1: nIn2_a |
| C00830/18 | MCTRL: nTorqueLimitAdapt_a |
| C00830/19 | Reserved |
| C00830/20 | MCTRL: nPosCtrlPAdapt_a |
| C00830/21 | MCTRL: nPosCtrlOutLimit_a |
| C00830/22 | MCTRL: nSpeedSetValue_a |
| C00830/23 | MCTRL: nSpeedLowLimit_a |
| C00830/24 | MCTRL: nSpeedCtrlI_a |
| C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| C00830/26 | MCTRL: nBoost_a |
| C00830/27 | MCTRL: nTorqueSetValue_a |
| C00830/28 | MCTRL: nTorqueGenLimit_a |
| C00830/29 | MCTRL: nTorqueMotLimit_a |
| C00830/30 | Reserved |
| C00830/31 | MCTRL: nVoltageAdd_a |
| C00830/32 | MCTRL: nPWMAngleOffset_a |
| C00830/33 | L_NSet_1: nCInhVal_a |
| C00830/34 | L_NSet_1: nNSet_a |
| C00830/35 | L_NSet_1: nSet_a |
| C00830/36 | L_NSet_1: nNAdd_a |
| C00830/37 | DCTRL: wCANControl |
| C00830/38 | DCTRL: wCCMControl |
| C00830/39 | L_NLim_1: nIn_a |
| C00830/40 | Reserved |
| C00830/41 | L_Compare_2: nIn1_a |
| C00830/42 | L_Compare_2: nIn2_a |
| C00830/43 | L_Compare_3: nIn1_a |
| C00830/44 | L_Compare_3: nIn2_a |
| C00830/45 | L_AnalogSwitch_2: nIn1_a |
| C00830/46 | L_AnalogSwitch_2: nIn2_a |
| C00830/47 | L_AnalogSwitch_3: nIn1_a |
| C00830/48 | L_AnalogSwitch_3: nIn2_a |

| Parameter \| Name:<br>**C00830 \| 16-bit inputs [%]** | Data type: INTEGER_16<br>Index: $23745_d = 5CC1_h$ |
|---|---|
| C00830/49 | L_Arithmetik_2: nIn1_a |
| C00830/50 | L_Arithmetik_2: nIn2_a |
| C00830/51 | Reserved |
| C00830/52 | Reserved |
| C00830/53 | L_GainOffset_2: nIn_a |
| C00830/54 | L_GainOffset_2: nGain_a |
| C00830/55 | L_GainOffset_2: nOffset_a |
| C00830/56 | L_OffsetGainP_1: nIn_a |
| C00830/57 | L_OffsetGainP_2: nIn_a |
| C00830/58 | L_OffsetGain_2: nIn_a |
| C00830/59 | L_OffsetGain_2: nOffset_a |
| C00830/60 | L_OffsetGain_2: nGain_a |
| C00830/61 | L_PCTRL_1: nAct_a |
| C00830/62 | L_PCTRL_1: nAdapt_a |
| C00830/63 | L_PCTRL_1: nSet_a |
| C00830/64 | L_PCTRL_1: nInfluence_a |
| C00830/65 | MCK: nSpeedCtrlI_a |
| C00830/66 | MCK: nPWMAngleOffset_a |
| C00830/67 | Reserved |
| C00830/68 | MCK: nMBrakeAddValue_a |
| C00830/69 | MCK: nTorqueSetValue_a |
| C00830/70 | MCK: nTorqueLimitAdapt_a |
| C00830/71 | MCK: nSRampOverride_a |
| C00830/72 | MCK: nSpeedSetValue_a |
| C00830/73 | MCK: wMotionCtrl2 |
| C00830/74 | MCK: wMotionCtrl1 |
| C00830/75 | MCK: nSpeedOverride_a |
| C00830/76 | MCK: nAccOverride_a |
| C00830/77 | MCK: nSpeedAdd_v |
| C00830/78 | MCK: wAuxCtrl |
| C00830/79 | MCK: wSMCtrl |
| C00830/80 | L_OffsetGainP_3: nIn_a |
| C00830/81 | L_MPot_1: nIn_a |
| C00830/82 | L_MulDiv_1: nIn_a |
| C00830/83 | LS_DataAccess: wIn1<br>(Lenze-internal) |
| C00830/84 | LS_DataAccess: wIn2<br>(Lenze-internal) |
| C00830/85 | LS_DataAccess: wIn3<br>(Lenze-internal) |
| C00830/86 | LS_DataAccess: wIn4<br>(Lenze-internal) |
| C00830/87 | L_PT1_1: nIn_a |
| C00830/88 | MCTRL: nSpeedHighLimit_a |
| C00830/89 | L_PCTRL_1: nNSet_a |

| Parameter \| Name: **C00830 \| 16-bit inputs [%]** | Data type: INTEGER_16 Index: $23745_d = 5CC1_h$ |
|---|---|
| C00830/90 | L_PCTRL_1: nISet_a |
| C00830/91 | L_Interpolator_1: nPhdIn_v |
| C00830/92 | L_Interpolator_1: nNIn_a |
| C00830/93 | Reserved |
| C00830/94 | Reserved |
| C00830/95 | Reserved |
| C00830/96 | MCTRL: nInertiaAdapt_a |
| C00830/97 | MCTRL: nSpeedSetValueInertia_a |
| C00830/98 | MCK: nProcessIn1_a |
| C00830/99 | MCK: nProcessIn2_a |
| C00830/100 | MCK: nProcessIn3_a |
| C00830/101 | MCK: nProcessIn4_a |
| C00830/102 | MCK: nProcessIn5_a |
| C00830/103 | MCK: nProcessIn6_a |
| C00830/104 | MCK: nProcessIn7_a |
| C00830/105 | MCK: nProcessIn8_a |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | |

**C00831**

| Parameter \| Name: **C00831 \| 16-bit inputs** | | | Data type: UNSIGNED_16 Index: $23744_d = 5CC0_h$ |
|---|---|---|---|
| Decimal/hexadecimal/bit coded display of 16 bit input values of various blocks | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00831/1 | | | L_Absolut_1: nIn_a |
| C00831/2 | | | L_AddSub_1: nIn1_a |
| C00831/3 | | | L_AddSub_1: nIn2_a |
| C00831/4 | | | L_AddSub_1: nIn3_a |
| C00831/5 | | | L_OffsetGain_1: nIn_a |
| C00831/6 | | | L_OffsetGain_1: nOffset_a |
| C00831/7 | | | L_OffsetGain_1: nGain_a |
| C00831/8 | | | L_Negation_1: nIn_a |
| C00831/9 | | | L_GainOffset_1: nIn_a |
| C00831/10 | | | L_GainOffset_1: nGain_a |
| C00831/11 | | | L_GainOffset_1: nOffset_a |
| C00831/12 | | | L_Arithmetik_1: nIn1_a |
| C00831/13 | | | L_Arithmetik_1: nIn2_a |
| C00831/14 | | | L_AnalogSwitch_1: nIn1_a |
| C00831/15 | | | L_AnalogSwitch_1: nIn2_a |

| Parameter \| Name:<br>**C00831 \| 16-bit inputs** | Data type: UNSIGNED_16<br>Index: $23744_d = 5CC0_h$ |
|---|---|
| C00831/16 | L_Compare_1: nIn1_a |
| C00831/17 | L_Compare_1: nIn2_a |
| C00831/18 | MCTRL: nTorqueLimitAdapt_a |
| C00831/19 | Reserved |
| C00831/20 | MCTRL: nPosCtrlPAdapt_a |
| C00831/21 | MCTRL: nPosCtrlOutLimit_a |
| C00831/22 | MCTRL: nSpeedSetValue_a |
| C00831/23 | MCTRL: nSpeedLowLimit_a |
| C00831/24 | MCTRL: nSpeedCtrlI_a |
| C00831/25 | MCTRL: nSpeedCtrlPAdapt_a |
| C00831/26 | MCTRL: nBoost_a |
| C00831/27 | MCTRL: nTorqueSetValue_a |
| C00831/28 | MCTRL: nTorqueGenLimit_a |
| C00831/29 | MCTRL: nTorqueMotLimit_a |
| C00831/30 | Reserved |
| C00831/31 | MCTRL: nVoltageAdd_a |
| C00831/32 | MCTRL: nPWMAngleOffset_a |
| C00831/33 | L_NSet_1: nCInhVal_a |
| C00831/34 | L_NSet_1: nNSet_a |
| C00831/35 | L_NSet_1: nSet_a |
| C00831/36 | L_NSet_1: nNAdd_a |
| C00831/37 | DCTRL: wCANControl |
| C00831/38 | DCTRL: wMCIControl |
| C00831/39 | L_NLim_1: nIn_a |
| C00831/40 | Reserved |
| C00831/41 | L_Compare_2: nIn1_a |
| C00831/42 | L_Compare_2: nIn2_a |
| C00831/43 | L_Compare_3: nIn1_a |
| C00831/44 | L_Compare_3: nIn2_a |
| C00831/45 | L_AnalogSwitch_2: nIn1_a |
| C00831/46 | L_AnalogSwitch_2: nIn2_a |
| C00831/47 | L_AnalogSwitch_3: nIn1_a |
| C00831/48 | L_AnalogSwitch_3: nIn2_a |
| C00831/49 | L_Arithmetik_2: nIn1_a |
| C00831/50 | L_Arithmetik_2: nIn2_a |
| C00831/51 | Reserved |
| C00831/52 | Reserved |
| C00831/53 | L_GainOffset_2: nIn_a |
| C00831/54 | L_GainOffset_2: nGain_a |
| C00831/55 | L_GainOffset_2: nOffset_a |
| C00831/56 | L_OffsetGainP_1: nIn_a |
| C00831/57 | L_OffsetGainP_2: nIn_a |
| C00831/58 | L_OffsetGain_2: nIn_a |
| C00831/59 | L_OffsetGain_2: nOffset_a |

| Parameter \| Name:<br>**C00831 \| 16-bit inputs** | Data type: UNSIGNED_16<br>Index: $23744_d = 5CC0_h$ |
|---|---|
| C00831/60 | L_OffsetGain_2: nGain_a |
| C00831/61 | L_PCTRL_1: nAct_a |
| C00831/62 | L_PCTRL_1: nAdapt_a |
| C00831/63 | L_PCTRL_1: nSet_a |
| C00831/64 | L_PCTRL_1: nInfluence_a |
| C00831/65 | MCK: nSpeedCtrlI_a |
| C00831/66 | MCK: nPWMAngleOffset_a |
| C00831/67 | Reserved |
| C00831/68 | MCK: nMBrakeAddValue_a |
| C00831/69 | MCK: nTorqueSetValue_a |
| C00831/70 | MCK: nTorqueLimitAdapt_a |
| C00831/71 | MCK: nSRampOverride_a |
| C00831/72 | MCK: nSpeedSetValue_a |
| C00831/73 | MCK: wMotionCtrl2 |
| C00831/74 | MCK: wMotionCtrl1 |
| C00831/75 | MCK: nSpeedOverride_a |
| C00831/76 | MCK: nAccOverride_a |
| C00831/77 | MCK: nSpeedAdd_v |
| C00831/78 | MCK: wAuxCtrl |
| C00831/79 | MCK: wSMCtrl |
| C00831/80 | L_OffsetGainP_3: nIn_a |
| C00831/81 | L_MPot_1: nIn_a |
| C00831/82 | L_MulDiv_1: nIn_a |
| C00831/83 | LS_DataAccess: wIn1<br>(Lenze-internal) |
| C00831/84 | LS_DataAccess: wIn2<br>(Lenze-internal) |
| C00831/85 | LS_DataAccess: wIn3<br>(Lenze-internal) |
| C00831/86 | LS_DataAccess: wIn4<br>(Lenze-internal) |
| C00831/87 | L_PT1_1: nIn_a |
| C00831/88 | MCTRL: nSpeedHighLimit_a |
| C00831/89 | L_PCTRL_1: nNSet_a |
| C00831/90 | L_PCTRL_1: nISet_a |
| C00831/91 | L_Interpolator_1: nPhdIn_v |
| C00831/92 | L_Interpolator_1: nNIn |
| C00831/93 | Reserved |
| C00831/94 | Reserved |
| C00831/95 | Reserved |
| C00831/96 | MCTRL: nInertiaAdapt_a |
| C00831/97 | MCTRL: nSpeedSetValueInertia_a |
| C00831/98 | MCK: nProcessIn1_a |
| C00831/99 | MCK: nProcessIn2_a |
| C00831/100 | MCK: nProcessIn3_a |

| Parameter \| Name: **C00831 \| 16-bit inputs** | Data type: UNSIGNED_16 Index: $23744_d = 5CC0_h$ |
|---|---|
| C00831/101 | MCK: nProcessIn4_a |
| C00831/102 | MCK: nProcessIn5_a |
| C00831/103 | MCK: nProcessIn6_a |
| C00831/104 | MCK: nProcessIn7_a |
| C00831/105 | MCK: nProcessIn8_a |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | |

**C00832**

| Parameter \| Name: **C00832 \| 16-bit inputs [incr./ms]** | | | Data type: INTEGER_16 Index: $23743_d = 5CBF_h$ |
|---|---|---|---|
| Display of 16-bit input values of different blocks in [rpm] | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -32767 | Incr./ms | 32767 | |
| **Subcodes** | | | **Info** |
| C00832/1 | | | L_Absolut_1: nIn_a |
| C00832/2 | | | L_AddSub_1: nIn1_a |
| C00832/3 | | | L_AddSub_1: nIn2_a |
| C00832/4 | | | L_AddSub_1: nIn3_a |
| C00832/5 | | | L_OffsetGain_1: nIn_a |
| C00832/6 | | | L_OffsetGain_1: nOffset_a |
| C00832/7 | | | L_OffsetGain_1: nGain_a |
| C00832/8 | | | L_Negation_1: nIn_a |
| C00832/9 | | | L_GainOffset_1: nIn_a |
| C00832/10 | | | L_GainOffset_1: nGain_a |
| C00832/11 | | | L_GainOffset_1: nOffset_a |
| C00832/12 | | | L_Arithmetik_1: nIn1_a |
| C00832/13 | | | L_Arithmetik_1: nIn2_a |
| C00832/14 | | | L_AnalogSwitch_1: nIn1_a |
| C00832/15 | | | L_AnalogSwitch_1: nIn2_a |
| C00832/16 | | | L_Compare_1: nIn1_a |
| C00832/17 | | | L_Compare_1: nIn2_a |
| C00832/18 | | | MCTRL: nTorqueLimitAdapt_a |
| C00832/19 | | | Reserved |
| C00832/20 | | | MCTRL: nPosCtrlPAdapt_a |
| C00832/21 | | | MCTRL: nPosCtrlOutLimit_a |
| C00832/22 | | | MCTRL: nSpeedSetValue_a |
| C00832/23 | | | MCTRL: nSpeedLowLimit_a |
| C00832/24 | | | MCTRL: nSpeedCtrlI_a |
| C00832/25 | | | MCTRL: nSpeedCtrlPAdapt_a |
| C00832/26 | | | MCTRL: nBoost_a |
| C00832/27 | | | MCTRL: nTorqueSetValue_a |
| C00832/28 | | | MCTRL: nTorqueGenLimit_a |
| C00832/29 | | | MCTRL: nTorqueMotLimit_a |
| C00832/30 | | | Reserved |

| Parameter \| Name:<br>**C00832 \| 16-bit inputs [incr./ms]** | Data type: INTEGER_16<br>Index: $23743_d = 5CBF_h$ |
|---|---|
| C00832/31 | MCTRL: nVoltageAdd_a |
| C00832/32 | MCTRL: nPWMAngleOffset_a |
| C00832/33 | L_NSet_1: nCInhVal_a |
| C00832/34 | L_NSet_1: nNSet_a |
| C00832/35 | L_NSet_1: nSet_a |
| C00832/36 | L_NSet_1: nNAdd_a |
| C00832/37 | DCTRL: wCANControl |
| C00832/38 | DCTRL: wCCMControl |
| C00832/39 | L_NLim_1: nIn_a |
| C00832/40 | Reserved |
| C00832/41 | L_Compare_2: nIn1_a |
| C00832/42 | L_Compare_2: nIn2_a |
| C00832/43 | L_Compare_3: nIn1_a |
| C00832/44 | L_Compare_3: nIn2_a |
| C00832/45 | L_AnalogSwitch_2: nIn1_a |
| C00832/46 | L_AnalogSwitch_2: nIn2_a |
| C00832/47 | L_AnalogSwitch_3: nIn1_a |
| C00832/48 | L_AnalogSwitch_3: nIn2_a |
| C00832/49 | L_Arithmetik_2: nIn1_a |
| C00832/50 | L_Arithmetik_2: nIn2_a |
| C00832/51 | Reserved |
| C00832/52 | Reserved |
| C00832/53 | L_GainOffset_2: nIn_a |
| C00832/54 | L_GainOffset_2: nGain_a |
| C00832/55 | L_GainOffset_2: nOffset_a |
| C00832/56 | L_OffsetGainP_1: nIn_a |
| C00832/57 | L_OffsetGainP_2: nIn_a |
| C00832/58 | L_OffsetGain_2: nIn_a |
| C00832/59 | L_OffsetGain_2: nOffset_a |
| C00832/60 | L_OffsetGain_2: nGain_a |
| C00832/61 | L_PCTRL_1: nAct_a |
| C00832/62 | L_PCTRL_1: nAdapt_a |
| C00832/63 | L_PCTRL_1: nSet_a |
| C00832/64 | L_PCTRL_1: nInfluence_a |
| C00832/65 | MCK: nSpeedCtrlI_a |
| C00832/66 | MCK: nPWMAngleOffset_a |
| C00832/67 | Reserved |
| C00832/68 | MCK: nMBrakeAddValue_a |
| C00832/69 | MCK: nTorqueSetValue_a |
| C00832/70 | MCK: nTorqueLimitAdapt_a |
| C00832/71 | MCK: nSRampOverride_a |
| C00832/72 | MCK: nSpeedSetValue_a |
| C00832/73 | MCK: wMotionCtrl2 |
| C00832/74 | MCK: wMotionCtrl1 |

| Parameter \| Name:<br>**C00832 \| 16-bit inputs [incr./ms]** | Data type: INTEGER_16<br>Index: $23743_d = 5CBF_h$ |
|---|---|
| C00832/75 | MCK: nSpeedOverride_a |
| C00832/76 | MCK: nAccOverride_a |
| C00832/77 | MCK: nSpeedAdd_v |
| C00832/78 | MCK: wAuxCtrl |
| C00832/79 | MCK: wSMCtrl |
| C00832/80 | L_OffsetGainP_3: nIn_a |
| C00832/81 | L_MPot_1: nIn_a |
| C00832/82 | L_MulDiv_1: nIn_a |
| C00832/83 | LS_DataAccess: wIn1<br>(Lenze-internal) |
| C00832/84 | LS_DataAccess: wIn2<br>(Lenze-internal) |
| C00832/85 | LS_DataAccess: wIn3<br>(Lenze-internal) |
| C00832/86 | LS_DataAccess: wIn4<br>(Lenze-internal) |
| C00832/87 | L_PT1_1: nIn_a |
| C00832/88 | MCTRL: nSpeedHighLimit_a |
| C00832/89 | L_PCTRL_1: nNSet_a |
| C00832/90 | L_PCTRL_1: nISet_a |
| C00832/91 | L_Interpolator_1: nPhdIn_v |
| C00832/92 | L_Interpolator_1: nNIn_a |
| C00832/93 | Reserved |
| C00832/94 | Reserved |
| C00832/95 | Reserved |
| C00832/96 | MCTRL: nInertiaAdapt_a |
| C00832/97 | MCTRL: nSpeedSetValueInertia_a |
| C00832/98 | MCK: nProcessIn1_a |
| C00832/99 | MCK: nProcessIn2_a |
| C00832/100 | MCK: nProcessIn3_a |
| C00832/101 | MCK: nProcessIn4_a |
| C00832/102 | MCK: nProcessIn5_a |
| C00832/103 | MCK: nProcessIn6_a |
| C00832/104 | MCK: nProcessIn7_a |
| C00832/105 | MCK: nProcessIn8_a |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00833**

| Parameter \| Name:<br>**C00833 \| Binary inputs** | Data type: UNSIGNED_8<br>Index: $23742_d = 5CBE_h$ |
|---|---|
| Display of the signal status of the binary inputs of different blocks | |
| **Selection list** | |
| 0 \| False | |
| 1 \| True | |
| **Subcodes** | **Info** |

| Parameter \| Name:<br>**C00833 \| Binary inputs** | Data type: UNSIGNED_8<br>Index: $23742_d = 5CBE_h$ |
|---|---|
| C00833/1 | L_And_1: bIn1 |
| C00833/2 | L_And_1: bIn2 |
| C00833/3 | L_And_1: bIn3 |
| C00833/4 | L_DFlipFlop_1: bD |
| C00833/5 | L_DFlipFlop_1: bClk |
| C00833/6 | L_DFlipFlop_1: bClr |
| C00833/7 | L_Not_1: bIn |
| C00833/8 | L_Or_1: bIn1 |
| C00833/9 | L_Or_1: bIn2 |
| C00833/10 | L_Or_1: bIn3 |
| C00833/11 | L_RLQ_1: bCw |
| C00833/12 | L_RLQ_1: bCcw |
| C00833/13 | L_AnalogSwitch_1: bSet |
| C00833/14 | L_NSet_1: bRfgStop |
| C00833/15 | L_NSet_1: bRfg0 |
| C00833/16 | L_NSet_1: bNSetInv |
| C00833/17 | L_NSet_1: bJog1 |
| C00833/18 | L_NSet_1: bJog2 |
| C00833/19 | L_NSet_1: bJog4 |
| C00833/20 | L_NSet_1: bJog8 |
| C00833/21 | L_NSet_1: bTi1 |
| C00833/22 | L_NSet_1: bTi2 |
| C00833/23 | L_NSet_1: bTi4 |
| C00833/24 | L_NSet_1: bTi8 |
| C00833/25 | L_NSet_1: bLoad |
| C00833/26 | L_NSet_1: bExternalCINH |
| C00833/27 | MCTRL: bPosCtrlOn |
| C00833/28 | MCTRL: bSpeedInterpolatorOn |
| C00833/29 | MCTRL: bTorqueInterpolatorOn |
| C00833/30 | MCTRL: bTorquemodeOn |
| C00833/31 | MCTRL: bSpeedCtrlIOn |
| C00833/32 | MCTRL: bAutoBoostOn |
| C00833/33 | MCTRL: bQSPOn |
| C00833/34 | MCTRL: bDcBrakeOn |
| C00833/35 | MCTRL: bDeltaPosOn |
| C00833/36 | DCTRL: bCINH |
| C00833/37 | DCTRL: bFailReset |
| C00833/38 | DCTRL: bStatus_B0 |
| C00833/39 | DCTRL: bStatus_B2 |
| C00833/40 | DCTRL: bStatus_B3 |
| C00833/41 | DCTRL: bStatus_B4 |
| C00833/42 | DCTRL: bStatus_B5 |
| C00833/43 | DCTRL: bStatus_B14 |
| C00833/44 | DCTRL: bStatus_B15 |

| Parameter \| Name:<br>**C00833 \| Binary inputs** | Data type: UNSIGNED_8<br>Index: $23742_d = 5CBE_h$ |
|---|---|
| C00833/45 | DCTRL: bFree_1 |
| C00833/46 | DCTRL: bFree_2 |
| C00833/47 | DCTRL: bFree_3 |
| C00833/48 | DCTRL: bFree_4 |
| C00833/49 | L_And_2: bIn1 |
| C00833/50 | L_And_2: bIn2 |
| C00833/51 | L_And_2: bIn3 |
| C00833/52 | L_And_3: bIn1 |
| C00833/53 | L_And_3: bIn2 |
| C00833/54 | L_And_3: bIn3 |
| C00833/55 | L_Or_2: bIn1 |
| C00833/56 | L_Or_2: bIn2 |
| C00833/57 | L_Or_2: bIn3 |
| C00833/58 | L_Or_3: bIn1 |
| C00833/59 | L_Or_3: bIn2 |
| C00833/60 | L_Or_3: bIn3 |
| C00833/61 | L_Not_2: bIn |
| C00833/62 | L_Not_3: bIn |
| C00833/63 | L_DigitalLogic_1: bIn1 |
| C00833/64 | L_DigitalLogic_1: bIn2 |
| C00833/65 | L_DigitalLogic_1: bIn3 |
| C00833/66 | L_DigitalDelay_1: bIn |
| C00833/67 | MCTRL: bPosDerivativeOn |
| C00833/68 | MCTRL: bMotorRefOffsetOn |
| C00833/69 | MCTRL: bSpeedCtrlPAdaptOn |
| C00833/70 | L_AnalogSwitch_2: bSet |
| C00833/71 | L_AnalogSwitch_3: bSet |
| C00833/72 | L_MPot_1: bUp |
| C00833/73 | L_MPot_1: bInAct |
| C00833/74 | L_MPot_1: bDown |
| C00833/75 | L_PCTRL_1: bPIDOff |
| C00833/76 | L_PCTRL_1: bInAct |
| C00833/77 | L_PCTRL_1: bIOff |
| C00833/78 | MCK: bSpeedCtrlIOn |
| C00833/79 | MCK: bDcBrakeOn |
| C00833/80 | MCK: bMBrakeRelease |
| C00833/81 | MCK: bMBrakeStartValue2 |
| C00833/82 | MCK: bMBrakeApplied |
| C00833/83 | MCK: bLimitSwitchPos |
| C00833/84 | MCK: bLimitSwitchNeg |
| C00833/85 | MCK: bPosCtrlOn |
| C00833/86 | MCK: bDeltaPosOn |
| C00833/87 | MCK: bPosDerivativeOn |
| C00833/88 | MCK: bMotorRefOffsetOn |

| Parameter \| Name:<br>**C00833 \| Binary inputs** | Data type: UNSIGNED_8<br>Index: $23742_d = 5CBE_h$ |
|---|---|
| C00833/89 | MCK: bQspOn |
| C00833/90 | MCK: bTorquemodeOn |
| C00833/91 | MCK: bTorqueLimitAdaptOn |
| C00833/92 | MCK: bHomMark |
| C00833/93 | L_Transient_1: bIn |
| C00833/94 | L_Transient_2: bIn |
| C00833/95 | L_Transient_3: bIn |
| C00833/96 | L_Transient_4: bIn |
| C00833/97 | Reserved |
| C00833/98 | MCTRL: bTorqueLimitAdaptOn |
| C00833/99 | L_NSet_1: bNAddInv |
| C00833/100 | L_MPot_1: bEnable |
| C00833/101 | L_NLim_1: bEnable |
| C00833/102 | LS_DataAccess: bEnableIn1<br>(Lenze-internal) |
| C00833/103 | LS_DataAccess: bEnableIn2<br>(Lenze-internal) |
| C00833/104 | LS_DataAccess: bEnableIn3<br>(Lenze-internal) |
| C00833/105 | LS_DataAccess: bEnableIn4<br>(Lenze-internal) |
| C00833/106 | L_PCTRL_1: bEnableInfluenceRamp |
| C00833/107 | LS_SetError_2: bSetError1 |
| C00833/108 | LS_SetError_2: bSetError2 |
| C00833/109 | LS_SetError_2: bSetError3 |
| C00833/110 | LS_SetError_2: bSetError4 |
| C00833/111 | L_JogCtrlExtension_1: bInputSel1 |
| C00833/112 | L_JogCtrlExtension_1: bInputSel2 |
| C00833/113 | L_JogCtrlExtension_1: bRfgIn |
| C00833/114 | L_JogCtrlExtension_1: bJog1In |
| C00833/115 | L_JogCtrlExtension_1: bJog2In |
| C00833/116 | L_JogCtrlExtension_1: bSlowDown1 |
| C00833/117 | L_JogCtrlExtension_1: bStop1 |
| C00833/118 | L_JogCtrlExtension_1: bSlowDown2 |
| C00833/119 | L_JogCtrlExtension_1: bStop2 |
| C00833/120 | L_JogCtrlExtension_1: bSlowDown3 |
| C00833/121 | L_JogCtrlExtension_1: bStop3 |
| C00833/122 | L_PCTRL_1: bISet |
| C00833/123 | L_Interpolator_1: bSpeedAct0 |
| C00833/124 | L_Or_4: bIn1 |
| C00833/125 | L_Or_4: bIn2 |
| C00833/126 | L_Or_4: bIn3 |
| C00833/127 | L_DigitalLogic_3: bIn1 |
| C00833/128 | L_DigitalLogic_3: bIn2 |
| C00833/129 | L_DigitalLogic_3: bIn3 |

| Parameter \| Name: **C00833 \| Binary inputs** | Data type: UNSIGNED_8 Index: $23742_d = 5CBE_h$ |
|---|---|
| C00833/130 | MCTRL: bBrakeChopperOn |
| C00833/131 | MCTRL: bVfcEcoDisable |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00834**

| Parameter \| Name: **C00834 \| 32-bit inputs [incr]** | | | Data type: INTEGER_32 Index: $23741_d = 5CBD_h$ |
|---|---|---|---|
| Display in [increments] of 32 bit input values of various blocks | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00834/1 | | | MCK: dnPosSetValue_p |
| C00834/2 | | | MCK: dnMotorRefOffset_p |
| C00834/3 | | | MCK: dnDeltaPos_p |
| C00834/4 | | | MCTRL: dnDeltaPos_p |
| C00834/5 | | | MCTRL: dnPosSetValue_p |
| C00834/6 | | | MCTRL: dnMotorRefOffset_p |
| C00834/7 | | | MCK: dnProfilePosition_p |
| C00834/8 | | | L_Interpolator_1: dnPhiIn_p |
| C00834/9 | | | MCK: dnProcessIn1_p |
| C00834/10 | | | MCK: dnProcessIn2_p |
| C00834/11 | | | MCK: dnProcessIn3_p |
| C00834/12 | | | MCK: dnProcessIn4_p |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00835**

| Parameter \| Name: **C00835 \| 16-bit inputs [%] (Set2)** | | | Data type: INTEGER_16 Index: $23740_d = 5CBC_h$ |
|---|---|---|---|
| Display in percent of 16-bit input values of different blocks | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00835/1 | | | L_Absolut_2: nIn_a |
| C00835/2 | | | L_AnalogSwitch_4: nIn1_a |
| C00835/3 | | | L_AnalogSwitch_4: nIn2_a |
| C00835/4 | | | L_AnalogSwitch_5: nIn1_a |
| C00835/5 | | | L_AnalogSwitch_5: nIn2_a |
| C00835/6 | | | L_Compare_4: nIn1_a |
| C00835/7 | | | L_Compare_4: nIn2_a |
| C00835/8 | | | L_Compare_5: nIn1_a |
| C00835/9 | | | L_Compare_5: nIn2_a |
| C00835/10 | | | L_Arithmetik_3: nIn1_a |
| C00835/11 | | | L_Arithmetik_3: nIn2_a |
| C00835/12 | | | L_Arithmetik_4: nIn1_a |

| Parameter \| Name:<br>**C00835 \| 16-bit inputs [%] (Set2)** | Data type: INTEGER_16<br>Index: $23740_d = 5CBC_h$ |
|---|---|
| C00835/13 | L_Arithmetik_4: nIn2_a |
| C00835/14 | L_Arithmetik_5: nIn1_a |
| C00835/15 | L_Arithmetik_5: nIn2_a |
| C00835/16 | L_Counter_2: wLdVal |
| C00835/17 | L_Counter_2: wCmpVal |
| C00835/18 | L_Counter_3: wLdVal |
| C00835/19 | L_Counter_3: wCmpVal |
| C00835/20 | L_PhaseIntK_1: nIn_v |
| C00835/21 | L_Negation_2: nIn_a |
| C00835/22 | L_NLim_2: nIn_a |
| C00835/23 | L_OffsetGain_3: nIn_a |
| C00835/24 | L_OffsetGain_3: nOffset_a |
| C00835/25 | L_OffsetGain_3: nGain_a |
| C00835/26 | L_PT1_2: nIn_a |
| C00835/27 | L_PT1_3: nIn_a |
| C00835/28 | L_PhaseIntK_2: nIn_v |
| C00835/29 | L_SampleHold_1: nIn_a |
| C00835/30 | L_SampleHold_2: nIn_a |
| C00835/31 | L_Mux_1: wInSelect |
| C00835/32 | L_GainOffset_3: nIn_a |
| C00835/33 | L_GainOffset_3: nGain_a |
| C00835/34 | L_GainOffset_3: nOffset_a |
| C00835/35 | L_MulDiv_2: nIn_a |
| C00835/36 | L_DT1_1: nIn_a |
| C00835/37 | L_Counter_1: wLdVal |
| C00835/38 | L_Counter_1: wCmpVal |
| C00835/39 | L_GainOffsetP_1: nIn_a |
| C00835/40 | L_GainOffsetP_2: nIn_a |
| C00835/41 | L_GainOffsetP_3: nIn_a |
| C00835/42 | L_Limit_1: nIn_a |
| C00835/43 | L_Limit_2: nIn_a |
| C00835/44 | L_MckCtrlInterface_1: wOperationMode |
| C00835/45 | L_MckCtrlInterface_1: wPosMode |
| C00835/46 | L_MckCtrlInterface_1: wProfileNo |
| C00835/47 | L_MckCtrlInterface_1: wInMckPosCtrl_1 |
| C00835/48 | L_MckCtrlInterface_1: wInMckPosCtrl_2 |
| C00835/49 | L_MckStateInterface_1: wInMckPosState_1 |
| C00835/50 | L_MckStateInterface_1: wInMckPosState_2 |
| C00835/51 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_1 |
| C00835/52 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_2 |
| C00835/53 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_3 |
| C00835/54 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_4 |
| C00835/55 | L_ConvWordToBits_1: wInput |
| C00835/56 | L_ConvWordToBits_2: wInput |

| Parameter \| Name:<br>**C00835 \| 16-bit inputs [%] (Set2)** | Data type: INTEGER_16<br>Index: $23740_d = 5CBC_h$ |
|---|---|
| C00835/57 | L_ConvWordToBits_3: wInput |
| C00835/58 | L_ConvWordsToDInt_1: wInLWord |
| C00835/59 | L_ConvWordsToDInt_1: wInHWord |
| C00835/60 | L_ConvWordsToDInt_2: wInLWord |
| C00835/61 | L_ConvWordsToDInt_2: wInHWord |
| C00835/62 | L_ConvWordsToDInt_3: wInLWord |
| C00835/63 | L_ConvWordsToDInt_3: wInHWord |
| C00835/64 | L_ConvUnitsToIncr_1: wInLWord |
| C00835/65 | L_ConvUnitsToIncr_1: wInHWord |
| C00835/66 | L_ConvUnitsToIncr_2: wInLWord |
| C00835/67 | L_ConvUnitsToIncr_2: wInHWord |
| C00835/68 | L_ConvUnitsToIncr_3: wInLWord |
| C00835/69 | L_ConvUnitsToIncr_3: wInHWord |
| C00835/70 | L_Curve_1: nIn_a |
| C00835/71 | L_ConvW_1: wIn |
| C00835/72 | L_ConvW_2: wIn |
| C00835/73 | L_ConvW_3: wIn |
| C00835/74 | L_ConvW_4: wIn |
| C00835/75 | L_MckCtrlInterface_1: wPosSetHW |
| C00835/76 | L_MckCtrlInterface_1: wPosSetLW |
| C00835/77 | L_PhaseDiff_1: nIn_v |
| C00835/78 | L_PhaseDiff_2: nIn_v |
| C00835/79 | L_SRFG_1: nIn_a |
| C00835/80 | L_SRFG_2: nIn_a |
| C00835/81 | L_SRFG_1: nSet_a |
| C00835/82 | L_SRFG_2: nSet_a |
| C00835/83 | L_SignalSwitch_1: wIn1 |
| C00835/84 | L_SignalSwitch_2: wIn1 |
| C00835/85 | L_SignalSwitch_3: wIn1 |
| C00835/86 | L_SignalSwitch_4: wIn1 |
| C00835/87 | L_SignalSwitch_1: wIn2 |
| C00835/88 | L_SignalSwitch_2: wIn2 |
| C00835/89 | L_SignalSwitch_3: wIn2 |
| C00835/90 | L_SignalSwitch_4: wIn2 |
| C00835/91 | L_Odometer_1: nInSpeed_v |
| C00835/92 | L_CalcDiameter_1: wDMax |
| C00835/93 | L_CalcDiameter_1: wDMin |
| C00835/94 | L_CalcDiameter_1: wVMax |
| C00835/95 | L_CalcDiameter_1: nVLine_a |
| C00835/96 | L_CalcDiameter_1: nMotorSpeedAct_v |
| C00835/97 | L_CalcDiameter_1: wGearNum |
| C00835/98 | L_CalcDiameter_1: wGearDenom |
| C00835/99 | L_CalcDiameter_1: nSetD_a |
| C00835/100 | L_ProcessCtrl_1: nVpAdapt_a |

| Parameter \| Name:<br>**C00835 \| 16-bit inputs [%] (Set2)** | Data type: INTEGER_16<br>Index: $23740_d = 5CBC_h$ |
|---|---|
| C00835/101 | L_ProcessCtrl_1: nSet_a |
| C00835/102 | L_ProcessCtrl_1: nAct_a |
| C00835/103 | L_ProcessCtrl_1: nRTimeAdapt_a |
| C00835/104 | L_ProcessCtrl_1: nInfluence_a |
| C00835/105 | L_PhiIntegrator_1: nIn_v |
| C00835/106 | L_PhiIntegrator_1: wGearNum |
| C00835/107 | L_PhiIntegrator_1: wGearDenom |
| C00835/108 | L_SwitchPoint_1: nActSpeed_v |
| C00835/109 | L_PhiIntegrator_1: nSpeedAdd_v |
| C00835/110 | L_DFSET_1: nSpeedTrim_v |
| C00835/111 | L_DFSET_1: nSpeedTrim_a |
| C00835/112 | L_DFSET_1: wGearNum |
| C00835/113 | L_DFSET_1: wGainNum |
| C00835/114 | L_DFSET_1: nSet_v |
| C00835/115 | L_DFSET_1: wGainDenom |
| C00835/116 | L_DFSET_1: wGearDenom |
| C00835/117 | L_DFSET_1: nPositionTrimming |
| C00835/118 | L_DFSET_1: nNAct_v |
| C00835/119 | L_DFRFG_1: nIn_v |
| C00835/120 | L_GearComp_1: nTorque_a |
| C00835/121 | L_ConvAP_1: nIn_a |
| C00835/122 | L_ConvAP_2: nIn_a |
| C00835/123 | L_ConvAP_3: nIn_a |
| C00835/124 | L_ConvX_1: nIn_a |
| C00835/125 | L_ConvX_1: nNum |
| C00835/126 | L_ConvX_1: wDenom |
| C00835/127 | L_ConvX_2: nIn_a |
| C00835/128 | L_ConvX_2: nNum |
| C00835/129 | L_ConvX_2: wDenom |
| C00835/130 | L_ConvX_3: nIn_a |
| C00835/131 | L_ConvX_3: nNum |
| C00835/132 | L_ConvX_3: wDenom |
| C00835/133 | L_ConvPP_1: nNum |
| C00835/134 | L_ConvPP_1: wDenom |
| C00835/135 | L_ConvPP_2: nNum |
| C00835/136 | L_ConvPP_2: wDenom |
| C00835/137 | L_ConvPP_3: nNum |
| C00835/138 | L_ConvPP_3: wDenom |
| C00835/139 | L_Curve_2: nIn_a |
| C00835/140 | L_Curve_3: nIn_a |
| C00835/141 | L_Sequencer_1: wStartStep |
| C00835/142 | L_Sequencer_1: wBranch1 |
| C00835/143 | L_Sequencer_1: wBranch2 |
| C00835/144 | L_Sequencer_1: wMotionState1 |

| Parameter \| Name:<br>**C00835 \| 16-bit inputs [%] (Set2)** | Data type: INTEGER_16<br>Index: $23740_d = 5CBC_h$ |
|---|---|
| C00835/145 | L_Sequencer_1: wMotionState2 |
| C00835/146 | L_Sequencer_1: wDigitalInputs |
| C00835/147 | L_ConvActPos_1: nSetPos_a |
| C00835/148 | L_ConvActPos_1: nPosIn_a |
| C00835/149 | L_ConvActPos_1: wVMax |
| C00835/150 | L_ConvActPos_1: nVLine_a |
| C00835/151 | L_MFail_1: nAdapt_a |
| C00835/152 | L_MFail_1: nConst_a |
| C00835/153 | L_MFail_1: nDCSet_a |
| C00835/154 | L_MFail_1: nNSet_a |
| C00835/155 | L_MFail_1: nThreshold_a |
| C00835/156 | L_MFail_1: nNAct_a |
| C00835/157 | L_MFail_1: nSet_a |
| C00835/158 | L_MFail_1: nDCVoltAct_a |
| C00835/159 | L_Curve_3: nCurveCtrl_a |
| C00835/160 | L_Curve_3: nD0_a |
| C00835/161 | L_Curve_3: nDiameter_a |
| C00835/162 | L_SwitchPointPar_1: nActSpeed_v |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | |

**C00836**

| Parameter \| Name:<br>**C00836 \| 16-bit inputs (Set2)** | | | Data type: UNSIGNED_16<br>Index: $23739_d = 5CBB_h$ |
|---|---|---|---|
| Decimal/hexadecimal/bit coded display of 16 bit input values of various blocks | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00836/1 | | | L_Absolut_2: nIn_a |
| C00836/2 | | | L_AnalogSwitch_4: nIn1_a |
| C00836/3 | | | L_AnalogSwitch_4: nIn2_a |
| C00836/4 | | | L_AnalogSwitch_5: nIn1_a |
| C00836/5 | | | L_AnalogSwitch_5: nIn2_a |
| C00836/6 | | | L_Compare_4: nIn1_a |
| C00836/7 | | | L_Compare_4: nIn2_a |
| C00836/8 | | | L_Compare_5: nIn1_a |
| C00836/9 | | | L_Compare_5: nIn2_a |
| C00836/10 | | | L_Arithmetik_3: nIn1_a |
| C00836/11 | | | L_Arithmetik_3: nIn2_a |
| C00836/12 | | | L_Arithmetik_4: nIn1_a |
| C00836/13 | | | L_Arithmetik_4: nIn2_a |

| Parameter \| Name:<br>**C00836 \| 16-bit inputs (Set2)** | Data type: UNSIGNED_16<br>Index: $23739_d = 5CBB_h$ |
|---|---|
| C00836/14 | L_Arithmetik_5: nIn1_a |
| C00836/15 | L_Arithmetik_5: nIn2_a |
| C00836/16 | L_Counter_2: wLdVal |
| C00836/17 | L_Counter_2: wCmpVal |
| C00836/18 | L_Counter_3: wLdVal |
| C00836/19 | L_Counter_3: wCmpVal |
| C00836/20 | L_PhaseIntK_1: nIn_v |
| C00836/21 | L_Negation_2: nIn_a |
| C00836/22 | L_NLim_2: nIn_a |
| C00836/23 | L_OffsetGain_3: nIn_a |
| C00836/24 | L_OffsetGain_3: nOffset_a |
| C00836/25 | L_OffsetGain_3: nGain_a |
| C00836/26 | L_PT1_2: nIn_a |
| C00836/27 | L_PT1_3: nIn_a |
| C00836/28 | L_PhaseIntK_2: nIn_v |
| C00836/29 | L_SampleHold_1: nIn_a |
| C00836/30 | L_SampleHold_2: nIn_a |
| C00836/31 | L_Mux_1: wInSelect |
| C00836/32 | L_GainOffset_3: nIn_a |
| C00836/33 | L_GainOffset_3: nGain_a |
| C00836/34 | L_GainOffset_3: nOffset_a |
| C00836/35 | L_MulDiv_2: nIn_a |
| C00836/36 | L_DT1_1: nIn_a |
| C00836/37 | L_Counter_1: wLdVal |
| C00836/38 | L_Counter_1: wCmpVal |
| C00836/39 | L_GainOffsetP_1: nIn_a |
| C00836/40 | L_GainOffsetP_2: nIn_a |
| C00836/41 | L_GainOffsetP_3: nIn_a |
| C00836/42 | L_Limit_1: nIn_a |
| C00836/43 | L_Limit_2: nIn_a |
| C00836/44 | L_MckCtrlInterface_1: wOperationMode |
| C00836/45 | L_MckCtrlInterface_1: wPosMode |
| C00836/46 | L_MckCtrlInterface_1: wProfileNo |
| C00836/47 | L_MckCtrlInterface_1: wInMckPosCtrl_1 |
| C00836/48 | L_MckCtrlInterface_1: wInMckPosCtrl_2 |
| C00836/49 | L_MckStateInterface_1: wInMckPosState_1 |
| C00836/50 | L_MckStateInterface_1: wInMckPosState_2 |
| C00836/51 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_1 |
| C00836/52 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_2 |
| C00836/53 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_3 |
| C00836/54 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_4 |
| C00836/55 | L_ConvWordToBits_1: wInput |
| C00836/56 | L_ConvWordToBits_2: wInput |
| C00836/57 | L_ConvWordToBits_3: wInput |

| Parameter \| Name:<br>**C00836 \| 16-bit inputs (Set2)** | Data type: UNSIGNED_16<br>Index: $23739_d = 5CBB_h$ |
|---|---|
| C00836/58 | L_ConvWordsToDInt_1: wInLWord |
| C00836/59 | L_ConvWordsToDInt_1: wInHWord |
| C00836/60 | L_ConvWordsToDInt_2: wInLWord |
| C00836/61 | L_ConvWordsToDInt_2: wInHWord |
| C00836/62 | L_ConvWordsToDInt_3: wInLWord |
| C00836/63 | L_ConvWordsToDInt_3: wInHWord |
| C00836/64 | L_ConvUnitsToIncr_1: wInLWord |
| C00836/65 | L_ConvUnitsToIncr_1: wInHWord |
| C00836/66 | L_ConvUnitsToIncr_2: wInLWord |
| C00836/67 | L_ConvUnitsToIncr_2: wInHWord |
| C00836/68 | L_ConvUnitsToIncr_3: wInLWord |
| C00836/69 | L_ConvUnitsToIncr_3: wInHWord |
| C00836/70 | L_Curve_1: nIn_a |
| C00836/71 | L_ConvW_1: wIn |
| C00836/72 | L_ConvW_2: wIn |
| C00836/73 | L_ConvW_3: wIn |
| C00836/74 | L_ConvW_4: wIn |
| C00836/75 | L_MckCtrlInterface_1: wPosSetHW |
| C00836/76 | L_MckCtrlInterface_1: wPosSetLW |
| C00836/77 | L_PhaseDiff_1: nIn_v |
| C00836/78 | L_PhaseDiff_2: nIn_v |
| C00836/79 | L_SRFG_1: nIn_a |
| C00836/80 | L_SRFG_2: nIn_a |
| C00836/81 | L_SRFG_1: nSet_a |
| C00836/82 | L_SRFG_2: nSet_a |
| C00836/83 | L_SignalSwitch_1: wIn1 |
| C00836/84 | L_SignalSwitch_2: wIn1 |
| C00836/85 | L_SignalSwitch_3: wIn1 |
| C00836/86 | L_SignalSwitch_4: wIn1 |
| C00836/87 | L_SignalSwitch_1: wIn2 |
| C00836/88 | L_SignalSwitch_2: wIn2 |
| C00836/89 | L_SignalSwitch_3: wIn2 |
| C00836/90 | L_SignalSwitch_4: wIn2 |
| C00836/91 | L_Odometer_1: nInSpeed_v |
| C00836/92 | L_CalcDiameter_1: wDMax |
| C00836/93 | L_CalcDiameter_1: wDMin |
| C00836/94 | L_CalcDiameter_1: wVMax |
| C00836/95 | L_CalcDiameter_1: nVLine_a |
| C00836/96 | L_CalcDiameter_1: nMotorSpeedAct_v |
| C00836/97 | L_CalcDiameter_1: wGearNum |
| C00836/98 | L_CalcDiameter_1: wGearDenom |
| C00836/99 | L_CalcDiameter_1: nSetD_a |
| C00836/100 | L_ProcessCtrl_1: nVpAdapt_a |
| C00836/101 | L_ProcessCtrl_1: nSet_a |

| Parameter \| Name:<br>**C00836 \| 16-bit inputs (Set2)** | Data type: UNSIGNED_16<br>Index: $23739_d = 5CBB_h$ |
|---|---|
| C00836/102 | L_ProcessCtrl_1: nAct_a |
| C00836/103 | L_ProcessCtrl_1: nRTimeAdapt_a |
| C00836/104 | L_ProcessCtrl_1: nInfluence_a |
| C00836/105 | L_PhiIntegrator_1: nIn_v |
| C00836/106 | L_PhiIntegrator_1: wGearNum |
| C00836/107 | L_PhiIntegrator_1: wGearDenom |
| C00836/108 | L_SwitchPoint_1: nActSpeed_v |
| C00836/109 | L_PhiIntegrator_1: nSpeedAdd_v |
| C00836/110 | L_DFSET_1: nSpeedTrim_v |
| C00836/111 | L_DFSET_1: nSpeedTrim_a |
| C00836/112 | L_DFSET_1: wGearNum |
| C00836/113 | L_DFSET_1: wGainNum |
| C00836/114 | L_DFSET_1: nSet_v |
| C00836/115 | L_DFSET_1: wGainDenom |
| C00836/116 | L_DFSET_1: wGearDenom |
| C00836/117 | L_DFSET_1: nPositionTrimming |
| C00836/118 | L_DFSET_1: nNAct_v |
| C00836/119 | L_DFRFG_1: nIn_v |
| C00836/120 | L_GearComp_1: nTorque_a |
| C00836/121 | L_ConvAP_1: nIn_a |
| C00836/122 | L_ConvAP_2: nIn_a |
| C00836/123 | L_ConvAP_3: nIn_a |
| C00836/124 | L_ConvX_1: nIn_a |
| C00836/125 | L_ConvX_1: nNum |
| C00836/126 | L_ConvX_1: wDenom |
| C00836/127 | L_ConvX_2: nIn_a |
| C00836/128 | L_ConvX_2: nNum |
| C00836/129 | L_ConvX_2: wDenom |
| C00836/130 | L_ConvX_3: nIn_a |
| C00836/131 | L_ConvX_3: nNum |
| C00836/132 | L_ConvX_3: wDenom |
| C00836/133 | L_ConvPP_1: nNum |
| C00836/134 | L_ConvPP_1: wDenom |
| C00836/135 | L_ConvPP_2: nNum |
| C00836/136 | L_ConvPP_2: wDenom |
| C00836/137 | L_ConvPP_3: nNum |
| C00836/138 | L_ConvPP_3: wDenom |
| C00836/139 | L_Curve_2: nIn_a |
| C00836/140 | L_Curve_3: nIn_a |
| C00836/141 | L_Sequencer_1: wStartStep |
| C00836/142 | L_Sequencer_1: wBranch1 |
| C00836/143 | L_Sequencer_1: wBranch2 |
| C00836/144 | L_Sequencer_1: wMotionState1 |
| C00836/145 | L_Sequencer_1: wMotionState2 |

| Parameter \| Name:<br>**C00836 \| 16-bit inputs (Set2)** | Data type: UNSIGNED_16<br>Index: $23739_d = 5CBB_h$ |
|---|---|
| C00836/146 | L_Sequencer_1: wDigitalInputs |
| C00836/147 | L_ConvActPos_1: nSetPos_a |
| C00836/148 | L_ConvActPos_1: nPosIn_a |
| C00836/149 | L_ConvActPos_1: wVMax |
| C00836/150 | L_ConvActPos_1: nVLine_a |
| C00836/151 | L_MFail_1: nAdapt_a |
| C00836/152 | L_MFail_1: nConst_a |
| C00836/153 | L_MFail_1: nDCSet_a |
| C00836/154 | L_MFail_1: nNSet_a |
| C00836/155 | L_MFail_1: nThreshold_a |
| C00836/156 | L_MFail_1: nNAct_a |
| C00836/157 | L_MFail_1: nSet_a |
| C00836/158 | L_MFail_1: nDCVoltAct_a |
| C00836/159 | L_Curve_3: nCurveCtrl_a |
| C00836/160 | L_Curve_3: nD0_a |
| C00836/161 | L_Curve_3: nDiameter_a |
| C00836/162 | L_SwitchPointPar_1: nActSpeed_v |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | |

**C00837**

| Parameter \| Name:<br>**C00837 \| 16-bit inputs [incr./ms] (Set2)** | | | Data type: INTEGER_16<br>Index: $23738_d = 5CBA_h$ |
|---|---|---|---|
| Display of 16-bit input values of different blocks in [rpm] | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -32767 | Incr./ms | 32767 | |
| **Subcodes** | | | **Info** |
| C00837/1 | | | L_Absolut_2: nIn_a |
| C00837/2 | | | L_AnalogSwitch_4: nIn1_a |
| C00837/3 | | | L_AnalogSwitch_4: nIn2_a |
| C00837/4 | | | L_AnalogSwitch_5: nIn1_a |
| C00837/5 | | | L_AnalogSwitch_5: nIn2_a |
| C00837/6 | | | L_Compare_4: nIn1_a |
| C00837/7 | | | L_Compare_4: nIn2_a |
| C00837/8 | | | L_Compare_5: nIn1_a |
| C00837/9 | | | L_Compare_5: nIn2_a |
| C00837/10 | | | L_Arithmetik_3: nIn1_a |
| C00837/11 | | | L_Arithmetik_3: nIn2_a |
| C00837/12 | | | L_Arithmetik_4: nIn1_a |
| C00837/13 | | | L_Arithmetik_4: nIn2_a |
| C00837/14 | | | L_Arithmetik_5: nIn1_a |
| C00837/15 | | | L_Arithmetik_5: nIn2_a |
| C00837/16 | | | L_Counter_2: wLdVal |
| C00837/17 | | | L_Counter_2: wCmpVal |
| C00837/18 | | | L_Counter_3: wLdVal |

| Parameter \| Name:<br>**C00837 \| 16-bit inputs [incr./ms] (Set2)** | Data type: INTEGER_16<br>Index: $23738_d = 5CBA_h$ |
|---|---|
| C00837/19 | L_Counter_3: wCmpVal |
| C00837/20 | L_PhaseIntK_1: nIn_v |
| C00837/21 | L_Negation_2: nIn_a |
| C00837/22 | L_NLim_2: nIn_a |
| C00837/23 | L_OffsetGain_3: nIn_a |
| C00837/24 | L_OffsetGain_3: nOffset_a |
| C00837/25 | L_OffsetGain_3: nGain_a |
| C00837/26 | L_PT1_2: nIn_a |
| C00837/27 | L_PT1_3: nIn_a |
| C00837/28 | L_PhaseIntK_2: nIn_v |
| C00837/29 | L_SampleHold_1: nIn_a |
| C00837/30 | L_SampleHold_2: nIn_a |
| C00837/31 | L_Mux_1: wInSelect |
| C00837/32 | L_GainOffset_3: nIn_a |
| C00837/33 | L_GainOffset_3: nGain_a |
| C00837/34 | L_GainOffset_3: nOffset_a |
| C00837/35 | L_MulDiv_2: nIn_a |
| C00837/36 | L_DT1_1: nIn_a |
| C00837/37 | L_Counter_1: wLdVal |
| C00837/38 | L_Counter_1: wCmpVal |
| C00837/39 | L_GainOffsetP_1: nIn_a |
| C00837/40 | L_GainOffsetP_2: nIn_a |
| C00837/41 | L_GainOffsetP_3: nIn_a |
| C00837/42 | L_Limit_1: nIn_a |
| C00837/43 | L_Limit_2: nIn_a |
| C00837/44 | L_MckCtrlInterface_1: wOperationMode |
| C00837/45 | L_MckCtrlInterface_1: wPosMode |
| C00837/46 | L_MckCtrlInterface_1: wProfileNo |
| C00837/47 | L_MckCtrlInterface_1: wInMckPosCtrl_1 |
| C00837/48 | L_MckCtrlInterface_1: wInMckPosCtrl_2 |
| C00837/49 | L_MckStateInterface_1: wInMckPosState_1 |
| C00837/50 | L_MckStateInterface_1: wInMckPosState_2 |
| C00837/51 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_1 |
| C00837/52 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_2 |
| C00837/53 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_3 |
| C00837/54 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_4 |
| C00837/55 | L_ConvWordToBits_1: wInput |
| C00837/56 | L_ConvWordToBits_2: wInput |
| C00837/57 | L_ConvWordToBits_3: wInput |
| C00837/58 | L_ConvWordsToDInt_1: wInLWord |
| C00837/59 | L_ConvWordsToDInt_1: wInHWord |
| C00837/60 | L_ConvWordsToDInt_2: wInLWord |
| C00837/61 | L_ConvWordsToDInt_2: wInHWord |
| C00837/62 | L_ConvWordsToDInt_3: wInLWord |

| Parameter \| Name:<br>**C00837 \| 16-bit inputs [incr./ms] (Set2)** | Data type: INTEGER_16<br>Index: $23738_d = 5CBA_h$ |
|---|---|
| C00837/63 | L_ConvWordsToDInt_3: wInHWord |
| C00837/64 | L_ConvUnitsToIncr_1: wInLWord |
| C00837/65 | L_ConvUnitsToIncr_1: wInHWord |
| C00837/66 | L_ConvUnitsToIncr_2: wInLWord |
| C00837/67 | L_ConvUnitsToIncr_2: wInHWord |
| C00837/68 | L_ConvUnitsToIncr_3: wInLWord |
| C00837/69 | L_ConvUnitsToIncr_3: wInHWord |
| C00837/70 | L_Curve_1: nIn_a |
| C00837/71 | L_ConvW_1: wIn |
| C00837/72 | L_ConvW_2: wIn |
| C00837/73 | L_ConvW_3: wIn |
| C00837/74 | L_ConvW_4: wIn |
| C00837/75 | L_MckCtrlInterface_1: wPosSetHW |
| C00837/76 | L_MckCtrlInterface_1: wPosSetLW |
| C00837/77 | L_PhaseDiff_1: nIn_v |
| C00837/78 | L_PhaseDiff_2: nIn_v |
| C00837/79 | L_SRFG_1: nIn_a |
| C00837/80 | L_SRFG_2: nIn_a |
| C00837/81 | L_SRFG_1: nSet_a |
| C00837/82 | L_SRFG_2: nSet_a |
| C00837/83 | L_SignalSwitch_1: wIn1 |
| C00837/84 | L_SignalSwitch_2: wIn1 |
| C00837/85 | L_SignalSwitch_3: wIn1 |
| C00837/86 | L_SignalSwitch_4: wIn1 |
| C00837/87 | L_SignalSwitch_1: wIn2 |
| C00837/88 | L_SignalSwitch_2: wIn2 |
| C00837/89 | L_SignalSwitch_3: wIn2 |
| C00837/90 | L_SignalSwitch_4: wIn2 |
| C00837/91 | L_Odometer_1: nInSpeed_v |
| C00837/92 | L_CalcDiameter_1: wDMax |
| C00837/93 | L_CalcDiameter_1: wDMin |
| C00837/94 | L_CalcDiameter_1: wVMax |
| C00837/95 | L_CalcDiameter_1: nVLine_a |
| C00837/96 | L_CalcDiameter_1: nMotorSpeedAct_v |
| C00837/97 | L_CalcDiameter_1: wGearNum |
| C00837/98 | L_CalcDiameter_1: wGearDenom |
| C00837/99 | L_CalcDiameter_1: nSetD_a |
| C00837/100 | L_ProcessCtrl_1: nVpAdapt_a |
| C00837/101 | L_ProcessCtrl_1: nSet_a |
| C00837/102 | L_ProcessCtrl_1: nAct_a |
| C00837/103 | L_ProcessCtrl_1: nRTimeAdapt_a |
| C00837/104 | L_ProcessCtrl_1: nInfluence_a |
| C00837/105 | L_PhiIntegrator_1: nIn_v |
| C00837/106 | L_PhiIntegrator_1: wGearNum |

| Parameter \| Name:<br>**C00837 \| 16-bit inputs [incr./ms] (Set2)** | Data type: INTEGER_16<br>Index: $23738_d = 5CBA_h$ |
|---|---|
| C00837/107 | L_PhiIntegrator_1: wGearDenom |
| C00837/108 | L_SwitchPoint_1: nActSpeed_v |
| C00837/109 | L_PhiIntegrator_1: nSpeedAdd_v |
| C00837/110 | L_DFSET_1: nSpeedTrim_v |
| C00837/111 | L_DFSET_1: nSpeedTrim_a |
| C00837/112 | L_DFSET_1: wGearNum |
| C00837/113 | L_DFSET_1: wGainNum |
| C00837/114 | L_DFSET_1: nSet_v |
| C00837/115 | L_DFSET_1: wGainDenom |
| C00837/116 | L_DFSET_1: wGearDenom |
| C00837/117 | L_DFSET_1: nPositionTrimming |
| C00837/118 | L_DFSET_1: nNAct_v |
| C00837/119 | L_DFRFG_1: nIn_v |
| C00837/120 | L_GearComp_1: nTorque_a |
| C00837/121 | L_ConvAP_1: nIn_a |
| C00837/122 | L_ConvAP_2: nIn_a |
| C00837/123 | L_ConvAP_3: nIn_a |
| C00837/124 | L_ConvX_1: nIn_a |
| C00837/125 | L_ConvX_1: nNum |
| C00837/126 | L_ConvX_1: wDenom |
| C00837/127 | L_ConvX_2: nIn_a |
| C00837/128 | L_ConvX_2: nNum |
| C00837/129 | L_ConvX_2: wDenom |
| C00837/130 | L_ConvX_3: nIn_a |
| C00837/131 | L_ConvX_3: nNum |
| C00837/132 | L_ConvX_3: wDenom |
| C00837/133 | L_ConvPP_1: nNum |
| C00837/134 | L_ConvPP_1: wDenom |
| C00837/135 | L_ConvPP_2: nNum |
| C00837/136 | L_ConvPP_2: wDenom |
| C00837/137 | L_ConvPP_3: nNum |
| C00837/138 | L_ConvPP_3: wDenom |
| C00837/139 | L_Curve_2: nIn_a |
| C00837/140 | L_Curve_3: nIn_a |
| C00837/141 | L_Sequencer_1: wStartStep |
| C00837/142 | L_Sequencer_1: wBranch1 |
| C00837/143 | L_Sequencer_1: wBranch2 |
| C00837/144 | L_Sequencer_1: wMotionState1 |
| C00837/145 | L_Sequencer_1: wMotionState2 |
| C00837/146 | L_Sequencer_1: wDigitalInputs |
| C00837/147 | L_ConvActPos_1: nSetPos_a |
| C00837/148 | L_ConvActPos_1: nPosIn_a |
| C00837/149 | L_ConvActPos_1: wVMax |
| C00837/150 | L_ConvActPos_1: nVLine_a |

| Parameter \| Name:<br>**C00837 \| 16-bit inputs [incr./ms] (Set2)** | Data type: INTEGER_16<br>Index: $23738_d = 5CBA_h$ |
|---|---|
| C00837/151 | L_MFail_1: nAdapt_a |
| C00837/152 | L_MFail_1: nConst_a |
| C00837/153 | L_MFail_1: nDCSet_a |
| C00837/154 | L_MFail_1: nNSet_a |
| C00837/155 | L_MFail_1: nThreshold_a |
| C00837/156 | L_MFail_1: nNAct_a |
| C00837/157 | L_MFail_1: nSet_a |
| C00837/158 | L_MFail_1: nDCVoltAct_a |
| C00837/159 | L_Curve_3: nCurveCtrl_a |
| C00837/160 | L_Curve_3: nD0_a |
| C00837/161 | L_Curve_3: nDiameter_a |
| C00837/162 | L_SwitchPointPar_1: nActSpeed_v |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00838**

| Parameter \| Name:<br>**C00838 \| Binary inputs (Set2)** | Data type: UNSIGNED_8<br>Index: $23737_d = 5CB9_h$ |
|---|---|
| Display of the signal status of the binary inputs of different blocks<br>• In addition to the parameter C00833. | |
| **Selection list** | |
| 0 \| False | |
| 1 \| True | |
| **Subcodes** | **Info** |
| C00838/1 | L_And5_1: bIn1 |
| C00838/2 | L_And5_1: bIn2 |
| C00838/3 | L_And5_1: bIn3 |
| C00838/4 | L_And5_1: bIn4 |
| C00838/5 | L_And5_1: bIn5 |
| C00838/6 | L_And5_2: bIn1 |
| C00838/7 | L_And5_2: bIn2 |
| C00838/8 | L_And5_2: bIn3 |
| C00838/9 | L_And5_2: bIn4 |
| C00838/10 | L_And5_2: bIn5 |
| C00838/11 | L_AnalogSwitch_4: bSet |
| C00838/12 | L_AnalogSwitch_5: bSet |
| C00838/13 | L_DFlipFlop_2: bD |
| C00838/14 | L_DFlipFlop_2: bClk |
| C00838/15 | L_DFlipFlop_2: bClr |
| C00838/16 | L_DigitalDelay_2: bIn |
| C00838/17 | L_DigitalDelay_3: bIn |
| C00838/18 | L_PhaseIntK_1: bLoad |
| C00838/19 | L_PhaseIntK_2: bLoad |

| Parameter \| Name:<br>**C00838 \| Binary inputs (Set2)** | Data type: UNSIGNED_8<br>Index: $23737_d = 5CB9_h$ |
|---|---|
| C00838/20 | L_DigitalLogic_2: bIn1 |
| C00838/21 | L_DigitalLogic_2: bIn2 |
| C00838/22 | L_DigitalLogic_2: bIn3 |
| C00838/23 | L_DigitalLogic5_1: bIn1 |
| C00838/24 | L_DigitalLogic5_1: bIn2 |
| C00838/25 | L_DigitalLogic5_1: bIn3 |
| C00838/26 | L_DigitalLogic5_1: bIn4 |
| C00838/27 | L_DigitalLogic5_1: bIn5 |
| C00838/28 | L_DigitalLogic5_2: bIn1 |
| C00838/29 | L_DigitalLogic5_2: bIn2 |
| C00838/30 | L_DigitalLogic5_2: bIn3 |
| C00838/31 | L_DigitalLogic5_2: bIn4 |
| C00838/32 | L_DigitalLogic5_2: bIn5 |
| C00838/33 | L_NLim_2: bEnable |
| C00838/34 | L_Or5_1: bIn1 |
| C00838/35 | L_Or5_1: bIn2 |
| C00838/36 | L_Or5_1: bIn3 |
| C00838/37 | L_Or5_1: bIn4 |
| C00838/38 | L_Or5_1: bIn5 |
| C00838/39 | L_Or5_2: bIn1 |
| C00838/40 | L_Or5_2: bIn2 |
| C00838/41 | L_Or5_2: bIn3 |
| C00838/42 | L_Or5_2: bIn4 |
| C00838/43 | L_Or5_2: bIn5 |
| C00838/44 | L_Not_4: bIn |
| C00838/45 | L_Not_5: bIn |
| C00838/46 | L_Not_6: bIn |
| C00838/47 | L_Not_7: bIn |
| C00838/48 | L_RSFlipFlop_1: bSet |
| C00838/49 | L_RSFlipFlop_1: bReset |
| C00838/50 | L_RSFlipFlop_2: bSet |
| C00838/51 | L_RSFlipFlop_2: bReset |
| C00838/52 | L_SampleHold_1: bLoad |
| C00838/53 | L_SampleHold_2: bLoad |
| C00838/54 | L_Counter_2: bClkUp |
| C00838/55 | L_Counter_2: bClkDown |
| C00838/56 | L_Counter_2: bLoad |
| C00838/57 | L_Counter_3: bClkUp |
| C00838/58 | L_Counter_3: bClkDown |
| C00838/59 | L_Counter_3: bLoad |
| C00838/60 | L_Transient_5: bIn |
| C00838/61 | L_Transient_6: bIn |
| C00838/62 | L_Transient_7: bIn |
| C00838/63 | L_Transient_8: bIn |

| Parameter \| Name:<br>**C00838 \| Binary inputs (Set2)** | Data type: UNSIGNED_8<br>Index: $23737_d = 5CB9_h$ |
|---|---|
| C00838/64 | L_Counter_1: bCountUp |
| C00838/65 | L_Counter_1: bCountDown |
| C00838/66 | L_Counter_1: bLoad |
| C00838/67 | L_ConvBitsToWord_1: bBit0 |
| C00838/68 | L_ConvBitsToWord_1: bBit1 |
| C00838/69 | L_ConvBitsToWord_1: bBit2 |
| C00838/70 | L_ConvBitsToWord_1: bBit3 |
| C00838/71 | L_ConvBitsToWord_1: bBit4 |
| C00838/72 | L_ConvBitsToWord_1: bBit5 |
| C00838/73 | L_ConvBitsToWord_1: bBit6 |
| C00838/74 | L_ConvBitsToWord_1: bBit7 |
| C00838/75 | L_ConvBitsToWord_1: bBit8 |
| C00838/76 | L_ConvBitsToWord_1: bBit9 |
| C00838/77 | L_ConvBitsToWord_1: bBit10 |
| C00838/78 | L_ConvBitsToWord_1: bBit11 |
| C00838/79 | L_ConvBitsToWord_1: bBit12 |
| C00838/80 | L_ConvBitsToWord_1: bBit13 |
| C00838/81 | L_ConvBitsToWord_1: bBit14 |
| C00838/82 | L_ConvBitsToWord_1: bBit15 |
| C00838/83 | L_ConvBitsToWord_2: bBit0 |
| C00838/84 | L_ConvBitsToWord_2: bBit1 |
| C00838/85 | L_ConvBitsToWord_2: bBit2 |
| C00838/86 | L_ConvBitsToWord_2: bBit3 |
| C00838/87 | L_ConvBitsToWord_2: bBit4 |
| C00838/88 | L_ConvBitsToWord_2: bBit5 |
| C00838/89 | L_ConvBitsToWord_2: bBit6 |
| C00838/90 | L_ConvBitsToWord_2: bBit7 |
| C00838/91 | L_ConvBitsToWord_2: bBit8 |
| C00838/92 | L_ConvBitsToWord_2: bBit9 |
| C00838/93 | L_ConvBitsToWord_2: bBit10 |
| C00838/94 | L_ConvBitsToWord_2: bBit11 |
| C00838/95 | L_ConvBitsToWord_2: bBit12 |
| C00838/96 | L_ConvBitsToWord_2: bBit13 |
| C00838/97 | L_ConvBitsToWord_2: bBit14 |
| C00838/98 | L_ConvBitsToWord_2: bBit15 |
| C00838/99 | L_ConvBitsToWord_3: bBit0 |
| C00838/100 | L_ConvBitsToWord_3: bBit1 |
| C00838/101 | L_ConvBitsToWord_3: bBit2 |
| C00838/102 | L_ConvBitsToWord_3: bBit3 |
| C00838/103 | L_ConvBitsToWord_3: bBit4 |
| C00838/104 | L_ConvBitsToWord_3: bBit5 |
| C00838/105 | L_ConvBitsToWord_3: bBit6 |
| C00838/106 | L_ConvBitsToWord_3: bBit7 |
| C00838/107 | L_ConvBitsToWord_3: bBit8 |

| Parameter \| Name:<br>**C00838 \| Binary inputs (Set2)** | Data type: UNSIGNED_8<br>Index: $23737_d = 5CB9_h$ |
|---|---|
| C00838/108 | L_ConvBitsToWord_3: bBit9 |
| C00838/109 | L_ConvBitsToWord_3: bBit10 |
| C00838/110 | L_ConvBitsToWord_3: bBit11 |
| C00838/111 | L_ConvBitsToWord_3: bBit12 |
| C00838/112 | L_ConvBitsToWord_3: bBit13 |
| C00838/113 | L_ConvBitsToWord_3: bBit14 |
| C00838/114 | L_ConvBitsToWord_3: bBit15 |
| C00838/115 | L_MckCtrlInterface_1: bManJogPos |
| C00838/116 | L_MckCtrlInterface_1: bManJogNeg |
| C00838/117 | L_MckCtrlInterface_1: bManJogExecute2ndVel |
| C00838/118 | L_MckCtrlInterface_1: bReleaseLimitSwitch |
| C00838/119 | L_MckCtrlInterface_1: bHomingStartStop |
| C00838/120 | L_MckCtrlInterface_1: bHomingSetPos |
| C00838/121 | L_MckCtrlInterface_1: bHomingResetPos |
| C00838/122 | L_MckCtrlInterface_1: bEnableVelOverride |
| C00838/123 | L_MckCtrlInterface_1: bEnableAccOverride |
| C00838/124 | L_MckCtrlInterface_1: bDisableSShaping |
| C00838/125 | L_MckCtrlInterface_1: bPosExecute |
| C00838/126 | L_MckCtrlInterface_1: bPosExecuteFinish |
| C00838/127 | L_MckCtrlInterface_1: bPosDisableFollowProfile |
| C00838/128 | L_MckCtrlInterface_1: bPosStop |
| C00838/129 | L_MckCtrlInterface_1: bPosTeachSetPos |
| C00838/130 | L_MckCtrlInterface_1: bPosTeachActPos |
| C00838/131 | L_MckCtrlInterface_1: bProfileNo_1 |
| C00838/132 | L_MckCtrlInterface_1: bProfileNo_2 |
| C00838/133 | L_MckCtrlInterface_1: bProfileNo_4 |
| C00838/134 | L_MckCtrlInterface_1: bProfileNo_8 |
| C00838/135 | L_MckCtrlInterface_1: bOperationMode_1 |
| C00838/136 | L_MckCtrlInterface_1: bOperationMode_2 |
| C00838/137 | L_MckCtrlInterface_1: bOperationMode_4 |
| C00838/138 | L_MckCtrlInterface_1: bOperationMode_8 |
| C00838/139 | L_PhaseDiff_1: bEnable |
| C00838/140 | L_PhaseDiff_2: bEnable |
| C00838/141 | L_PhaseDiff_1: bReset |
| C00838/142 | L_PhaseDiff_2: bReset |
| C00838/143 | L_SRFG_1: bLoad |
| C00838/144 | L_SRFG_2: bLoad |
| C00838/145 | L_SignalSwitch_1: bSet |
| C00838/146 | L_SignalSwitch_2: bSet |
| C00838/147 | L_SignalSwitch_3: bSet |
| C00838/148 | L_SignalSwitch_4: bSet |
| C00838/149 | L_Odometer_1: bTriggerPulse |
| C00838/150 | L_Odometer_1: bReset |
| C00838/151 | L_FixSet_a_1: bSelect1 |

| Parameter \| Name:<br>**C00838 \| Binary inputs (Set2)** | Data type: UNSIGNED_8<br>Index: $23737_d = 5CB9_h$ |
|---|---|
| C00838/152 | L_FixSet_a_1: bSelect2 |
| C00838/153 | L_FixSet_a_1: bSelect4 |
| C00838/154 | L_FixSet_a_1: bSelect8 |
| C00838/155 | L_FixSet_w_1: bSelect1 |
| C00838/156 | L_FixSet_w_1: bSelect2 |
| C00838/157 | L_FixSet_w_1: bSelect4 |
| C00838/158 | L_FixSet_w_1: bSelect8 |
| C00838/159 | L_FixSet_w_2: bSelect1 |
| C00838/160 | L_FixSet_w_2: bSelect2 |
| C00838/161 | L_FixSet_w_2: bSelect4 |
| C00838/162 | L_FixSet_w_2: bSelect8 |
| C00838/163 | L_CalcDiameter_1: bResetPos |
| C00838/164 | L_CalcDiameter_1: bHoldD |
| C00838/165 | L_CalcDiameter_1: bUnidirect |
| C00838/166 | L_CalcDiameter_1: bUnwind |
| C00838/167 | L_CalcDiameter_1: bLoadDiameter |
| C00838/168 | L_CalcDiameter_1: bCalcRef |
| C00838/169 | L_ProcessCtrl_1: bLoadAct |
| C00838/170 | L_ProcessCtrl_1: bIOff |
| C00838/171 | L_ProcessCtrl_1: bReset |
| C00838/172 | L_PhiIntegrator_1: bTPReceived |
| C00838/173 | L_PhiIntegrator_1: bReset |
| C00838/174 | L_PhiIntegrator_1: bLoad |
| C00838/175 | L_PosCtrlLin_1: bExecute |
| C00838/176 | L_PosCtrlLin_1: bSetPos0 |
| C00838/177 | L_PosCtrlLin_1: bPosMode |
| C00838/178 | L_PosCtrlLin_1: bEnable |
| C00838/179 | L_PosCtrlLin_2: bExecute |
| C00838/180 | L_PosCtrlLin_2: bSetPos0 |
| C00838/181 | L_PosCtrlLin_2: bPosMode |
| C00838/182 | L_PosCtrlLin_2: bEnable |
| C00838/183 | L_SwitchPoint_1: bDisable |
| C00838/184 | L_DFSET_1: bZeroPulse |
| C00838/185 | L_DFSET_1: bSetTPReceived |
| C00838/186 | L_DFSET_1: bActTPReceived |
| C00838/187 | L_DFSET_1: bSetActIntegrator |
| C00838/188 | L_DFSET_1: bResetAllIntegrators |
| C00838/189 | L_DFRFG_1: bSetTPReceived |
| C00838/190 | L_DFRFG_1: bRfg0 |
| C00838/191 | L_DFRFG_1: bRfgStop |
| C00838/192 | L_DFRFG_1: bReset |
| C00838/193 | L_ConvX_1: bInvers |
| C00838/194 | L_ConvX_2: bInvers |
| C00838/195 | L_ConvX_3: bInvers |

| Parameter \| Name:<br>**C00838 \| Binary inputs (Set2)** | Data type: UNSIGNED_8<br>Index: $23737_d = 5CB9_h$ |
|---|---|
| C00838/196 | L_ConvPP_1: bAct |
| C00838/197 | L_ConvPP_2: bAct |
| C00838/198 | L_ConvPP_3: bAct |
| C00838/199 | L_SignalSwitch32_1: bSet |
| C00838/200 | L_SignalSwitch32_2: bSet |
| C00838/201 | L_SignalSwitch32_3: bSet |
| C00838/202 | L_Sequencer_1: bStart |
| C00838/203 | L_Sequencer_1: bPause |
| C00838/204 | L_Sequencer_1: bNextStep |
| C00838/205 | L_Sequencer_1: bCancel |
| C00838/206 | L_Sequencer_1: bBreak |
| C00838/207 | L_Sequencer_1: bReset |
| C00838/208 | L_ConvActPos_1: bWriteMaxPos |
| C00838/209 | L_ConvActPos_1: bWriteMinPos |
| C00838/210 | L_MFail_1: bFault |
| C00838/211 | L_MFail_1: bReset |
| C00838/212 | L_Curve_3: bEnableTensionCurve |
| C00838/213 | L_SwitchPointPar_1: bDisable |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00839**

| Parameter \| Name:<br>**C00839 \| 32-bit inputs [incr] (Set2)** | | | Data type: INTEGER_32<br>Index: $23736_d = 5CB8_h$ |
|---|---|---|---|
| Display in [increments] of 32 bit input values of various blocks<br>• In addition to the parameter C00834. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00839/1 | | | L_ComparePhi_1: dnIn1_p |
| C00839/2 | | | L_ComparePhi_1: dnIn2_p |
| C00839/3 | | | L_ComparePhi_2: dnIn1_p |
| C00839/4 | | | L_ComparePhi_2: dnIn2_p |
| C00839/5 | | | L_ComparePhi_3: dnIn1_p |
| C00839/6 | | | L_ComparePhi_3: dnIn2_p |
| C00839/7 | | | L_ComparePhi_4: dnIn1_p |
| C00839/8 | | | L_ComparePhi_4: dnIn2_p |
| C00839/9 | | | L_ComparePhi_5: dnIn1_p |
| C00839/10 | | | L_ComparePhi_5: dnIn2_p |
| C00839/11 | | | L_ArithmetikPhi_1: dnIn1_p |
| C00839/12 | | | L_ArithmetikPhi_1: dnIn2_p |
| C00839/13 | | | L_ArithmetikPhi_2: dnIn1_p |
| C00839/14 | | | L_ArithmetikPhi_2: dnIn2_p |
| C00839/15 | | | L_ArithmetikPhi_3: dnIn1_p |
| C00839/16 | | | L_ArithmetikPhi_3: dnIn2_p |

| Parameter \| Name:<br>**C00839 \| 32-bit inputs [incr] (Set2)** | Data type: INTEGER_32<br>Index: $23736_d = 5CB8_h$ |
|---|---|
| C00839/17 | L_GainOffsetPhiP_1: dnIn_p |
| C00839/18 | L_GainOffsetPhiP_2: dnIn_p |
| C00839/19 | L_LimitPhi_1: dnIn_p |
| C00839/20 | L_LimitPhi_2: dnIn_p |
| C00839/21 | L_LimitPhi_3: dnIn_p |
| C00839/22 | L_OffsetGainPhiP_1: dnIn_p |
| C00839/23 | L_OffsetGainPhiP_2: dnIn_p |
| C00839/24 | L_PhaseIntK_1: dnSet_p |
| C00839/25 | L_PhaseIntK_2: dnSet_p |
| C00839/26 | L_Mux_1: dnInput1_p |
| C00839/27 | L_Mux_1: dnInput2_p |
| C00839/28 | L_Mux_1: dnInput3_p |
| C00839/29 | L_Mux_1: dnInput4_p |
| C00839/30 | L_Mux_1: dnInput5_p |
| C00839/31 | L_Mux_1: dnInput6_p |
| C00839/32 | L_Mux_1: dnInput7_p |
| C00839/33 | L_Mux_1: dnInput8_p |
| C00839/34 | L_SQrt_1: dnInput_p |
| C00839/35 | L_ConvDIntToWords_1: dnInput_p |
| C00839/36 | L_ConvDIntToWords_2: dnInput_p |
| C00839/37 | L_ConvDIntToWords_3: dnInput_p |
| C00839/38 | L_MckCtrlInterface_1: dnPosSetIn_p |
| C00839/39 | L_PhaseDiff_1: dnSet_p |
| C00839/40 | L_PhaseDiff_2: dnSet_p |
| C00839/41 | L_PhaseDiff_1: dnAdd_p |
| C00839/42 | L_PhaseDiff_2: dnAdd_p |
| C00839/43 | L_MckStateInterface_1: dnPosIn_p |
| C00839/44 | L_Odometer_1: dnInPosition_p |
| C00839/45 | L_PhiIntegrator_1: dnTPPosition_p |
| C00839/46 | L_PhiIntegrator_1: dnLoadVal_p |
| C00839/47 | L_PosCtrlLin_1: dnSet_p |
| C00839/48 | L_PosCtrlLin_1: dnAct_p |
| C00839/49 | L_PosCtrlLin_2: dnSet_p |
| C00839/50 | L_PosCtrlLin_2: dnAct_p |
| C00839/51 | L_SwitchPoint_1: dnActPos_p |
| C00839/52 | L_SwitchPoint_1: dnSwitchPoint1_p |
| C00839/53 | L_SwitchPoint_1: dn2ndPoint_Size1_p |
| C00839/54 | L_SwitchPoint_1: dnSwitchPoint2_p |
| C00839/55 | L_SwitchPoint_1: dn2ndPoint_Size2_p |
| C00839/56 | L_SwitchPoint_1: dnSwitchPoint3_p |
| C00839/57 | L_SwitchPoint_1: dn2ndPoint_Size3_p |
| C00839/58 | L_SwitchPoint_1: dnSwitchPoint4_p |
| C00839/59 | L_SwitchPoint_1: dn2ndPoint_Size4_p |
| C00839/60 | L_DFSET_1: dnSetTPPos_p |

| Parameter \| Name: **C00839 \| 32-bit inputs [incr] (Set2)** | Data type: INTEGER_32 Index: $23736_d = 5CB8_h$ |
|---|---|
| C00839/61 | L_DFSET_1: dnActTPPos_p |
| C00839/62 | L_DFSET_1: dnPosOffset |
| C00839/63 | L_DFRFG_1: dnSetTPPos_p |
| C00839/64 | L_GearComp_1: dnPhiIn_p |
| C00839/65 | L_ConvPA_1: dnIn_p |
| C00839/66 | L_ConvPA_2: dnIn_p |
| C00839/67 | L_ConvPA_3: dnIn_p |
| C00839/68 | L_ConvPP_1: dnIn_p |
| C00839/69 | L_ConvPP_2: dnIn_p |
| C00839/70 | L_ConvPP_3: dnIn_p |
| C00839/71 | L_SignalSwitch32_1: dnIn1 |
| C00839/72 | L_SignalSwitch32_1: dnIn2 |
| C00839/73 | L_SignalSwitch32_2: dnIn1 |
| C00839/74 | L_SignalSwitch32_2: dnIn2 |
| C00839/75 | L_SignalSwitch32_3: dnIn1 |
| C00839/76 | L_SignalSwitch32_3: dnIn2 |
| C00839/77 | L_ArithmetikPhi_4: adnIn1_p |
| C00839/78 | L_ArithmetikPhi_4: adnIn2_p |
| C00839/79 | L_ArithmetikPhi_5: adnIn1_p |
| C00839/80 | L_ArithmetikPhi_5: adnIn2_p |
| C00839/81 | L_ArithmetikPhi_6: adnIn1_p |
| C00839/82 | L_ArithmetikPhi_6: adnIn2_p |
| C00839/83 | L_PhiDiv_1: dnIn_p |
| C00839/84 | L_PhiAdd_1: dnIn1_p |
| C00839/85 | L_PhiAdd_1: dnIn2_p |
| C00839/86 | L_PhiAdd_1: dnIn3_p |
| C00839/87 | L_DFSET_1: dnDeltaPosIn_p |
| C00839/88 | L_SwitchPoint_1: dnSwitchPoint5_p |
| C00839/89 | L_SwitchPoint_1: dn2ndPoint_Size5_p |
| C00839/90 | L_SwitchPoint_1: dnSwitchPoint6_p |
| C00839/91 | L_SwitchPoint_1: dn2ndPoint_Size6_p |
| C00839/92 | L_SwitchPoint_1: dnSwitchPoint7_p |
| C00839/93 | L_SwitchPoint_1: dn2ndPoint_Size7_p |
| C00839/94 | L_SwitchPoint_1: dnSwitchPoint8_p |
| C00839/95 | L_SwitchPoint_1: dn2ndPoint_Size8_p |
| C00839/96 | L_SwitchPointPar_1: dnActPos_p |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00840**

| Parameter \| Name: **C00840 \| 16-bit inputs I/O level [%]** | | | Data type: INTEGER_16 Index: $23735_d = 5CB7_h$ |
|---|---|---|---|
| Display in percent of 16 bit input values of various blocks of the I/O level | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |

| Parameter \| Name:<br>**C00840 \| 16-bit inputs I/O level [%]** | Data type: INTEGER_16<br>Index: $23735_d = 5CB7_h$ |
|---|---|
| **Subcodes** | **Info** |
| C00840/1 | LS_AnalogOutput: nOut1_a (V) |
| C00840/2 | LP_CanOut1: wState |
| C00840/3 | LP_CanOut1: wOut2 |
| C00840/4 | LP_CanOut1: wOut3 |
| C00840/5 | LP_CanOut1: wOut4 |
| C00840/6 | LP_CanOut2: wOut1 |
| C00840/7 | LP_CanOut2: wOut2 |
| C00840/8 | LP_CanOut2: wOut3 |
| C00840/9 | LP_CanOut2: wOut4 |
| C00840/10 | LP_CanOut3: wOut1 |
| C00840/11 | LP_CanOut3: wOut2 |
| C00840/12 | LP_CanOut3: wOut3 |
| C00840/13 | LP_CanOut3: wOut4 |
| C00840/14 | LS_DisFree_a: nDis1_a |
| C00840/15 | LS_DisFree_a: nDis2_a |
| C00840/16 | LS_DisFree_a: nDis3_a |
| C00840/17 | LS_DisFree_a: nDis4_a |
| C00840/18 | LS_DisFree: wDis1 |
| C00840/19 | LS_DisFree: wDis2 |
| C00840/20 | LS_DisFree: wDis3 |
| C00840/21 | LS_DisFree: wDis4 |
| C00840/22 | LP_MciOut: wState |
| C00840/23 | LP_MciOut: wOut2 |
| C00840/24 | LP_MciOut: wOut3 |
| C00840/25 | LP_MciOut: wOut4 |
| C00840/26 | LP_MciOut: wOut5 |
| C00840/27 | LP_MciOut: wOut6 |
| C00840/28 | LP_MciOut: wOut7 |
| C00840/29 | LP_MciOut: wOut8 |
| C00840/30 | LP_MciOut: wOut9 |
| C00840/31 | LP_MciOut: wOut10 |
| C00840/32 | LP_MciOut: wOut11 |
| C00840/33 | LP_MciOut: wOut12 |
| C00840/34 | LP_MciOut: wOut13 |
| C00840/35 | LP_MciOut: wOut14 |
| C00840/36 | LP_MciOut: wOut15 |
| C00840/37 | LP_MciOut: wOut16 |
| C00840/38 | LS_AnalogOutput: nOut2_a (V) |
| C00840/39 | LS_AnalogOutput: nOut1_a (I) |
| C00840/40 | LS_AnalogOutput: nOut2_a (I) |
| C00840/41 | LS_DisFree_a: nDis5_a |
| C00840/42 | LS_DisFree_a: nDis6_a |
| C00840/43 | LS_DisFree_a: nDis7_a |

| Parameter \| Name:<br>**C00840 \| 16-bit inputs I/O level [%]** | Data type: INTEGER_16<br>Index: $23735_d = 5CB7_h$ |
|---|---|
| C00840/44 | LS_DisFree_a: nDis8_a |
| C00840/45 | LS_DisFree: wDis5 |
| C00840/46 | LS_DisFree: wDis6 |
| C00840/47 | LS_DisFree: wDis7 |
| C00840/48 | LS_DisFree: wDis8 |
| C00840/49 | LS_ParReadWrite_1: wParIndex |
| C00840/50 | LS_ParReadWrite_1: wParSubindex |
| C00840/51 | LS_ParReadWrite_1: wInHWord |
| C00840/52 | LS_ParReadWrite_1: wInLWord |
| C00840/53 | LS_ParReadWrite_2: wParIndex |
| C00840/54 | LS_ParReadWrite_2: wParSubindex |
| C00840/55 | LS_ParReadWrite_2: wInHWord |
| C00840/56 | LS_ParReadWrite_2: wInLWord |
| C00840/57 | LS_ParReadWrite_3: wParIndex |
| C00840/58 | LS_ParReadWrite_3: wParSubindex |
| C00840/59 | LS_ParReadWrite_3: wInHWord |
| C00840/60 | LS_ParReadWrite_3: wInLWord |
| C00840/61 | LS_ParReadWrite_4: wParIndex |
| C00840/62 | LS_ParReadWrite_4: wParSubindex |
| C00840/63 | LS_ParReadWrite_4: wInHWord |
| C00840/64 | LS_ParReadWrite_4: wInLWord |
| C00840/65 | LS_ParReadWrite_5: wParIndex |
| C00840/66 | LS_ParReadWrite_5: wParSubindex |
| C00840/67 | LS_ParReadWrite_5: wInHWord |
| C00840/68 | LS_ParReadWrite_5: wInLWord |
| C00840/69 | LS_ParReadWrite_6: wParIndex |
| C00840/70 | LS_ParReadWrite_6: wParSubindex |
| C00840/71 | LS_ParReadWrite_6: wInHWord |
| C00840/72 | LS_ParReadWrite_6: wInLWord |
| C00840/73 | Reserved |
| C00840/74 | Reserved |
| C00840/75 | Reserved |
| C00840/76 | Reserved |
| C00840/77 | Reserved |
| C00840/78 | Reserved |
| C00840/79 | Reserved |
| C00840/80 | Reserved |
| C00840/81 | Reserved |
| C00840/82 | Reserved |
| C00840/83 | Reserved |
| C00840/84 | Reserved |
| C00840/85 | Reserved |
| C00840/86 | Reserved |
| C00840/87 | Reserved |

| Parameter \| Name:<br>**C00840 \| 16-bit inputs I/O level [%]** | Data type: INTEGER_16<br>Index: $23735_d = 5CB7_h$ |
|---|---|
| C00840/88 | Reserved |
| C00840/89 | LS_RetainData: wIn1 |
| C00840/90 | LS_RetainData: wIn2 |
| C00840/91 | LS_RetainData: wIn3 |
| C00840/92 | LS_RetainData: wIn4 |
| C00840/93 | LS_AxisBusOut: wLine1 |
| C00840/94 | LS_AxisBusOut: wLine2 |
| C00840/95 | LS_AxisBusOut: wLine3 |
| C00840/96 | LS_AxisBusOut: wCas1 |
| C00840/97 | LS_AxisBusOut: wCas2 |
| C00840/98 | LS_AxisBusOut: wCas3 |
| C00840/99 | LS_AxisBusOut: wCas4 |
| C00840/100 | LS_AxisBusAux: wAuxOut1 |
| C00840/101 | LS_AxisBusAux: wAuxOut2 |
| C00840/102 | LS_AxisBusAux: wAuxOut3 |
| C00840/103 | LS_AxisBusAux: wAuxOut4 |
| C00840/104 | LS_AxisBusAux: wSelectSlave |
| C00840/105 | LS_DFOut: nOut_v |
| C00840/106 | LS_MultiEncoder: wActPosExternalHW |
| C00840/107 | LS_MultiEncoder: wActPosExternalLW |
| C00840/108 | LP_CanOut4: wOut1 |
| C00840/109 | LP_CanOut4: wOut2 |
| C00840/110 | LP_CanOut4: wOut3 |
| C00840/111 | LP_CanOut4: wOut4 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | |

**C00841**

| Parameter \| Name:<br>**C00841 \| 16-bit inputs I/O level** | | | Data type: UNSIGNED_16<br>Index: $23734_d = 5CB6_h$ |
|---|---|---|---|
| Decimal/hexadecimal/bit coded display of 16 bit input values of various blocks of the I/O level | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00841/1 | | | LS_AnalogOutput: nOut1_a (V) |
| C00841/2 | | | LP_CanOut1: wState |
| C00841/3 | | | LP_CanOut1: wOut2 |
| C00841/4 | | | LP_CanOut1: wOut3 |
| C00841/5 | | | LP_CanOut1: wOut4 |
| C00841/6 | | | LP_CanOut2: wOut1 |
| C00841/7 | | | LP_CanOut2: wOut2 |

| Parameter \| Name:<br>**C00841 \| 16-bit inputs I/O level** | Data type: UNSIGNED_16<br>Index: $23734_d = 5CB6_h$ |
|---|---|
| C00841/8 | LP_CanOut2: wOut3 |
| C00841/9 | LP_CanOut2: wOut4 |
| C00841/10 | LP_CanOut3: wOut1 |
| C00841/11 | LP_CanOut3: wOut2 |
| C00841/12 | LP_CanOut3: wOut3 |
| C00841/13 | LP_CanOut3: wOut4 |
| C00841/14 | LS_DisFree_a: nDis1_a |
| C00841/15 | LS_DisFree_a: nDis2_a |
| C00841/16 | LS_DisFree_a: nDis3_a |
| C00841/17 | LS_DisFree_a: nDis4_a |
| C00841/18 | LS_DisFree: wDis1 |
| C00841/19 | LS_DisFree: wDis2 |
| C00841/20 | LS_DisFree: wDis3 |
| C00841/21 | LS_DisFree: wDis4 |
| C00841/22 | LP_MciOut: wState |
| C00841/23 | LP_MciOut: wOut2 |
| C00841/24 | LP_MciOut: wOut3 |
| C00841/25 | LP_MciOut: wOut4 |
| C00841/26 | LP_MciOut: wOut5 |
| C00841/27 | LP_MciOut: wOut6 |
| C00841/28 | LP_MciOut: wOut7 |
| C00841/29 | LP_MciOut: wOut8 |
| C00841/30 | LP_MciOut: wOut9 |
| C00841/31 | LP_MciOut: wOut10 |
| C00841/32 | LP_MciOut: wOut11 |
| C00841/33 | LP_MciOut: wOut12 |
| C00841/34 | LP_MciOut: wOut13 |
| C00841/35 | LP_MciOut: wOut14 |
| C00841/36 | LP_MciOut: wOut15 |
| C00841/37 | LP_MciOut: wOut16 |
| C00841/38 | LS_AnalogOutput: nOut2_a (V) |
| C00841/39 | LS_AnalogOutput: nOut1_a (I) |
| C00841/40 | LS_AnalogOutput: nOut2_a (I) |
| C00841/41 | LS_DisFree_a: nDis5_a |
| C00841/42 | LS_DisFree_a: nDis6_a |
| C00841/43 | LS_DisFree_a: nDis7_a |
| C00841/44 | LS_DisFree_a: nDis8_a |
| C00841/45 | LS_DisFree: wDis5 |
| C00841/46 | LS_DisFree: wDis6 |
| C00841/47 | LS_DisFree: wDis7 |
| C00841/48 | LS_DisFree: wDis8 |
| C00841/49 | LS_ParReadWrite_1: wParIndex |
| C00841/50 | LS_ParReadWrite_1: wParSubindex |
| C00841/51 | LS_ParReadWrite_1: wInHWord |

| Parameter \| Name:<br>**C00841 \| 16-bit inputs I/O level** | Data type: UNSIGNED_16<br>Index: $23734_d = 5CB6_h$ |
|---|---|
| C00841/52 | LS_ParReadWrite_1: wInLWord |
| C00841/53 | LS_ParReadWrite_2: wParIndex |
| C00841/54 | LS_ParReadWrite_2: wParSubindex |
| C00841/55 | LS_ParReadWrite_2: wInHWord |
| C00841/56 | LS_ParReadWrite_2: wInLWord |
| C00841/57 | LS_ParReadWrite_3: wParIndex |
| C00841/58 | LS_ParReadWrite_3: wParSubindex |
| C00841/59 | LS_ParReadWrite_3: wInHWord |
| C00841/60 | LS_ParReadWrite_3: wInLWord |
| C00841/61 | LS_ParReadWrite_4: wParIndex |
| C00841/62 | LS_ParReadWrite_4: wParSubindex |
| C00841/63 | LS_ParReadWrite_4: wInHWord |
| C00841/64 | LS_ParReadWrite_4: wInLWord |
| C00841/65 | LS_ParReadWrite_5: wParIndex |
| C00841/66 | LS_ParReadWrite_5: wParSubindex |
| C00841/67 | LS_ParReadWrite_5: wInHWord |
| C00841/68 | LS_ParReadWrite_5: wInLWord |
| C00841/69 | LS_ParReadWrite_6: wParIndex |
| C00841/70 | LS_ParReadWrite_6: wParSubindex |
| C00841/71 | LS_ParReadWrite_6: wInHWord |
| C00841/72 | LS_ParReadWrite_6: wInLWord |
| C00841/73 | Reserved |
| C00841/74 | Reserved |
| C00841/75 | Reserved |
| C00841/76 | Reserved |
| C00841/77 | Reserved |
| C00841/78 | Reserved |
| C00841/79 | Reserved |
| C00841/80 | Reserved |
| C00841/81 | Reserved |
| C00841/82 | Reserved |
| C00841/83 | Reserved |
| C00841/84 | Reserved |
| C00841/85 | Reserved |
| C00841/86 | Reserved |
| C00841/87 | Reserved |
| C00841/88 | Reserved |
| C00841/89 | LS_RetainData: wIn1 |
| C00841/90 | LS_RetainData: wIn2 |
| C00841/91 | LS_RetainData: wIn3 |
| C00841/92 | LS_RetainData: wIn4 |
| C00841/93 | LS_AxisBusOut: wLine1 |
| C00841/94 | LS_AxisBusOut: wLine2 |
| C00841/95 | LS_AxisBusOut: wLine3 |

| Parameter \| Name:<br>**C00841 \| 16-bit inputs I/O level** | Data type: UNSIGNED_16<br>Index: $23734_d = 5CB6_h$ |
|---|---|
| C00841/96 | LS_AxisBusOut: wCas1 |
| C00841/97 | LS_AxisBusOut: wCas2 |
| C00841/98 | LS_AxisBusOut: wCas3 |
| C00841/99 | LS_AxisBusOut: wCas4 |
| C00841/100 | LS_AxisBusAux: wAuxOut1 |
| C00841/101 | LS_AxisBusAux: wAuxOut2 |
| C00841/102 | LS_AxisBusAux: wAuxOut3 |
| C00841/103 | LS_AxisBusAux: wAuxOut4 |
| C00841/104 | LS_AxisBusAux: wSelectSlave |
| C00841/105 | LS_DFOut: nOut_v |
| C00841/106 | LS_MultiEncoder: wActPosExternalHW |
| C00841/107 | LS_MultiEncoder: wActPosExternalLW |
| C00841/108 | LP_CanOut4: wOut1 |
| C00841/109 | LP_CanOut4: wOut2 |
| C00841/110 | LP_CanOut4: wOut3 |
| C00841/111 | LP_CanOut4: wOut4 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | |

**C00843**

| Parameter \| Name:<br>**C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8<br>Index: $23732_d = 5CB4_h$ |
|---|---|
| Display of the signal status of the binary inputs of different I/O level blocks | |
| **Selection list** | |
| 0 \| False | |
| 1 \| True | |
| **Subcodes** | **Info** |
| C00843/1 | LS_DigitalOutput: bRelay |
| C00843/2 | LS_DigitalOutput: bOut1 |
| C00843/3 | LS_DigitalInput: bCountIn1_Reset |
| C00843/4 | LS_DigitalInput: bCountIn1_LoadStartValue |
| C00843/5 | LP_CanOut1: bState_B0 |
| C00843/6 | LP_CanOut1: bState_B1 |
| C00843/7 | LP_CanOut1: bState_B2 |
| C00843/8 | LP_CanOut1: bState_B3 |
| C00843/9 | LP_CanOut1: bState_B4 |
| C00843/10 | LP_CanOut1: bState_B5 |
| C00843/11 | LP_CanOut1: bState_B6 |
| C00843/12 | LP_CanOut1: bState_B7 |
| C00843/13 | LP_CanOut1: bState_B8 |
| C00843/14 | LP_CanOut1: bState_B9 |
| C00843/15 | LP_CanOut1: bState_B10 |
| C00843/16 | LP_CanOut1: bState_B11 |
| C00843/17 | LP_CanOut1: bState_B12 |
| C00843/18 | LP_CanOut1: bState_B13 |

| Parameter \| Name:<br>**C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8<br>Index: $23732_d = 5CB4_h$ |
|---|---|
| C00843/19 | LP_CanOut1: bState_B14 |
| C00843/20 | LP_CanOut1: bState_B15 |
| C00843/21 | LS_DisFree_b: bDis1 |
| C00843/22 | LS_DisFree_b: bDis2 |
| C00843/23 | LS_DisFree_b: bDis3 |
| C00843/24 | LS_DisFree_b: bDis4 |
| C00843/25 | LS_DisFree_b: bDis5 |
| C00843/26 | LS_DisFree_b: bDis6 |
| C00843/27 | LS_DisFree_b: bDis7 |
| C00843/28 | LS_DisFree_b: bDis8 |
| C00843/29 | LP_CanOut2: bOut1_B0 |
| C00843/30 | LP_CanOut2: bOut1_B1 |
| C00843/31 | LP_CanOut2: bOut1_B2 |
| C00843/32 | LP_CanOut2: bOut1_B3 |
| C00843/33 | LP_CanOut2: bOut1_B4 |
| C00843/34 | LP_CanOut2: bOut1_B5 |
| C00843/35 | LP_CanOut2: bOut1_B6 |
| C00843/36 | LP_CanOut2: bOut1_B7 |
| C00843/37 | LP_CanOut2: bOut1_B8 |
| C00843/38 | LP_CanOut2: bOut1_B9 |
| C00843/39 | LP_CanOut2: bOut1_B10 |
| C00843/40 | LP_CanOut2: bOut1_B11 |
| C00843/41 | LP_CanOut2: bOut1_B12 |
| C00843/42 | LP_CanOut2: bOut1_B13 |
| C00843/43 | LP_CanOut2: bOut1_B14 |
| C00843/44 | LP_CanOut2: bOut1_B15 |
| C00843/45 | LP_CanOut3: bOut1_B0 |
| C00843/46 | LP_CanOut3: bOut1_B1 |
| C00843/47 | LP_CanOut3: bOut1_B2 |
| C00843/48 | LP_CanOut3: bOut1_B3 |
| C00843/49 | LP_CanOut3: bOut1_B4 |
| C00843/50 | LP_CanOut3: bOut1_B5 |
| C00843/51 | LP_CanOut3: bOut1_B6 |
| C00843/52 | LP_CanOut3: bOut1_B7 |
| C00843/53 | LP_CanOut3: bOut1_B8 |
| C00843/54 | LP_CanOut3: bOut1_B9 |
| C00843/55 | LP_CanOut3: bOut1_B10 |
| C00843/56 | LP_CanOut3: bOut1_B11 |
| C00843/57 | LP_CanOut3: bOut1_B12 |
| C00843/58 | LP_CanOut3: bOut1_B13 |
| C00843/59 | LP_CanOut3: bOut1_B14 |
| C00843/60 | LP_CanOut3: bOut1_B15 |
| C00843/61 | LP_MciOut: bState_B0 |
| C00843/62 | LP_MciOut: bState_B1 |

| Parameter \| Name:<br>**C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8<br>Index: $23732_d = 5CB4_h$ |
|---|---|
| C00843/63 | LP_MciOut: bState_B2 |
| C00843/64 | LP_MciOut: bState_B3 |
| C00843/65 | LP_MciOut: bState_B4 |
| C00843/66 | LP_MciOut: bState_B5 |
| C00843/67 | LP_MciOut: bState_B6 |
| C00843/68 | LP_MciOut: bState_B7 |
| C00843/69 | LP_MciOut: bState_B8 |
| C00843/70 | LP_MciOut: bState_B9 |
| C00843/71 | LP_MciOut: bState_B10 |
| C00843/72 | LP_MciOut: bState_B11 |
| C00843/73 | LP_MciOut: bState_B12 |
| C00843/74 | LP_MciOut: bState_B13 |
| C00843/75 | LP_MciOut: bState_B14 |
| C00843/76 | LP_MciOut: bState_B15 |
| C00843/77 | LP_MciOut: bOut2_B0 |
| C00843/78 | LP_MciOut: bOut2_B1 |
| C00843/79 | LP_MciOut: bOut2_B2 |
| C00843/80 | LP_MciOut: bOut2_B3 |
| C00843/81 | LP_MciOut: bOut2_B4 |
| C00843/82 | LP_MciOut: bOut2_B5 |
| C00843/83 | LP_MciOut: bOut2_B6 |
| C00843/84 | LP_MciOut: bOut2_B7 |
| C00843/85 | LP_MciOut: bOut2_B8 |
| C00843/86 | LP_MciOut: bOut2_B9 |
| C00843/87 | LP_MciOut: bOut2_B10 |
| C00843/88 | LP_MciOut: bOut2_B11 |
| C00843/89 | LP_MciOut: bOut2_B12 |
| C00843/90 | LP_MciOut: bOut2_B13 |
| C00843/91 | LP_MciOut: bOut2_B14 |
| C00843/92 | LP_MciOut: bOut2_B15 |
| C00843/93 | LS_SetError_1: bSetError1 |
| C00843/94 | LS_SetError_1: bSetError2 |
| C00843/95 | LS_SetError_1: bSetError3 |
| C00843/96 | LS_SetError_1: bSetError4 |
| C00843/97 | LS_DigitalInput: bCountIn6_Reset |
| C00843/98 | LS_DigitalInput: bCountIn6_LoadStartValue |
| C00843/99 | LS_DigitalOutput: bOut2 |
| C00843/100 | LS_DigitalOutput: bOut3 |
| C00843/101 | LS_DigitalOutput: bOut HighCurrent |
| C00843/102 | LS_DisFree_b: bDis9 |
| C00843/103 | LS_DisFree_b: bDis10 |
| C00843/104 | LS_DisFree_b: bDis11 |
| C00843/105 | LS_DisFree_b: bDis12 |
| C00843/106 | LS_DisFree_b: bDis13 |

| Parameter \| Name:<br>**C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8<br>Index: $23732_d = 5CB4_h$ |
|---|---|
| C00843/107 | LS_DisFree_b: bDis14 |
| C00843/108 | LS_DisFree_b: bDis15 |
| C00843/109 | LS_DisFree_b: bDis16 |
| C00843/110 | Reserved |
| C00843/111 | LS_ParReadWrite_1: bExecute |
| C00843/112 | LS_ParReadWrite_1: bReadWrite |
| C00843/113 | LS_ParReadWrite_2: bExecute |
| C00843/114 | LS_ParReadWrite_2: bReadWrite |
| C00843/115 | LS_ParReadWrite_3: bExecute |
| C00843/116 | LS_ParReadWrite_3: bReadWrite |
| C00843/117 | LS_ParReadWrite_4: bExecute |
| C00843/118 | LS_ParReadWrite_4: bReadWrite |
| C00843/119 | LS_ParReadWrite_5: bExecute |
| C00843/120 | LS_ParReadWrite_5: bReadWrite |
| C00843/121 | LS_ParReadWrite_6: bExecute |
| C00843/122 | LS_ParReadWrite_6: bReadWrite |
| C00843/123 | LS_WriteParamList: bExecute |
| C00843/124 | LS_WriteParamList: bSelectWriteValue_1 |
| C00843/125 | LS_WriteParamList: bSelectWriteValue_2 |
| C00843/126 | LS_CANManagement: bResetNode |
| C00843/127 | LS_CANManagement: bReInitCAN |
| C00843/128 | LS_DigitalInput: bPosIn12_Load |
| C00843/129 | Reserved |
| C00843/130 | Reserved |
| C00843/131 | Reserved |
| C00843/132 | Reserved |
| C00843/133 | Reserved |
| C00843/134 | Reserved |
| C00843/135 | Reserved |
| C00843/136 | Reserved |
| C00843/137 | Reserved |
| C00843/138 | Reserved |
| C00843/139 | Reserved |
| C00843/140 | Reserved |
| C00843/141 | Reserved |
| C00843/142 | Reserved |
| C00843/143 | Reserved |
| C00843/144 | Reserved |
| C00843/145 | Reserved |
| C00843/146 | Reserved |
| C00843/147 | LS_TouchProbe: bDisableTPDigIn3 |
| C00843/148 | LS_TouchProbe: bDisableTPDigIn4 |
| C00843/149 | LS_TouchProbe: bDisableTPDigIn5 |
| C00843/150 | LS_TouchProbe: bDisableTPDigIn6 |

| Parameter \| Name:<br>**C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8<br>Index: $23732_d = 5CB4_h$ |
|---|---|
| C00843/151 | LS_TouchProbe: bDisableTPDigIn7 |
| C00843/152 | LS_TouchProbe: bDisableTPDigIn3_Rising |
| C00843/153 | LS_TouchProbe: bDisableTPDigIn3_Falling |
| C00843/154 | LS_TouchProbe: bDisableTPDigIn4_Rising |
| C00843/155 | LS_TouchProbe: bDisableTPDigIn4_Falling |
| C00843/156 | LS_TouchProbe: bDisableTPDigIn5_Rising |
| C00843/157 | LS_TouchProbe: bDisableTPDigIn5_Falling |
| C00843/158 | LS_TouchProbe: bDisableTPDigIn3Window |
| C00843/159 | LS_TouchProbe: bDisableTPDigIn4Window |
| C00843/160 | LS_TouchProbe: bDisableTPDigIn5Window |
| C00843/161 | LS_AxisBusIO: bSetFail |
| C00843/162 | LS_AxisBusIO: bResetFail |
| C00843/163 | Reserved |
| C00843/164 | LS_MultiEncoder: bSetRef |
| C00843/165 | LS_RetainData: bSetRetain_1 |
| C00843/166 | LS_RetainData: bSetRetain_2 |
| C00843/167 | LS_RetainData: bSetRetain_3 |
| C00843/168 | LS_RetainData: bLoadParams |
| C00843/169 | LS_RetainData: bIn1 |
| C00843/170 | LS_RetainData: bIn2 |
| C00843/171 | LS_RetainData: bIn3 |
| C00843/172 | LS_RetainData: bIn4 |
| C00843/173 | LS_AxisBusAux: bReadWrite |
| C00843/174 | LS_AxisBusAux: bExecute |
| C00843/175 | LS_AxisBusAux: bStop |
| C00843/176 | LS_DigitalOutput: bUserLED |
| C00843/177 | LS_TouchProbe: bDisableTPEncoderWindow |
| C00843/178 | LS_TouchProbe: bDisableTPResolverWindow |
| C00843/179 | LS_TouchProbe: bDisableTPEncoder |
| C00843/180 | LS_TouchProbe: bDisableTPResolver |
| C00843/181 | LS_DFOut: bSynRdy |
| C00843/182 | Reserved |
| C00843/183 | Reserved |
| C00843/184 | LP_CanOut4: bOut1_B0 |
| C00843/185 | LP_CanOut4: bOut1_B1 |
| C00843/186 | LP_CanOut4: bOut1_B2 |
| C00843/187 | LP_CanOut4: bOut1_B3 |
| C00843/188 | LP_CanOut4: bOut1_B4 |
| C00843/189 | LP_CanOut4: bOut1_B5 |
| C00843/190 | LP_CanOut4: bOut1_B6 |
| C00843/191 | LP_CanOut4: bOut1_B7 |
| C00843/192 | LP_CanOut4: bOut1_B8 |
| C00843/193 | LP_CanOut4: bOut1_B9 |
| C00843/194 | LP_CanOut4: bOut1_B10 |

| Parameter \| Name:<br>**C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8<br>Index: $23732_d = 5CB4_h$ |
|---|---|
| C00843/195 | LP_CanOut4: bOut1_B11 |
| C00843/196 | LP_CanOut4: bOut1_B12 |
| C00843/197 | LP_CanOut4: bOut1_B13 |
| C00843/198 | LP_CanOut4: bOut1_B14 |
| C00843/199 | LP_CanOut4: bOut1_B15 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00844**

| Parameter \| Name:<br>**C00844 \| 32-bit inputs I/O level [incr]** | | | Data type: INTEGER_32<br>Index: $23731_d = 5CB3_h$ |
|---|---|---|---|
| Display of 32-bit input values of different I/O level blocks in [increments] | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00844/1 | | | LS_DisFree_p: dnDis1_p |
| C00844/2 | | | LS_DisFree_p: dnDis2_p |
| C00844/3 | | | LS_DisFree_p: dnDis3_p |
| C00844/4 | | | LS_DisFree_p: dnDis4_p |
| C00844/5 | | | LS_DisFree_p: dnDis5_p |
| C00844/6 | | | LS_DisFree_p: dnDis6_p |
| C00844/7 | | | LS_DisFree_p: dnDis7_p |
| C00844/8 | | | LS_DisFree_p: dnDis8_p |
| C00844/9 | | | LP_CanOut1: dnOut34_p |
| C00844/10 | | | LP_CanOut2: dnOut34_p |
| C00844/11 | | | LP_CanOut3: dnOut34_p |
| C00844/12 | | | LP_MciOut: dnOut34_p |
| C00844/13 | | | LS_DigitalInput: dnPosIn12_Set_p |
| C00844/14 | | | Reserved |
| C00844/15 | | | Reserved |
| C00844/16 | | | LS_RetainData: dnIn1 |
| C00844/17 | | | LS_RetainData: dnIn2 |
| C00844/18 | | | LS_RetainData: dnIn3 |
| C00844/19 | | | LS_RetainData: dnIn4 |
| C00844/20 | | | LS_AxisBusOut: dnLine12 |
| C00844/21 | | | LS_AxisBusOut: dnCas12 |
| C00844/22 | | | LS_AxisBusOut: dnCas34 |
| C00844/23 | | | LS_AxisBusAux: dnAuxOut12 |
| C00844/24 | | | LS_AxisBusAux: dnAuxOut34 |
| C00844/25 | | | LP_CanOut4: dnOut34_p |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00866**

| Parameter \| Name: **C00866 \| CAN input words** | | | Data type: UNSIGNED_16 Index: $23709_d = 5C9D_h$ |
|---|---|---|---|
| Display of the 16 bit input values of the CAN interface<br>▸ "CAN on board" system bus | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |

| Subcodes | Info |
|---|---|
| C00866/1 | LP_CanIn1: wCtrl |
| C00866/2 | LP_CanIn1: wIn2 |
| C00866/3 | LP_CanIn1: wIn3 |
| C00866/4 | LP_CanIn1: wIn4 |
| C00866/5 | LP_CanIn2: wIn1 |
| C00866/6 | LP_CanIn2: wIn2 |
| C00866/7 | LP_CanIn2: wIn3 |
| C00866/8 | LP_CanIn2: wIn4 |
| C00866/9 | LP_CanIn3: wIn1 |
| C00866/10 | LP_CanIn3: wIn2 |
| C00866/11 | LP_CanIn3: wIn3 |
| C00866/12 | LP_CanIn3: wIn4 |
| C00866/13 | LP_CanIn4: wIn1<br>• From version 15.00.00 |
| C00866/14 | LP_CanIn4: wIn2<br>• From version 15.00.00 |
| C00866/15 | LP_CanIn4: wIn3<br>• From version 15.00.00 |
| C00866/16 | LP_CanIn4: wIn4<br>• From version 15.00.00 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT

**C00868**

| Parameter \| Name: **C00868 \| CAN output words** | | | Data type: UNSIGNED_16 Index: $23707_d = 5C9B_h$ |
|---|---|---|---|
| Display of the 16 bit output values of the CAN interface<br>▸ "CAN on board" system bus | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| Bit 1 | Active | | |
| Bit 2 | Active | | |
| Bit 3 | Active | | |
| Bit 4 | Active | | |
| Bit 5 | Active | | |
| Bit 6 | Active | | |
| Bit 7 | Active | | |
| Bit 8 | Active | | |
| Bit 9 | Active | | |
| Bit 10 | Active | | |
| Bit 11 | Active | | |
| Bit 12 | Active | | |
| Bit 13 | Active | | |
| Bit 14 | Active | | |
| Bit 15 | Active | | |

| Subcodes | Info |
|---|---|
| C00868/1 | LP_CanOut1: wState |
| C00868/2 | LP_CanOut1: wOut2 |
| C00868/3 | LP_CanOut1: wOut3 |
| C00868/4 | LP_CanOut1: wOut4 |
| C00868/5 | LP_CanOut2: wOut1 |
| C00868/6 | LP_CanOut2: wOut2 |
| C00868/7 | LP_CanOut2: wOut3 |
| C00868/8 | LP_CanOut2: wOut4 |
| C00868/9 | LP_CanOut3: wOut1 |
| C00868/10 | LP_CanOut3: wOut2 |
| C00868/11 | LP_CanOut3: wOut3 |
| C00868/12 | LP_CanOut3: wOut4 |
| C00868/13 | LP_CanOut4: wOut1<br>• From version 15.00.00 |
| C00868/14 | LP_CanOut4: wOut2<br>• From version 15.00.00 |
| C00868/15 | LP_CanOut4: wOut3<br>• From version 15.00.00 |
| C00868/16 | LP_CanOut4: wOut4<br>• From version 15.00.00 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT

**C00876**

| Parameter \| Name: **C00876 \| MCI input words** | | | Data type: UNSIGNED_16 Index: $23699_d = 5C93_h$ |
|---|---|---|---|
| Display of the 16 bit input values of the communication module | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |

| Subcodes | Info |
|---|---|
| C00876/1 | LP_MciIn: wCtrl |
| C00876/2 | LP_MciIn: wIn2 |
| C00876/3 | LP_MciIn: wIn3 |
| C00876/4 | LP_MciIn: wIn4 |
| C00876/5 | LP_MciIn: wIn5 |
| C00876/6 | LP_MciIn: wIn6 |
| C00876/7 | LP_MciIn: wIn7 |
| C00876/8 | LP_MciIn: wIn8 |
| C00876/9 | LP_MciIn: wIn9 |
| C00876/10 | LP_MciIn: wIn10 |
| C00876/11 | LP_MciIn: wIn11 |
| C00876/12 | LP_MciIn: wIn12 |
| C00876/13 | LP_MciIn: wIn13 |
| C00876/14 | LP_MciIn: wIn14 |
| C00876/15 | LP_MciIn: wIn15 |
| C00876/16 | LP_MciIn: wIn16 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT

**C00877**

| Parameter \| Name: **C00877 \| MCI output words** | | | Data type: UNSIGNED_16 Index: $23698_d = 5C92_h$ |
|---|---|---|---|
| Display of the 16 bit output values of the communication module | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00877/1 | | | LP_MciOut: wState |
| C00877/2 | | | LP_MciOut: wOut2 |
| C00877/3 | | | LP_MciOut: wOut3 |
| C00877/4 | | | LP_MciOut: wOut4 |
| C00877/5 | | | LP_MciOut: wOut5 |
| C00877/6 | | | LP_MciOut: wOut6 |
| C00877/7 | | | LP_MciOut: wOut7 |
| C00877/8 | | | LP_MciOut: wOut8 |
| C00877/9 | | | LP_MciOut: wOut9 |
| C00877/10 | | | LP_MciOut: wOut10 |
| C00877/11 | | | LP_MciOut: wOut11 |
| C00877/12 | | | LP_MciOut: wOut12 |
| C00877/13 | | | LP_MciOut: wOut13 |
| C00877/14 | | | LP_MciOut: wOut14 |
| C00877/15 | | | LP_MciOut: wOut15 |
| C00877/16 | | | LP_MciOut: wOut16 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00890**

| Parameter \| Name: **C00890 \| MCI_InOut: Inversion** | | | Data type: UNSIGNED_16 Index: $23685_d = 5C85_h$ |
|---|---|---|---|
| This parameter serves to invert the control/status bits of the MCI port blocks. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Active | | Bit set = inversion active |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00890/1 | 0x0000 | | LP_MciIn: Invert.Ctrl_B0..15 |
| C00890/2 | 0x0000 | | LP_MciOut: Invert.State_B0..15 |
| C00890/3 | 0x0000 | | LP_MciIn: Invert. In2_B0..15 |
| C00890/4 | 0x0000 | | LP_MciOut: Invert.Out2_B0..15 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00905**

| Parameter \| Name: **C00905 \| Motor phase direction of rotation** | Data type: UNSIGNED_8 Index: $23670_d = 5C76_h$ |
|---|---|

To correct such misconnected motor phases, the rotating field of the inverter's output can be reversed by selecting "1: Inverted". In this case, a phase will be reversed at the output of the inverter.

**Note:**

Up to and including version 07.00.00, this function may only be activated for the following motor control types:

- V/f characteristic control (VFCplus)
- Energy-saving V/f characteristic control (VFCplusEco)

For all other motor control types, this function must not be activated because the set control mode would not work in that case!

From version 12.00.00, this function may be activated for all motor control types with the exception of the ones for synchronous motor.

The activation of this function does not affect the control types for synchronous motor since these control types require an in-phase connection of the synchronous motor.

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **not inverted** |
| 1 | inverted |

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00909**

| Parameter \| Name: **C00909 \| Speed limitation** | Data type: INTEGER_16 Index: $23666_d = 5C72_h$ |
|---|---|

Max. positive/negative speed for all motor control modes

| **Setting range** (min. value \| unit \| max. value) | | |
|---|---|---|
| 0.00 | % | 175.00 |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C00909/1 | 120.00 % | Max. pos. speed |
| C00909/2 | 120.00 % | Max. neg. speed |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100

**C00910**

| Parameter \| Name: **C00910 \| Frequency limitation** | Data type: UNSIGNED_16 Index: $23665_d = 5C71_h$ |
|---|---|

Max. positive/negative output frequency for all motor control modes

- For dependencies see chapters "Override" and "Extended profile entry".

| **Setting range** (min. value \| unit \| max. value) | | |
|---|---|---|
| 0 | Hz | 1300 |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C00910/1 | 1000 Hz | Max. pos. output frequency |
| C00910/2 | 1000 Hz | Max. neg. output frequency |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00915**

| Parameter \| Name: **C00915 \| Motor cable length** | | | Data type: UNSIGNED_16 Index: $23660_d = 5C6C_h$ |
|---|---|---|---|
| Single motor cable length for calculating the motor cable resistance<br>• The calculated motor cable resistance is displayed in C00917. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | m | 1000.0 | **5.0 m** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00916**

| Parameter \| Name: **C00916 \| Motor cable cross-section** | | | Data type: UNSIGNED_16 Index: $23659_d = 5C6B_h$ |
|---|---|---|---|
| Motor cable cross-section of a phase/cable for calculating the motor cable resistance<br>• The calculated motor cable resistance is displayed in C00917. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.50 | mm^2 | 100.00 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00917**

| Parameter \| Name: **C00917 \| Motor cable resistance** | | | Data type: UNSIGNED_16 Index: $23658_d = 5C6A_h$ |
|---|---|---|---|
| Display of the motor cable resistance of a motor cable phase<br>• The motor cable resistance is calculated from the motor cable length set in C00915 and the motor cable cross-section set in C00916. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | mOhm | 64000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00918**

| Parameter \| Name: **C00918 \| SC: Start motor magnetising current** | | | Data type: UNSIGNED_8 Index: $23657_d = 5C69_h$ |
|---|---|---|---|
| If the servo control (SC) mode is selected and the inverter is enabled, the speed setpoint for motor control will only be enabled when the motor magnetising current has reached the threshold value set here.<br>• The adjustable percentage threshold value refers to the rated magnetising current (C00095). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | % | 90 | **87 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00919**

| Parameter \| Name: C00919 \| Moment of inertia from load | | | Data type: UNSIGNED_32 Index: $23656_d = 5C68_h$ |
|---|---|---|---|
| From version 12.00.00<br>In order to take account of mass inertias changing during the process (e.g. reels) when optimising the response to setpoint changes, the maximum value of the changing moment of inertia must be set here. The process signal *nInertiaAdapt_a* of the LS_MotorInterface SB can then be used in the process to dynamically control what percentage of it is to be taken into account for the setpoint feedforward control. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | kg cm^2 | 6000000.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00919/1 | 0.00 kg cm^2 | | Load moment of inertia |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00920**

| Parameter \| Name: C00920 \| Rated device currents | | | Data type: UNSIGNED_16 Index: $23655_d = 5C67_h$ |
|---|---|---|---|
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.0 | A | 6000.0 | |
| **Subcodes** | | | **Info** |
| C00920/1 | | | Rated current 3ph 400V/1ph 230V |
| C00920/2 | | | Rated current 3ph 440V |
| C00920/3 | | | Rated current 3ph 480V |
| C00920/4 | | | Rated current 3ph 500V |
| C00920/5 | | | Increased rated current 3ph 400V/1ph 230V |
| C00920/6 | | | Increased rated current 3ph 440V |
| C00920/7 | | | Increased rated curent 3ph 480V |
| C00920/8 | | | Increased rated current 3ph 500V |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00922**

| Parameter \| Name: C00922 \| ICM_DiagnosticCounter | Data type: UNSIGNED_16 Index: $23653_d = 5C65_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00925**

| Parameter \| Name: C00925 \| LS_Resolver: Number of pole pairs | | | Data type: UNSIGNED_8 Index: $23650_d = 5C62_h$ |
|---|---|---|---|
| | | | ▸ Encoder/feedback system |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | | 10 | **1** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00926**

| Parameter \| Name: C00926 \| Pole position | | | Data type: INTEGER_16 Index: $23649_d = 5C61_h$ |
|---|---|---|---|
| | | | ▸ Encoder/feedback system |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -179.9 | ° | 179.9 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00926/1 | -90.0 ° | | LS_Resolver: Pole position |
| C00926/2 | 0.0 ° | | LS_MultiEncoder: Pole position |
| C00926/3 | 0.0 ° | | LS_Resolver: NP Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00927**

| Parameter \| Name: C00927 \| Motor rotor position | | | Data type: UNSIGNED_16 Index: $23648_d = 5C60_h$ |
|---|---|---|---|
| | | | ▸ Encoder/feedback system |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 2047 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00937**

| Parameter \| Name: C00937 \| Field-oriented motor currents | | | Data type: INTEGER_16 Index: $23638_d = 5C56_h$ |
|---|---|---|---|
| From version 02.00.00 | | | ▸ Field weakening for synchronous motors |
| **Display area** (min. value \| unit \| max. value) | | | |
| -320.00 | A | 320.00 | |
| **Subcodes** | | | **Info** |
| C00937/1 | | | Field-producing current |
| C00937/2 | | | Torque-producing current |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00938**

| Parameter \| Name: C00938 \| PSM: Maximum motor current field weakening | | | Data type: UNSIGNED_16 Index: $23637_d = 5C55_h$ |
|---|---|---|---|
| From version 02.00.00 | | | ▸ Field weakening for synchronous motors |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 500.00 | **30.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00939**

| Parameter \| Name: **C00939 \| Ultimate motor current** | | | Data type: UNSIGNED_16 Index: $23636_d = 5C54_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | A | 3000.0 | **3000.0 A** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00940**

| Parameter \| Name: **C00940 \| L_ConvW numerator** | | | Data type: INTEGER_16 Index: $23635_d = 5C53_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00940/1 | 1 | | L_ConvW_1: Numerator |
| C00940/2 | 1 | | L_ConvW_2: Numerator |
| C00940/3 | 1 | | L_ConvW_3: Numerator |
| C00940/4 | 1 | | L_ConvW_4: Numerator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00941**

| Parameter \| Name: **C00941 \| L_ConvW denominator** | | | Data type: INTEGER_16 Index: $23634_d = 5C52_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00941/1 | 1 | | L_ConvW_1: Denominator |
| C00941/2 | 1 | | L_ConvW_2: Denominator |
| C00941/3 | 1 | | L_ConvW_3: Denominator |
| C00941/4 | 1 | | L_ConvW_4: Denominator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00942**

| Parameter \| Name: **C00942 \| L_ConvW conversion method** | | Data type: UNSIGNED_8 Index: $23633_d = 5C51_h$ |
|---|---|---|
| **Selection list** | | |
| 0 | no conversion | |
| 1 | from [%] into [incr./ms] | |
| 2 | from [incr./ms] into [%] | |
| 3 | Factors signed | |
| 4 | Factors unsigned | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00942/1 | 0: No conversion | L_ConvW_1: Conversion method |
| C00942/2 | 0: No conversion | L_ConvW_2: Conversion method |
| C00942/3 | 0: No conversion | L_ConvW_3: Conversion method |
| C00942/4 | 0: No conversion | L_ConvW_4: Conversion method |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00950**

| Parameter \| Name: **C00950 \| L_Interpolator_1: Activation FB functions** | | Data type: UNSIGNED_8 Index: $23625_d = 5C49_h$ |
|---|---|---|
| The L_Interpolator_1 FB: Activation of signal interpolation and signal monitoring | | |
| **Selection list** | | |
| 0 | Off | |
| 1 | On | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00950/1 | 0: Off | L_Interpolator_1: Signal interpolation |
| C00950/2 | 0: Off | L_Interpolator_1: Signal monitoring |
| C00950/3 | 0: Off | L_Interpolator_1: Master value monitoring |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00951**

| Parameter \| Name: **C00951 \| L_Interpolator_1: No. of interpolation steps** | | | Data type: UNSIGNED_16 Index: $23624_d = 5C48_h$ |
|---|---|---|---|
| The L_Interpolator_1 FB: No. of interpolation steps | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 65535 | **1** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00952**

| Parameter \| Name: **C00952 \| L_Interpolator_1: Limit value - error cycles** | | | Data type: UNSIGNED_16 Index: $23623_d = 5C47_h$ |
|---|---|---|---|
| The L_Interpolator_1 FB: Limit value for missing data telegrams | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 65535 | **5** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00953**

| Parameter \| Name: **C00953 \| L_Interpolator_1: Speed-up** | | | Data type: UNSIGNED_8 Index: $23622_d = 5C46_h$ |
|---|---|---|---|
| FB L_Interpolator_1: Limitation of the number of correction increments per cycle (speed-up) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 100 | **0** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00959**

| Parameter \| Name: **C00959 \| L_Curve: Current output value** | | | Data type: INTEGER_16 Index: $23616_d = 5C40_h$ |
|---|---|---|---|
| From version 02.00.00<br>FB L_Curve_1: Display of the current output value *nOut_a* | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00959/1 | | | L_Curve_1: Current output value |
| C00959/2 | | | L_Curve_2: Current output value |
| C00959/3 | | | L_Curve_3: Current output value |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00960**

| Parameter \| Name: **C00960 \| L_Curve_1: Selected curve type** | | Data type: UNSIGNED_8 Index: $23615_d = 5C3F_h$ |
|---|---|---|
| FB L_Curve_1: Selected curve type | | |
| **Selection list** | | |
| 0 | Out = 0 | |
| 1 | Out = In | |
| 2 | Out = f(In) | |
| 3 | Out = f(table) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00960/1 | 1: Out = In | L_Curve_1: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00961**

| Parameter \| Name: **C00961 \| L_Curve_1: Input limitation** | | | Data type: INTEGER_16 Index: $23614_d = 5C3E_h$ |
|---|---|---|---|
| FB L_Curve_1: Upper and lower limit for input value | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00961/1 | 199.99 % | | L_Curve_1: Max. input |
| C00961/2 | -199.99 % | | L_Curve_1: Min. input |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00963**

| Parameter \| Name: **C00963 \| L_Curve_1: Table X-values** | | | Data type: INTEGER_16 Index: $23612_d = 5C3C_h$ |
|---|---|---|---|
| FB L_Curve_1: X-values for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00963/1 | 0 | | X values 1 ... 32 for characteristic function |
| C00963/... | | | |
| C00963/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00964**

| Parameter \| Name: **C00964 \| L_Curve_1: Table Y-values** | | | Data type: INTEGER_16 Index: $23611_d = 5C3B_h$ |
|---|---|---|---|
| FB L_Curve_1: Y-value for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00964/1 | 0 | | Y values 1 ... 32 for characteristic function |
| C00964/... | | | |
| C00964/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00965**

| Parameter \| Name: **C00965 \| Max. motor speed** | | | Data type: UNSIGNED_16 Index: $23610_d = 5C3A_h$ |
|---|---|---|---|
| When the drive reaches the motor speed set here:<br>• The "Fault" error response takes place, i.e. the motor is shut down immediately.<br>• The error message "oS2: Max. motor speed reached" is entered into the logbook. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 50 | rpm | 65000 | **60000 rpm** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00966**

| Parameter \| Name: **C00966 \| VFC: Time const. slip comp.** | | | Data type: UNSIGNED_16 Index: $23609_d = 5C39_h$ |
|---|---|---|---|
| Filter time constant of the slip compensation for V/f characteristic control (VFCplus)<br>• The time constant of slip compensation serves to specify the dynamics of slip compensation for V/f characteristic control without feedback.<br>• The lower the selected time constant, the higher the dynamic performance of the slip compensation. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | ms | 6000 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00967**

| Parameter \| Name: **C00967 \| VFC: Frequency interpol. point n** | | | Data type: INTEGER_16 Index: $23608_d = 5C38_h$ |
|---|---|---|---|
| Selection of the interpolation points (frequency values) for the V/f characteristic control (VFCplus) with user-definable V/f characteristic (C00006 = "10") | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2600.0 | Hz | 2600.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00967/1 | -50.0 Hz | | VFC: Frequency interpol. point 1 |
| C00967/2 | -40.0 Hz | | VFC: Frequency interpol. point 2 |
| C00967/3 | -30.0 Hz | | VFC: Frequency interpol. point 3 |
| C00967/4 | -20.0 Hz | | VFC: Frequency interpol. point 4 |
| C00967/5 | -10.0 Hz | | VFC: Frequency interpol. point 5 |
| C00967/6 | 0.0 Hz | | VFC: Frequency interpol. point 6 |
| C00967/7 | 10.0 Hz | | VFC: Frequency interpol. point 7 |
| C00967/8 | 20.0 Hz | | VFC: Frequency interpol. point 8 |
| C00967/9 | 30.0 Hz | | VFC: Frequency interpol. point 9 |
| C00967/10 | 40.0 Hz | | VFC: Frequency interpol. point 10 |
| C00967/11 | 50.0 Hz | | VFC: Frequency interpol. point 11 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00968**

| Parameter \| Name: **C00968 \| VFC: Voltage interpol. point n** | | | Data type: UNSIGNED_16 Index: $23607_d = 5C37_h$ |
|---|---|---|---|
| Selection of the interpolation points (voltage values) for the V/f characteristic control (VFCplus) with user-definable V/f characteristic (C00006 = "10") | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | V | 600.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00968/1 | 400.00 V | | VFC: Voltage interpol. point 1 |
| C00968/2 | 320.00 V | | VFC: Voltage interpol. point 2 |
| C00968/3 | 240.00 V | | VFC: Voltage interpol. point 3 |
| C00968/4 | 160.00 V | | VFC: Voltage interpol. point 4 |
| C00968/5 | 80.00 V | | VFC: Voltage interpol. point 5 |
| C00968/6 | 0.00 V | | VFC: Voltage interpol. point 6 |
| C00968/7 | 80.00 V | | VFC: Voltage interpol. point 7 |
| C00968/8 | 160.00 V | | VFC: Voltage interpol. point 8 |
| C00968/9 | 240.00 V | | VFC: Voltage interpol. point 9 |
| C00968/10 | 320.00 V | | VFC: Tension interpol. point 10 |
| C00968/11 | 400.00 V | | VFC: Voltage interpol. point 11 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00969**

| Parameter \| Name: **C00969 \| Motorparameter** | | | Data type: UNSIGNED_8 Index: $23606_d = 5C36_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 1 | | 255 | |
| **Subcodes** | | | **Info** |
| C00969/1 | | | Motor - number of pole pairs |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00971**

| Parameter \| Name: **C00971 \| VFC: Limitation V/f +encoder** | | | Data type: UNSIGNED_16 Index: $23604_d = 5C34_h$ |
|---|---|---|---|
| Limitation of the output frequency of the slip regulator and limitation of the injected stator frequency for the V/f control (VFCplus+encoder) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | Hz | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00971/1 | 10.00 Hz | | Maximum output / correcting variable of the slip regulator<br>• The slip regulator output is limited to the value set here in motor/generator mode.<br>• It is recommended to select 1 to 3 times the slip frequency of the motor as limit value. |
| C00971/2 | 100.00 Hz | | Maximum frequency deviation between the rotational frequency (speed) measured mechanically by the encoder and the injected stator frequency.<br>• A limitation may e.g. avoid overcurrent interruption when traversing to a fixed limit stop. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00972**

| Parameter \| Name: **C00972 \| VFC: Vp V/f +encoder** | | | Data type: UNSIGNED_16 Index: $23603_d = 5C33_h$ |
|---|---|---|---|
| Proportional gain of the slip regulator for V/f control (VFCplus+encoder)<br>• The gain must be selected depending on the drive system and the sensor resolution (range: 0.005 ... 5).<br>• A high gain requires a high number of increments. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | Hz/Hz | 64.000 | **0.100 Hz/Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1000 | | | |

**C00973**

| Parameter \| Name: **C00973 \| VFC: Ti V/f +encoder** | | | Data type: UNSIGNED_16 Index: $23602_d = 5C32_h$ |
|---|---|---|---|
| Integral time constant of the slip regulator for V/f control (VFCplus+encoder)<br>• In general, the time constant should be selected in a range of 20 ms (high dynamics) to 200 (low dynamics). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | ms | 6000.0 | **100.0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00975**

| Parameter \| Name: **C00975 \| VFC-ECO: Vp CosPhi controller** | | | Data type: UNSIGNED_16 Index: $23600_d = 5C30_h$ |
|---|---|---|---|
| Proportional gain of the Cos-Phi controller for energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | Hz/Hz | 64.000 | **0.500 Hz/Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1000 | | | |

**C00976**

| Parameter \| Name: **C00976 \| VFC-ECO: Ti CosPhi controller** | | | Data type: UNSIGNED_16 Index: $23599_d = 5C2F_h$ |
|---|---|---|---|
| Reset time of the Cos-Phi controller for energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | ms | 6000.0 | **200.0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00977**

| Parameter \| Name: **C00977 \| VFC-ECO: Minimum voltage V/f** | | | Data type: INTEGER_16 Index: $23598_d = 5C2E_h$ |
|---|---|---|---|
| Minimum voltage V/f of the Cos-Phi controller for energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 20.00 | % | 100.00 | **20.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00978**

| Parameter \| Name: **C00978 \| VFC-ECO: Voltage reduction** | | | Data type: INTEGER_16 Index: $23597_d = 5C2D_h$ |
|---|---|---|---|
| Display of the voltage reduction with energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -1000 | V | 1000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00979**

| Parameter \| Name: **C00979 \| Cosine phi** | | | Data type: INTEGER_16 Index: $23596_d = 5C2C_h$ |
|---|---|---|---|
| Display of the cosφ setpoint and actual value with energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -1.00 | | 1.00 | |
| **Subcodes** | | | **Info** |
| C00979/1 | | | Cosine phi act |
| C00979/2 | | | Cosine phi set |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00980**

| Parameter \| Name: C00980 \| Output power | | | Data type: INTEGER_32 Index: $23595_d = 5C2B_h$ |
|---|---|---|---|
| Display parameter for an energy analysis in the prevailing application. From this, decisions can be deduced whether a measure for energy optimisation is economic. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -32.000 | kW | 32.000 | |
| **Subcodes** | | | **Info** |
| C00980/1 | | | Active output power |
| C00980/2 | | | Apparent output power |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00981**

| Parameter \| Name: C00981 \| Energy display | | | Data type: INTEGER_32 Index: $23594_d = 5C2A_h$ |
|---|---|---|---|
| Display parameter for an energy analysis in the prevailing application. From this, decisions can be deduced whether a measure for energy optimisation is economic.<br>• The values are saved to the device by switching off the mains and cannot be reset. | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | kWh | 21474836.47 | |
| **Subcodes** | | | **Info** |
| C00981/1 | | | Output energy in motor mode |
| C00981/2 | | | Output energy in generator mode |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00982**

| Parameter \| Name: C00982 \| VFC-ECO: Voltage reduction ramp | | | Data type: UNSIGNED_8 Index: $23593_d = 5C29_h$ |
|---|---|---|---|
| Voltage ramp for cancelling V-Sub with energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | s | 5.0 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00983**

| Parameter \| Name: C00983 \| Delay | | | Data type: UNSIGNED_16 Index: $23592_d = 5C28_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 4000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00983/1 | 50 ms | | Delay Imax |
| C00983/2 | 1000 ms | | VFC: Voltage ramp after overvoltage oU<br>• From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00985**

| Parameter \| Name: **C00985 \| SLVC: Gain of field current controller** | | | Data type: INTEGER_16 Index: $23590_d = 5C26_h$ |
|---|---|---|---|
| Gain of the direct-axis current difference (Id) between setpoint and actual current for the voltage model of the sensorless vector control (SLVC)<br>• The gain should be selected within the range 0 ...1 %. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 20.00 | **0.50 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00986**

| Parameter \| Name: **C00986 \| SLVC: Gain of cross current controller** | | | Data type: INTEGER_16 Index: $23589_d = 5C25_h$ |
|---|---|---|---|
| Gain of the IQ difference for the voltage model of the sensorless vector control (SLVC) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 20.00 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00987**

| Parameter \| Name: **C00987 \| Inverter motor brake: nAdd** | | | Data type: INTEGER_16 Index: $23588_d = 5C24_h$ |
|---|---|---|---|
| Speed lift which is connected in pulses to the brake ramp when the motor is braked.<br>▸ Inverter motor brake | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | rpm | 1000 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00988**

| Parameter \| Name: **C00988 \| Inverter motor brake: PT1 filter time** | | | Data type: INTEGER_16 Index: $23587_d = 5C23_h$ |
|---|---|---|---|
| PT1 filter time for smoothing the speed lift which is added in pulses (C00987)<br>▸ Inverter motor brake | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | ms | 100.0 | **0.0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C00989**

| Parameter \| Name: **C00989 \| Restart on the fly: Flying restart frequency fd_add** | | | Data type: INTEGER_16 Index: $23586_d = 5C22_h$ |
|---|---|---|---|
| From version 15.00.00<br>▸ Flying restart fct. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | Hz | 5.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00989/1 | 0.00 Hz | | Restart on the fly: Flying restart frequency fd_add |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00990**

| Parameter \| Name: **C00990 \| Flying restart fct.: Activation** | | | Data type: UNSIGNED_8 Index: $23585_d = 5C21_h$ |
|---|---|---|---|
| Switch on/activate flying restart circuit for non-feedback drive systems<br>▸ Flying restart fct. | | | |
| **Selection list** (Lenze setting printed in bold) | | | |
| **0** | **Off** | | |
| 1 | On | | |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00991**

| Parameter \| Name: **C00991 \| Flying restart fct.: Process** | | Data type: UNSIGNED_16 Index: $23584_d = 5C20_h$ |
|---|---|---|
| Selection of the starting value and the speed search range for the flying restart function<br>▸ Flying restart fct. | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | 0...+n \| Start: +10 Hz | Search positive speed range (0 ... +n) with a start frequency of +10 Hz |
| 1 | -n...0 \| Start: -10 Hz | Search negative speed range (-n ... 0) with a start frequency of -10 Hz |
| **2** | **-n...+n \| Start: +10 Hz** | Search negative and positive speed range (-n ... n) with a start frequency of +10 Hz |
| 3 | -n...+n \| Start: -10 Hz | Search negative and positive speed range (-n ... n) with a start frequency of -10 Hz |
| 4 | -n...+n \| Start: Cx992 | Search the negative and positive speed range (-n ... n) with the start frequency set in C00992 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00992**

| Parameter \| Name: **C00992 \| Flying restart fct.: Start frequency** | | | Data type: INTEGER_16 Index: $23583_d = 5C1F_h$ |
|---|---|---|---|
| Manual selection of the starting value for the flying restart function<br>• Only active if C00991 = 4<br>▸ Flying restart fct. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -200 | Hz | 200 | **10 Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00993**

| Parameter \| Name: **C00993 \| Flying restart fct: Int. time** | | | Data type: UNSIGNED_16 Index: $23582_d = 5C1E_h$ |
|---|---|---|---|
| Time constant of the angular difference controller of the flying restart function<br>• The time constant is to amount between 60 ... 300 ms.<br>▸ Flying restart fct. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | ms | 6000.0 | power-related (see table) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00994**

| Parameter \| Name: C00994 \| Flying restart fct.: Current | | | Data type: INTEGER_16 Index: $23581_d = 5C1D_h$ |
|---|---|---|---|
| Current to be injected during the flying restart process<br>• 100 % ≡ rated motor current (C00088).<br>• The flying restart current should amount to 10 ... 25 % of the rated motor current.<br>▸ Flying restart fct. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **25.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00995**

| Parameter \| Name: C00995 \| SLPSM: Controlled current setpoint | | | Data type: UNSIGNED_16 Index: $23580_d = 5C1C_h$ |
|---|---|---|---|
| ▸ Sensorless control for synchronous motors (SLPSM) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 5.00 | % | 400.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00995/1 | 100.00 % | | SLPSM: Controlled accelerating current |
| C00995/2 | 20.00 % | | SLPSM: Controlled standstill current |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00996**

| Parameter \| Name: C00996 \| SLPSM: Switching speed | | | Data type: INTEGER_16 Index: $23579_d = 5C1B_h$ |
|---|---|---|---|
| ▸ Sensorless control for synchronous motors (SLPSM) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00996/1 | 13.00 % | | SLPSM: Switching speed, closed-loop control |
| C00996/2 | 8.00 % | | SLPSM: Switching speed, open-loop control |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00997**

| Parameter \| Name: C00997 \| SLPSM: Filter cutoff frequency | | | Data type: INTEGER_16 Index: $23578_d = 5C1A_h$ |
|---|---|---|---|
| ▸ Sensorless control for synchronous motors (SLPSM) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **5.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00998**

| Parameter \| Name: **C00998 \| SLPSM: Filter time rotor position** | | | Data type: INTEGER_16 Index: $23577_d = 5C19_h$ |
|---|---|---|---|
| | | | ▸ Sensorless control for synchronous motors (SLPSM) |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.5 | ms | 20.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00998/1 | 3.0 ms | | SLPSM: Filter time rotor position |
| C00998/2 | 5.0 ms | | SLPSM: Filter time actual speed value |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00999**

| Parameter \| Name: **C00999 \| SLPSM: PLL gain** | | | Data type: INTEGER_16 Index: $23576_d = 5C18_h$ |
|---|---|---|---|
| | | | ▸ Sensorless control for synchronous motors (SLPSM) |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | % | 1000 | **400 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C01000**

| Parameter \| Name: **C01000 \| MCTRL: Status** | | Data type: UNSIGNED_16 Index: $23575_d = 5C17_h$ |
|---|---|---|
| **Display area** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** | | **Info** |
| Bit 0 | SLPSM: Speed closed-loop control | 1 ≡ The sensorless control for synchronous motors is in the closed-loop controlled operation ($\|n_{Setpoint}\| > n_{C00996}$).<br>▸ Sensorless control for synchronous motors (SLPSM) |
| Bit 1 | SLPSM: Speed open-loop control | From version 12.00.00<br>1 ≡ The sensorless control for synchronous motors is in the open-loop controlled operation ($\|n_{Setpoint}\| > n_{C00996}$).<br>▸ Sensorless control for synchronous motors (SLPSM) |
| Bit 2 | Motor ASM | From version 12.00.00<br>1 ≡ Set motor type is "Asynchronous motor" |
| Bit 3 | Motor PSM | From version 12.00.00<br>1 ≡ Set motor type is "Synchronous motor" |
| Bit 4 | Encoder open circuit: V/f linear active | From version 15.00.00<br>1 ≡ Due to an encoder open circuit, it has been internally switched to the encoderless V/f characteristic control to avoid impermissible motor movements. |
| Bit 5 | Speed limit. C909/C910/fdmax | |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT | | |

**C01001**

| Parameter \| Name: **C01001 \| Manual entry of motor type** | | Data type: UNSIGNED_8 Index: $23574_d = 5C16_h$ |
|---|---|---|
| From version 14.00.00 | | |
| **Selection list** | | |
| 0 | Automatic | |
| 1 | ASM | |
| 2 | PSM | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01001/1 | 0: Automatic | Manual entry of motor type |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C01010**

| Parameter \| Name: C01010 \| L_ArithmetikPhi 1-3: Function | | Data type: UNSIGNED_8 Index: $23565_d = 5C0D_h$ |
|---|---|---|
| Selection of the internal arithmetics | | |
| **Selection list** | | |
| 0 | dnIn1_p | |
| 1 | dnIn1_p + dnIn2_p | |
| 2 | dnIn1_p - dnIn2_p | |
| 3 | dnIn1_p * dnIn2_p | |
| 4 | dnIn1_p / dnIn2_p | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01010/1 | 0: dnIn1_p | L_ArithmetikPhi_1: Function |
| C01010/2 | 0: dnIn1_p | L_ArithmetikPhi_2: Function |
| C01010/3 | 0: dnIn1_p | L_ArithmetikPhi_3: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01011**

| Parameter \| Name: C01011 \| L_ArithmetikPhi 4-6: Function | | Data type: UNSIGNED_8 Index: $23564_d = 5C0C_h$ |
|---|---|---|
| From version 12.00.00<br>Selection of the internal arithmetics | | |
| **Selection list** | | |
| 0 | dnIn1_p | |
| 1 | dnIn1_p + dnIn2_p | |
| 2 | dnIn1_p - dnIn2_p | |
| 3 | dnIn1_p * dnIn2_p | |
| 4 | dnIn1_p / dnIn2_p | |
| 5 | dnIn1_p % dnIn2_p | |
| 6 | dnIn1_p + dnIn2_p (no limit.) | |
| 7 | dnIn1_p - dnIn2_p (no limit.) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01011/1 | 0: dnIn1_p | L_ArithmetikPhi_4: Function |
| C01011/2 | 0: dnIn1_p | L_ArithmetikPhi_5: Function |
| C01011/3 | 0: dnIn1_p | L_ArithmetikPhi_6: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01012**

| Parameter \| Name: C01012 \| L_PhiDiv: byDivision | | | Data type: INTEGER_8 Index: $23563_d = 5C0B_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -31 | | 31 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01012/1 | 0 | | L_PhiDiv_1: byDivision |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01020**

| Parameter \| Name: **C01020 \| L_Odometer_1: Memory length** | Data type: UNSIGNED_8 Index: $23555_d = 5C03_h$ |
|---|---|
| FB L_Odometer_1: No. of measurements | |
| **Selection list** (Lenze setting printed in bold) | |
| 1 | 1 measurement |
| 2 | 2 measurements |
| 3 | 3 measurements |
| 4 | 4 measurements |
| 5 | 5 measurements |
| 6 | 6 measurements |
| **7** | **7 measurements** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C01021**

| Parameter \| Name: **C01021 \| L_Odometer_1: Memory type** | Data type: UNSIGNED_8 Index: $23554_d = 5C02_h$ |
|---|---|
| FB L_Odometer_1: If "ring buffer" is selected, it is started again after the measurements set in C01020 and the old values are overwritten. Otherwise, the measurement stops. | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **No ring buffer** |
| 1 | Ring buffer |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C01022**

| Parameter \| Name: **C01022 \| L_Odometer_1: Input selection** | Data type: UNSIGNED_8 Index: $23553_d = 5C01_h$ |
|---|---|
| FB L_Odometer_1: Selection of position or speed input | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **Pos input** |
| 1 | V input |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C01023**

| Parameter \| Name: **C01023 \| L_Odometer_1: Edge selection** | Data type: UNSIGNED_8 Index: $23552_d = 5C00_h$ |
|---|---|
| FB L_Odometer_1: No. of the edge triggering the measurement | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **High edge** |
| 1 | Low edge |
| 2 | High and low edge |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C01025**

| Parameter \| Name: **C01025 \| L_Curve_2: Selected curve type** | | Data type: UNSIGNED_8 Index: $23550_d = 5BFE_h$ |
|---|---|---|
| From version 02.00.00<br>FB L_Curve_2: Selected curve type | | |
| **Selection list** | | |
| 0 | Out = 0 | |
| 1 | Out = In | |
| 2 | Out = f(In) | |
| 3 | Out = f(table) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01025/1 | 1: Out = In | L_Curve_2: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01026**

| Parameter \| Name: **C01026 \| L_Curve_2: Input limitation** | | | Data type: INTEGER_16 Index: $23549_d = 5BFD_h$ |
|---|---|---|---|
| From version 02.00.00<br>FB L_Curve_2: Upper and lower limit for input value | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01026/1 | 199.99 % | | L_Curve_2: Max. input |
| C01026/2 | -199.99 % | | L_Curve_2: Min. input |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01028**

| Parameter \| Name: **C01028 \| L_Curve_2: Table X-values** | | | Data type: INTEGER_16 Index: $23547_d = 5BFB_h$ |
|---|---|---|---|
| From version 02.00.00<br>FB L_Curve_2: X-values for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01028/1 | 0 | | X values 1 ... 32 for characteristic function |
| C01028/... | | | |
| C01028/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01029**

| Parameter \| Name: **C01029 \| L_Curve_2: Table Y-values** | | | Data type: INTEGER_16 Index: $23546_d = 5BFA_h$ |
|---|---|---|---|
| From version 02.00.00<br>FB L_Curve_2: Y-value for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01029/1 | 0 | | Y values 1 ... 32 for characteristic function |
| C01029/... | | | |
| C01029/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01030**

| Parameter \| Name: **C01030 \| L_Curve_3: Selected curve type** | | Data type: UNSIGNED_8 Index: $23545_d = 5BF9_h$ |
|---|---|---|
| From version 02.00.00<br>FB L_Curve_3: Selected curve type | | |
| **Selection list** | | **Info** |
| 0 | Out = 0 | |
| 1 | Out = In | |
| 2 | Out = f(In) | |
| 3 | Out = f(table) | |
| 4 | Out = f(characteristic) | From version 12.00.00 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01030/1 | 1: Out = In | L_Curve_3: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01031**

| Parameter \| Name: **C01031 \| L_Curve_3: Input limitation** | | | Data type: INTEGER_16 Index: $23544_d = 5BF8_h$ |
|---|---|---|---|
| From version 02.00.00<br>FB L_Curve_3: Upper and lower limit for input value | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01031/1 | 199.99 % | | L_Curve_3: Max. input |
| C01031/2 | -199.99 % | | L_Curve_3: Min. input |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01033**

| Parameter \| Name: **C01033 \| L_Curve_3: Table X-values** | | | Data type: INTEGER_16 Index: $23542_d = 5BF6_h$ |
|---|---|---|---|
| From version 02.00.00<br>FB L_Curve_3: X-values for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01033/1 | 0 | | X values 1 ... 32 for characteristic function |
| C01033/... | | | |
| C01033/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01034**

| Parameter \| Name: **C01034 \| L_Curve_3: Table Y-values** | | | Data type: INTEGER_16 Index: $23541_d = 5BF5_h$ |
|---|---|---|---|
| From version 02.00.00<br>FB L_Curve_3: Y-value for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01034/1 | 0 | | Y values 1 ... 32 for characteristic function |
| C01034/... | | | |
| C01034/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01035**

| Parameter \| Name: **C01035 \| L_Curve_3: SelectCurve** | | Data type: UNSIGNED_16 Index: $23540_d = 5BF4_h$ |
|---|---|---|
| From version 12.00.00<br>FB L_Curve_3: Selection of the tensile force profile | | |
| **Selection list** | | **Info** |
| 0 | Linear tensile force profile | ▸ Use of L_Curve_3 for tensile force characteristic |
| 1 | Linear torque profile | |
| 2 | Tensile force profile according to characteristic | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01035/1 | 0: Linear tensile force profile | L_Curve_3: SelectCurve |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01040**

| Parameter \| Name: **C01040 \| L_SRFG_1..2 linear ramp time** | | | Data type: UNSIGNED_32 Index: $23535_d = 5BEF_h$ |
|---|---|---|---|
| Symmetrical acceleration/deceleration time for the ramp function generator | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.001 | s | 999.999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01040/1 | 100.000 s | | L_SRFG_1: linear ramp time |
| C01040/2 | 100.000 s | | L_SRFG_2: linear ramp time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01041**

| Parameter \| Name: **C01041 \| L_SRFG_1..2 S-ramp time** | | | Data type: UNSIGNED_32 Index: $23534_d = 5BEE_h$ |
|---|---|---|---|
| S-ramp time for jerk-free acceleration | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.001 | s | 50.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01041/1 | 0.200 s | | L_SRFG_1: S-ramp time |
| C01041/2 | 0.200 s | | L_SRFG_2: S-ramp time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01042**

| Parameter \| Name: **C01042 \| L_SRFG_1..2 limitations of output values** | | | Data type: INTEGER_16 Index: $23533_d = 5BED_h$ |
|---|---|---|---|
| Limitation of the output values | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01042/1 | 100.00 % | | L_SRFG_1: Pos. Limit |
| C01042/2 | -100.00 % | | L_SRFG_1: Neg. Limit |
| C01042/3 | 100.00 % | | L_SRFG_2: Pos. Limit |
| C01042/4 | -100.00 % | | L_SRFG_2: Neg. Limit |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01045**

| Parameter \| Name: **C01045 \| L_ConvAP 1-3: Numerator/denominator** | | | Data type: INTEGER_16 Index: $23530_d = 5BEA_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01045/1 | 1 | | L_ConvAP_1: Numerator |
| C01045/2 | 1 | | L_ConvAP_1: Denominator |
| C01045/3 | 1 | | L_ConvAP_2: Numerator |
| C01045/4 | 1 | | L_ConvAP_2: Denominator |
| C01045/5 | 1 | | L_ConvAP_3: Numerator |
| C01045/6 | 1 | | L_ConvAP_3: Denominator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01046**

| Parameter \| Name: **C01046 \| L_ConvPA 1-3: byDivision** | | | Data type: INTEGER_8 Index: $23529_d = 5BE9_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 31 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01046/1 | 1 | | L_ConvPA_1: Division factor |
| C01046/2 | 1 | | L_ConvPA_2: Division factor |
| C01046/3 | 1 | | L_ConvPA_3: Division factor |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01047**

| Parameter \| Name: **C01047 \| L_GearComp_1: Offset** | | | Data type: INTEGER_16 Index: $23528_d = 5BE8_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -16383 | | 16383 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01047/1 | 0 | | L_GearComp_1: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01048**

| Parameter \| Name: **C01048 \| L_GearComp_1: Num_Denom** | | | Data type: INTEGER_16 Index: $23527_d = 5BE7_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01048/1 | 1 | | L_GearComp_1: Numerator |
| C01048/2 | 1 | | L_GearComp_1: Denominator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01049**

| Parameter \| Name: **C01049 \| L_CalcDiameter_1: Status** | | | Data type: INTEGER_16 Index: $23526_d = 5BE6_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -10 | | 30 | |
| **Subcodes** | | | **Info** |
| C01049/1 | | | L_CalcDiameter_1: Status |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01050**

| Parameter \| Name: **C01050 \| L_CalcDiameter_1: Diameter recalculation** | | | Data type: UNSIGNED_32 Index: $23525_d = 5BE5_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.001 | Rev. | 1000.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01050/1 | 1.000 rev. | | L_CalcDiameter_1: Diameter recalculation 0 |
| C01050/2 | 0.100 rev. | | L_CalcDiameter_1: Diameter recalculation 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01051**

| Parameter \| Name: **C01051 \| L_CalcDiameter_1: Filter time constant** | | | Data type: UNSIGNED_16 Index: $23524_d = 5BE4_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.010 | s | 3.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01051/1 | 1.000 s | | L_CalcDiameter_1: Filter time constant |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01052**

| Parameter \| Name: C01052 \| L_CalcDiameter_1: Web break monitoring | | | Data type: INTEGER_16 Index: $23523_d = 5BE3_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01052/1 | 10.00 % | | L_CalcDiameter_1: Permissible diameter change |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01053**

| Parameter \| Name: C01053 \| L_ProcessCtrl_1: Controller times | | | Data type: UNSIGNED_16 Index: $23522_d = 5BE2_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 30.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01053/1 | 0.000 s | | L_ProcessCtrl_1: Acceleration/deceleration time |
| C01053/2 | 0.000 s | | L_ProcessCtrl_1: Filter time constant |
| C01053/3 | 0.000 s | | L_ProcessCtrl_1: Rate time |
| C01053/4 | 1.000 s | | L_ProcessCtrl_1: Reset time |
| C01053/5 | 0.000 s | | L_ProcessCtrl_1: Rate action |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01054**

| Parameter \| Name: C01054 \| L_ProcessCtrl_1: System deviation | | | Data type: INTEGER_16 Index: $23521_d = 5BE1_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01054/1 | 100.00 % | | L_ProcessCtrl_1: Gain of system deviation |
| C01054/2 | 0.00 % | | L_ProcessCtrl_1: Area of system deviation |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01055**

| Parameter \| Name: C01055 \| L_ProcessCtrl_1: Correcting variable limitation | | Data type: UNSIGNED_8 Index: $23520_d = 5BE0_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01055/1 | 0: FALSE | L_ProcessCtrl_1: Correcting variable limitation |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01056**

| Parameter \| Name: **C01056 \| L_ProcessCtrl_1: Controller gain** | | | Data type: UNSIGNED_16 Index: $23519_d = 5BDF_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01056/1 | 0.10 | | L_ProcessCtrl_1: Controller gain |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01057**

| Parameter \| Name: **C01057 \| L_CalcDiameter_1: Current diameter** | | | Data type: UNSIGNED_32 Index: $23518_d = 5BDE_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.000 | mm | 10000.000 | |
| **Subcodes** | | | **Info** |
| C01057/1 | | | L_CalcDiameter_1: Current diameter |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01058**

| Parameter \| Name: **C01058 \| L_PosCtrlLin 1-2: Limit stop** | | Data type: UNSIGNED_8 Index: $23517_d = 5BDD_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | deactivated | |
| 1 | activated | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01058/1 | 0: Deactivated | L_PosCtrlLin_1: Limit stop |
| C01058/2 | 0: Deactivated | L_PosCtrlLin_2: Limit stop |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01059**

| Parameter \| Name: **C01059 \| L_PosCtrlLin 1-2: Positioning behaviour** | | Data type: UNSIGNED_8 Index: $23516_d = 5BDC_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | dnOut_p = 0 | |
| 1 | dnOut_p/nOut_v follow dnAct_p | |
| 2 | dnOut_p/nOut_v follow dnSet_p | |
| 3 | dnOut_p/nOut_v follow dnAct_p (without limitation) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01059/1 | 0: dnOut_p = 0 | L_PosCtrlLin_1: Positioning behaviour |
| C01059/2 | 0: dnOut_p = 0 | L_PosCtrlLin_2: Positioning behaviour |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01060**

| Parameter \| Name: **C01060 \| L_PosCtrlLin 1-2: Ramps** | | | Data type: INTEGER_32 Index: $23515_d = 5BDB_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.010 | s | 130.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01060/1 | 1.000 s | | L_PosCtrlLin_1: Deceleration of set position |
| C01060/2 | 1.000 s | | L_PosCtrlLin_1: Acceleration ramp |
| C01060/3 | 1.000 s | | L_PosCtrlLin_1: Deceleration ramp |
| C01060/4 | 1.000 s | | L_PosCtrlLin_2: Deceleration of set position |
| C01060/5 | 1.000 s | | L_PosCtrlLin_2: Acceleration ramp |
| C01060/6 | 1.000 s | | L_PosCtrlLin_2: Deceleration ramp |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01061**

| Parameter \| Name: **C01061 \| L_PosCtrlLin 1-2: Traversing speeds** | | | Data type: INTEGER_16 Index: $23514_d = 5BDA_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -15000 | rpm | 15000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01061/1 | 199 rpm | | L_PosCtrlLin_1: Forward motion |
| C01061/2 | 199 rpm | | L_PosCtrlLin_1: Return motion |
| C01061/3 | 199 rpm | | L_PosCtrlLin_2: Forward motion |
| C01061/4 | 199 rpm | | L_PosCtrlLin_2: Return motion |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01062**

| Parameter \| Name: **C01062 \| L_SwitchPoint_1: Dead time** | | | Data type: UNSIGNED_16 Index: $23513_d = 5BD9_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01062/1 | 0 | | L_SwitchPoint_1: Dead time 1 |
| C01062/2 | 0 | | L_SwitchPoint_1: Dead time 2 |
| C01062/3 | 0 | | L_SwitchPoint_1: Dead time 3 |
| C01062/4 | 0 | | L_SwitchPoint_1: Dead time 4 |
| C01062/5 | 0 | | L_SwitchPoint_1: Dead time 5 |
| C01062/6 | 0 | | L_SwitchPoint_1: Dead time 6 |
| C01062/7 | 0 | | L_SwitchPoint_1: Dead time 7 |
| C01062/8 | 0 | | L_SwitchPoint_1: Dead time 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01063**

| Parameter \| Name: **C01063 \| L_SwitchPoint_1: Hysteresis** | | | Data type: UNSIGNED_16 Index: $23512_d = 5BD8_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01063/1 | 0 incr. | | L_SwitchPoint_1: Hysteresis 1 |
| C01063/2 | 0 incr. | | L_SwitchPoint_1: Hysteresis 2 |
| C01063/3 | 0 incr. | | L_SwitchPoint_1: Hysteresis 3 |
| C01063/4 | 0 incr. | | L_SwitchPoint_1: Hysteresis 4 |
| C01063/5 | 0 incr. | | L_SwitchPoint_1: Hysteresis 5 |
| C01063/6 | 0 incr. | | L_SwitchPoint_1: Hysteresis 6 |
| C01063/7 | 0 incr. | | L_SwitchPoint_1: Hysteresis 7 |
| C01063/8 | 0 incr. | | L_SwitchPoint_1: Hysteresis 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01064**

| Parameter \| Name: **C01064 \| L_SwitchPoint_1: CenterMode** | | Data type: UNSIGNED_8 Index: $23511_d = 5BD7_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01064/1 | 0: FALSE | L_SwitchPoint_1: CenterMode 1 |
| C01064/2 | 0: FALSE | L_SwitchPoint_1: CenterMode 2 |
| C01064/3 | 0: FALSE | L_SwitchPoint_1: CenterMode 3 |
| C01064/4 | 0: FALSE | L_SwitchPoint_1: CenterMode 4 |
| C01064/5 | 0: FALSE | L_SwitchPoint_1: CenterMode 5 |
| C01064/6 | 0: FALSE | L_SwitchPoint_1: CenterMode 6 |
| C01064/7 | 0: FALSE | L_SwitchPoint_1: CenterMode 7 |
| C01064/8 | 0: FALSE | L_SwitchPoint_1: CenterMode 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01065**

| Parameter \| Name: **C01065 \| L_SwitchPoint_1: Running time** | | | Data type: UNSIGNED_16 Index: $23510_d = 5BD6_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01065/1 | 0 ms | | L_SwitchPoint_1: Running time 1 |
| C01065/2 | 0 ms | | L_SwitchPoint_1: Running time 2 |
| C01065/3 | 0 ms | | L_SwitchPoint_1: Running time 3 |
| C01065/4 | 0 ms | | L_SwitchPoint_1: Running time 4 |
| C01065/5 | 0 ms | | L_SwitchPoint_1: Running time 5 |
| C01065/6 | 0 ms | | L_SwitchPoint_1: Running time 6 |
| C01065/7 | 0 ms | | L_SwitchPoint_1: Running time 7 |
| C01065/8 | 0 ms | | L_SwitchPoint_1: Running time 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01066**

| Parameter \| Name: **C01066 \| L_SwitchPoint_1: Status** | | Data type: INTEGER_16 Index: $23509_d = 5BD5_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | OK | |
| 10 | FB not active | |
| 100 | Switching points not plausible | |
| **Subcodes** | | **Info** |
| C01066/1 | | L_SwitchPoint_1: Status 1 |
| C01066/2 | | L_SwitchPoint_1: Status 2 |
| C01066/3 | | L_SwitchPoint_1: Status 3 |
| C01066/4 | | L_SwitchPoint_1: Status 4 |
| C01066/5 | | L_SwitchPoint_1: Status 5 |
| C01066/6 | | L_SwitchPoint_1: Status 6 |
| C01066/7 | | L_SwitchPoint_1: Status 7 |
| C01066/8 | | L_SwitchPoint_1: Status 8 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01067**

| Parameter \| Name: **C01067 \| Inversion of gearbox stages** | | Data type: UNSIGNED_8 Index: $23508_d = 5BD4_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | not inverted | |
| 1 | inverted | |
| 2 | Automatically from MCK | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01067/1 | 0: Not inverted | L_PhiIntegrator_1: Invert. gearbox nSet_v |
| C01067/2 | 0: Not inverted | L_DFSET_1: Invert. gearbox nSet_v |
| C01067/3 | 0: Not inverted | L_CalcDiameter_1: Invert. gearbox nMotorSpeedAct_v |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01068**

| Parameter \| Name: **C01068 \| L_SwitchPoint_1: Invert** | | Data type: UNSIGNED_16 Index: $23507_d = 5BD3_h$ |
|---|---|---|
| From version 12.00.00 | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | 0xFFFF | |
| **Value is bit-coded:** | | **Info** |
| Bit 0 | Invert Output1 | Bit set = inversion active |
| Bit 1 | Invert Output2 | |
| Bit 2 | Invert Output3 | |
| Bit 3 | Invert Output4 | |
| Bit 4 | Invert Output5 | |
| Bit 5 | Invert Output6 | |
| Bit 6 | Invert Output7 | |
| Bit 7 | Invert Output8 | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01068/1 | 0x0000 | L_SwitchPoint_1: Inversion of outputs |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C01069**

| Parameter \| Name: **C01069 \| L_DFSET_1: Ramp settings** | | | Data type: UNSIGNED_16 Index: $23506_d = 5BD2_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr./ms | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01069/1 | 10 incr./ms | | L_DFSET_1: Ramp SpeedTrim (1/100)<br>• From version 12.00.00 onwards, this setting has a resolution increased by the factor 100: A value of 32767 is internally evaluated with 327.67. Hence, the unit is [incr./100 ms]. |
| C01069/2 | 100 incr./ms | | L_DFSET_1: Ramp angle compensation |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01070**

| Parameter \| Name: **C01070 \| L_DFSET_1: Angular trimming** | | | Data type: INTEGER_32 Index: $23505_d = 5BD1_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -134217728 | Incr. | 134217728 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01070/1 | 0 incr. | | L_DFSET_1: Speed-dependent angle adjustment |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01071**

| Parameter \| Name: **C01071 \| L_DFSET_1: Following error limit** | | | Data type: UNSIGNED_32 Index: $23504_d = 5BD0_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 10 | Incr. | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01071/1 | 32768 incr. | | L_DFSET_1: Following error limit |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01072**

| Parameter \| Name: **C01072 \| L_DFSET_1: Multiplier - angular trimming** | | | Data type: INTEGER_16 Index: $23503_d = 5BCF_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -20000 | | 20000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01072/1 | 1 | | L_DFSET_1: Multiplier - angular trimming |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01073**

| Parameter \| Name: **C01073 \| L_DFSET_1: Adjustment** | | | Data type: UNSIGNED_8 Index: $23502_d = 5BCE_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x00 | | 0xFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Setpoint angle without gearbox factor | | 0 ≡ Evaluation of the setpoint angle with gearbox factor<br>1 ≡ Evaluation of the setpoint angle without gearbox factor |
| Bit 1 | Angle correction with polynomial | | 0 ≡ Angle correction without downstream polynomial<br>1 ≡ Angle correction with downstream polynomial |
| Bit 2 | External angle correction | | 0 ≡ Internal angle correction<br>1 ≡ External angle correction<br>• From version 13.00.00 |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01073/1 | 0x02 | | L_DFSET_1: Adjustment |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01074**

| Parameter \| Name: **C01074 \| L_DFSET_1: Zero pulse divider** | | | Data type: UNSIGNED_16 Index: $23501_d = 5BCD_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 16384 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01074/1 | 0 | | L_DFSET_1: Divider for actual value zero pulse |
| C01074/2 | 0 | | L_DFSET_1: Divider for setpoint zero pulse |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01075**

| Parameter \| Name: **C01075 \| L_DFSET_1: Synchronisation mode** | | Data type: UNSIGNED_8 Index: $23500_d = 5BCC_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | Inactive | |
| 1 | Permanent sync. without enable (bZeroPulse) | |
| 2 | Permanent sync. with enable (bZeroPulse) | |
| 10 | 1x sync. - Angular diff. shortest path setpoint | |
| 11 | 1x Sync. - Angular diff. Cw | |
| 12 | 1x Sync. - Angular diff. Ccw | |
| 13 | 1x Sync. - Angular diff. shortest path act. value | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01075/1 | 0: Inactive | L_DFSET_1: Synchronisation mode |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01076**

| Parameter \| Name: **C01076 \| L_DFRFG_1: Times** | | | Data type: UNSIGNED_32 Index: $23499_d = 5BCB_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.900 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01076/1 | 1.000 s | | L_DFRFG_1: Acceleration and deceleration time |
| C01076/2 | 0.000 s | | L_DFRFG_1: Deceleration time for quick stop |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01077**

| Parameter \| Name: **C01077 \| L_DFRFG_1: Max. speed-up** | | | Data type: INTEGER_16 Index: $23498_d = 5BCA_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | rpm | 15000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01077/1 | 3000 rpm | | L_DFRFG_1: Max. speed-up |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01078**

| Parameter \| Name: **C01078 \| L_DFRFG_1: Following error** | | | Data type: UNSIGNED_32 Index: $23497_d = 5BC9_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 10 | Incr. | 2000000000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01078/1 | 2000000000 incr. | | L_DFRFG_1: Following error limit |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01079**

| Parameter \| Name: **C01079 \| L_DFRFG_1: Synchronisation window** | | | Data type: UNSIGNED_16 Index: $23496_d = 5BC8_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01079/1 | 100 incr. | | L_DFRFG_1: Synchronisation window (position) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01080**

| Parameter \| Name: **C01080 \| L_DFRFG_1: Offset** | | | Data type: INTEGER_32 Index: $23495_d = 5BC7_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01080/1 | 0 incr. | | L_DFRFG_1: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01081**

| Parameter \| Name: **C01081 \| L_DFRFG_1: Sync. direction / TP function** | | Data type: UNSIGNED_8 Index: $23494_d = 5BC6_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 1 | cw/ccw - without TP | |
| 2 | cw - without TP | |
| 3 | ccw - without TP | |
| 4 | cw/ccw - with TP | |
| 5 | cw - with TP | |
| 6 | ccw - with TP | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01081/1 | 1: cw/ccw - without TP | L_DFRFG_1: Sync. direction / TP function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01082**

| Parameter \| Name: **C01082 \| LS_WriteParamList: Execute Mode** | Data type: UNSIGNED_8 Index: $23493_d = 5BC5_h$ |
|---|---|
| Parameter change-over: Selection of the activation method | |

| Selection list (Lenze setting printed in bold) | | Info |
|---|---|---|
| **0** | **by Execute** | The writing of the parameter list is activated by a FALSE/TRUE edge at the *bExecute* input. |
| 1 | by Input Select | The writing of the parameter list is carried out if a change is made at the select inputs and if the inverter is initialised. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01083**

| Parameter \| Name: **C01083 \| LS_WriteParamList: Error status** | Data type: UNSIGNED_16 Index: $23492_d = 5BC4_h$ |
|---|---|

Parameter change-over: Error status:
- 0 = no error
- 33803 | 0x840B = invalid data type (e.g. STRING)
- 33804 | 0x840C = limit violation
- 33806 | 0x840E = invalid code
- 33813 | 0x8415 = no element of the selection list
- 33815 | 0x8417 = writing of the parameter not permitted
- 33816 | 0x8418 = writing of the parameter only permitted if controller is inhibited
- 33829 | 0x8425 = invalid subcode
- 33865 | 0x8449 = no parameter with subcodes

| Display area (min. value \| unit \| max. value) | | |
|---|---|---|
| 0 | | 34000 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01084**

| Parameter \| Name: **C01084 \| LS_WriteParamList: Error line** | Data type: UNSIGNED_8 Index: $23491_d = 5BC3_h$ |
|---|---|
| Parameter change-over: Display of the number of list entry where the error occurred (in connection with the value set selected via *bSelectWriteValue_1* and *bSelectWriteValue_2*). | |

| Display area (min. value \| unit \| max. value) | | |
|---|---|---|
| 0 | | 32 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01085**

| Parameter \| Name: **C01085 \| LS_WriteParamList: Index** | Data type: INTEGER_32 Index: $23490_d = 5BC2_h$ |
|---|---|
| Parameter change-over: Parameter for entry 1 ... 32 | |

| Setting range (min. value \| unit \| max. value) | | |
|---|---|---|
| 0.000 | | 16000.000 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01085/1 | 0.000 | Parameter for entries 1 ... 32<br>• Format: <code number>.<subcode number><br>• Examples: "12.000" = C00012; "26.001" = C00026/1 |
| C01085/... | | |
| C01085/32 | | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C01086**

| Parameter \| Name: **C01086 \| LS_WriteParamList: WriteValue_1** | | | Data type: INTEGER_32 Index: $23489_d = 5BC1_h$ |
|---|---|---|---|
| Parameter change-over: Parameter values - value set 1 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01086/1 | 0 | | Parameter values - value set 1<br>• Parameter values for the parameters defined in C01085/1 ... 32. |
| C01086/... | | | |
| C01086/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01087**

| Parameter \| Name: **C01087 \| LS_WriteParamList: WriteValue_2** | | | Data type: INTEGER_32 Index: $23488_d = 5BC0_h$ |
|---|---|---|---|
| Parameter change-over: Parameter values - value set 2 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01087/1 | 0 | | Parameter values - value set 2<br>• Parameter values for the parameters defined in C01085/1 ... 32. |
| C01087/... | | | |
| C01087/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01088**

| Parameter \| Name: **C01088 \| LS_WriteParamList: WriteValue_3** | | | Data type: INTEGER_32 Index: $23487_d = 5BBF_h$ |
|---|---|---|---|
| Parameter change-over: Parameter values - value set 3 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01088/1 | 0 | | Parameter values - value set 3<br>• Parameter values for the parameters defined in C01085/1 ... 32. |
| C01088/... | | | |
| C01088/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01089**

| Parameter \| Name: **C01089 \| LS_WriteParamList: WriteValue_4** | | | Data type: INTEGER_32 Index: $23486_d = 5BBE_h$ |
|---|---|---|---|
| Parameter change-over: Parameter values - value set 4 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01089/1 | 0 | | Parameter values - value set 4<br>• Parameter values for the parameters defined in C01085/1 ... 32. |
| C01089/... | | | |
| C01089/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01090**

| Parameter \| Name: **C01090 \| LS_ParReadWrite 1-6: Index** | | | Data type: INTEGER_32 Index: $23485_d = 5BBD_h$ |
|---|---|---|---|
| Parameter to be read or written.<br>• Format: <code number>,<subcode number><br>• For a setting of "0,000", inputs *wParIndex* and *wParSubindex* are effective for addressing purposes instead. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | | 16000.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01090/1 | 0.000 | | LS_ParReadWrite_1: Index |
| C01090/2 | 0.000 | | LS_ParReadWrite_2: Index |
| C01090/3 | 0.000 | | LS_ParReadWrite_3: Index |
| C01090/4 | 0.000 | | LS_ParReadWrite_4: Index |
| C01090/5 | 0.000 | | LS_ParReadWrite_5: Index |
| C01090/6 | 0.000 | | LS_ParReadWrite_6: Index |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01091**

| Parameter \| Name: **C01091 \| LS_ParReadWrite 1-6: Cycle time** | | Data type: UNSIGNED_16 Index: $23484_d = 5BBC_h$ |
|---|---|---|
| Time interval for cyclic reading/writing | | |
| **Selection list** | | |
| 0 | 0 (by Execute) | |
| 20 | 20 ms | |
| 50 | 50 ms | |
| 100 | 100 ms | |
| 200 | 200 ms | |
| 500 | 500 ms | |
| 1000 | 1 s | |
| 2000 | 2 s | |
| 5000 | 5 s | |
| 10000 | 10 s | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01091/1 | 0: 0 (by Execute) | LS_ParReadWrite_1: Cycle time |
| C01091/2 | 0: 0 (by Execute) | LS_ParReadWrite_2: Cycle time |
| C01091/3 | 0: 0 (by Execute) | LS_ParReadWrite_3: Cycle time |
| C01091/4 | 0: 0 (by Execute) | LS_ParReadWrite_4: Cycle time |
| C01091/5 | 0: 0 (by Execute) | LS_ParReadWrite_5: Cycle time |
| C01091/6 | 0: 0 (by Execute) | LS_ParReadWrite_6: Cycle time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01092**

| Parameter \| Name: **C01092 \| LS_ParReadWrite 1-6: FailState** | | Data type: UNSIGNED_16 Index: $23483_d = 5BBB_h$ |
|---|---|---|
| Error status:<br>• 0 = no error<br>• 33803 \| 0x840B = invalid data type (e.g. STRING)<br>• 33804 \| 0x840C = limit violation<br>• 33806 \| 0x840E = invalid code<br>• 33813 \| 0x8415 = no element of the selection list<br>• 33815 \| 0x8417 = writing of the parameter not permitted<br>• 33816 \| 0x8418 = writing of the parameter only permitted if controller is inhibited<br>• 33829 \| 0x8425 = invalid subcode<br>• 33865 \| 0x8449 = no parameter with subcodes | | |
| **Display area** (min. value \| unit \| max. value) | | |
| 0 | | 34000 |
| **Subcodes** | | **Info** |
| C01092/1 | | LS_ParReadWrite_1: Error status |
| C01092/2 | | LS_ParReadWrite_2: Error status |
| C01092/3 | | LS_ParReadWrite_3: Error status |
| C01092/4 | | LS_ParReadWrite_4: Error status |
| C01092/5 | | LS_ParReadWrite_5: Error status |
| C01092/6 | | LS_ParReadWrite_6: Error status |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01093**

| Parameter \| Name: **C01093 - LS_ParReadWrite 1-6: Arithmetic mode** | | Data type: UNSIGNED_8 Index: $23482_d = 5BBA_h$ |
|---|---|---|
| The integrated arithmetic function allows for easy arithmetic conversion of the process value to be written or which was read into the format of the target parameter via parameterisable factors and without the need for an additional arithmetic FB. | | |
| **Selection list** | | |
| 0 | No arithmetic | |
| 1 | In16Bit: LW=+/-32767 | |
| 2 | In16Bit: HW=+/-; LW=0..65535 | |
| 3 | In32Bit: HW_LW=+/-2147483647 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01093/1 | 0: no arithmetic | LS_ParReadWrite_1: Arithmetic mode |
| C01093/2 | 0: no arithmetic | LS_ParReadWrite_2: Arithmetic mode |
| C01093/3 | 0: no arithmetic | LS_ParReadWrite_3: Arithmetic mode |
| C01093/4 | 0: no arithmetic | LS_ParReadWrite_4: Arithmetic mode |
| C01093/5 | 0: no arithmetic | LS_ParReadWrite_5: Arithmetic mode |
| C01093/6 | 0: no arithmetic | LS_ParReadWrite_6: Arithmetic mode |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01094**

| Parameter \| Name: **C01094 - LS_ParReadWrite 1-6: Numerator** | | Data type: INTEGER_16 Index: $23481_d = 5BB9_h$ |
|---|---|---|
| Arithmetic function: Factor (numerator) for internal conversion in arithmetic modes 1 ... 3. | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -32767 | 32767 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01094/1 | 1 | LS_ParReadWrite_1: Numerator |
| C01094/2 | 1 | LS_ParReadWrite_2: Numerator |
| C01094/3 | 1 | LS_ParReadWrite_3: Numerator |
| C01094/4 | 1 | LS_ParReadWrite_4: Numerator |
| C01094/5 | 1 | LS_ParReadWrite_5: Numerator |
| C01094/6 | 1 | LS_ParReadWrite_6: Numerator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01095**

| Parameter \| Name: **C01095 - LS_ParReadWrite 1-6: Denominator** | | | Data type: INTEGER_16 Index: $23480_d = 5BB8_h$ |
|---|---|---|---|
| Arithmetic function: Factor (denominator) for internal conversion in arithmetic modes 1 ... 3. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01095/1 | 1 | | LS_ParReadWrite_1: Denominator |
| C01095/2 | 1 | | LS_ParReadWrite_2: Denominator |
| C01095/3 | 1 | | LS_ParReadWrite_3: Denominator |
| C01095/4 | 1 | | LS_ParReadWrite_4: Denominator |
| C01095/5 | 1 | | LS_ParReadWrite_5: Denominator |
| C01095/6 | 1 | | LS_ParReadWrite_6: Denominator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01098**

| Parameter \| Name: **C01098 \| LS_ParReadWrite 1-6: Configuration** | | | Data type: UNSIGNED_16 Index: $23477_d = 5BB5_h$ |
|---|---|---|---|
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Keeping the value | | |
| Bit 1 | Keeping the value in the event of an error | | |
| Bit 2 | Reserved | | |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01098/1 | 0x0000 | | LS_ParReadWrite_1: Configuration |
| C01098/2 | 0x0000 | | LS_ParReadWrite_2: Configuration |
| C01098/3 | 0x0000 | | LS_ParReadWrite_3: Configuration |
| C01098/4 | 0x0000 | | LS_ParReadWrite_4: Configuration |
| C01098/5 | 0x0000 | | LS_ParReadWrite_5: Configuration |
| C01098/6 | 0x0000 | | LS_ParReadWrite_6: Configuration |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01100**

| Parameter \| Name: **C01100 \| Function L_Counter 1-3** | | Data type: UNSIGNED_8 Index: $23475_d = 5BB3_h$ |
|---|---|---|
| Selection of reset function | | |
| **Selection list** | | |
| 0 | Normal counting | |
| 1 | Auto reset | |
| 2 | Manual reset | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01100/1 | 0: Normal counting | L_Counter_1: Function |
| C01100/2 | 0: Normal counting | L_Counter_2: Function |
| C01100/3 | 0: Normal counting | L_Counter_3: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01101**

| Parameter \| Name: **C01101 \| Comparison L_Counter 1-3** | | Data type: UNSIGNED_8 Index: $23474_d = 5BB2_h$ |
|---|---|---|
| Selection of comparison operation | | |
| **Selection list** | | |
| 0 | Greater than or equal to | |
| 1 | Less than or equal to | |
| 2 | equal to | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01101/1 | 0: Greater than or equal to | L_Counter_1: Comparison |
| C01101/2 | 0: Greater than or equal to | L_Counter_2: Comparison |
| C01101/3 | 0: Greater than or equal to | L_Counter_3: Comparison |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01108**

| Parameter \| Name: **C01108 \| L_SwitchPoint_1: Dead time factor** | | Data type: UNSIGNED_8 Index: $23467_d = 5BAB_h$ |
|---|---|---|
| From version 14.00.00 | | |
| **Selection list** | | |
| 1 | 1 µs | |
| 2 | 10 µs | |
| 3 | 100 µs | |
| 4 | 1000 µs | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01108/1 | 2: 10 µs | L_SwitchPoint_1: Deadtime factor 1 |
| C01108/2 | 2: 10 µs | L_SwitchPoint_1: Deadtime factor 2 |
| C01108/3 | 2: 10 µs | L_SwitchPoint_1: Deadtime factor 3 |
| C01108/4 | 2: 10 µs | L_SwitchPoint_1: Deadtime factor 4 |
| C01108/5 | 2: 10 µs | L_SwitchPoint_1: Deadtime factor 5 |
| C01108/6 | 2: 10 µs | L_SwitchPoint_1: Deadtime factor 6 |
| C01108/7 | 2: 10 µs | L_SwitchPoint_1: Deadtime factor 7 |
| C01108/8 | 2: 10 µs | L_SwitchPoint_1: Deadtime factor 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01109**

| Parameter \| Name: **C01109 \| L_SwitchPointPar_1: Dead time factor** | | Data type: UNSIGNED_8 Index: $23466_d = 5BAA_h$ |
|---|---|---|
| From version 14.00.00 | | |
| **Selection list** | | |
| 1 | 1 µs | |
| 2 | 10 µs | |
| 3 | 100 µs | |
| 4 | 1000 µs | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01109/1 | 2: 10 µs | L_SwitchPointPar_1: Dead time factor 1 |
| C01109/2 | 2: 10 µs | L_SwitchPointPar_1: Dead time factor 2 |
| C01109/3 | 2: 10 µs | L_SwitchPointPar_1: Dead time factor 3 |
| C01109/4 | 2: 10 µs | L_SwitchPointPar_1: Dead time factor 4 |
| C01109/5 | 2: 10 µs | L_SwitchPointPar_1: Dead time factor 5 |
| C01109/6 | 2: 10 µs | L_SwitchPointPar_1: Dead time factor 6 |
| C01109/7 | 2: 10 µs | L_SwitchPointPar_1: Dead time factor 7 |
| C01109/8 | 2: 10 µs | L_SwitchPointPar_1: Dead time factor 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01110**

| Parameter \| Name: C01110 \| LS_MultiEncoder: Solid measure | | Data type: UNSIGNED_8 Index: $23465_d = 5BA9_h$ |
|---|---|---|
| From version 02.00.00<br>Configuration examples can be found in the following chapters:<br>• SinCos absolute value encoder with HIPERFACE® protocol<br>• SSI absolute value encoder with Stegmann-SSI protocol | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **rotatively unipolar** | |
| 1 | linearly unipolar | |
| 2 | rotatively bipolar | |
| 3 | linearly bipolar | |
| 4 | Inverted rotatively in unipolar mode | From version 15.00.00 |
| 5 | Inverted linearly in unipolar mode | |
| 6 | Inverted rotatively in bipolar mode | From version 15.00.00 |
| 7 | Inverted linearly in bipolar mode | From version 15.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C01111**

| Parameter \| Name: C01111 \| LS_MultiEncoder: Encoder constant | | Data type: INTEGER_32 Index: $23464_d = 5BA8_h$ |
|---|---|---|
| From version 02.00.00<br>Configuration examples can be found in the following chapters:<br>• SinCos absolute value encoder with HIPERFACE® protocol<br>• SSI absolute value encoder with Stegmann-SSI protocol | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 1 | 2147483647 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01111/1 | 1 | LS_MultiEncoder: Difference - traverse path |
| C01111/2 | 1 | LS_MultiEncoder: Difference - encoder value |
| C01111/3 | 1 | LS_MultiEncoder: Traverse path - encoder increment |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C01112**

| Parameter \| Name: C01112 \| LS_MultiEncoder: Position values | | Data type: INTEGER_32 Index: $23463_d = 5BA7_h$ |
|---|---|---|
| ▸ Encoder/feedback system: Multi-Encoder | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -214748.3647 units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01112/1 | 0.0000 units | LS_MultiEncoder: Position offset |
| C01112/2 | -214748.3647 units | LS_MultiEncoder: Data area min |
| C01112/3 | 214748.3647 units | LS_MultiEncoder: Data area max |
| C01112/4 | 0.0000 units | LS_MultiEncoder: Max. position jump<br>• From version 14.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10000 | | |

**C01119**

| Parameter \| Name: **C01119 \| LS_MultiEncoder: Current position** | | | Data type: INTEGER_32 Index: $23456_d = 5BA0_h$ |
|---|---|---|---|
| From version 02.00.00 | | | ▸ Encoder/feedback system: Multi-Encoder |
| **Display area** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01119/1 | | | LS_MultiEncoder: Current position |
| C01119/2 | | | LS_MultiEncoder: Maximum travel distance |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01120**

| Parameter \| Name: **C01120 \| Sync signal source** | | Data type: UNSIGNED_8 Index: $23455_d = 5B9F_h$ |
|---|---|---|
| Selection of the signal source for device synchronisation<br>• The drive can only be synchronised by one source. | | ▸ Synchronisation of the internal time base |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Off** | Synchronisation off |
| 1 | CAN on board | Synchronisation via CAN bus<br>▸ Sync telegram |
| 2 | AxisBusIO | Synchronisation via axis bus<br>▸ Axis bus |
| 4 | MCI | Synchronisation via MCI (communication module) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01121**

| Parameter \| Name: **C01121 \| Sync cycle time setpoint** | | | Data type: UNSIGNED_16 Index: $23454_d = 5B9E_h$ |
|---|---|---|---|
| Cycle time setpoint for device synchronisation<br>• Time interval at which the phase control loop (PLL) in the inverter expects the synchronisation signals.<br>• The cycle time setpoint must be set according to the cycle of the respective synchronisation source.<br>**Note:**<br>• Only integer multiples of 1000 µs can be set.<br>• Intelligent communication modules usually define the cycle time setpoint derived from the bus cycle. In this case, a manual change is not possible.<br>Example: For the CAN bus, a distance of 2 ms has been set between two synchronisation signals. If the CAN bus is to be used as synchronisation source, a synchronisation cycle of 2000 µs must be set in C01121. | | | ▸ Synchronisation of the internal time base |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1000 | µs | 20000 | **1000 µs** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01122**

| Parameter \| Name: **C01122 \| Sync phase position** | Data type: UNSIGNED_16 Index: $23453_d = 5B9D_h$ |
|---|---|

Phase position for device synchronisation
- The phase position determines the zero-time of the internal system cycle with regard to the synchronisation signal (bus cycle). Since PDO processing is an inherent part of the system part of the application, the instant of acceptance of the PDOs is postponed as well by a changed phase position.
- With a setting = 0, the system cycle starts simultaneously with the synchronisation signal.
- With a setting > 0, the internal system cycle starts earlier by the set time with regard to the synchronisation signal (the phase position acts negatively).
- Intelligent communication modules define the optimal time with activated synchronisation by themselves. In this case, a manual change is not possible.
- The decisive factor for defining C01122 is the time where all nodes are provided with valid PDOs.

Example: If the phase position is set to 550 µs, the system part of the application starts 550 µs before the arrival of the synchronisation signal.

▸ Synchronisation of the internal time base

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0 | µs | 1000 | **0 µs** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01123**

| Parameter \| Name: **C01123 \| Sync window** | Data type: UNSIGNED_16 Index: $23452_d = 5B9C_h$ |
|---|---|

Time slot for monitoring the synchronisation signal or the phase position
- The synchronisation signal or the current phase position must be within this time slot around the corresponding expected value (C01122).
- With the setting "1000 µs" there will be no monitoring.

▸ Synchronisation of the internal time base

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0 | µs | 10000 | **100 µs** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01124**

| Parameter \| Name: **C01124 \| Sync correction width** | Data type: UNSIGNED_8 Index: $23451_d = 5B9B_h$ |
|---|---|

Correction increment for device synchronisation
- If the cycle times of the synchronisation signal differs and phase-locken loop (PLL) differ from each other, this setting defines the measure the phase-locking loop is reset with.
- If synchronisation is not reached, select a higher correction constant.
- The optimum setting depends on quartz precision and must be determined empirically if required.

▸ Synchronisation of the internal time base

| Selection list (Lenze setting printed in bold) | |
|---|---|
| 1 | 80ns |
| 2 | 160ns |
| 3 | 240ns |
| **4** | **320ns** |
| 5 | 400ns |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01130**

| Parameter \| Name: **C01130 \| LS_RetainData: Selection** | | | Data type: UNSIGNED_16 Index: $23445_d = 5B95_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | wIn1 | | |
| Bit 1 | wIn2 | | |
| Bit 2 | wIn3 | | |
| Bit 3 | wIn4 | | |
| Bit 4 | dnIn1 | | |
| Bit 5 | dnIn2 | | |
| Bit 6 | dnIn3 | | |
| Bit 7 | dnIn4 | | |
| Bit 8 | bIn1 | | |
| Bit 9 | bIn2 | | |
| Bit 10 | bIn3 | | |
| Bit 11 | bIn4 | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01130/1 | 0x000F | | LS_RetainData: Selection bSetRetain_1 |
| C01130/2 | 0x00F0 | | LS_RetainData: Selection bSetRetain_2 |
| C01130/3 | 0x0F00 | | LS_RetainData: Selection bSetRetain_3 |
| C01130/4 | 0x0FFF | | LS_RetainData: Selection bLoadParams |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01131**

| Parameter \| Name: **C01131 \| LS_RetainData: 16Bit data** | | | Data type: UNSIGNED_16 Index: $23444_d = 5B94_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01131/1 | 0 | | LS_RetainData: wInitValue1 |
| C01131/2 | 0 | | LS_RetainData: wInitValue2 |
| C01131/3 | 0 | | LS_RetainData: wInitValue3 |
| C01131/4 | 0 | | LS_RetainData: wInitValue4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01132**

| Parameter \| Name: C01132 \| LS_RetainData: 32Bit data | | | Data type: INTEGER_32 Index: $23443_d = 5B93_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01132/1 | 0 | | LS_RetainData: dnInitValue1 |
| C01132/2 | 0 | | LS_RetainData: dnInitValue2 |
| C01132/3 | 0 | | LS_RetainData: dnInitValue3 |
| C01132/4 | 0 | | LS_RetainData: dnInitValue4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01133**

| Parameter \| Name: C01133 \| LS_RetainData: Bool data | | Data type: UNSIGNED_8 Index: $23442_d = 5B92_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01133/1 | 0: FALSE | LS_RetainData: bInitValue1 |
| C01133/2 | 0: FALSE | LS_RetainData: bInitValue2 |
| C01133/3 | 0: FALSE | LS_RetainData: bInitValue3 |
| C01133/4 | 0: FALSE | LS_RetainData: bInitValue4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01134**

| Parameter \| Name: C01134 \| LS_RetainData: 16bit data | | | Data type: UNSIGNED_16 Index: $23441_d = 5B91_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01134/1 | 0 | | LS_RetainData: wValue1 |
| C01134/2 | 0 | | LS_RetainData: wValue2 |
| C01134/3 | 0 | | LS_RetainData: wValue3 |
| C01134/4 | 0 | | LS_RetainData: wValue4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01135**

| Parameter \| Name: **C01135 \| LS_RetainData: 32bit data** | | Data type: INTEGER_32 Index: $23440_d = 5B90_h$ |
|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | |
| -2147483647 | 2147483647 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01135/1 | 0 | LS_RetainData: dnValue1 |
| C01135/2 | 0 | LS_RetainData: dnValue2 |
| C01135/3 | 0 | LS_RetainData: dnValue3 |
| C01135/4 | 0 | LS_RetainData: dnValue4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01136**

| Parameter \| Name: **C01136 \| LS_RetainData: Bool data** | | Data type: UNSIGNED_8 Index: $23439_d = 5B8F_h$ |
|---|---|---|
| **Selection list** | | **Info** |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01136/1 | 0: FALSE | LS_RetainData: bValue1 |
| C01136/2 | 0: FALSE | LS_RetainData: bValue2 |
| C01136/3 | 0: FALSE | LS_RetainData: bValue3 |
| C01136/4 | 0: FALSE | LS_RetainData: bValue4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01138**

| Parameter \| Name: **C01138 \| L_Transient 1-4: Function** | | Data type: UNSIGNED_8 Index: $23437_d = 5B8D_h$ |
|---|---|---|
| Selection of edge evaluation | | |
| **Selection list** | | |
| 0 | High edge | |
| 1 | Low edge | |
| 2 | High and low edge | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01138/1 | 0: High edge | L_Transient_1: Function |
| C01138/2 | 0: High edge | L_Transient_2: Function |
| C01138/3 | 0: High edge | L_Transient_3: Function |
| C01138/4 | 0: High edge | L_Transient_4: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01139**

| Parameter \| Name: **C01139 \| L_Transient 1-4: Pulse duration** | | | Data type: UNSIGNED_16 Index: $23436_d = 5B8C_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 60.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01139/1 | 0.000 s | | L_Transient_1: Pulse duration |
| C01139/2 | 0.000 s | | L_Transient_2: Pulse duration |
| C01139/3 | 0.000 s | | L_Transient_3: Pulse duration |
| C01139/4 | 0.000 s | | L_Transient_4: Pulse duration |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01140**

| Parameter \| Name: **C01140 \| L_Transient 5-8: Function** | | Data type: UNSIGNED_8 Index: $23435_d = 5B8B_h$ |
|---|---|---|
| Selection of edge evaluation | | |
| **Selection list** | | |
| 0 | High edge | |
| 1 | Low edge | |
| 2 | High and low edge | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01140/1 | 0: High edge | L_Transient_5: Function |
| C01140/2 | 0: High edge | L_Transient_6: Function |
| C01140/3 | 0: High edge | L_Transient_7: Function |
| C01140/4 | 0: High edge | L_Transient_8: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01141**

| Parameter \| Name: **C01141 \| L_Transient 5-8 pulse duration** | | | Data type: UNSIGNED_16 Index: $23434_d = 5B8A_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 60.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01141/1 | 0.000 s | | L_Transient_5: Pulse duration |
| C01141/2 | 0.000 s | | L_Transient_6: Pulse duration |
| C01141/3 | 0.000 s | | L_Transient_7: Pulse duration |
| C01141/4 | 0.000 s | | L_Transient_8: Pulse duration |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01150**

| Parameter \| Name: **C01150 \| L_PhaseIntK: Function** | | Data type: UNSIGNED_8 Index: $23425_d = 5B81_h$ |
|---|---|---|
| Loading behaviour of the integrator | | |
| **Selection list** | | **Info** |
| 0 | Loading with level | Load integrator with TRUE level at the input *bLoad* |
| 1 | Loading with edge | Load integrator with FALSE/TRUE edge at the input *bLoad* |
| 2 | Loading with level + reset | Load integrator when the comparison value (C01151) is reached or with TRUE level at the *bLoad* input |
| **Subcodes** | **Lenze setting** | **Info** |
| C01150/1 | 0: Loading with level | L_PhaseIntK_1: Function |
| C01150/2 | 0: Loading with level | L_PhaseIntK_2: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01151**

| Parameter \| Name: **C01151 \| L_PhaseIntK: Compare** | | | Data type: INTEGER_32 Index: $23424_d = 5B80_h$ |
|---|---|---|---|
| Comparison value | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 2147418112 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01151/1 | 0 | | L_PhaseIntK_1: Comparison value |
| C01151/2 | 0 | | L_PhaseIntK_2: Comparison value |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01152**

| Parameter \| Name: **C01152 \| L_SwitchPointPar_1: Dead time** | | | Data type: UNSIGNED_16 Index: $23423_d = 5B7F_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01152/1 | 0 | | L_SwitchPointPar_1: Dead time 1 |
| C01152/2 | 0 | | L_SwitchPointPar_1: Dead time 2 |
| C01152/3 | 0 | | L_SwitchPointPar_1: Dead time 3 |
| C01152/4 | 0 | | L_SwitchPointPar_1: Dead time 4 |
| C01152/5 | 0 | | L_SwitchPointPar_1: Dead time 5 |
| C01152/6 | 0 | | L_SwitchPointPar_1: Dead time 6 |
| C01152/7 | 0 | | L_SwitchPointPar_1: Dead time 7 |
| C01152/8 | 0 | | L_SwitchPointPar_1: Dead time 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01153**

| Parameter \| Name: **C01153 \| L_SwitchPointPar_1: Hysteresis** | | | Data type: UNSIGNED_16 Index: $23422_d = 5B7E_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01153/1 | 0 incr. | | L_SwitchPointPar_1: Hysteresis 1 |
| C01153/2 | 0 incr. | | L_SwitchPointPar_1: Hysteresis 2 |
| C01153/3 | 0 incr. | | L_SwitchPointPar_1: Hysteresis 3 |
| C01153/4 | 0 incr. | | L_SwitchPointPar_1: Hysteresis 4 |
| C01153/5 | 0 incr. | | L_SwitchPointPar_1: Hysteresis 5 |
| C01153/6 | 0 incr. | | L_SwitchPointPar_1: Hysteresis 6 |
| C01153/7 | 0 incr. | | L_SwitchPointPar_1: Hysteresis 7 |
| C01153/8 | 0 incr. | | L_SwitchPointPar_1: Hysteresis 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01154**

| Parameter \| Name: **C01154 \| L_SwitchPointPar_1: CenterMode** | | Data type: UNSIGNED_8 Index: $23421_d = 5B7D_h$ |
|---|---|---|
| From version 12.00.00 | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01154/1 | 0: FALSE | L_SwitchPointPar_1: CenterMode 1 |
| C01154/2 | 0: FALSE | L_SwitchPointPar_1: CenterMode 2 |
| C01154/3 | 0: FALSE | L_SwitchPointPar_1: CenterMode 3 |
| C01154/4 | 0: FALSE | L_SwitchPointPar_1: CenterMode 4 |
| C01154/5 | 0: FALSE | L_SwitchPointPar_1: CenterMode 5 |
| C01154/6 | 0: FALSE | L_SwitchPointPar_1: CenterMode 6 |
| C01154/7 | 0: FALSE | L_SwitchPointPar_1: CenterMode 7 |
| C01154/8 | 0: FALSE | L_SwitchPointPar_1: CenterMode 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01155**

| Parameter \| Name: **C01155 \| L_SwitchPointPar_1: Running time** | | | Data type: UNSIGNED_16 Index: $23420_d = 5B7C_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01155/1 | 0 ms | | L_SwitchPointPar_1: Running time 1 |
| C01155/2 | 0 ms | | L_SwitchPointPar_1: Running time 2 |
| C01155/3 | 0 ms | | L_SwitchPointPar_1: Running time 3 |
| C01155/4 | 0 ms | | L_SwitchPointPar_1: Running time 4 |
| C01155/5 | 0 ms | | L_SwitchPointPar_1: Running time 5 |
| C01155/6 | 0 ms | | L_SwitchPointPar_1: Running time 6 |
| C01155/7 | 0 ms | | L_SwitchPointPar_1: Running time 7 |
| C01155/8 | 0 ms | | L_SwitchPointPar_1: Running time 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01156**

| Parameter \| Name: **C01156 \| L_SwitchPointPar_1: Status** | | Data type: INTEGER_16 Index: $23419_d = 5B7B_h$ |
|---|---|---|
| From version 12.00.00 | | |
| **Selection list** | | |
| 0 | OK | |
| 10 | FB not active | |
| 100 | Switching points not plausible | |
| **Subcodes** | | **Info** |
| C01156/1 | | L_SwitchPointPar_1: Status 1 |
| C01156/2 | | L_SwitchPointPar_1: Status 2 |
| C01156/3 | | L_SwitchPointPar_1: Status 3 |
| C01156/4 | | L_SwitchPointPar_1: Status 4 |
| C01156/5 | | L_SwitchPointPar_1: Status 5 |
| C01156/6 | | L_SwitchPointPar_1: Status 6 |
| C01156/7 | | L_SwitchPointPar_1: Status 7 |
| C01156/8 | | L_SwitchPointPar_1: Status 8 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01157**

| Parameter \| Name: **C01157 \| L_SwitchPointPar_1:Position** | | | Data type: INTEGER_32 Index: $23418_d = 5B7A_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01157/1 | 0.0000 units | | L_SwitchPointPar_1: dnSwitchPoint1_p |
| C01157/2 | 0.0000 units | | L_SwitchPointPar_1: dn2ndPoint_Size1_p |
| C01157/3 | 0.0000 units | | L_SwitchPointPar_1: dnSwitchPoint2_p |
| C01157/4 | 0.0000 units | | L_SwitchPointPar_1: dn2ndPoint_Size2_p |
| C01157/5 | 0.0000 units | | L_SwitchPointPar_1: dnSwitchPoint3_p |
| C01157/6 | 0.0000 units | | L_SwitchPointPar_1: dn2ndPoint_Size3_p |
| C01157/7 | 0.0000 units | | L_SwitchPointPar_1: dnSwitchPoint4_p |
| C01157/8 | 0.0000 units | | L_SwitchPointPar_1: dn2ndPoint_Size4_p |
| C01157/9 | 0.0000 units | | L_SwitchPointPar_1: dnSwitchPoint5_p |
| C01157/10 | 0.0000 units | | L_SwitchPointPar_1: dn2ndPoint_Size5_p |
| C01157/11 | 0.0000 units | | L_SwitchPointPar_1: dnSwitchPoint6_p |
| C01157/12 | 0.0000 units | | L_SwitchPointPar_1: dn2ndPoint_Size6_p |
| C01157/13 | 0.0000 units | | L_SwitchPointPar_1: dnSwitchPoint7_p |
| C01157/14 | 0.0000 units | | L_SwitchPointPar_1: dn2ndPoint_Size7_p |
| C01157/15 | 0.0000 units | | L_SwitchPointPar_1: dnSwitchPoint8_p |
| C01157/16 | 0.0000 units | | L_SwitchPointPar_1: dn2ndPoint_Size8_p |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01158**

| Parameter \| Name: **C01158 \| L_SwitchPointPar_1: Invert** | | | Data type: UNSIGNED_16 Index: $23417_d = 5B79_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Invert Output1 | | Bit set = inversion active |
| Bit 1 | Invert Output2 | | |
| Bit 2 | Invert Output3 | | |
| Bit 3 | Invert Output4 | | |
| Bit 4 | Invert Output5 | | |
| Bit 5 | Invert Output6 | | |
| Bit 6 | Invert Output7 | | |
| Bit 7 | Invert Output8 | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01158/1 | 0x0000 | | L_SwitchPointPar_1: Inversion of outputs |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01190**

| Parameter \| Name: **C01190 \| Motor thermal sensor** | | Data type: UNSIGNED_8 Index: $23385_d = 5B59_h$ |
|---|---|---|
| Selection of the thermal sensor for monitoring the motor temperature<br>▸ Encoder/feedback system: Motor temperature monitoring (KTY) | | |
| **Selection list** | | **Info** |
| 0 | KTY83-110 | Lenze standard KTY83-110 (MDSKX, MCS06) |
| 1 | Spec. characteristic | Characteristic defined via C01191 and C01192 |
| 2 | KTY83-110 + 2 x PTC | Lenze standard KTY83-110 + 2 x PTC 150°C (MCS09-MCS19) |
| 4 | KTY84-130 | Lenze standard KTY83-130 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01190/1 | 0: KTY83-110 | Type of motor temperature sensor resolver |
| C01190/2 | 0: KTY83-110 | Type of motor temperature sensor MultiEncoder |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C01191**

| Parameter \| Name: **C01191 \| PTC characteristic: Temperature 1/2** | | Data type: UNSIGNED_8<br>Index: $23384_d = 5B58_h$ |
|---|---|---|
| The spec. thermal sensor characteristic is selected through the setting C01190="1"<br>▸ Encoder/feedback system: Motor temperature monitoring (KTY) | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | °C | 255 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01191/1 | 100 °C | PTC characteristic: Temperature 1 |
| C01191/2 | 150 °C | PTC characteristic: Temperature 2 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C01192**

| Parameter \| Name: **C01192 \| PTC characteristic: Resistance 1/2** | | Data type: INTEGER_32<br>Index: $23383_d = 5B57_h$ |
|---|---|---|
| The spec. thermal sensor characteristic is selected through the setting C01190="1"<br>▸ Encoder/feedback system: Motor temperature monitoring (KTY) | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | Ohm | 30000 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01192/1 | 1070 Ohms | PTC characteristic: Resistance 1 |
| C01192/2 | 2225 Ohm | PTC characteristic: Resistance 2 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C01193**

| Parameter \| Name: **C01193 \| Motor temp. feedback system** | Data type: UNSIGNED_8<br>Index: $23382_d = 5B56_h$ |
|---|---|
| Selection of the feedback system for monitoring the motor temperature<br>▸ Encoder/feedback system: Motor temperature monitoring (KTY) | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **Speed feedback** |
| 1 | Resolver input |
| 2 | Encoder input |
| 5 | Res. and enc. parallel |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C01201**

| Parameter \| Name: **C01201 \| MCK: Cycle** | | Data type: INTEGER_32<br>Index: $23374_d = 5B4E_h$ |
|---|---|---|
| Cycle for Modulo measuring system<br>• When the cycle is set to 0 units (Lenze setting), the traversing range is unlimited (classical measuring system).<br>▸ Activation of the Modulo measuring system | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0.0000 | units | 214748.3647 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01201/1 | 0.0000 units | MCK: Cycle |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C01202**

| Parameter \| Name: **C01202 \| MCK: iM motor/process** | | | Data type: UNSIGNED_16 Index: $23373_d = 5B4D_h$ |
|---|---|---|---|
| Gearbox factor - motor<br>• Entry of the gearbox factor as numerator/denominator ratio (numerator = motor speed and denominator = output speed of gearbox) or from the number of teeth of the gearbox arrangement.<br>▸ Machine parameter | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01202/1 | 1 | | MCK: iM: Numerator gearbox factor Z2 |
| C01202/2 | 1 | | MCK: iM: Denominator gearbox factor Z1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01203**

| Parameter \| Name: **C01203 \| MCK: iG motor/position encoder** | | | Data type: UNSIGNED_16 Index: $23372_d = 5B4C_h$ |
|---|---|---|---|
| Gearbox factor - position encoder<br>• Entry of the gearbox factor as numerator/denominator ratio, with numerator = motor speed and denominator = position encoder speed.<br>▸ Machine parameter | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01203/1 | 1 | | MCK: iG: Numerator (motor speed) |
| C01203/2 | 1 | | MCK: iG: Denominator (encoder speed) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01204**

| Parameter \| Name: **C01204 \| MCK: Feed constant** | | | Data type: INTEGER_32 Index: $23371_d = 5B4B_h$ |
|---|---|---|---|
| The feed constant corresponds to the movement of the machine during one revolution of the gearbox output shaft.<br>• The value is entered in application units referred to one revolution.<br>▸ Machine parameter | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0001 | units/rev. | 214748.3647 | **360.0000 units/rev.** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01205**

| Parameter \| Name: **C01205 \| MCK: Position resolution** | | | Data type: INTEGER_32 Index: $23370_d = 5B4A_h$ |
|---|---|---|---|
| Display of the number of increments corresponding to one unit.<br>• 1 motor revolution ≡ 65536 increments<br>▸ Machine parameter | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.0000 | Incr./unit | 214748.3647 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01206**

| Parameter \| Name: **C01206 \| MCK: Mounting direction** | | Data type: UNSIGNED_8 Index: $23369_d = 5B49_h$ |
|---|---|---|
| Inversion for mirrored motor and encoder mounting ▸ Machine parameter | | |
| **Selection list** | | |
| 0 | not inverted | |
| 1 | inverted | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01206/1 | 0: Not inverted | Motor mounting direction<br>• Setting for motor mounting turned by 180°. |
| C01206/2 | 0: Not inverted | Position encoder mounting direction<br>• Setting for position encoder system mounting turned by 180°. |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C01210**

| Parameter \| Name: **C01210 \| MCK: Current positions** | | | Data type: INTEGER_32 Index: $23365_d = 5B45_h$ |
|---|---|---|---|
| Display of current position data of the Motion Control Kernel | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01210/1 | | | MCK: Feed<br>• Display of the current feed for positioning profiles as a relative distance. |
| C01210/2 | | | MCK: Set position<br>• Display of the current setpoint position calculated by the MCK. |
| C01210/3 | | | MCK: Actual position<br>• Display of the current actual position calculated by an optional encoder system. |
| C01210/4 | | | MCK: Following error<br>• Display of the current following error as a difference between setpoint position and actual position. |
| C01210/5 | | | MCK: Positioning accuracy<br>• Display of the current positioning accuracy referred to the actual number of increments of the position encoder. |
| C01210/6 | | | MCK: Target position |
| C01210/7 | | | MCK: Set position modulo |
| C01210/8 | | | MCK: Actual position modulo<br>• From version 12.00.00 |
| C01210/9 | | | MCK: Dist. ref. mark and Ref-TP<br>• From version 14.00.00<br>• Display of the distance between pre-stop mark/limit switch and TP/MP while homing.<br>▸ Distance measurement |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01211**

| Parameter \| Name: **C01211 \| MCK: Speed** | | | Data type: INTEGER_32 Index: $23364_d = 5B44_h$ |
|---|---|---|---|
| Display of the current speed data | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units/s | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01211/1 | | | MCK: Max. traversing speed 100%_C11<br>• Display of the maximum traversing speed based on the reference speed set in C00011. |
| C01211/2 | | | MCK: Set speed<br>• From version 12.00.00 |
| C01211/3 | | | MCK: Actual speed<br>• From version 12.00.00 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01213**

| Parameter \| Name: **C01213 \| MCK: Max. traversing distance** | | | Data type: INTEGER_32 Index: $23362_d = 5B42_h$ |
|---|---|---|---|
| Display of current position limits | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -2147480000 | units | 2147480000 | |
| **Subcodes** | | | **Info** |
| C01213/1 | | | MCK: Max. traversing distance<br>• Display of the maximum distance to be traversed referred to the 32-bit display area (max. 2147483647 increments). |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01215**

| Parameter \| Name: **C01215 \| MCK: Following error** | | | Data type: INTEGER_32 Index: $23360_d = 5B40_h$ |
|---|---|---|---|
| Limits for following error monitoring system<br>▸ Following error monitoring system | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0000 | units | 214748.0000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01215/1 | 0.0000 units | | MCK: Following error limit 1<br>• First limit of the maximum following error for monitoring and response activation. |
| C01215/2 | 0.0000 units | | MCK: Following error limit 2<br>• Second limit of the maximum following error for monitoring and response activation. |
| C01215/3 | 0.0000 units | | MCK: Following error SetPos=ActPos<br>• From version 14.00.00<br>▸ Monitoring of the following error in case of controller inhibit |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01216**

| Parameter \| Name: **C01216 \| MCK: Positioning setting** | | | Data type: UNSIGNED_8 Index: $23359_d = 5B3F_h$ |
|---|---|---|---|
| | | | ▸ Positioning |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x01** (decimal: 1) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☑ | PosAbort at PosInit | | 1 ≡ When a changeover to "positioning" mode is made, ramping down at the rate of deceleration set in C01251for normal stopping is carried out. |
| Bit 1 ☐ | PosExecute active at PosInit | | 1 ≡When a changeover is made to the "positioning" mode, the specified profile is immediately executed if the "MCK PosExecute" control bit has also been set. If the MCK "PosExecute" control bit has not been set, the setpoint is continued. |
| Bit 2 ☐ | Reserved | | |
| Bit 3 ☐ | Reserved | | |
| Bit 4 ☐ | ProfilStart at PosInit | | 1 ≡When it is changed to the "Positioning" operating mode, the defined profile is executed immediately without the need to set the MCK control bit "PosExecute". |
| Bit 5 ☐ | Maximum jerk | | 1 ≡ The maximum jerk is calculated based on the acceleration and S-ramp time of the current profile and the sequence profile. This is then used in an accelerated drive for reducing the acceleration to 0 or for acceleration of the sequence profile.<br>• From version 14.00.00<br>▸ Setting or activation of maximum jerk for traversing profiles |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01218**

| Parameter \| Name: **C01218 \| MCK: Position follower setting** | | | Data type: UNSIGNED_8 Index: $23357_d = 5B3D_h$ |
|---|---|---|---|
| Settings for "Position follower" mode<br>▸ Position follower | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x0C** (decimal: 12) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | Speed FF control.: nSpeedSetValue_a | | 1 ≡ speed feedforward control value comes from main setpoint *nSpeedSetValue_a* |
| Bit 1 ☐ | Speed FF control: nSpeedAddValue_v | | 1 ≡ speed feedforward control value comes from additive speed value *nSpeedAddValue_v* |
| Bit 2 ☑ | HW limit switch on | | 1 ≡ Travel range monitoring via hardware limit switch is active. The error response can be parameterised in C00595/1 and C00595/2. |
| Bit 3 ☑ | SW limit switch on | | 1 ≡ Travel range monitoring via parameterised software limit positions is active. The error response can be parameterised in C00595/3 and C00595/4. |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Position controller off | | 1 ≡ Position controller is deactivated. Thus, the compensation of the following error is switched off. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01219**

| Parameter \| Name: **C01219 \| MCK: Speed follower setting** | | | Data type: UNSIGNED_8 Index: $23356_d = 5B3C_h$ |
|---|---|---|---|
| Settings for "Speed follower" mode<br>▸ Speed follower | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x0C** (decimal: 12) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | Reserved | | |
| Bit 1 ☐ | Reserved | | |
| Bit 2 ☑ | HW limit switch on | | 1 ≡ Travel range monitoring via hardware limit switch is active. The error response can be parameterised in C00595/1 and C00595/2. |
| Bit 3 ☑ | SW limit switch on | | 1 ≡ Travel range monitoring via parameterised software limit positions is active. The error response can be parameterised in C00595/3 and C00595/4. |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Position controller off | | 1 ≡ Position controller is deactivated. Thus, the compensation of the following error is switched off. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01220**

| Parameter \| Name: **C01220 \| MCK: Ref. setting** | | Data type: UNSIGNED_16 Index: $23355_d = 5B3B_h$ |
|---|---|---|
| From version 14.00.00<br>Settings for "homing" mode<br>▸ Homing | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |

| **Value is bit-coded:** | | **Info** |
|---|---|---|
| Bit 0 | Reserved | |
| Bit 1 | Reserved | |
| Bit 2 | Reserved | |
| Bit 3 | Reserved | |
| Bit 4 | Start deletes bHomePosAvailable | 1 ≡ When homing is started (mode 4 ... 15), the reference information is deleted. |
| Bit 5 | Maximum jerk | 1 ≡ The maximum jerk is calculated based on the acceleration and S-ramp time of the current profile and the sequence profile. This is then used in an accelerated drive for reducing the acceleration to 0 or for acceleration of the sequence profile.<br>▸ Setting or activation of maximum jerk for traversing profiles |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C01220/1 | 0x0000 | MCK: Ref. setting |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

**C01221**

| Parameter \| Name: **C01221 \| MCK: Ref. mode** | | Data type: UNSIGNED_8 Index: $23354_d = 5B3A_h$ |
|---|---|---|
| Selection of the homing mode.<br>▸ Homing | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 4 | >_Rp_<_TP | Positive direction - reversing on pre-stop mark (pos. edge) - to touch probe<br>• From version 14.00.00 |
| 5 | <_Rp_>_TP | Negative direction - reversing on pre-stop mark (pos. edge) - to touch probe<br>• From version 14.00.00 |
| 6 | >_Rn_>_TP | Positive direction - via pre-stop mark (neg. edge) - continuing in positive direction - to touch probe |
| 7 | <_Rn_<_TP | Negative direction - via pre-stop mark (neg. edge) - continuing in negativee direction - to touch probe |
| 8 | >_TP | Positive direction - to touch probe |
| 9 | <_TP | Negative direction - to touch probe |
| 10 | >_Lp_<_TP | Positive direction - reversing on pos. limit switch - to touch probe |
| 11 | <_Ln_>_TP | Negative direction - reversing on neg. limit switch - to touch probe |
| **12** | **>_Lp** | Positive direction - to pos. limit switch |
| 13 | <_Ln | Negative direction - to neg. limit switch |
| 14 | >_Mlim | Positive direction towards torque limit (C01222) |
| 15 | <_Mlim | Negative direction towards torque limit (C01222) |
| 100 | SetRef | Direct acceptance and setting of the home position<br>• The measuring system is set based on the home position parameterised in C01227/2 when the drive is at standstill. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01222**

| Parameter \| Name: **C01222 \| MCK: Ref. M limit mode 14/15** | | | Data type: INTEGER_16 Index: $23353_d = 5B39_h$ |
|---|---|---|---|
| Torque limit for homing modes 14 and 15 (homing towards positive stop)<br>• 100 % ≡ maximum torque (C00057)<br>▸ Homing | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 199.99 | **10.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01223**

| Parameter \| Name: **C01223 \| MCK: Ref. waiting time mode 14/15** | | | Data type: UNSIGNED_16 Index: $23352_d = 5B38_h$ |
|---|---|---|---|
| Blocking time for homing modes 14 and 15 (homing towards positive stop)<br>• The reference is set if an excess of the torque limit set in C01222 has been deteced over a time period defined here.<br>▸ Homing | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | ms | 65000 | **100 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01224**

| Parameter \| Name: **C01224 \| MCK: Ref. speeds** | | | Data type: INTEGER_32 Index: $23351_d = 5B37_h$ |
|---|---|---|---|
| Speeds for reference search/homing<br>▸ Homing | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0000 | unit/s | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01224/1 | 720.0000 unit/s | | MCK: Ref. initial speed<br>• Start speed for quick approach of the limit switch/pre-stop mark (depending on the selected homing mode). |
| C01224/2 | 180.0000 unit/s | | MCK: Ref. search speed<br>• Search speed for slower - but more precise - approaching of the touch probe sensor (sensor reference signal). |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01225**

| Parameter \| Name: **C01225 \| MCK: Ref. accelerations** | | | Data type: INTEGER_32 Index: $23350_d = 5B36_h$ |
|---|---|---|---|
| Accelerations for reference search/homing<br>▸ Homing | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0000 | unit/s2 | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01225/1 | 720.0000 unit/s2 | | MCK: Ref. initial acceleration<br>• Starting acceleration for the starting speed ramps. |
| C01225/2 | 720.0000 unit/s2 | | MCK: Ref. search acceleration<br>• Search acceleration for the search speed ramps. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01226**

| Parameter \| Name: **C01226 \| MCK: Ref. S-ramp time** | | | Data type: UNSIGNED_16 Index: $23349_d = 5B35_h$ |
|---|---|---|---|
| S-ramp time for reference search/homing<br>• Setting "0 s" ≡ no rounding<br>▸ Homing | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 10.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01226/1 | 0.000 s | | MCK: Ref. S-ramp time<br>• S-ramp time for the starting and search speed ramps. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01227**

| Parameter \| Name: **C01227 \| MCK: Ref. positions** | | | Data type: INTEGER_32 Index: $23348_d = 5B34_h$ |
|---|---|---|---|
| Positions for determining the zero position of the reference measuring system<br>▸ Homing | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01227/1 | 0.0000 unit | | MCK: Ref. offset reference degree<br>• Relative traverse path by which the drive traverses correctly signed with search speed after detection of the sensor reference signal. |
| C01227/2 | 0.0000 unit | | MCK: Ref. home position<br>• Position with which setpoint and actual position are loaded after completion of homing. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01228**

| Parameter \| Name: **C01228 \| MCK: Ref. sequence profile** | | | Data type: UNSIGNED_8 Index: $23347_d = 5B33_h$ |
|---|---|---|---|
| Number of the sequence profile the absolute position of which will be approached after homing.<br>▸ Homing | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 15 | **0** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01229**

| Parameter \| Name: **C01229 \| MCK: Position limiting values** | | | Data type: INTEGER_32 Index: $23346_d = 5B32_h$ |
|---|---|---|---|
| Software limit position for limiting the valid traversing range<br>• The error response to leaving the valid traversing range can be parameterised in C00595/3 and C00595/4.<br><br>**Note:**<br>For limiting the traversing range by means of software limit positions, the home position must be known and the positive software limit position must be higher than the negative software limit position!<br>▸ Limit position monitoring | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01229/1 | 0.0000 units | | MCK: Positive SW limit position (positive travel range limit) |
| C01229/2 | 0.0000 units | | MCK: Negative SW limit position (negative travel range limit) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01230**

| Parameter \| Name: **C01230 \| Manual jog: Setting** | | | Data type: UNSIGNED_8 Index: $23345_d = 5B31_h$ |
|---|---|---|---|
| Settings for "Manual jog" mode<br><br>▸ Manual jog | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x00** (decimal: 0) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | Breakpoints 1..4 on | | 1 ≡ approach of the parameterised breakpoint in manual jog direction |
| Bit 1 ☐ | Time-based start of 2nd speed | | 1 ≡ Automatic change-over to second manual speed after the wating time set in C01235/1. |
| Bit 2 ☐ | HW limit switch on | | 1 ≡ Travel range monitoring via hardware limit switch is active. The error response can be parameterised in C00595/1 and C00595/2. |
| Bit 3 ☐ | SW limit switch on | | 1 ≡ Travel range monitoring via parameterised software limit positions is active. The error response can be parameterised in C00595/3 and C00595/4. |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Maximum jerk | | 1 ≡ The maximum jerk is calculated based on the acceleration and S-ramp time of the current profile and the sequence profile. This is then used in an accelerated drive for reducing the acceleration to 0 or for acceleration of the sequence profile.<br>• From version 14.00.00<br>▸ Setting or activation of maximum jerk for traversing profiles |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Position controller off | | 1 ≡ Position controller is deactivated. Thus, the compensation of the following error is switched off. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01231**

| Parameter \| Name: **C01231 \| MCK: Manual jog speeds** | | | Data type: INTEGER_32 Index: $23344_d = 5B30_h$ |
|---|---|---|---|
| Manual jog speeds<br>▸ Manual jog | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units/s | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01231/1 | 360.0000 units/s | | Manual jog: speed 1<br>• Regular manual speed |
| C01231/2 | 720.0000 units/s | | Manual jog: Speed 2<br>• Second manual speed (can be activated via MCK control word) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01232**

| Parameter \| Name: **C01232 \| MCK: Manual jog accelerations** | | | Data type: INTEGER_32 Index: $23343_d = 5B2F_h$ |
|---|---|---|---|
| Acceleration/deceleration for manual jog<br>▸ Manual jog | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units/s2 | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01232/1 | 720.0000 units/s2 | | Manual jog: Acceleration<br>• Acceleration for ramp-up to manual speed. |
| C01232/2 | 720.0000 units/s2 | | Manual jog: Deceleration<br>• Deceleration for manual speed ramp-down to standstill. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01233**

| Parameter \| Name: **C01233 \| MCK: Manual jog S-ramp time** | | | Data type: UNSIGNED_16 Index: $23342_d = 5B2E_h$ |
|---|---|---|---|
| S-ramp time for manual jog<br>• Setting "0 s" ≡ no rounding<br>▸ Manual jog | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 10.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01233/1 | 0.000 s | | Manual jog: S-ramp time<br>• S-ramp time for the manual speed ramps. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01234**

| Parameter \| Name: **C01234 \| MCK: Manual jog breakpoints** | | | Data type: INTEGER_32 Index: $23341_d = 5B2D_h$ |
|---|---|---|---|
| Breakpoint positions for manual jog<br>• The drive stops at the parameterised positions if these are located in the manual jog direction and the approaching of the breakpoints is switched on in C01230 via bit 0.<br>▸ Manual jog | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01234/1 | 0.0000 unit | | Manual jog: Breakpoint 1 |
| C01234/2 | 0.0000 unit | | Manual jog: Breakpoint 2 |
| C01234/3 | 0.0000 unit | | Manual jog: Breakpoint 3 |
| C01234/4 | 0.0000 unit | | Manual jog: Breakpoint 4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01235**

| Parameter \| Name: **C01235 \| MCK: Manual jog waiting times** | | | Data type: UNSIGNED_16 Index: $23340_d = 5B2C_h$ |
|---|---|---|---|
| Waiting times for manual jog<br>▸ Manual jog | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 65.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01235/1 | 5.000 s | | Manual jog: Waiting time 2nd speed<br>• Time after which the first manual speed (C01231/1) is accelerated to the second manual speed (C01231/2). |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01236**

| Parameter \| Name: **C01236 \| MCK: Speed follower** | | | Data type: INTEGER_32 Index: $23339_d = 5B2B_h$ |
|---|---|---|---|
| Speeds for position follower<br>▸ Position follower | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units/s | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01236/1 | 360.0000 units/s | | PosFollower: Sync. speed<br>• Synchronisation speed for approaching the setpoint position of the master. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01237**

| Parameter \| Name: **C01237 \| MCK: Acceleration follower** | | | Data type: INTEGER_32 Index: $23338_d = 5B2A_h$ |
|---|---|---|---|
| Accelerations for position follower<br>▸ Position follower | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units/s2 | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01237/1 | 720.0000 units/s2 | | Pos follower: Sync. accel.<br>• Acceleration for ramp-up to synchronisation speed. |
| C01237/2 | 720.0000 units/s2 | | Pos follower: Sync. decel.<br>• Deceleration for synchronisation speed ramp-down to standstill (to the setpoint position of the master). |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01238**

| Parameter \| Name: **C01238 \| MCK: S-ramp time follower** | | | Data type: UNSIGNED_16 Index: $23337_d = 5B29_h$ |
|---|---|---|---|
| S-ramp times for position follower<br>• Setting "0 s" ≡ no rounding<br>▸ Position follower | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 10.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01238/1 | 0.000 s | | Pos follower: Sync. S-ramp time<br>• S-ramp time for the synchronisation speed ramps. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01239**

| Parameter \| Name: **C01239 \| Setpoint holding** | | | Data type: UNSIGNED_32 Index: $23336_d = 5B28_h$ |
|---|---|---|---|
| From version 12.00.00<br>▸ Setpoint holding for bus runtime compensation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | ms | 32 | **0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01240**

| Parameter \| Name: **C01240 \| MCK: Control word** | | | Data type: UNSIGNED_32 Index: $23335_d = 5B27_h$ |
|---|---|---|---|
| Bit-coded status of the MCK control word<br>• The MCK control word is used to control the functions of the Motion Control Kernel.<br>▸ MCK control word | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x00000000 | | 0xFFFFFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | OpMode_Bit0 | | Active operating mode - control bit 0 |
| Bit 1 | OpMode_Bit1 | | Active operating mode - control bit 1 |

| Parameter \| Name: **C01240 \| MCK: Control word** | | Data type: UNSIGNED_32 Index: $23335_d = 5B27_h$ |
|---|---|---|
| Bit 2 | OpMode_Bit2 | Active operating mode - control bit 2 |
| Bit 3 | OpMode_Bit3 | Active operating mode - control bit 4 |
| Bit 4 | ManJogPos | Control bit for manual jog in positive direction |
| Bit 5 | ManJogNeg | Control bit for manual jog in negative direction |
| Bit 6 | ManExecute2ndSpeed | Control bit for activation of second manual speed |
| Bit 7 | ReleaseLimitSwitch | Control bit for retracting the hardware limit positions |
| Bit 8 | HomStartStop | Control bit for start/stop homing |
| Bit 9 | HomSetPos | Control bit for setting of home position<br>• 0->1 ≡ sets position setpoint and actual position value to the home position at the MCK.<br>• This function is independent of the operating modes. |
| Bit 10 | HomResetPos | Control bit for reset of home position<br>• 0->1 ≡ resets the information bit for the status "Home position known" (home position not known).<br>• The position displays are not influenced by this. |
| Bit 11 | EnableSpeedOverride | Control bit for activation of speed override<br>• 1 ≡ activation of the override speed at the MCK for acceptance in a running speed profile. |
| Bit 12 | EnableAccOverride | Control bit for activation of acceleration override<br>• 1 ≡ activation of the override acceleration at the MCK for acceptance in a running speed profile. |
| Bit 13 | EnableSRampOverride | Control bit for deactivation of S-shaping for speed profiles<br>• 1 ≡ deactivates S-shaping for speed profiles |
| Bit 14 | PosTeachSetPos | Control bit for accepting the selected setpoint position in the selected profile<br>• 0->1 ≡ accepts the setpoint position at the MCK in the profile with the profile number defined via the MCK control word. |
| Bit 15 | PosTeachActPos | Control bit for accepting the current actual position in the selected profile<br>• 0->1 ≡ accepts the current actual position in the profile with the profile number defined via the MCK control word. |
| Bit 16 | PosExecute | Control bit for starting a positioning profile<br>• 0->1 ≡ starts a positioning profile selected via the control word in the "Positioning" mode. |
| Bit 17 | PosFinishTarget | Control bit for completing processing of an interrupted positioning profile<br>• 0->1 ≡ processing of a positioning profile which has already been started and has been interrupted because of a cancellation condition or a change of the operating mode is completed in the "Positioning" mode when the home position is known.<br>**Note:**<br>Completion of profile processing through this control is based on the target position detected at the profile start! |
| Bit 18 | PosDisableFollowProfile | Control bit for suppression of the sequence profile<br>• 1 ≡ positioning profiles whose profile numbers are entered in profile data sets for sequence profiles are not started after completion of the corresponding profile. |

| Parameter \| Name: **C01240 \| MCK: Control word** | | Data type: UNSIGNED_32 Index: $23335_d = 5B27_h$ |
|---|---|---|
| Bit 19 | PosStop | Control bit for stopping the positioning profile<br>• 0->1 ≡ ("Positioning" mode): Interrupts a running profile through deceleration along the set deceleration ramp to standstill.<br>• 0->1 ≡ ("Stop" mode): Starts a new deceleration to standstill with the parameterised stop ramp. |
| Bit 20 | PosModeBit0 | Active positioning mode - control bit 0 |
| Bit 21 | PosModeBit1 | Active positioning mode - control bit 1 |
| Bit 22 | PosModeBit2 | Active positioning mode - control bit 2 |
| Bit 23 | PosModeBit3 | Active positioning mode - control bit 3 |
| Bit 24 | ProfileNo_Bit0 | Active positioning profile number - control bit 0 |
| Bit 25 | ProfileNo_Bit1 | Active positioning profile number - control bit 1 |
| Bit 26 | ProfileNo_Bit2 | Active positioning profile number - control bit 2 |
| Bit 27 | ProfileNo_Bit3 | Active positioning profile number - control bit 3 |
| Bit 28 | ProfileNo_Bit4 | Active positioning profile number - control bit 4 |
| Bit 29 | ProfileNo_Bit5 | Active positioning profile number - control bit 5 |
| Bit 30 | ProfileNo_Bit6 | Active positioning profile number - control bit 6 |
| Bit 31 | ProfileNo_Bit7 | Active positioning profile number - control bit 7 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | |

**C01241**

| Parameter \| Name: **C01241 \| MCK: Status word** | | | Data type: UNSIGNED_32 Index: $23334_d = 5B26_h$ |
|---|---|---|---|
| Bit-coded status of the MCK status word<br>• The MCK status word is used to output status messages of the MotionControlKernel.<br>▸ MCK status word | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x00000000 | | 0xFFFFFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | ActOpModeBit00 | | Active operating mode - status bit 0 |
| Bit 1 | ActOpModeBit01 | | Active operating mode - status bit 1 |
| Bit 2 | ActOpModeBit02 | | Active operating mode - status bit 2 |
| Bit 3 | ActOpModeBit03 | | Active operating mode - status bit 4 |
| Bit 4 | Busy | | Status bit - profile generation active<br>• 1 ≡ Internal profile generation is active. A speed profile is being generated. |
| Bit 5 | Done | | Status bit - profile generation completed<br>• 1 ≡ Generation of a speed profile with the selected position has been completed. |
| Bit 6 | AcceleratingActive | | Status bit - acceleration process for profile generation active<br>• 1 ≡ Profile generation phase is in the acceleration process. |
| Bit 7 | ConstSpeedDuty | | Status bit - constant speed for profile generation active<br>• 1 ≡ Profile generation phase at constant speed active. |
| Bit 8 | DeceleratingActive | | Status bit - deceleration process for profile generation active<br>• 1 ≡ Profile generation phase is in the deceleration process. |

| Parameter \| Name:<br>**C01241 \| MCK: Status word** | | Data type: UNSIGNED_32<br>Index: $23334_d = 5B26_h$ |
|---|---|---|
| Bit 9 | S_ShapingActive | Status bit - rounding for profile generation active<br>• 1 ≡ Rounding during acceleration/deceleration active. |
| Bit 10 | Pos. HW-Limit Detected | Status bit - positive hardware limit detected<br>• 1 ≡ Positive limit switch has triggered.<br>• Reset only possible via"Manual jog" mode! |
| Bit 11 | Neg. HW-Limit Detected | Status bit - negative hardware limit detected<br>• 1 ≡ Negative limit switch has triggered.<br>• Reset only possible via"Manual jog" mode! |
| Bit 12 | HomPosDone | Status bit - homing completed<br>• 1 ≡ Homing has been completed. |
| Bit 13 | HomPosAvailable | Status bit - home position known<br>• 1 ≡ The home position has been detected and is known in the drive. |
| Bit 14 | Pos. SW-Limit Detected | Status bit - positive software limit detected<br>• 1 ≡ Positive software limit position overtravelled. |
| Bit 15 | Neg. SW-Limit Detected | Status bit - negative software limit detected<br>• 1 ≡ Negative software limit position overtravelled. |
| Bit 16 | DwellTime | Status bit - transient effects in target position active<br>• 1 ≡ Dwell time after reaching the setpoint position is active. |
| Bit 17 | InTarget | Status bit - actual position is in the target window<br>• 1 ≡ Dwell time has expired and current actual position is in the set target window. |
| Bit 18 | PosDone | Status bit - positioning process completed<br>• 1 ≡ Positioning profile has been completed in the "Positioning" mode. Setpoint position is in target. |
| Bit 19 | ReadyToOperate | Status bit - readiness to process setpoint signals or setpoint commands |
| Bit 20 | ActPosMode_Bit00 | Active positioning mode - status bit 0 |
| Bit 21 | ActPosMode_Bit01 | Active positioning mode - status bit 1 |
| Bit 22 | ActPosMode_Bit02 | Active positioning mode - status bit 2 |
| Bit 23 | ActPosMode_Bit03 | Active positioning mode - status bit 3 |
| Bit 24 | ActProfileNo_Bit00 | Active positioning profile - status bit 0 |
| Bit 25 | ActProfileNo_Bit01 | Active positioning profile - status bit 1 |
| Bit 26 | ActProfileNo_Bit02 | Active positioning profile - status bit 2 |
| Bit 27 | ActProfileNo_Bit03 | Active positioning profile - status bit 3 |
| Bit 28 | ActProfileNo_Bit04 | Active positioning profile - status bit 4 |
| Bit 29 | ActProfileNo_Bit05 | Active positioning profile - status bit 5 |
| Bit 30 | ActProfileNo_Bit06 | Active positioning profile - status bit 6 |
| Bit 31 | ActProfileNo_Bit07 | Active positioning profile - status bit 7 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | |

**C01242**

| Parameter \| Name: **C01242 \| MCK: Current pos profile number** | | | Data type: UNSIGNED_8 Index: $23333_d = 5B25_h$ |
|---|---|---|---|
| Display of the current profile number of the active profile in the "Positioning" mode<br>▸ Positioning | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 255 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01243**

| Parameter \| Name: **C01243 \| MCK: Current operating mode** | | Data type: UNSIGNED_8 Index: $23332_d = 5B24_h$ |
|---|---|---|
| Display of the active operating mode of the Motion Control Kernel<br>▸ Basic drive functions | | |
| **Selection list** (read only) | | **Info** |
| 0 | Speed follower | Drive traverses according to a preselected speed setpoint |
| 1 | Homing | Drive finds its reference measuring system by setting the home position or homing |
| 2 | ManualJog | Drive can be traversed manually via initiators, e.g. for cleaning purposes or tool changes. |
| 3 | Positioning | Drive traverses according to defined travel profiles and motion processes |
| 4 | Stop | Drive is decelerated to standstill along a parameterised deceleration ramp |
| 5 | Position follower | Drive traverses according to a preselected position setpoint |
| 15 | StandBy | Internal operating mode for quick stop and pulse inhibit |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01244**

| Parameter \| Name: **C01244 \| MCK: Target detection - times** | | | Data type: UNSIGNED_16 Index: $23331_d = 5B23_h$ |
|---|---|---|---|
| Timing for various MCK functions | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01244/1 | 100 ms | | MCK: Dwell time - target position<br>• Time expiring after reaching the setpoint position for positioning processes and enabling the actual position detection in the target position window.<br>▸ Target position monitoring |
| C01244/2 | 0 ms | | MCK: Waiting time following error 1<br>• From version 02.00.00<br>▸ Following error monitoring system |
| C01244/3 | 0 ms | | MCK: Waiting time following error 2<br>• From version 02.00.00<br>▸ Following error monitoring system |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01245**

| Parameter \| Name: **C01245 \| MCK: Target detection - positions** | | | Data type: INTEGER_32 Index: $23330_d = 5B22_h$ |
|---|---|---|---|
| Position settings for various MCK functions | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0000 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01245/1 | 1.0000 units | | MCK: Window target position<br>• Window around the target position for comparison with the actual position to see whether the drive is in target.<br>▸ Target position monitoring |
| C01245/2 | 1.0000 units | | MCK: Blocking zone Modulo<br>▸ Activation of the Modulo measuring system |
| C01245/3 | 0.0000 units | | MCKI: Hysteresis position change |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01246**

| Parameter \| Name: **C01246 \| MCK: Select signal source** | | Data type: UNSIGNED_8 Index: $23329_d = 5B21_h$ |
|---|---|---|
| Selection of the touch probe signal source for reference search with touch probe detection<br>• If the reference signal is to follow a real touch probe, the Touch-Probe-Interface must be configured accordingly.<br>▸ Homing | | |
| **Selection list** | | **Info** |
| 0 | No TP | |
| 3 | TP-DigIn3 | |
| 4 | TP-DigIn4 | |
| 5 | TP-DigIn5 | |
| 6 | TP-DigIn6 | |
| 7 | TP-DigIn7 | |
| 8 | TP-Z-trace encoder | From version 12.00.00 |
| 9 | TP-Z-trace resolver | From version 12.00.00 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01246/1 | 0: No TP | MCK: Ref. TP signal source<br>**Note:**<br>In case of setting "0: No TP", the digital input DI3 is used as signal source for reference search. An inversion of DI3 via C00114 is considered. |
| C01246/2 | 0: No TP | MCK: Set.Ref. signal source |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01247**

| Parameter \| Name: **C01247 \| MCK: Diagnostic word** | | | Data type: UNSIGNED_32 Index: $23328_d = 5B20_h$ |
|---|---|---|---|
| From version 15.00.00 | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x00000000 | | 0xFFFFFFFF | |
| **Value is bit-coded:** | | | **Info** |

| Parameter \| Name: **C01247 \| MCK: Diagnostic word** | | Data type: UNSIGNED_32 Index: $23328_d = 5B20_h$ |
|---|---|---|
| Bit 0 | Traversing range: Limited | 1 ≡ The traversing range is limited:<br>• The software limit positions are set.<br>• The reference is known.<br>• There is no modulo measuring system available (C01201/1 = 0).<br>**Note:**<br>This status does not consider whether the software limit positions are activated or deactivated by the settings in the prevailing operating mode. |
| Bit 1 | Traversing range: Modulo | 1 ≡ A modulo measuring system is available as a cycle length has been set in C01201/1. |
| Bit 2 | Pos. HW limit switch active | Evaluation of travel range limit switch active in positive direction |
| Bit 3 | Neg. HW limit switch active | Evaluation of travel range limit switch active in negative direction |
| Bit 4 | Pos. SW-Limit active | Evaluation of software limit positions active in positive direction |
| Bit 5 | Neg. SW-Limit active | Evaluation of software limit positions active in negative direction |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | SpeedNormConstant: Low-Limit | 1 ≡ Due to the set machine parameters, an internal calculation constant is limited to a possible minimum.<br>Remedy: Increase reference speed in C00011. |
| Bit 9 | SpeedNormConstant: Up-Limit | 1 ≡ Due to the set machine parameters, an internal calculation constant is limited to a possible maximum.<br>Remedy: Reduce reference speed in C00011. |
| Bit 10 | Max. speed limit | 1 ≡ Due to the set machine parameters, the maximum speed for specifications in [units/s] are limited to the factor 150 %. |
| Bit 11 | Quick stop is active | 1 ≡ Quick stop function is active. |
| Bit 12 | Quick stop standstill is active | 1 ≡ Standstill during active quick stop function (no ramp generation). |
| Bit 13 | Max. speed is pending | 1 ≡ The speed setpoint has exceeded the limit of ± 199.99 %.<br>Possible cause:<br>• Too high influence due to speed override<br>• Increased speed by reduced acceleration with little jerk |
| Bit 14 | Position resolution limit | 1 ≡ The position resolution calculated by machine data entries has been limited to the maximum display area of 214748.3647 [Incr./Unit] with reference to the display in C01205/0. The real value is higher! |
| Bit 15 | Reserved | |
| Bit 16 | Reserved | |
| Bit 17 | Reserved | |
| Bit 18 | Reserved | |
| Bit 19 | Reserved | |
| Bit 20 | Reserved | |
| Bit 21 | Reserved | |
| Bit 22 | Reserved | |
| Bit 23 | Reserved | |
| Bit 24 | Reserved | |

| Parameter \| Name: C01247 \| MCK: Diagnostic word | | Data type: UNSIGNED_32 Index: $23328_d = 5B20_h$ |
|---|---|---|
| Bit 25 | Reserved | |
| Bit 26 | Reserved | |
| Bit 27 | Reserved | |
| Bit 28 | Reserved | |
| Bit 29 | Reserved | |
| Bit 30 | Reserved | |
| Bit 31 | Reserved | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | |

**C01248**

| Parameter \| Name: C01248 \| MCK: Ref. step | | Data type: UNSIGNED_8 Index: $23327_d = 5B1F_h$ |
|---|---|---|
| From version 16.00.00 | | |
| **Selection list** | | **Info** |
| 0 | No Operation | Initial step when changing to the homing mode |
| 1 | RefMarke: Search | Search travel until reference pre-stop mark |
| 2 | RefMarke: Stop | Stop when reference pre-stop mark has been detected |
| 3 | RefMarke: Reversal | Reversal of reference pre-stop mark |
| 4 | HW-Limit: Search | Search travel until limit switch |
| 5 | HW-Limit: Stop | Stop when limit switch has been detected |
| 6 | HW-Limit: Reversal | Reversal of limit switch |
| 7 | M-Limit: Search | Search travel until torque limit has been detected |
| 8 | M-Limit: Stop | Detection of torque limit, tolerance time is running |
| 9 | Reference: Search | Search travel until reference initiator |
| 10 | Reference: Stop | Stop when reference initiator has been detected |
| 11 | Reference: Offset | Travel offset path active |
| 12 | Reference: Sequence profile | Travel sequence profile after homing |
| 13 | Reference: Position | Travel to home position |
| 14 | Reference: Abort | Homing process aborted |
| 15 | Finish | Homing process executed |
| **Subcodes** | | **Info** |
| C01248/1 | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01251**

| Parameter \| Name: **C01251 \| MCK: Acceleration stop** | | | Data type: INTEGER_32 Index: $23324_d = 5B1C_h$ |
|---|---|---|---|
| Acceleration parameter for stop<br>▸ Stop | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units/s2 | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01251/1 | 720.0000 units/s2 | | MCK: Stop: Decel.<br>• Deceleration for setpoint speed ramp-down to standstill. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01252**

| Parameter \| Name: **C01252 \| MCK: S-ramp times stop** | | | Data type: UNSIGNED_16 Index: $23323_d = 5B1B_h$ |
|---|---|---|---|
| S-ramp times for stop<br>• Setting "0 s" ≡ no rounding<br>▸ Stop | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 10.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01252/1 | 0.000 s | | MCK: Stop: S-ramp time<br>• S-ramp time for setpoint speed ramp-down to standstill. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01293**

| Parameter \| Name: **C01293 \| L_MckStateInterface_1: Status** | | Data type: UNSIGNED_8 Index: $23282_d = 5AF2_h$ |
|---|---|---|
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x00 | | 0xFF |
| **Value is bit-coded:** | | **Info** |
| Bit 0 | ActOpModeBit00 | |
| Bit 1 | ActOpModeBit01 | |
| Bit 2 | ActOpModeBit02 | |
| Bit 3 | ActOpModeBit03 | |
| Bit 4 | Busy | |
| Bit 5 | Done | |
| Bit 6 | HomPosAvailable | |
| Bit 7 | ReadyToOperate | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01293/1 | 0x00 | L_MckStateInterface_1: Status 1 |
| C01293/2 | 0x01 | L_MckStateInterface_1: Status 2 |
| C01293/3 | 0x02 | L_MckStateInterface_1: Status 3 |
| C01293/4 | 0x03 | L_MckStateInterface_1: Status 4 |
| C01293/5 | 0x04 | L_MckStateInterface_1: Status 5 |
| C01293/6 | 0x05 | L_MckStateInterface_1: Status 6 |
| C01293/7 | 0x0F | L_MckStateInterface_1: Status 7 |
| C01293/8 | 0x0E | L_MckStateInterface_1: Status 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C01294**

| Parameter \| Name: **C01294 \| Mode: Position calculation** | | Data type: UNSIGNED_8 Index: $23281_d = 5AF1_h$ |
|---|---|---|
| From version 12.00.00<br>Selection of the conversion mode | | |
| **Selection list** | | |
| 0 | dnOut_p=ConvWordsToDInt(HW_LW) | |
| 1 | 16 bits: LW=+/-32767 | |
| 2 | 16 bits: HW=+/-; LW=0..65535 | |
| 3 | 32Bit: HW_LW=+/-214748.3647 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01294/1 | 0: dnOut_p=ConvWordsToDInt(HW_LW) | L_ConvUnitsToIncr_1: PosCalcMode |
| C01294/2 | 0: dnOut_p=ConvWordsToDInt(HW_LW) | L_ConvUnitsToIncr_2: PosCalcMode |
| C01294/3 | 0: dnOut_p=ConvWordsToDInt(HW_LW) | L_ConvUnitsToIncr_3: PosCalcMode |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01295**

| Parameter \| Name: **C01295 \| L_MCKStateInterface_1: Pos. selection** | | Data type: UNSIGNED_8 Index: $23280_d = 5AF0_h$ |
|---|---|---|
| Selection of the position to be provided at the *dnPosOut_p* output of the FB L_MckStateInterface | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | dnPosIn_p | |
| 1 | Current feed | |
| 2 | dnSetPos_p | |
| **3** | **dnActPos_p** | |
| 4 | dnDeltaPos_p | |
| 5 | dnTargetPos_p | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01296**

| Parameter \| Name: **C01296 \| Mode: Position calculation** | | Data type: UNSIGNED_8 Index: $23279_d = 5AEF_h$ |
|---|---|---|
| | | ▸ MCKInterface |
| **Selection list** | | |
| 0 | dnPosOut_p=dnPosIn_p | |
| 1 | 16 bits: LW=+/-32767 | |
| 2 | 16 bits: HW=+/-; LW=0..65535 | |
| 3 | 32Bit: HW_LW=+/-214748.3647 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01296/1 | 0: dnPosOut_p=dnPosIn_p | L_MckCtrlInterface_1: PosCalcMode |
| C01296/2 | 1: 16Bit: LW=+/-32767 | L_MckStateInterface_1: PosDisplayMode |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01297**

| Parameter \| Name: **C01297 \| L_MckCtrlInterface_1: Alternative function** | | | Data type: UNSIGNED_8 Index: $23278_d = 5AEE_h$ |
|---|---|---|---|
| Selection of the alternative function for bit 16 (PosExecute) in the MCK control word ▸ Alternative functions for "PosExecute" control bit | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x03** (decimal: 3) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☑ | PosStop with PosExecute = FALSE | | 1 ≡ Abort positioning with PosExecute<br>• PosExecute="0" stops a running positioning process by ramp-down to standstill. |
| Bit 1 ☑ | HomingStartStop with PosExecute | | 1 ≡ Start/stop homing with PosExecute<br>• PosExecute="1" starts homing, PosExecute="0" stops homing in the "Homing" mode. |
| Bit 2 ☐ | SetProfilPosition with PosExecute | | 1 ≡ Teach set position with PosExecute<br>• PosExecute="0->1" accepts the setpoint position at the MCK in the profile with the specified profile number. |
| Bit 3 ☐ | SetProfilPosition at position change | | 1 ≡ Automatic acceptance of set position<br>• The set position pending at the MCKInterface is automatically transferred to the profile with the preset profile number if a data change at the corresponding input for the setpoijnt is detected. |
| Bit 4 ☐ | Profile start at position change | | From version 02.00.00<br>1 ≡ Automatic profile start at position change<br>• An automatic PosExecute takes place if the incremental position selection changes and this change is higher than set in the hysteresis for position change (C01245/3). |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01298**

| Parameter \| Name: **C01298 \| MCK: Operating mode change with profile no.** | | Data type: UNSIGNED_8 Index: $23277_d = 5AED_h$ |
|---|---|---|
| FB L_MckCtrlInterface_1: Assignment of an operating mode to a profile | | |
| **Selection list** | | **Info** |
| 0 | Op.Mode inputs | There will be no changed operating mode when the profile is changed. Instead, the operating mode requested "from outside" via the MCK control word applies. |
| 1 | Speed follower | Change to the "Speed follower" mode |
| 2 | Homing | Change to the "Homing" mode |
| 3 | Manual jog | Change to the "Manual jog" mode |
| 4 | Positioning | Change to the "Positioning" mode |
| 5 | Stop | Change to the "Stop" mode |
| 6 | Position follower | Change to the "Position follower" mode |
| **Subcodes** | **Lenze setting** | **Info** |
| C01298/1 | 1: Speed follower | MCK operating mode at profile no. 0<br>• Selection of the operating mode for profile 0 at L_MckCtrlInterface_1. |
| C01298/2 | 2: Homing | MCK operating mode at profile no. 1<br>• Selection of the operating mode for profile 1 at L_MckCtrlInterface_1. |
| C01298/3 | 3: Manual jog | MCK operating mode at profile no. 2<br>• Selection of the operating mode for profile 2 at L_MckCtrlInterface_1. |
| C01298/4 | 4: Positioning | MCK operating mode at profile no. 3...15<br>• Selection of the operating mode for profile 3 ... 15 at L_MckCtrlInterface_1. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01299**

| Parameter \| Name: **C01299 \| MCKI: Status MCKInterface** | | | Data type: UNSIGNED_8 Index: $23276_d = 5AEC_h$ |
|---|---|---|---|
| FB L_MckCtrlInterface_1: Status messages | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x00 | | 0xFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | InvalidOperationMode | | Invalid operating mode selection<br>• 1 ≡ The selected operating mode is not defined/invalid. |
| Bit 1 | InvalidPosMode | | Invalid positioning mode selection<br>• 1 ≡ The selected positioning mode is not defined/invalid. |
| Bit 2 | InvalidProfileNo | | Invalid profile number selection<br>• 1 ≡ The selected profile number refers to a profile data set that does not exist. |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C01300**

| Parameter \| Name: **C01300 \| Profile data: Positioning mode** | | Data type: UNSIGNED_8 Index: $23275_d = 5AEB_h$ |
|---|---|---|
| Selection of the positioning mode for profiles 1 ... 15<br>▸ Profile entry | | |
| **Selection list** | | **Info** |
| 1 | Absolute (shortest path) | Absolute positioning<br>• The profile position corresponds to the target position. |
| 2 | Continuous | Continuous positioning without approaching a target position |
| 3 | Relative | Relative positioning<br>• The profile position determines the distance to be traversed. |
| 4 | absolute (Cw) | Clockwise absolute positioning<br>• The zero position of the axis can be exceed in this direction.<br>• The profile position corresponds to the target position. |
| 5 | absolute (Ccw) | Counter-clockwise absolute positioning<br>• The zero position of the axis can be exceed in this direction.<br>• The profile position corresponds to the target position. |
| 8 | Absolute (shortest path) to TP | Absolute positioning<br>• The profile position corresponds to the target position.<br>• After a touch probe is detected, the TP profile (C01308/1...15) is executed. |

| Parameter \| Name: **C01300 \| Profile data: Positioning mode** | | Data type: UNSIGNED_8 Index: $23275_d = 5AEB_h$ |
|---|---|---|
| 9 | Continuous to TP | Continuous positioning without approaching a target position<br>• After a touch probe is detected, the TP profile (C01308/1...15) is executed. |
| 10 | Relative to TP | Relative positioning<br>• The profile position determines the distance to be traversed.<br>• After a touch probe is detected, the TP profile (C01308/1...15) is executed. |
| 11 | Absolute (Cw) on TP | Clockwise absolute positioning<br>• The zero position of the axis can be exceed in this direction.<br>• The profile position corresponds to the target position.<br>• After a touch probe is detected, the TP profile (C01308/1...15) is executed. |
| 12 | Absolute (Ccw) on TP | Counter-clockwise absolute positioning<br>• The zero position of the axis can be exceed in this direction.<br>• The profile position corresponds to the target position.<br>• After a touch probe is detected, the TP profile (C01308/1...15) is executed. |
| **Subcodes** | **Lenze setting** | **Info** |
| C01300/1 | 1: absolute (beeline) | Profiles 1 ... 15: Positioning mode |
| C01300/... | | |
| C01300/15 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01301**

| Parameter \| Name: **C01301 \| Profile data: Position** | | | Data type: INTEGER_32 Index: $23274_d = 5AEA_h$ |
|---|---|---|---|
| Selection of the positions for profiles 1 ... 15<br>▸ Profile entry | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01301/1 | 360.0000 unit | | Profiles 1 ... 15: Position |
| C01301/... | | | |
| C01301/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01302**

| Parameter \| Name: **C01302 \| Profile data: Speed** | | | Data type: INTEGER_32 Index: $23273_d = 5AE9_h$ |
|---|---|---|---|
| Selection of the maximum speeds for profiles 1 ... 15 | | | ▸ Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit/s | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01302/1 | 360.0000 unit/s | | Profiles 1 ... 15: Speed |
| C01302/... | | | |
| C01302/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01303**

| Parameter \| Name: **C01303 \| Profile data: Acceleration** | | | Data type: INTEGER_32 Index: $23272_d = 5AE8_h$ |
|---|---|---|---|
| Selection of the accelerations for profiles 1 ... 15 | | | ▸ Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit/s2 | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01303/1 | 720.0000 unit/s2 | | Profiles 1 ... 15: Acceleration |
| C01303/... | | | |
| C01303/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01304**

| Parameter \| Name: **C01304 \| Profile data: Deceleration** | | | Data type: INTEGER_32 Index: $23271_d = 5AE7_h$ |
|---|---|---|---|
| Selection of the decelerations for profiles 1 ... 15 | | | ▸ Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit/s2 | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01304/1 | 720.0000 unit/s2 | | Profiles 1 ... 15: Deceleration |
| C01304/... | | | |
| C01304/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01305**

| Parameter \| Name: **C01305 \| Profile data: Final speed** | | | Data type: INTEGER_32 Index: $23270_d = 5AE6_h$ |
|---|---|---|---|
| Selection of the final speeds for profiles 1 ... 15<br>• For profile linkage with overchange.<br>▸ Profile entry | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit/s | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01305/1 | 0.0000 unit/s | | Profiles 1 ... 15: Final speed |
| C01305/... | | | |
| C01305/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01306**

| Parameter \| Name: **C01306 \| Profile data: S-ramp time** | | | Data type: UNSIGNED_16 Index: $23269_d = 5AE5_h$ |
|---|---|---|---|
| Selection of the S-ramp times for profiles 1 ... 15<br>• The maximum S-ramp time is the time in which the maximum acceleration or the maximum deceleration is reached along a ramp (jerk limitation).<br>▸ Profile entry | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 10.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01306/1 | 0.000 s | | Profiles 1 ... 15: S-ramp time |
| C01306/... | | | |
| C01306/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01307**

| Parameter \| Name: **C01307 \| Profile data: Sequence profile** | | | Data type: UNSIGNED_8 Index: $23268_d = 5AE4_h$ |
|---|---|---|---|
| Selection of the sequence profile numbers for profiles 1 ... 15<br>• For profile linkage<br>▸ Profile entry | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 15 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01307/1 | 0 | | Profiles 1 ... 15: Sequence profile |
| C01307/... | | | |
| C01307/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01308**

| Parameter \| Name: **C01308 \| Profile data: TP profile** | | | Data type: UNSIGNED_8 Index: $23267_d = 5AE3_h$ |
|---|---|---|---|
| Selection of the touch-probe profile numbers for profile 1 ... 15<br>• Only relevant for positioning modes with touch-probe.<br>▸ Profile entry | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 15 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01308/1 | 0 | | Profile 1 ... 15: TP profile |
| C01308/... | | | |
| C01308/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01309**

| Parameter \| Name: **C01309 \| Profile data: TP signal source** | | Data type: UNSIGNED_8 Index: $23266_d = 5AE2_h$ |
|---|---|---|
| Selection of the touch-probe signal sources for profile 1 ... 15<br>• Only relevant for positioning modes with touch-probe.<br>▸ Profile entry | | |
| **Selection list** | | |
| 3 | TP-DigIn3 | |
| 4 | TP-DigIn4 | |
| 5 | TP-DigIn5 | |
| 6 | TP-DigIn6 | |
| 7 | TP-DigIn7 | |
| 8 | TP-Z-trace encoder | |
| 9 | TP-Z-trace resolver | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01309/1 | 3: TP-DigIn3 | Profiles 1 ... 15: Sequence profile |
| C01309/... | | |
| C01309/15 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01310**

| Parameter \| Name: **C01310 \| Profile data: PI position** | | Data type: UNSIGNED_8 Index: $23265_d = 5AE1_h$ |
|---|---|---|
| **Selection list** | | **Info** |
| 0 | Parameter value (C1301/n) | |
| 1 | dnProcessIn1_p | |
| 2 | dnProcessIn2_p | |
| 3 | dnProcessIn3_p | |
| 4 | dnProcessIn4_p | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01310/1 | 0: Parameter value (C1301/n) | Profile 1: PI position |
| C01310/2 | 0: Parameter value (C1301/n) | Profil 2: PI-Position |
| C01310/3 | 0: Parameter value (C1301/n) | Profile 3: PI position |
| C01310/4 | 0: Parameter value (C1301/n) | Profile 4: PI position |
| C01310/5 | 0: Parameter value (C1301/n) | Profile 5: PI position |
| C01310/6 | 0: Parameter value (C1301/n) | Profile 6: PI position |
| C01310/7 | 0: Parameter value (C1301/n) | Profile 7: PI position |
| C01310/8 | 0: Parameter value (C1301/n) | Profile 8: PI position |
| C01310/9 | 0: Parameter value (C1301/n) | Profile 9: PI position |
| C01310/10 | 0: Parameter value (C1301/n) | Profile 10: PI position |
| C01310/11 | 0: Parameter value (C1301/n) | Profile 11: PI position |
| C01310/12 | 0: Parameter value (C1301/n) | Profile 12: PI position |
| C01310/13 | 0: Parameter value (C1301/n) | Profile 13: PI position |
| C01310/14 | 0: Parameter value (C1301/n) | Profile 14: PI position |
| C01310/15 | 0: Parameter value (C1301/n) | Profile 15: PI position |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01311**

| Parameter \| Name: **C01311 \| Profile data: PI speed** | | Data type: UNSIGNED_8 Index: $23264_d = 5AE0_h$ |
|---|---|---|
| **Selection list** | | **Info** |
| 0 | Parameter value (C1302/n) | |
| 1 | nProcessIn1_a x (C1302/n) | |
| 2 | nProcessIn2_a x (C1302/n) | |
| 3 | nProcessIn3_a x (C1302/n) | |
| 4 | nProcessIn4_a x (C1302/n) | |
| 5 | nProcessIn5_a x (C1302/n) | |
| 6 | nProcessIn6_a x (C1302/n) | |
| 7 | nProcessIn7_a x (C1302/n) | |
| 8 | nProcessIn8_a x (C1302/n) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01311/1 | 0: Parameter value (C1302/n) | Profile 1: PI speed |
| C01311/2 | 0: Parameter value (C1302/n) | Profile 2: PI speed |
| C01311/3 | 0: Parameter value (C1302/n) | Profile 3: PI speed |
| C01311/4 | 0: Parameter value (C1302/n) | Profile 4: PI speed |
| C01311/5 | 0: Parameter value (C1302/n) | Profile 5: PI speed |
| C01311/6 | 0: Parameter value (C1302/n) | Profile 6: PI speed |
| C01311/7 | 0: Parameter value (C1302/n) | Profile 7: PI speed |
| C01311/8 | 0: Parameter value (C1302/n) | Profile 8: PI speed |
| C01311/9 | 0: Parameter value (C1302/n) | Profile 9: PI speed |
| C01311/10 | 0: Parameter value (C1302/n) | Profile 10: PI speed |
| C01311/11 | 0: Parameter value (C1302/n) | Profile 11: PI speed |
| C01311/12 | 0: Parameter value (C1302/n) | Profile 12: PI speed |
| C01311/13 | 0: Parameter value (C1302/n) | Profile 13: PI speed |
| C01311/14 | 0: Parameter value (C1302/n) | Profile 14: PI speed |
| C01311/15 | 0: Parameter value (C1302/n) | Profile 15: PI speed |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01312**

| Parameter \| Name: **C01312 \| Profile data: PI accel.** | | Data type: UNSIGNED_8 Index: $23263_d = 5ADF_h$ |
|---|---|---|
| **Selection list** | | **Info** |
| 0 | Parameter value (C1303/n) | |
| 1 | \| nProcessIn1_a x (C1303/n) \| | |
| 2 | \| nProcessIn2_a x (C1303/n) \| | |
| 3 | \| nProcessIn3_a x (C1303/n) \| | |
| 4 | \| nProcessIn4_a x (C1303/n) \| | |
| 5 | \| nProcessIn5_a x (C1303/n) \| | |
| 6 | \| nProcessIn6_a x (C1303/n) \| | |
| 7 | \| nProcessIn7_a x (C1303/n) \| | |
| 8 | \| nProcessIn8_a x (C1303/n) \| | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01312/1 | 0: Parameter value (C1303/n) | Profile 1: PI accel. |
| C01312/2 | 0: Parameter value (C1303/n) | Profile 2: PI accel. |
| C01312/3 | 0: Parameter value (C1303/n) | Profile 3: PI accel. |
| C01312/4 | 0: Parameter value (C1303/n) | Profile 4: PI accel. |
| C01312/5 | 0: Parameter value (C1303/n) | Profile 5: PI accel. |
| C01312/6 | 0: Parameter value (C1303/n) | Profile 6: PI accel. |
| C01312/7 | 0: Parameter value (C1303/n) | Profile 7: PI accel. |
| C01312/8 | 0: Parameter value (C1303/n) | Profile 8: PI accel. |
| C01312/9 | 0: Parameter value (C1303/n) | Profile 9: PI accel. |
| C01312/10 | 0: Parameter value (C1303/n) | Profile 10: PI accel. |
| C01312/11 | 0: Parameter value (C1303/n) | Profile 11: PI accel. |
| C01312/12 | 0: Parameter value (C1303/n) | Profile 12: PI accel. |
| C01312/13 | 0: Parameter value (C1303/n) | Profile 13: PI accel. |
| C01312/14 | 0: Parameter value (C1303/n) | Profile 14: PI accel. |
| C01312/15 | 0: Parameter value (C1303/n) | Profile 15: PI accel. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01313**

| Parameter \| Name: **C01313 \| Profile data: PI decel.** | | Data type: UNSIGNED_8 Index: $23262_d = 5ADE_h$ |
|---|---|---|
| **Selection list** | | **Info** |
| 0 | Parameter value (C1304/n) | |
| 1 | \| nProcessIn1_a x (C1304/n) \| | |
| 2 | \| nProcessIn2_a x (C1304/n) \| | |
| 3 | \| nProcessIn3_a x (C1304/n) \| | |
| 4 | \| nProcessIn4_a x (C1304/n) \| | |
| 5 | \| nProcessIn5_a x (C1304/n) \| | |
| 6 | \| nProcessIn6_a x (C1304/n) \| | |
| 7 | \| nProcessIn7_a x (C1304/n) \| | |
| 8 | \| nProcessIn8_a x (C1304/n) \| | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01313/1 | 0: Parameter value (C1304/n) | Profile 1: PI decel. |
| C01313/2 | 0: Parameter value (C1304/n) | Profile 2: PI decel. |
| C01313/3 | 0: Parameter value (C1304/n) | Profile 3: PI decel. |
| C01313/4 | 0: Parameter value (C1304/n) | Profile 4: PI decel. |
| C01313/5 | 0: Parameter value (C1304/n) | Profile 5: PI decel. |
| C01313/6 | 0: Parameter value (C1304/n) | Profile 6: PI decel. |
| C01313/7 | 0: Parameter value (C1304/n) | Profile 7: PI decel. |
| C01313/8 | 0: Parameter value (C1304/n) | Profile 8: PI decel. |
| C01313/9 | 0: Parameter value (C1304/n) | Profile 9: PI decel. |
| C01313/10 | 0: Parameter value (C1304/n) | Profile 10: PI decel. |
| C01313/11 | 0: Parameter value (C1304/n) | Profile 11: PI decel. |
| C01313/12 | 0: Parameter value (C1304/n) | Profile 12: PI decel. |
| C01313/13 | 0: Parameter value (C1304/n) | Profile 13: PI decel. |
| C01313/14 | 0: Parameter value (C1304/n) | Profile 14: PI decel. |
| C01313/15 | 0: Parameter value (C1304/n) | Profile 15: PI decel. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01314**

| Parameter \| Name: **C01314 \| Profile data: PI final speed** | | Data type: UNSIGNED_8 Index: $23261_d = 5ADD_h$ |
|---|---|---|
| **Selection list** | | **Info** |
| 0 | Parameter value (C1305/n) | |
| 1 | nProcessIn1_a x (C1305/n) | |
| 2 | nProcessIn2_a x (C1305/n) | |
| 3 | nProcessIn3_a x (C1305/n) | |
| 4 | nProcessIn4_a x (C1305/n) | |
| 5 | nProcessIn5_a x (C1305/n) | |
| 6 | nProcessIn6_a x (C1305/n) | |
| 7 | nProcessIn7_a x (C1305/n) | |
| 8 | nProcessIn8_a x (C1305/n) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01314/1 | 0: Parameter value (C1305/n) | Profile 1: PI final speed |
| C01314/2 | 0: Parameter value (C1305/n) | Profile 2: PI final speed |
| C01314/3 | 0: Parameter value (C1305/n) | Profile 3: PI final speed |
| C01314/4 | 0: Parameter value (C1305/n) | Profile 4: PI final speed |
| C01314/5 | 0: Parameter value (C1305/n) | Profile 5: PI final speed |
| C01314/6 | 0: Parameter value (C1305/n) | Profile 6: PI final speed |
| C01314/7 | 0: Parameter value (C1305/n) | Profile 7: PI final speed |
| C01314/8 | 0: Parameter value (C1305/n) | Profile 8: PI final speed |
| C01314/9 | 0: Parameter value (C1305/n) | Profile 9: PI final speed |
| C01314/10 | 0: Parameter value (C1305/n) | Profile 10: PI final speed |
| C01314/11 | 0: Parameter value (C1305/n) | Profile 11: PI final speed |
| C01314/12 | 0: Parameter value (C1305/n) | Profile 12: PI final speed |
| C01314/13 | 0: Parameter value (C1305/n) | Profile 13: PI final speed |
| C01314/14 | 0: Parameter value (C1305/n) | Profile 14: PI final speed |
| C01314/15 | 0: Parameter value (C1305/n) | Profile 15: PI final speed |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01315**

| Parameter \| Name: **C01315 \| Profile data: PI S-ramp time** | | Data type: UNSIGNED_8 Index: $23260_d = 5ADC_h$ |
|---|---|---|
| **Selection list** | | **Info** |
| 0 | Parameter value (C1306/n) | |
| 1 | \| nProcessIn1_a \| x (C1306/n) | |
| 2 | \| nProcessIn2_a \| x (C1306/n) | |
| 3 | \| nProcessIn3_a \| x (C1306/n) | |
| 4 | \| nProcessIn4_a \| x (C1306/n) | |
| 5 | \| nProcessIn5_a \| x (C1306/n) | |
| 6 | \| nProcessIn6_a \| x (C1306/n) | |
| 7 | \| nProcessIn7_a \| x (C1306/n) | |
| 8 | \| nProcessIn8_a \| x (C1306/n) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01315/1 | 0: Parameter value (C1306/n) | Profile 1: PI S-ramp time |
| C01315/2 | 0: Parameter value (C1306/n) | Profile 2: PI S-ramp time |
| C01315/3 | 0: Parameter value (C1306/n) | Profile 3: PI S-ramp time |
| C01315/4 | 0: Parameter value (C1306/n) | Profile 4: PI S-ramp time |
| C01315/5 | 0: Parameter value (C1306/n) | Profile 5: PI S-ramp time |
| C01315/6 | 0: Parameter value (C1306/n) | Profile 6: PI S-ramp time |
| C01315/7 | 0: Parameter value (C1306/n) | Profile 7: PI S-ramp time |
| C01315/8 | 0: Parameter value (C1306/n) | Profile 8: PI S-ramp time |
| C01315/9 | 0: Parameter value (C1306/n) | Profile 9: PI S-ramp time |
| C01315/10 | 0: Parameter value (C1306/n) | Profile 10: PI S-ramp time |
| C01315/11 | 0: Parameter value (C1306/n) | Profile 11: PI S-ramp time |
| C01315/12 | 0: Parameter value (C1306/n) | Profile 12: PI S-ramp time |
| C01315/13 | 0: Parameter value (C1306/n) | Profile 13: PI S-ramp time |
| C01315/14 | 0: Parameter value (C1306/n) | Profile 14: PI S-ramp time |
| C01315/15 | 0: Parameter value (C1306/n) | Profile 15: PI S-ramp time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01320**

| Parameter \| Name: **C01320 \| Profile data: Position** | | | Data type: INTEGER_32 Index: $23255_d = 5AD7_h$ |
|---|---|---|---|
| **Display area** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01320/1 | | | Profile 1: Position |
| C01320/2 | | | Profile 2: Position |
| C01320/3 | | | Profile 3: Position |
| C01320/4 | | | Profile 4: Position |
| C01320/5 | | | Profile 5: Position |
| C01320/6 | | | Profile 6: Position |
| C01320/7 | | | Profile 7: Position |
| C01320/8 | | | Profile 8: Position |
| C01320/9 | | | Profile 9: Position |
| C01320/10 | | | Profile 10: Position |
| C01320/11 | | | Profile 11: Position |
| C01320/12 | | | Profile 12: Position |
| C01320/13 | | | Profile 13: Position |
| C01320/14 | | | Profile 14: Position |
| C01320/15 | | | Profile 15: Position |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01321**

| Parameter \| Name: **C01321 \| Profile data: Speed** | | | Data type: INTEGER_32 Index: $23254_d = 5AD6_h$ |
|---|---|---|---|
| **Display area** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit/s | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01321/1 | | | Profile 1: Speed |
| C01321/2 | | | Profile 2: Speed |
| C01321/3 | | | Profile 3: Speed |
| C01321/4 | | | Profile 4: Speed |
| C01321/5 | | | Profile 5: Speed |
| C01321/6 | | | Profile 6: Speed |
| C01321/7 | | | Profile 7: Speed |
| C01321/8 | | | Profile 8: Speed |
| C01321/9 | | | Profile 9: Speed |
| C01321/10 | | | Profile 10: Speed |
| C01321/11 | | | Profile 11: Speed |
| C01321/12 | | | Profile 12: Speed |
| C01321/13 | | | Profile 13: Speed |
| C01321/14 | | | Profile 14: Speed |
| C01321/15 | | | Profile 15: Speed |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01322**

| Parameter \| Name: **C01322 \| Profile data: Acceleration** | | | Data type: INTEGER_32 Index: $23253_d = 5AD5_h$ |
|---|---|---|---|
| **Display area** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit/s2 | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01322/1 | | | Profile 1: Accel. |
| C01322/2 | | | Profile 2: Accel. |
| C01322/3 | | | Profile 3: Accel. |
| C01322/4 | | | Profile 4: Accel. |
| C01322/5 | | | Profile 5: Accel. |
| C01322/6 | | | Profile 6: Accel. |
| C01322/7 | | | Profile 7: Accel. |
| C01322/8 | | | Profile 8: Accel. |
| C01322/9 | | | Profile 9: Accel. |
| C01322/10 | | | Profile 10: Accel. |
| C01322/11 | | | Profile 11: Accel. |
| C01322/12 | | | Profile 12: Accel. |
| C01322/13 | | | Profile 13: Accel. |
| C01322/14 | | | Profile 14: Accel. |
| C01322/15 | | | Profile 15: Accel. |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01323**

| Parameter \| Name: **C01323 \| Profile data: Deceleration** | | | Data type: INTEGER_32 Index: $23252_d = 5AD4_h$ |
|---|---|---|---|
| **Display area** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit/s2 | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01323/1 | | | Profile 1: Decel. |
| C01323/2 | | | Profile 2: Decel. |
| C01323/3 | | | Profile 3: Decel. |
| C01323/4 | | | Profile 4: Decel. |
| C01323/5 | | | Profile 5: Decel. |
| C01323/6 | | | Profile 6: Decel. |
| C01323/7 | | | Profile 7: Decel. |
| C01323/8 | | | Profile 8: Decel. |
| C01323/9 | | | Profile 9: Decel. |
| C01323/10 | | | Profile 10: Decel. |
| C01323/11 | | | Profile 11: Decel. |
| C01323/12 | | | Profile 12: Decel. |
| C01323/13 | | | Profile 13: Decel. |
| C01323/14 | | | Profile 14: Decel. |
| C01323/15 | | | Profile 15: Decel. |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01324**

| Parameter \| Name: **C01324 \| Profile data: Final speed** | | | Data type: INTEGER_32 Index: $23251_d = 5AD3_h$ |
|---|---|---|---|
| **Display area** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit/s | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01324/1 | | | Profile 1: Final speed |
| C01324/2 | | | Profile 2: Final speed |
| C01324/3 | | | Profile 3: Final speed |
| C01324/4 | | | Profile 4: Final speed |
| C01324/5 | | | Profile 5: Final speed |
| C01324/6 | | | Profile 6: Final speed |
| C01324/7 | | | Profile 7: Final speed |
| C01324/8 | | | Profile 8: Final speed |
| C01324/9 | | | Profile 9: Final speed |
| C01324/10 | | | Profile 10: Final speed |
| C01324/11 | | | Profile 11: Final speed |
| C01324/12 | | | Profile 12: Final speed |
| C01324/13 | | | Profile 13: Final speed |
| C01324/14 | | | Profile 14: Final speed |
| C01324/15 | | | Profile 15: Final speed |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01325**

| Parameter \| Name: **C01325 \| Profile data: S-ramp time** | | | Data type: UNSIGNED_16 Index: $23250_d = 5AD2_h$ |
|---|---|---|---|
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 10.000 | |
| **Subcodes** | | | **Info** |
| C01325/1 | | | Profile 1: S-ramp time |
| C01325/2 | | | Profile 2: S-ramp time |
| C01325/3 | | | Profile 3: S-ramp time |
| C01325/4 | | | Profile 4: S-ramp time |
| C01325/5 | | | Profile 5: S-ramp time |
| C01325/6 | | | Profile 6: S-ramp time |
| C01325/7 | | | Profile 7: S-ramp time |
| C01325/8 | | | Profile 8: S-ramp time |
| C01325/9 | | | Profile 9: S-ramp time |
| C01325/10 | | | Profile 10: S-ramp time |
| C01325/11 | | | Profile 11: S-ramp time |
| C01325/12 | | | Profile 12: S-ramp time |
| C01325/13 | | | Profile 13: S-ramp time |
| C01325/14 | | | Profile 14: S-ramp time |
| C01325/15 | | | Profile 15: S-ramp time |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01350**

| Parameter \| Name: **C01350 \| ACDrive: Drive mode** | | Data type: UNSIGNED_8 Index: $23225_d = 5AB9_h$ |
|---|---|---|
| **Selection list** | | |
| 1 | Speed mode | |
| 3 | Torque mode | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01350/1 | 1: Speed mode | ACDrive: Drive mode |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01351**

| Parameter \| Name: **C01351 \| ACDrive: Control word** | | | Data type: UNSIGNED_16 Index: $23224_d = 5AB8_h$ |
|---|---|---|---|
| From version 13.00.00<br>Display of the "AC Drive Profile" control word for the inverter<br>• Detailed information on the "AC Drive Profile" can be found in the EtherNet/IP™ communication manual.<br>▸ Actuating drive speed (AC Drive Profile) | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Run Forward | | Relationships between Run1 and Run2 and trigger events can be found in the chapter "Run/Stop Event". |
| Bit 1 | Run Backward | | |
| Bit 2 | Fault Reset | | 0->1 ≡ Reset error<br>0 ≡ No response |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | NetCtrl | | Run/Stop control:<br>0 ≡ via local setting in the device or terminal<br>1 ≡ via fieldbus (e.g. by the scanner) |
| Bit 6 | NetRef | | Reference speed/reference torque:<br>0 ≡ via local setting in the device or terminal<br>1 ≡ via fieldbus (e.g. by the scanner) |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | | | **Info** |
| C01351/1 | | | ACDrive: Control word |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C01352**

| Parameter \| Name: **C01352 \| ACDrive: Status word** | | | Data type: UNSIGNED_16 Index: $23223_d = 5AB7_h$ |
|---|---|---|---|
| From version 13.00.00<br>Display of the "AC Drive Profile" status word of the inverter<br>• Detailed information on the "AC Drive Profile" can be found in the EtherNet/IP™ communication manual.<br>▸ Actuating drive speed (AC Drive Profile) | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Faulted | | 0 ≡ No errors<br>1 ≡ Errors have occurred |
| Bit 1 | Warning | | 0 ≡ No warnings<br>1 ≡ Warnings have occurred |
| Bit 2 | Running1 (Fwd) | | Relationships between Run1 and Run2 and trigger events can be found in the chapter "Run/Stop Event". |
| Bit 3 | Running2 (Rev) | | |
| Bit 4 | Ready | | 0 ≡ Different status than in case of "1"<br>1 ≡ Ready or Enabled or Stopping |
| Bit 5 | Ctrl from Net | | Run/Stop control:<br>0 ≡ via local setting in the device or terminal<br>1 ≡ via fieldbus (e.g. by the scanner) |
| Bit 6 | Ref from Net | | Reference speed/reference torque:<br>0 ≡ via local setting in the device or terminal<br>1 ≡ via fieldbus (e.g. by the scanner) |
| Bit 7 | At Reference | | 1 ≡ Currently, the inverter runs with the reference speed or reference torque (depending on the "drive mode" set in C01350/1). |
| Bit 8 | DriveState_0 | | The "Drive State" is coded as follows:<br>0: Manufacturer-specific (not used with 8400)<br>1: Start-up (drive initialisation)<br>2: Not_Ready (mains voltage switched off)<br>3: Ready (mains voltage switched-on)<br>4: Enabled (drive has received "Run" command)<br>5: Stopping (drive has received "Stop" command and is stopped)<br>6: Fault_Stop (drive is stopped due to an error)<br>7: Faulted (errors have occurred) |
| Bit 9 | DriveState_1 | | |
| Bit 10 | DriveState_2 | | |
| Bit 11 | DriveState_3 | | |
| Bit 12 | DriveState_4 | | |
| Bit 13 | DriveState_5 | | |
| Bit 14 | DriveState_6 | | |
| Bit 15 | DriveState_7 | | |
| **Subcodes** | | | **Info** |
| C01352/1 | | | ACDrive: Status word |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C01353**

| Parameter \| Name: **C01353 \| ACDrive: Setpoint scaling** | | | Data type: INTEGER_8 Index: $23222_d = 5AB6_h$ |
|---|---|---|---|
| From version 13.00.00 | | | |
| ▸ Actuating drive speed (AC Drive Profil): Scaling of the speed and torque values | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -128 | | 127 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01353/1 | 0 | | ACDrive: Speed scaling |
| C01353/2 | 0 | | ACDrive: Torque scaling |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01400**

| Parameter \| Name: **C01400 \| L_Sequencer_1: Sequence step** | | | Data type: UNSIGNED_16 Index: $23175_d = 5A87_h$ |
|---|---|---|---|
| From version 12.00.00<br>In the subcodes, the calls of the actions required for the positioning program are stored. In this way, the basic sequence (with the exception of the branches) is defined.<br>▸ L_Sequencer_1 (sequence control) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 10005 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01400/1 | 0 | | L_Sequencer_1: Action in sequence step 1 |
| C01400/... | | | ... |
| C01400/100 | | | L_Sequencer_1: Action in sequence step 100 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01401**

| Parameter \| Name: **C01401 \| L_Sequencer_1: Current step** | | | Data type: UNSIGNED_16 Index: $23174_d = 5A86_h$ |
|---|---|---|---|
| From version 12.00.00<br>▸ L_Sequencer_1 (sequence control) | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | | | **Info** |
| C01401/1 | | | L_Sequencer_1: Current step |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01402**

| Parameter \| Name: **C01402 \| L_Sequencer_1: Sequence control: Status** | | | Data type: UNSIGNED_16 Index: $23173_d = 5A85_h$ |
|---|---|---|---|
| From version 12.00.00<br>0 - READY: Positioning program ready to start/Program end reached.<br>1 - RUN: Positioning program running.<br>2 - PAUSE: Current step interrupted, pause active.<br>3 - BREAK: Positioning program interrupted.<br>4 - RESET: Positioning program reset.<br>▸ L_Sequencer_1 (sequence control) | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 4 | |
| **Subcodes** | | | **Info** |
| C01402/1 | | | L_Sequencer_1: Sequence control: Status |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01403**

| Parameter \| Name: **C01403 \| L_Sequencer_1: Sequence control: Control word** | | | Data type: UNSIGNED_16 Index: $23172_d = 5A84_h$ |
|---|---|---|---|
| From version 12.00.00<br>Bit-coded code for controlling the sequencer with a PC or master control via the parameter channel. Setting a bit to "1" activates the corresponding function:<br>• Bit 0: Start<br>• Bit 1: Pause<br>• Bit 2: Break<br>• Bit 3: Cancel<br>• Bit 4: Reset<br>• Bit 5: NextStep<br>▸ L_Sequencer_1 (sequence control) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 64 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01403/1 | 0 | | L_Sequencer_1: Sequence control: Control word |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01404**

| Parameter \| Name: **C01404 \| L_Sequencer_1: Step for bCancel = TRUE** | | | Data type: UNSIGNED_16 Index: $23171_d = 5A83_h$ |
|---|---|---|---|
| From version 12.00.00<br>▸ L_Sequencer_1 (sequence control) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 101 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01404/1 | 0 | | L_Sequencer_1: Step for bCancel = TRUE |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01405**

| Parameter \| Name: **C01405 \| L_Sequencer_1: PSInput** | | Data type: UNSIGNED_16 Index: $23170_d = 5A82_h$ |
|---|---|---|
| From version 12.00.00 | | ▸ L_Sequencer_1: "Positioning" action |
| **Selection list** | | |
| 0 | Waiting function deactivated | |
| 1 | Input 1(wDigitalInp. Bit00) | |
| 2 | Input 2(wDigitalInp. Bit01) | |
| 3 | Input 3(wDigitalInp. Bit02) | |
| 4 | Input 4(wDigitalInp. Bit03) | |
| 5 | Input 5(wDigitalInp. Bit04) | |
| 6 | Input 6(wDigitalInp. Bit05) | |
| 7 | Input 7(wDigitalInp. Bit06) | |
| 8 | Input 8(wDigitalInp. Bit07) | |
| 9 | Input 9(wDigitalInp. Bit08) | |
| 10 | Input 10(wDigitalInp. Bit09) | |
| 11 | Input 11(wDigitalInp. Bit10) | |
| 12 | Input 12(wDigitalInp. Bit11) | |
| 13 | Input 13(wDigitalInp. Bit12) | |
| 14 | Input 14(wDigitalInp. Bit13) | |
| 15 | Input 15(wDigitalInp. Bit14) | |
| 16 | Input 16(wDigitalInp. Bit15) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01405/1 | 0: Waiting function deactivated | L_Sequencer_1: Pos. action 1: Start with |
| C01405/... | | ... |
| C01405/50 | | L_Sequencer_1: Pos. action 50: Start with |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01406**

| Parameter \| Name: **C01406 \| L_Sequencer_1: PSLevel** | | | Data type: UNSIGNED_8 Index: $23169_d = 5A81_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Positioning" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 1 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01406/1 | 0 | | L_Sequencer_1: Pos. action 1: input polarity |
| C01406/... | | | ... |
| C01406/50 | | | L_Sequencer_1: Pos. action 50: input polarity |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01407**

| Parameter \| Name: **C01407 \| L_Sequencer_1: PSProfil** | | | Data type: UNSIGNED_16 Index: $23168_d = 5A80_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Positioning" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 15 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01407/1 | 0 | | L_Sequencer_1: Pos. action 1: Profile number |
| C01407/... | | | ... |
| C01407/50 | | | L_Sequencer_1: Pos. action 50: Profile number |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01408**

| Parameter \| Name: **C01408 \| L_Sequencer_1: PSStep** | | | Data type: UNSIGNED_16 Index: $23167_d = 5A7F_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Positioning" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01408/1 | 0 | | L_Sequencer_1: Pos. action 1: Jump destination |
| C01408/... | | | ... |
| C01408/50 | | | L_Sequencer_1: Pos. action 50: Jump destination |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01409**

| Parameter \| Name: **C01409 \| L_Sequencer_1: PS_WD_Time** | | | Data type: UNSIGNED_32 Index: $23166_d = 5A7E_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Positioning" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | | 2147480.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01409/1 | 0.000 | | L_Sequencer_1: Pos. action 1: Monitoring time |
| C01409/... | | | ... |
| C01409/50 | | | L_Sequencer_1: Pos. action 50: Monitoring time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01410**

| Parameter \| Name: **C01410 \| L_Sequencer_1: PS_WD_Step** | | | Data type: UNSIGNED_16 Index: $23165_d = 5A7D_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Positioning" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01410/1 | 0 | | L_Sequencer_1: Pos. action 1: Jump destination monit. |
| C01410/... | | | ... |
| C01410/50 | | | L_Sequencer_1: Pos. action 50: Jump destination monit. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01411**

| Parameter \| Name: **C01411 \| L_Sequencer_1: SwitchOut1** | | Data type: UNSIGNED_16 Index: $23164_d = 5A7C_h$ |
|---|---|---|
| From version 12.00.00 | | ▸ L_Sequencer_1: "Switching" action |
| **Selection list** | | |
| 0 | Deactivated | |
| 1 | Output 1 (wDigitalOutp. Bit00) | |
| 2 | Output 2 (wDigitalOutp. Bit01) | |
| 3 | Output 3 (wDigitalOutp. Bit02) | |
| 4 | Output 4 (wDigitalOutp. Bit03) | |
| 5 | Output 5 (wDigitalOutp. Bit04) | |
| 6 | Output 6 (wDigitalOutp. Bit05) | |
| 7 | Output 7 (wDigitalOutp. Bit06) | |
| 8 | Output 8 (wDigitalOutp. Bit07) | |
| 9 | Output 9 (wDigitalOutp. Bit08) | |
| 10 | Output 10 (wDigitalOutp. Bit09) | |
| 11 | Output 11 (wDigitalOutp. Bit10) | |
| 12 | Output 12 (wDigitalOutp. Bit11) | |
| 13 | Output 13 (wDigitalOutp. Bit12) | |
| 14 | Output 14 (wDigitalOutp. Bit13) | |
| 15 | Output 15 (wDigitalOutp. Bit14) | |
| 16 | Output 16 (wDigitalOutp. Bit15) | |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C01411/1 | 0: Deactivated | L_Sequencer_1: Switch. act. 1: output switch. A |
| C01411/... | | ... |
| C01411/16 | | L_Sequencer_1: Switch. act. 16: output switch. A |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01412**

| Parameter \| Name: **C01412 \| L_Sequencer_1: SwitchLevel1** | | | Data type: UNSIGNED_8 Index: $23163_d = 5A7B_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Switching" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 1 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01412/1 | 0 | | L_Sequencer_1: Switch. act. 1: pol. switch. A |
| C01412/... | | | ... |
| C01412/16 | | | L_Sequencer_1: Switch. act. 16: pol. switch. A |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01413**

| Parameter \| Name: **C01413 \| L_Sequencer_1: SwitchOut2** | | Data type: UNSIGNED_16 Index: $23162_d = 5A7A_h$ |
|---|---|---|
| From version 12.00.00 | | ▸ L_Sequencer_1: "Switching" action |
| **Selection list** | | |
| 0 | Deactivated | |
| 1 | Output 1 (wDigitalOutp. Bit00) | |
| 2 | Output 2 (wDigitalOutp. Bit01) | |
| 3 | Output 3 (wDigitalOutp. Bit02) | |
| 4 | Output 4 (wDigitalOutp. Bit03) | |
| 5 | Output 5 (wDigitalOutp. Bit04) | |
| 6 | Output 6 (wDigitalOutp. Bit05) | |
| 7 | Output 7 (wDigitalOutp. Bit06) | |
| 8 | Output 8 (wDigitalOutp. Bit07) | |
| 9 | Output 9 (wDigitalOutp. Bit08) | |
| 10 | Output 10 (wDigitalOutp. Bit09) | |
| 11 | Output 11 (wDigitalOutp. Bit10) | |
| 12 | Output 12 (wDigitalOutp. Bit11) | |
| 13 | Output 13 (wDigitalOutp. Bit12) | |
| 14 | Output 14 (wDigitalOutp. Bit13) | |
| 15 | Output 15 (wDigitalOutp. Bit14) | |
| 16 | Output 16 (wDigitalOutp. Bit15) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01413/1 | 0: Deactivated | L_Sequencer_1: Switch. act. 1: output switch. B |
| C01413/... | | ... |
| C01413/16 | | L_Sequencer_1: Switch. act. 16: output switch. B |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01414**

| Parameter \| Name: **C01414 \| L_Sequencer_1: SwitchLevel2** | | | Data type: UNSIGNED_8 Index: $23161_d = 5A79_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Switching" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 1 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01414/1 | 0 | | L_Sequencer_1: Switch. act. 1: pol. switch. B |
| C01414/... | | | ... |
| C01414/16 | | | L_Sequencer_1: Switch. act. 16: pol. switch. B |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01415**

| Parameter \| Name: **C01415 \| L_Sequencer_1: BranchIn** | | Data type: UNSIGNED_16 Index: $23160_d = 5A78_h$ |
|---|---|---|
| From version 12.00.00 | | ▸ L_Sequencer_1: "Branch" action |
| **Selection list** | | |
| 0 | Unconditional branch | |
| 1 | Input 1(wDigitalInp. Bit00) | |
| 2 | Input 2(wDigitalInp. Bit01) | |
| 3 | Input 3(wDigitalInp. Bit02) | |
| 4 | Input 4(wDigitalInp. Bit03) | |
| 5 | Input 5(wDigitalInp. Bit04) | |
| 6 | Input 6(wDigitalInp. Bit05) | |
| 7 | Input 7(wDigitalInp. Bit06) | |
| 8 | Input 8(wDigitalInp. Bit07) | |
| 9 | Input 9(wDigitalInp. Bit08) | |
| 10 | Input 10(wDigitalInp. Bit09) | |
| 11 | Input 11(wDigitalInp. Bit10) | |
| 12 | Input 12(wDigitalInp. Bit11) | |
| 13 | Input 13(wDigitalInp. Bit12) | |
| 14 | Input 14(wDigitalInp. Bit13) | |
| 15 | Input 15(wDigitalInp. Bit14) | |
| 16 | Input 16(wDigitalInp. Bit15) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01415/1 | 0: Unconditional branch | L_Sequencer_1: Branch act. 1: Input for jump |
| C01415/... | | ... |
| C01415/16 | | L_Sequencer_1: Branch act. 16: Input for jump |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01416**

| Parameter \| Name: **C01416 \| L_Sequencer_1: BranchLevel** | | | Data type: UNSIGNED_8 Index: $23159_d = 5A77_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 1 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01416/1 | 0 | | L_Sequencer_1: Branch act. 1: Input polarity |
| C01416/... | | | ... |
| C01416/16 | | | L_Sequencer_1: Branch act. 16: Input polarity |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01417**

| Parameter \| Name: **C01417 \| L_Sequencer_1: BranchStep** | | | Data type: UNSIGNED_16 Index: $23158_d = 5A76_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01417/1 | 0 | | L_Sequencer_1: Branch act.1: Jump destination |
| C01417/... | | | ... |
| C01417/16 | | | L_Sequencer_1: Branch act. 16: Jump destination |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01418**

| Parameter \| Name: **C01418 \| L_Sequencer_1: BranchStep1** | | | Data type: UNSIGNED_16 Index: $23157_d = 5A75_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01418/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 1 |
| C01418/2 | 0 | | L_Sequencer_1: Var. branch2: Jump destination 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01419**

| Parameter \| Name: C01419 \| L_Sequencer_1: BranchStep2 | | | Data type: UNSIGNED_16 Index: $23156_d = 5A74_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01419/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 2 |
| C01419/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 2 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01420**

| Parameter \| Name: C01420 \| L_Sequencer_1: BranchStep3 | | | Data type: UNSIGNED_16 Index: $23155_d = 5A73_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01420/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 3 |
| C01420/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01421**

| Parameter \| Name: C01421 \| L_Sequencer_1: BranchStep4 | | | Data type: UNSIGNED_16 Index: $23154_d = 5A72_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01421/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 4 |
| C01421/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01422**

| Parameter \| Name: C01422 \| L_Sequencer_1: BranchStep5 | | | Data type: UNSIGNED_16 Index: $23153_d = 5A71_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01422/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 5 |
| C01422/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 5 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01423**

| Parameter \| Name: **C01423 \| L_Sequencer_1: BranchStep6** | | | Data type: UNSIGNED_16 Index: $23152_d = 5A70_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01423/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 6 |
| C01423/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 6 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01424**

| Parameter \| Name: **C01424 \| L_Sequencer_1: BranchStep7** | | | Data type: UNSIGNED_16 Index: $23151_d = 5A6F_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01424/1 | 0 | | L_Sequencer_1: Var. branch1: Jump destination 7 |
| C01424/2 | 0 | | L_Sequencer_1: Var. branch2: Jump destination 7 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01425**

| Parameter \| Name: **C01425 \| L_Sequencer_1: BranchStep8** | | | Data type: UNSIGNED_16 Index: $23150_d = 5A6E_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01425/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 8 |
| C01425/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 8 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01426**

| Parameter \| Name: **C01426 \| L_Sequencer_1: BranchStep9** | | | Data type: UNSIGNED_16 Index: $23149_d = 5A6D_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01426/1 | 0 | | L_Sequencer_1: Var. branch1: Jump destination 9 |
| C01426/2 | 0 | | L_Sequencer_1: Var. branch2: Jump destination 9 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01427**

| Parameter \| Name: C01427 \| L_Sequencer_1: BranchStep10 | | | Data type: UNSIGNED_16 Index: $23148_d = 5A6C_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01427/1 | 0 | | L_Sequencer_1: Var. branch1: Jump destin. 10 |
| C01427/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 10 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01428**

| Parameter \| Name: C01428 \| L_Sequencer_1: BranchStep11 | | | Data type: UNSIGNED_16 Index: $23147_d = 5A6B_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01428/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 11 |
| C01428/2 | 0 | | L_Sequencer_1: Var. branch2: Jump destination 11 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01429**

| Parameter \| Name: C01429 \| L_Sequencer_1: BranchStep12 | | | Data type: UNSIGNED_16 Index: $23146_d = 5A6A_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01429/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 12 |
| C01429/2 | 0 | | L_Sequencer_1: Var. branch2: Jump destin. 12 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01430**

| Parameter \| Name: C01430 \| L_Sequencer_1: BranchStep13 | | | Data type: UNSIGNED_16 Index: $23145_d = 5A69_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01430/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 13 |
| C01430/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 13 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01431**

| Parameter \| Name: **C01431 \| L_Sequencer_1: BranchStep14** | | | Data type: UNSIGNED_16 Index: $23144_d = 5A68_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01431/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 14 |
| C01431/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 14 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01432**

| Parameter \| Name: **C01432 \| L_Sequencer_1: BranchStep15** | | | Data type: UNSIGNED_16 Index: $23143_d = 5A67_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01432/1 | 0 | | L_Sequencer_1: Var. branch1: Jump destination 15 |
| C01432/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 15 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01433**

| Parameter \| Name: **C01433 \| L_Sequencer_1: BranchStep16** | | | Data type: UNSIGNED_16 Index: $23142_d = 5A66_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01433/1 | 0 | | L_Sequencer_1: Var. branch1: Jump destination 16 |
| C01433/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 16 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01434**

| Parameter \| Name: **C01434 \| L_Sequencer_1: BranchStep17** | | | Data type: UNSIGNED_16 Index: $23141_d = 5A65_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01434/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 17 |
| C01434/2 | 0 | | L_Sequencer_1: Var. branch2: Jump destination 17 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01435**

| Parameter \| Name: **C01435 \| L_Sequencer_1: BranchStep18** | | | Data type: UNSIGNED_16 Index: $23140_d = 5A64_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01435/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 18 |
| C01435/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 18 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01436**

| Parameter \| Name: **C01436 \| L_Sequencer_1: BranchStep19** | | | Data type: UNSIGNED_16 Index: $23139_d = 5A63_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01436/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 19 |
| C01436/2 | 0 | | L_Sequencer_1: Var. branch2: Jump destination 19 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01437**

| Parameter \| Name: **C01437 \| L_Sequencer_1: BranchStep20** | | | Data type: UNSIGNED_16 Index: $23138_d = 5A62_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Variable branch" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01437/1 | 0 | | L_Sequencer_1: Var. branch 1: Jump destin. 20 |
| C01437/2 | 0 | | L_Sequencer_1: Var. branch 2: Jump destin. 20 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01438**

| Parameter \| Name: **C01438 \| L_Sequencer_1: Wartezeit** | | | Data type: UNSIGNED_32 Index: $23137_d = 5A61_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Waiting" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 2127480.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01438/1 | 0.000 s | | L_Sequencer_1: Wait. act. 1: Waiting time |
| C01438/... | | | ... |
| C01438/8 | | | L_Sequencer_1: Wait. act. 8: Waiting time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01439**

| Parameter \| Name: **C01439 \| L_Sequencer_1: Wait Eingang f. Weiter** | | Data type: UNSIGNED_16 Index: $23136_d = 5A60_h$ |
|---|---|---|
| From version 12.00.00 | | ▸ L_Sequencer_1: "Waiting" action |
| **Selection list** | | |
| 0 | Input deactivated | |
| 1 | Input 1(wDigitalInp. Bit00) | |
| 2 | Input 2(wDigitalInp. Bit01) | |
| 3 | Input 3(wDigitalInp. Bit02) | |
| 4 | Input 4(wDigitalInp. Bit03) | |
| 5 | Input 5(wDigitalInp. Bit04) | |
| 6 | Input 6(wDigitalInp. Bit05) | |
| 7 | Input 7(wDigitalInp. Bit06) | |
| 8 | Input 8(wDigitalInp. Bit07) | |
| 9 | Input 9(wDigitalInp. Bit08) | |
| 10 | Input 10(wDigitalInp. Bit09) | |
| 11 | Input 11(wDigitalInp. Bit10) | |
| 12 | Input 12(wDigitalInp. Bit11) | |
| 13 | Input 13(wDigitalInp. Bit12) | |
| 14 | Input 14(wDigitalInp. Bit13) | |
| 15 | Input 15(wDigitalInp. Bit14) | |
| 16 | Input 16(wDigitalInp. Bit15) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01439/1 | 0: Input deactivated | L_Sequencer_1: Wait. act. 1: Input for "Next" |
| C01439/... | | ... |
| C01439/8 | | L_Sequencer_1: Wait. act. 8: Input for "Next" |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01440**

| Parameter \| Name: **C01440 \| L_Sequencer_1: WaitLevel** | | | Data type: UNSIGNED_8 Index: $23135_d = 5A5F_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Waiting" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 1 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01440/1 | 0 | | L_Sequencer_1: Wait. act. 1: Input polarity |
| C01440/... | | | ... |
| C01440/8 | | | L_Sequencer_1: Wait. act. 8: Input polarity |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01441**

| Parameter \| Name: **C01441 \| L_Sequencer_1: SetZählernummer** | | | Data type: UNSIGNED_16 Index: $23134_d = 5A5E_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Counter setting" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 5 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01441/1 | 1 | | L_Sequencer_1: Counter set act. 1: Counter no. |
| C01441/... | | | ... |
| C01441/5 | | | L_Sequencer_1: Counter set act. 5: Counter no. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01442**

| Parameter \| Name: **C01442 \| L_Sequencer_1: SetZählerStartwert** | | | Data type: INTEGER_32 Index: $23133_d = 5A5D_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Counter setting" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01442/1 | 0 | | L_Sequencer_1: Counter set act. 1: Starting value |
| C01442/... | | | ... |
| C01442/5 | | | L_Sequencer_1: Counter set act. 5: Starting value |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01443**

| Parameter \| Name: **C01443 \| L_Sequencer_1: Aktueller Zählerstand** | | | Data type: INTEGER_32 Index: $23132_d = 5A5C_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Counter setting" action |
| **Display area** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | | | **Info** |
| C01443/1 | | | L_Sequencer_1: Counter 1: Current counter content |
| C01443/... | | | ... |
| C01443/5 | | | L_Sequencer_1: Counter 5: Current counter content |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01444**

| Parameter \| Name: **C01444 \| L_Sequencer_1: ZählerNummer** | | | Data type: UNSIGNED_16 Index: $23131_d = 5A5B_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Counting" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 5 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01444/1 | 1 | | L_Sequencer_1: Count. act. 1: Counter no. |
| C01444/... | | | ... |
| C01444/8 | | | L_Sequencer_1: Count. act. 8: Counter no. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01445**

| Parameter \| Name: **C01445 \| L_Sequencer_1: SchrittwertZähler** | | | Data type: INTEGER_32 Index: $23130_d = 5A5A_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Counting" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01445/1 | 1 | | L_Sequencer_1: Count. act. 1: Step value |
| C01445/... | | | ... |
| C01445/8 | | | L_Sequencer_1: Count. act. 8: Step value |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01446**

| Parameter \| Name: **C01446 \| L_Sequencer_1: ZählerVergleichswert** | | | Data type: INTEGER_32 Index: $23129_d = 5A59_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Counting" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01446/1 | 0 | | L_Sequencer_1: Count. act. 1: comparison value |
| C01446/... | | | ... |
| C01446/8 | | | L_Sequencer_1: Count. act. 8: comparison value |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01447**

| Parameter \| Name: **C01447 \| L_Sequencer_1: ZählerSprungziel** | | | Data type: UNSIGNED_16 Index: $23128_d = 5A58_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Counting" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01447/1 | 0 | | L_Sequencer_1: Count. act. 1: Jump destination |
| C01447/... | | | ... |
| C01447/8 | | | L_Sequencer_1: Count. act. 8: Jump destination |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01448**

| Parameter \| Name: **C01448 \| L_Sequencer_1: ZählerVergleichsfunktion** | | Data type: UNSIGNED_16 Index: $23127_d = 5A57_h$ |
|---|---|---|
| From version 12.00.00 | | ▸ L_Sequencer_1: "Counting" action |
| **Selection list** | | |
| 1 | Counter content = comparison value | |
| 2 | Counter content > comparison value | |
| 3 | Counter content >= comparison value | |
| 4 | Counter content < comparison value | |
| 5 | Counter content <= comparison value | |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C01448/1 | 1: Counter content = comparison value | L_Sequencer_1: Count. act. 1: Comparison op. |
| C01448/... | | ... |
| C01448/8 | | L_Sequencer_1: Count. act. 8: Comaprison op. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01449**

| Parameter \| Name: **C01449 \| L_Sequencer_1: Standby Eingang f. Ende** | | Data type: UNSIGNED_16 Index: $23126_d = 5A56_h$ |
|---|---|---|
| From version 12.00.00 | | ▸ L_Sequencer_1: "Standby" action |
| **Selection list** | | |
| 1 | Input 1(wDigitalInp. Bit00) | |
| 2 | Input 2(wDigitalInp. Bit01) | |
| 3 | Input 3(wDigitalInp. Bit02) | |
| 4 | Input 4(wDigitalInp. Bit03) | |
| 5 | Input 5(wDigitalInp. Bit04) | |
| 6 | Input 6(wDigitalInp. Bit05) | |
| 7 | Input 7(wDigitalInp. Bit06) | |
| 8 | Input 8(wDigitalInp. Bit07) | |
| 9 | Input 9(wDigitalInp. Bit08) | |
| 10 | Input 10(wDigitalInp. Bit09) | |
| 11 | Input 11(wDigitalInp. Bit10) | |
| 12 | Input 12(wDigitalInp. Bit11) | |
| 13 | Input 13(wDigitalInp. Bit12) | |
| 14 | Input 14(wDigitalInp. Bit13) | |
| 15 | Input 15(wDigitalInp. Bit14) | |
| 16 | Input 16(wDigitalInp. Bit15) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01449/1 | 1: Input 1(wDigitalInp. Bit00) | L_Sequencer_1: Standby act. 1: Input for "End" |
| C01449/... | | ... |
| C01449/5 | | L_Sequencer_1: Standby act. 5: Input for "End" |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01450**

| Parameter \| Name: **C01450 \| L_Sequencer_1: Standby PolaritätEingang** | | | Data type: UNSIGNED_8 Index: $23125_d = 5A55_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Standby" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 1 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01450/1 | 0 | | L_Sequencer_1: Standby act. 1: Input polarity |
| C01450/... | | | ... |
| C01450/5 | | | L_Sequencer_1: Standby act. 5: Input polarity |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01451**

| Parameter \| Name: C01451 \| L_Sequencer_1: Standby Modusfolger | | Data type: UNSIGNED_16 Index: $23124_d = 5A54_h$ |
|---|---|---|
| From version 12.00.00 | | ▸ L_Sequencer_1: "Standby" action |
| **Selection list** | | |
| 0 | Speed follower | |
| 1 | Position follower | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01451/1 | 0: Speed follower | L_Sequencer_1: Standby act. 1: Follower mode |
| C01451/... | | ... |
| C01451/5 | | L_Sequencer_1: Standby act. 5: Follower mode |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01452**

| Parameter \| Name: C01452 \| L_Sequencer_1: StandbySollwert | | | Data type: INTEGER_16 Index: $23123_d = 5A53_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1: "Standby" action |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01452/1 | 0.00 % | | L_Sequencer_1: Standby act. 1: Setpoint |
| C01452/... | | | ... |
| C01452/5 | | | L_Sequencer_1: Standby act. 5: Setpoint |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01453**

| Parameter \| Name: **C01453 \| L_Sequencer_1: Bool signal** | | | Data type: UNSIGNED_8 Index: $23122_d = 5A52_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1 (sequence control) |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 1 | |
| **Subcodes** | | | **Info** |
| C01453/1 | | | L_Sequencer_1: bStart |
| C01453/2 | | | L_Sequencer_1: bPause |
| C01453/3 | | | L_Sequencer_1: bNextStep |
| C01453/4 | | | L_Sequencer_1: bCancel |
| C01453/5 | | | L_Sequencer_1: bBreak |
| C01453/6 | | | L_Sequencer_1: bReset |
| C01453/7 | | | L_Sequencer_1: bProgramBusy |
| C01453/8 | | | L_Sequencer_1: bStateReady |
| C01453/9 | | | L_Sequencer_1: bStateRun |
| C01453/10 | | | L_Sequencer_1: bStatePause |
| C01453/11 | | | L_Sequencer_1: bStateBreak |
| C01453/12 | | | L_Sequencer_1: bStateReset |
| C01453/13 | | | L_Sequencer_1: bStateDone |
| C01453/14 | | | L_Sequencer_1: bWatchDogAcitve |
| C01453/15 | | | L_Sequencer_1: bErrBranch |
| C01453/16 | | | L_Sequencer_1: bErrActive |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01454**

| Parameter \| Name: **C01454 \| L_Sequencer_1: INT signal** | | | Data type: INTEGER_16 Index: $23121_d = 5A51_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1 (sequence control) |
| **Display area** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C01454/1 | | | L_Sequencer_1: nSet_a |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01455**

| Parameter \| Name: **C01455 \| L_Sequencer_1: Word signal** | | | Data type: UNSIGNED_16 Index: $23120_d = 5A50_h$ |
|---|---|---|---|
| From version 12.00.00 | | | ▸ L_Sequencer_1 (sequence control) |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | | | **Info** |
| C01455/1 | | | L_Sequencer_1: wStartStep |
| C01455/2 | | | L_Sequencer_1: wBranch1 |
| C01455/3 | | | L_Sequencer_1: wBranch2 |
| C01455/4 | | | L_Sequencer_1: wDigitalInputs |
| C01455/5 | | | L_Sequencer_1: wMotionState1 |
| C01455/6 | | | L_Sequencer_1: wMotionState2 |
| C01455/7 | | | L_Sequencer_1: wMckPosCtrl1 |
| C01455/8 | | | L_Sequencer_1: wMckPosCtrl2 |
| C01455/9 | | | L_Sequencer_1: wAuxCtrl |
| C01455/10 | | | L_Sequencer_1: wState |
| C01455/11 | | | L_Sequencer_1: wActStep |
| C01455/12 | | | L_Sequencer_1: wDigitalOutputs |
| C01455/13 | | | L_Sequencer_1: wProfileNumber |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01460**

| Parameter \| Name: **C01460 \| L_ConvActPos: TConst** | | | Data type: UNSIGNED_16 Index: $23115_d = 5A4B_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 1000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01460/1 | 10 ms | | L_ConvActPos_1: TConst |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01461**

| Parameter \| Name: **C01461 \| L_ConvActPos: MaxPos/MinPos** | | | Data type: INTEGER_16 Index: $23114_d = 5A4A_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01461/1 | 100.00 % | | L_ConvActPos_1: MaxPos |
| C01461/2 | -100.00 % | | L_ConvActPos_1: MinPos |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01462**

| Parameter \| Name: C01462 \| L_ConvActPos: Length | | | Data type: UNSIGNED_32 Index: $23113_d = 5A49_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | mm | 100000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01462/1 | 0 mm | | L_ConvActPos_1: Length |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01463**

| Parameter \| Name: C01463 \| L_ConvActPos: UpperRef/LowerRef | | | Data type: INTEGER_16 Index: $23112_d = 5A48_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01463/1 | 90.00 % | | L_ConvActPos_1: UpperRef |
| C01463/2 | -90.01 % | | L_ConvActPos_1: LowerRef |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01464**

| Parameter \| Name: C01464 \| L_ConvActPos: WindowSetPos | | | Data type: INTEGER_16 Index: $23111_d = 5A47_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01464/1 | 10.00 % | | L_ConvActPos_1: WindowSetPos |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01465**

| Parameter \| Name: C01465 \| L_ConvActPos: DelayTime | | | Data type: UNSIGNED_16 Index: $23110_d = 5A46_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 50.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01465/1 | 0.100 s | | L_ConvActPos_1: DelayTime |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01466**

| Parameter \| Name: C01466 \| L_ConvActPos: TConstVAdd | | | Data type: UNSIGNED_16 Index: $23109_d = 5A45_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 1000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01466/1 | 10 ms | | L_ConvActPos_1: TConstVAdd |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01467**

| Parameter \| Name: C01467 \| L_ConvActPos: DeadBand | | | Data type: INTEGER_16 Index: $23108_d = 5A44_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01467/1 | 1.00 % | | L_ConvActPos_1: DeadBand |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01468**

| Parameter \| Name: C01468 \| L_ConvActPos: Damping | | | Data type: UNSIGNED_16 Index: $23107_d = 5A43_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | | 10.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01468/1 | 0.00 | | L_ConvActPos_1: Damping |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01469**

| Parameter \| Name: C01469 \| L_MFail: Vp | | | Data type: UNSIGNED_16 Index: $23106_d = 5A42_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.001 | | 31.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01469/1 | 1.000 | | L_MFail_1: Vp |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01470**

| Parameter \| Name: **C01470 \| L_MFail: Tn** | | | Data type: UNSIGNED_16 Index: $23105_d = 5A41_h$ |
|---|---|---|---|
| From version 12.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01470/1 | 20 ms | | L_MFail_1: U DC-bus controller reset time |
| C01470/2 | 20 ms | | L_MFail_1: Acceleration time |
| C01470/3 | 20 ms | | L_MFail_1: Retrigger time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01501**

| Parameter \| Name: **C01501 \| Resp. to communication error with MCI** | | Data type: UNSIGNED_8 Index: $23074_d = 5A22_h$ |
|---|---|---|
| Configuration of monitoring functions for the communication module | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01501/1 | 0: No Reaction | Resp. to MCI connection error<br>• Response to a communication error of the attached communication module. |
| C01501/2 | 0: No Reaction | Resp. to MCI invalid module<br>• Response to an unplugged or incompatible communication module |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01670**

| Parameter \| Name: **C01670 \| L_ComparePhi 1-5: Function** | | Data type: UNSIGNED_8 Index: $22905_d = 5979_h$ |
|---|---|---|
| Selection of comparison operation<br>• If the statement of the selected comparison operation is true, the binary *bOut* output will be set to TRUE. | | |
| **Selection list** | | |
| 1 | In1 = In2 | |
| 2 | In1 > In2 | |
| 3 | In1 < In2 | |
| 4 | \|In1\| = \|In2\| | |
| 5 | \|In1\| > \|In2\| | |
| 6 | \|In1\| < \|In2\| | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01670/1 | 1: In1 = In2 | L_ComparePhi_1: Function |
| C01670/2 | 1: In1 = In2 | L_ComparePhi_2: Function |
| C01670/3 | 1: In1 = In2 | L_ComparePhi_3: Function |
| C01670/4 | 1: In1 = In2 | L_ComparePhi_4: Function |
| C01670/5 | 1: In1 = In2 | L_ComparePhi_5: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01671**

| Parameter \| Name: **C01671 \| L_ComparePhi 1-5: Hysteresis** | | | Data type: INTEGER_32 Index: $22904_d = 5978_h$ |
|---|---|---|---|
| Hysteresis for the comparison function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 1073741824 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01671/1 | 0 incr. | | L_ComparePhi_1: Hysteresis |
| C01671/2 | 0 incr. | | L_ComparePhi_2: Hysteresis |
| C01671/3 | 0 incr. | | L_ComparePhi_3: Hysteresis |
| C01671/4 | 0 incr. | | L_ComparePhi_4: Hysteresis |
| C01671/5 | 0 incr. | | L_ComparePhi_5: Hysteresis |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01672**

| Parameter \| Name: C01672 \| L_ComparePhi 1-5: Window | | | Data type: INTEGER_32 Index: $22903_d = 5977_h$ |
|---|---|---|---|
| Window for the comparison operation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 1073741824 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01672/1 | 0 incr. | | L_ComparePhi_1: Window |
| C01672/2 | 0 incr. | | L_ComparePhi_2: Window |
| C01672/3 | 0 incr. | | L_ComparePhi_3: Window |
| C01672/4 | 0 incr. | | L_ComparePhi_4: Window |
| C01672/5 | 0 incr. | | L_ComparePhi_5: Window |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01751**

| Parameter \| Name: C01751 \| Service code inverter characteristic | Data type: UNSIGNED_8 Index: $22824_d = 5928_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01752**

| Parameter \| Name: C01752 \| Service par. inverter charact. function | Data type: UNSIGNED_8 Index: $22823_d = 5927_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01755**

| Parameter \| Name: C01755 \| Service par. inverter charact. factor | Data type: INTEGER_16 Index: $22820_d = 5924_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01763**

| Parameter \| Name: C01763 \| Service code -clamp threshold | Data type: INTEGER_16 Index: $22812_d = 591C_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01764**

| Parameter \| Name: C01764 \| Service par. clamp time | Data type: UNSIGNED_8 Index: $22811_d = 591B_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01765**

| Parameter \| Name: C01765 \| Service code - difference threshold UG | Data type: UNSIGNED_16 Index: $22810_d = 591A_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01770**

| Parameter \| Name: **C01770 \| Filter time - earth-fault detect. is running** | | | Data type: UNSIGNED_8 Index: $22805_d = 5915_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | ms | 250 | **2 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C01771**

| Parameter \| Name: **C01771 \| BU-Osc: Cycle time selection** | Data type: UNSIGNED_8 Index: $22804_d = 5914_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01772**

| Parameter \| Name: **C01772 \| BU-Osc: Trigger channel selection** | Data type: UNSIGNED_8 Index: $22803_d = 5913_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01773**

| Parameter \| Name: **C01773 \| BU-Osc: Trigger command** | Data type: UNSIGNED_8 Index: $22802_d = 5912_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01774**

| Parameter \| Name: **C01774 \| BU-Osc: Trigger mode** | Data type: UNSIGNED_8 Index: $22801_d = 5911_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01775**

| Parameter \| Name: **C01775 \| BU-Osc: Trigger** | Data type: INTEGER_16 Index: $22800_d = 5910_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01902**

| Parameter \| Name: **C01902 \| Diagnostics X6: Max. baud rate** | Data type: UNSIGNED_16 Index: $22673_d = 5891_h$ |
|---|---|
| Maximally permissible baud rate in the standard device after determination of the baud rate at the diagnostic interface X6 | |
| **Selection list** (Lenze setting printed in bold) | |
| 192 | 19.200 Bd |
| 384 | 38.400 Bd |
| 576 | 57.600 Bd |
| 750 | 75.000 Bd |
| 1152 | 115.200 Bd |
| 1500 | 150.000 Bd |
| 2500 | 250.000 Bd |
| 3750 | 375.000 Bd |
| **7500** | **750.000 Bd** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | |

**C01903**

| Parameter \| Name: **C01903 \| Diagnostics X6: Change baud rate** | Data type: UNSIGNED_8 Index: $22672_d = 5890_h$ |
|---|---|
| New baud rate determination at the diagnostic interface X6 | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **Ignore changes** |
| 1 | Negotiate baud rate |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | |

**C01905**

| Parameter \| Name: **C01905 \| Diagnostics X6: Current baud rate** | | | Data type: UNSIGNED_32 Index: $22670_d = 588E_h$ |
|---|---|---|---|
| Current baud rate at the diagnostic interface X6 | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | Bd | 3000000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C02200**

| Parameter \| Name: **C02200 \| LS_WriteParamList: Function** | | Data type: UNSIGNED_8 Index: $22375_d = 5767_h$ |
|---|---|---|
| From version 12.00.00<br>Configuration of which parameter lists are to be subject to the "Parameter change-over" function. | | |
| **Selection list** | | |
| 0 | Off | |
| 1 | On | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02200/1 | 1: On | Definable parameter list |
| C02200/2 | 0: Off | Motor data parameter list |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02210**

| Parameter \| Name: **C02210 \| LS_WriteParamList: Motor control** | | Data type: UNSIGNED_8 Index: $22365_d = 575D_h$ |
|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | |
| **Selection list** | | |
| 1 | SC: Servo control PSM | |
| 2 | SC: Servo control ASM | |
| 3 | SLPSM: Sensorless PSM | |
| 4 | SLVC: Vector control | |
| 6 | VFCplus: V/f linear | |
| 7 | VFCplus: V/f linear + encoder | |
| 8 | VFCplus: V/f quadr | |
| 9 | VFCplus: V/f quadr + encoder | |
| 10 | VFCplus: V/f definable | |
| 11 | VFCplusEco: V/f energy-saving | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02210/1 | 6: VFCplus: V/f linear | LS_WriteParamList: c6_val1 |
| C02210/2 | 6: VFCplus: V/f linear | LS_WriteParamList: c6_val2 |
| C02210/3 | 6: VFCplus: V/f linear | LS_WriteParamList: c6_val3 |
| C02210/4 | 6: VFCplus: V/f linear | LS_WriteParamList: c6_val3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02212**

| Parameter \| Name: **C02212 \| LS_WriteParamList: VFC: V/f base frequency** | | | Data type: UNSIGNED_16 Index: $22363_d = 575B_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 7.5 | Hz | 2600.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02212/1 | 50.0 Hz | | LS_WriteParamList: c15_val1 |
| C02212/2 | 50.0 Hz | | LS_WriteParamList: c15_val2 |
| C02212/3 | 50.0 Hz | | LS_WriteParamList: c15_val3 |
| C02212/4 | 50.0 Hz | | LS_WriteParamList: c15_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02213**

| Parameter \| Name: **C02213 \| LS_WriteParamList: VFC: Vmin boost** | | | Data type: UNSIGNED_16 Index: $22362_d = 575A_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02213/1 | 1.60 % | | LS_WriteParamList: c16_val1 |
| C02213/2 | 1.60 % | | LS_WriteParamList: c16_val2 |
| C02213/3 | 1.60 % | | LS_WriteParamList: c16_val3 |
| C02213/4 | 1.60 % | | LS_WriteParamList: c16_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02214**

| Parameter \| Name: **C02214 \| LS_WriteParamList: Switching frequency** | | Data type: UNSIGNED_8 Index: $22361_d = 5759_h$ |
|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | |
| **Selection list** | | |
| 1 | 4 kHz var./drive-optimised | |
| 2 | 8 kHz var./drive-optimised | |
| 3 | 16 kHz var./drive-optimised | |
| 5 | 2 kHz constant/drive-optimised | |
| 6 | 4 kHz constant/drive-optimised | |
| 7 | 8 kHz constant/drive-optimised | |
| 8 | 16 kHz constant/drive-optimised | |
| 11 | 4 kHz var./min. Pv | |
| 12 | 8 kHz var./min. Pv | |
| 13 | 16 kHz var./min. Pv | |
| 15 | 2 kHz constant/min. Pv | |
| 16 | 4 kHz constant/min. Pv | |
| 17 | 8 kHz constant/min. Pv | |
| 18 | 16 kHz constant/min. Pv | |
| 21 | 8 kHz var./drive-opt./4 kHz min | |
| 22 | 16 kHz var./drive-opt./4 kHz min | |
| 23 | 16 kHz var./drive-opt./8 kHz min | |
| 31 | 8 kHz var./min. Pv/4 kHz min | |
| 32 | 16 kHz var./min. Pv/4 kHz min | |
| 33 | 16 kHz var./min. Pv/8 kHz min | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02214/1 | 2: 8 kHz var./drive-opt. | LS_WriteParamList: c18_val1 |
| C02214/2 | 2: 8 kHz var./drive-opt. | LS_WriteParamList: c18_val2 |
| C02214/3 | 2: 8 kHz var./drive-opt. | LS_WriteParamList: c18_val3 |
| C02214/4 | 2: 8 kHz var./drive-opt. | LS_WriteParamList: c18_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02215**

| Parameter \| Name: **C02215 \| LS_WriteParamList: Auto-DCB: Threshold** | | | Data type: UNSIGNED_16 Index: $22360_d = 5758_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | rpm | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02215/1 | 3 rpm | | LS_WriteParamList: c19_val1 |
| C02215/2 | 3 rpm | | LS_WriteParamList: c19_val2 |
| C02215/3 | 3 rpm | | LS_WriteParamList: c19_val3 |
| C02215/4 | 3 rpm | | LS_WriteParamList: c19_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02216**

| Parameter \| Name: **C02216 \| LS_WriteParamList: Slip compensation** | | | Data type: INTEGER_16 Index: $22359_d = 5757_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -100.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02216/1 | 2.67 % | | LS_WriteParamList: c21_val1 |
| C02216/2 | 2.67 % | | LS_WriteParamList: c21_val2 |
| C02216/3 | 2.67 % | | LS_WriteParamList: c21_val3 |
| C02216/4 | 2.67 % | | LS_WriteParamList: c21_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02217**

| Parameter \| Name: **C02217 \| LS_WriteParamList: Imax in motor mode** | | | Data type: UNSIGNED_16 Index: $22358_d = 5756_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | A | 655.35 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02217/1 | 47.00 A | | LS_WriteParamList: c22_val1 |
| C02217/2 | 47.00 A | | LS_WriteParamList: c22_val2 |
| C02217/3 | 47.00 A | | LS_WriteParamList: c22_val3 |
| C02217/4 | 47.00 A | | LS_WriteParamList: c22_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02218**

| Parameter \| Name: **C02218 \| LS_WriteParamList: Imax in generator mode** | | | Data type: INTEGER_16 Index: $22357_d = 5755_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02218/1 | 100.00 % | | LS_WriteParamList: c23_val1 |
| C02218/2 | 100.00 % | | LS_WriteParamList: c23_val2 |
| C02218/3 | 100.00 % | | LS_WriteParamList: c23_val3 |
| C02218/4 | 100.00 % | | LS_WriteParamList: c23_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02219**

| Parameter \| Name: **C02219 \| LS_WriteParamList: DC braking: Current** | | | Data type: INTEGER_16 Index: $22356_d = 5754_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 200.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02219/1 | 50.00 % | | LS_WriteParamList: c36_val1 |
| C02219/2 | 50.00 % | | LS_WriteParamList: c36_val2 |
| C02219/3 | 50.00 % | | LS_WriteParamList: c36_val3 |
| C02219/4 | 50.00 % | | LS_WriteParamList: c36_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02220**

| Parameter \| Name: **C02220 \| LS_WriteParamList: Vp speed controller** | | | Data type: UNSIGNED_16 Index: $22355_d = 5753_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | | 600.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02220/1 | 15.00 | | LS_WriteParamList: c70_s1_val1 |
| C02220/2 | 15.00 | | LS_WriteParamList: c70_s1_val2 |
| C02220/3 | 15.00 | | LS_WriteParamList: c70_s1_val3 |
| C02220/4 | 15.00 | | LS_WriteParamList: c70_s1_val4 |
| C02220/5 | 6.00 | | LS_WriteParamList: c70_s2_val1 |
| C02220/6 | 6.00 | | LS_WriteParamList: c70_s2_val2 |
| C02220/7 | 6.00 | | LS_WriteParamList: c70_s2_val3 |
| C02220/8 | 6.00 | | LS_WriteParamList: c70_s2_val4 |
| C02220/9 | 3.00 | | LS_WriteParamList: c70_s3_val1 |
| C02220/10 | 3.00 | | LS_WriteParamList: c70_s3_val2 |
| C02220/11 | 3.00 | | LS_WriteParamList: c70_s3_val3 |
| C02220/12 | 3.00 | | LS_WriteParamList: c70_s3_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02221**

| Parameter \| Name: C02221 \| LS_WriteParamList: Ti speed controller | | | Data type: UNSIGNED_16 Index: $22354_d = 5752_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | ms | 6000.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02221/1 | 100.0 ms | | LS_WriteParamList: c71_s1_val1 |
| C02221/2 | 100.0 ms | | LS_WriteParamList: c71_s1_val2 |
| C02221/3 | 100.0 ms | | LS_WriteParamList: c71_s1_val3 |
| C02221/4 | 100.0 ms | | LS_WriteParamList: c71_s1_val4 |
| C02221/5 | 50.0 ms | | LS_WriteParamList: c71_s2_val1 |
| C02221/6 | 50.0 ms | | LS_WriteParamList: c71_s2_val2 |
| C02221/7 | 50.0 ms | | LS_WriteParamList: c71_s2_val3 |
| C02221/8 | 50.0 ms | | LS_WriteParamList: c71_s2_val4 |
| C02221/9 | 50.0 ms | | LS_WriteParamList: c71_s3_val1 |
| C02221/10 | 50.0 ms | | LS_WriteParamList: c71_s3_val2 |
| C02221/11 | 50.0 ms | | LS_WriteParamList: c71_s3_val3 |
| C02221/12 | 50.0 ms | | LS_WriteParamList: c71_s3_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02222**

| Parameter \| Name: C02222 \| LS_WriteParamList: SC: Tdn speed controller | | | Data type: UNSIGNED_16 Index: $22353_d = 5751_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | ms | 3.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02222/1 | 0.00 ms | | LS_WriteParamList: c72_val1 |
| C02222/2 | 0.00 ms | | LS_WriteParamList: c72_val2 |
| C02222/3 | 0.00 ms | | LS_WriteParamList: c72_val3 |
| C02222/4 | 0.00 ms | | LS_WriteParamList: c72_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02223**

| Parameter \| Name: C02223 \| LS_WriteParamList: Imax/M controller gain | | | Data type: UNSIGNED_16 Index: $22352_d = 5750_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02223/1 | 0.25 | | LS_WriteParamList: c73_s1_val1 |
| C02223/2 | 0.25 | | LS_WriteParamList: c73_s1_val2 |
| C02223/3 | 0.25 | | LS_WriteParamList: c73_s1_val3 |
| C02223/4 | 0.25 | | LS_WriteParamList: c73_s1_val4 |
| C02223/5 | 1.25 | | LS_WriteParamList: c73_s2_val1 |
| C02223/6 | 1.25 | | LS_WriteParamList: c73_s2_val2 |
| C02223/7 | 1.25 | | LS_WriteParamList: c73_s2_val3 |
| C02223/8 | 1.25 | | LS_WriteParamList: c73_s2_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02224**

| Parameter \| Name: C02224 \| LS_WriteParamList: Imax/M controller reset time | | | Data type: UNSIGNED_16 Index: $22351_d = 574F_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 9990 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02224/1 | 65 ms | | LS_WriteParamList: c74_s1_val1 |
| C02224/2 | 65 ms | | LS_WriteParamList: c74_s1_val2 |
| C02224/3 | 65 ms | | LS_WriteParamList: c74_s1_val3 |
| C02224/4 | 65 ms | | LS_WriteParamList: c74_s1_val4 |
| C02224/5 | 30 ms | | LS_WriteParamList: c74_s2_val1 |
| C02224/6 | 30 ms | | LS_WriteParamList: c74_s2_val2 |
| C02224/7 | 30 ms | | LS_WriteParamList: c74_s2_val3 |
| C02224/8 | 30 ms | | LS_WriteParamList: c74_s2_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02225**

| Parameter \| Name: C02225 \| LS_WriteParamList: Vp current controller | | | Data type: UNSIGNED_16 Index: $22350_d = 574E_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | V/A | 500.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02225/1 | 7.00 V/A | | LS_WriteParamList: c75_val1 |
| C02225/2 | 7.00 V/A | | LS_WriteParamList: c75_val2 |
| C02225/3 | 7.00 V/A | | LS_WriteParamList: c75_val3 |
| C02225/4 | 7.00 V/A | | LS_WriteParamList: c75_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02226**

| Parameter \| Name: **C02226 \| LS_WriteParamList: Ti current controller** | | | Data type: UNSIGNED_16 Index: $22349_d = 574D_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | ms | 500.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02226/1 | 10.61 ms | | LS_WriteParamList: c76_val1 |
| C02226/2 | 10.61 ms | | LS_WriteParamList: c76_val2 |
| C02226/3 | 10.61 ms | | LS_WriteParamList: c76_val3 |
| C02226/4 | 10.61 ms | | LS_WriteParamList: c76_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02227**

| Parameter \| Name: **C02227 \| LS_WriteParamList: SC: Vp field controller** | | | Data type: UNSIGNED_16 Index: $22348_d = 574C_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | | 500.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02227/1 | 12.80 | | LS_WriteParamList: c77_val1 |
| C02227/2 | 12.80 | | LS_WriteParamList: c77_val2 |
| C02227/3 | 12.80 | | LS_WriteParamList: c77_val3 |
| C02227/4 | 12.80 | | LS_WriteParamList: c77_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02228**

| Parameter \| Name: **C02228 \| LS_WriteParamList: SC: Tn field controller** | | | Data type: UNSIGNED_16 Index: $22347_d = 574B_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | ms | 6000.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02228/1 | 256.0 ms | | LS_WriteParamList: c78_val1 |
| C02228/2 | 256.0 ms | | LS_WriteParamList: c78_val2 |
| C02228/3 | 256.0 ms | | LS_WriteParamList: c78_val3 |
| C02228/4 | 256.0 ms | | LS_WriteParamList: c78_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02229**

| Parameter \| Name: **C02229 \| LS_WriteParamList: SC: Settings** | | Data type: UNSIGNED_8 Index: $22346_d = 574A_h$ |
|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | |
| **Selection list** | | |
| 0 | Off | |
| 1 | On | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02229/1 | 0: Off | LS_WriteParamList: c79_s1_val1 |
| C02229/2 | 0: Off | LS_WriteParamList: c79_s1_val2 |
| C02229/3 | 0: Off | LS_WriteParamList: c79_s1_val3 |
| C02229/4 | 0: Off | LS_WriteParamList: c79_s1_val4 |
| C02229/5 | 1: On | LS_WriteParamList: c79_s2_val1 |
| C02229/6 | 1: On | LS_WriteParamList: c79_s2_val2 |
| C02229/7 | 1: On | LS_WriteParamList: c79_s2_val3 |
| C02229/8 | 1: On | LS_WriteParamList: c79_s2_val4 |
| C02229/9 | 0: Off | LS_WriteParamList: c79_s3_val1 |
| C02229/10 | 0: Off | LS_WriteParamList: c79_s3_val2 |
| C02229/11 | 0: Off | LS_WriteParamList: c79_s3_val3 |
| C02229/12 | 0: Off | LS_WriteParamList: c79_s3_val4 |
| C02229/13 | 1: On | LS_WriteParamList: c79_s4_val1 |
| C02229/14 | 1: On | LS_WriteParamList: c79_s4_val2 |
| C02229/15 | 1: On | LS_WriteParamList: c79_s4_val3 |
| C02229/16 | 1: On | LS_WriteParamList: c79_s4_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02230**

| Parameter \| Name: **C02230 \| LS_WriteParamList: Override point of field weakening** | | | Data type: INTEGER_16 Index: $22345_d = 5749_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -500 | Hz | 500 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02230/1 | 0 Hz | | LS_WriteParamList: c80_val1 |
| C02230/2 | 0 Hz | | LS_WriteParamList: c80_val2 |
| C02230/3 | 0 Hz | | LS_WriteParamList: c80_val3 |
| C02230/4 | 0 Hz | | LS_WriteParamList: c80_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02231**

| Parameter \| Name: **C02231 \| LS_WriteParamList: Rated motor power** | | | Data type: UNSIGNED_16 Index: $22344_d = 5748_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | kW | 500.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02231/1 | 11.00 kW | | LS_WriteParamList: c81_val1 |
| C02231/2 | 11.00 kW | | LS_WriteParamList: c81_val2 |
| C02231/3 | 11.00 kW | | LS_WriteParamList: c81_val3 |
| C02231/4 | 11.00 kW | | LS_WriteParamList: c81_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02232**

| Parameter \| Name: **C02232 \| LS_WriteParamList: Motor rotor resistance** | | | Data type: UNSIGNED_32 Index: $22343_d = 5747_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | mOhm | 200000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02232/1 | 276 mOhm | | LS_WriteParamList: c82_val1 |
| C02232/2 | 276 mOhm | | LS_WriteParamList: c82_val2 |
| C02232/3 | 276 mOhm | | LS_WriteParamList: c82_val3 |
| C02232/4 | 276 mOhm | | LS_WriteParamList: c82_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02233**

| Parameter \| Name: **C02233 \| LS_WriteParamList: Motor stator resistance** | | | Data type: UNSIGNED_32 Index: $22342_d = 5746_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | mOhm | 200000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02233/1 | 330 mOhm | | LS_WriteParamList: c84_val1 |
| C02233/2 | 330 mOhm | | LS_WriteParamList: c84_val2 |
| C02233/3 | 330 mOhm | | LS_WriteParamList: c84_val3 |
| C02233/4 | 330 mOhm | | LS_WriteParamList: c84_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02234**

| Parameter \| Name: **C02234 \| LS_WriteParamList: Motor stator leakage inductance** | | | Data type: UNSIGNED_16 Index: $22341_d = 5745_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | mH | 650.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02234/1 | 3.50 mH | | LS_WriteParamList: c85_val1 |
| C02234/2 | 3.50 mH | | LS_WriteParamList: c85_val2 |
| C02234/3 | 3.50 mH | | LS_WriteParamList: c85_val3 |
| C02234/4 | 3.50 mH | | LS_WriteParamList: c85_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02236**

| Parameter \| Name: **C02236 \| LS_WriteParamList: Rated motor speed** | | | Data type: UNSIGNED_16 Index: $22339_d = 5743_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 50 | rpm | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02236/1 | 1460 rpm | | LS_WriteParamList: c87_val1 |
| C02236/2 | 1460 rpm | | LS_WriteParamList: c87_val2 |
| C02236/3 | 1460 rpm | | LS_WriteParamList: c87_val3 |
| C02236/4 | 1460 rpm | | LS_WriteParamList: c87_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02237**

| Parameter \| Name: **C02237 \| LS_WriteParamList: Rated motor current** | | | Data type: UNSIGNED_16 Index: $22338_d = 5742_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.20 | A | 320.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02237/1 | 21.00 A | | LS_WriteParamList: c88_val1 |
| C02237/2 | 21.00 A | | LS_WriteParamList: c88_val2 |
| C02237/3 | 21.00 A | | LS_WriteParamList: c88_val3 |
| C02237/4 | 21.00 A | | LS_WriteParamList: c88_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02238**

| Parameter \| Name: C02238 \| LS_WriteParamList: Rated motor frequency | | | Data type: UNSIGNED_16 Index: $22337_d = 5741_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | Hz | 1000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02238/1 | 50 Hz | | LS_WriteParamList: c89_val1 |
| C02238/2 | 50 Hz | | LS_WriteParamList: c89_val2 |
| C02238/3 | 50 Hz | | LS_WriteParamList: c89_val3 |
| C02238/4 | 50 Hz | | LS_WriteParamList: c89_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02239**

| Parameter \| Name: C02239 \| LS_WriteParamList: Rated motor voltage | | | Data type: UNSIGNED_16 Index: $22336_d = 5740_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | V | 5000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02239/1 | 400 V | | LS_WriteParamList: c90_val1 |
| C02239/2 | 400 V | | LS_WriteParamList: c90_val2 |
| C02239/3 | 400 V | | LS_WriteParamList: c90_val3 |
| C02239/4 | 400 V | | LS_WriteParamList: c90_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02240**

| Parameter \| Name: C02240 \| LS_WriteParamList: Motor cosine phi | | | Data type: UNSIGNED_8 Index: $22335_d = 573F_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.20 | | 1.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02240/1 | 0.85 | | LS_WriteParamList: c91_val1 |
| C02240/2 | 0.85 | | LS_WriteParamList: c91_val2 |
| C02240/3 | 0.85 | | LS_WriteParamList: c91_val3 |
| C02240/4 | 0.85 | | LS_WriteParamList: c91_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02241**

| Parameter \| Name: **C02241 \| LS_WriteParamList: Motor magnetizing inductance** | | | Data type: UNSIGNED_16 Index: $22334_d = 573E_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | mH | 6500.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02241/1 | 81.0 mH | | LS_WriteParamList: c92_val1 |
| C02241/2 | 81.0 mH | | LS_WriteParamList: c92_val2 |
| C02241/3 | 81.0 mH | | LS_WriteParamList: c92_val3 |
| C02241/4 | 81.0 mH | | LS_WriteParamList: c92_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02242**

| Parameter \| Name: **C02242 \| LS_WriteParamList: Motor magnetising current** | | | Data type: UNSIGNED_16 Index: $22333_d = 573D_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | A | 320.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02242/1 | 8.50 A | | LS_WriteParamList: c95_val1 |
| C02242/2 | 8.50 A | | LS_WriteParamList: c95_val2 |
| C02242/3 | 8.50 A | | LS_WriteParamList: c95_val3 |
| C02242/4 | 8.50 A | | LS_WriteParamList: c95_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02244**

| Parameter \| Name: **C02244 \| LS_WriteParamList: Auto-DCB: Hold time** | | | Data type: UNSIGNED_32 Index: $22331_d = 573B_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02244/1 | 0.500 s | | LS_WriteParamList: c106_val1 |
| C02244/2 | 0.500 s | | LS_WriteParamList: c106_val2 |
| C02244/3 | 0.500 s | | LS_WriteParamList: c106_val3 |
| C02244/4 | 0.500 s | | LS_WriteParamList: c106_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C02245**

| Parameter \| Name: C02245 \| LS_WriteParamList: DC braking: Hold time | | | Data type: UNSIGNED_32 Index: $22330_d = 573A_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02245/1 | 998.999 s | | LS_WriteParamList: c107_val1 |
| C02245/2 | 998.999 s | | LS_WriteParamList: c107_val2 |
| C02245/3 | 998.999 s | | LS_WriteParamList: c107_val3 |
| C02245/4 | 998.999 s | | LS_WriteParamList: c107_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C02246**

| Parameter \| Name: C02246 \| LS_WriteParamList: Setting of motor overload (I²xt) | | | Data type: INTEGER_16 Index: $22329_d = 5739_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 250.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02246/1 | 100.00 % | | LS_WriteParamList: c120_val1 |
| C02246/2 | 100.00 % | | LS_WriteParamList: c120_val2 |
| C02246/3 | 100.00 % | | LS_WriteParamList: c120_val3 |
| C02246/4 | 100.00 % | | LS_WriteParamList: c120_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02249**

| Parameter \| Name: C02249 \| LS_WriteParamList: Oscillation damping influence | | | Data type: UNSIGNED_16 Index: $22326_d = 5736_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 250.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02249/1 | 5.00 % | | LS_WriteParamList: c234_val1 |
| C02249/2 | 5.00 % | | LS_WriteParamList: c234_val2 |
| C02249/3 | 5.00 % | | LS_WriteParamList: c234_val3 |
| C02249/4 | 5.00 % | | LS_WriteParamList: c234_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02250**

| Parameter \| Name: **C02250 \| LS_WriteParamList: Filter time - oscill. damping** | | | Data type: UNSIGNED_8 Index: $22325_d = 5735_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 2 | ms | 250 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02250/1 | 32 ms | | LS_WriteParamList: c235_val1 |
| C02250/2 | 32 ms | | LS_WriteParamList: c235_val2 |
| C02250/3 | 32 ms | | LS_WriteParamList: c235_val3 |
| C02250/4 | 32 ms | | LS_WriteParamList: c235_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02251**

| Parameter \| Name: **C02251 \| LS_WriteParamList: Oscillation damping field weakening** | | | Data type: UNSIGNED_8 Index: $22324_d = 5734_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 40 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02251/1 | 14 | | LS_WriteParamList: c236_val1 |
| C02251/2 | 14 | | LS_WriteParamList: c236_val2 |
| C02251/3 | 14 | | LS_WriteParamList: c236_val3 |
| C02251/4 | 14 | | LS_WriteParamList: c236_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02252**

| Parameter \| Name: **C02252 \| LS_WriteParamList: Kp position controller** | | | Data type: UNSIGNED_16 Index: $22323_d = 5733_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | 1/s | 500.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02252/1 | 5.00 1/s | | LS_WriteParamList: c254_val1 |
| C02252/2 | 5.00 1/s | | LS_WriteParamList: c254_val2 |
| C02252/3 | 5.00 1/s | | LS_WriteParamList: c254_val3 |
| C02252/4 | 5.00 1/s | | LS_WriteParamList: c254_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02256**

| Parameter \| Name: **C02256 \| LS_WriteParamList: Moment of inertia** | | | Data type: UNSIGNED_32 Index: $22319_d = 572F_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | kg cm^2 | 6000000.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02256/1 | 0.00 kg cm^2 | | LS_WriteParamList: c273_val1 |
| C02256/2 | 0.00 kg cm^2 | | LS_WriteParamList: c273_val2 |
| C02256/3 | 0.00 kg cm^2 | | LS_WriteParamList: c273_val3 |
| C02256/4 | 0.00 kg cm^2 | | LS_WriteParamList: c273_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02260**

| Parameter \| Name: **C02260 \| LS_WriteParamList: Speed sensor selection** | | Data type: UNSIGNED_8 Index: $22315_d = 572B_h$ |
|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | |
| **Selection list** | | |
| 0 | No sensor | |
| 1 | Sensor signal FreqIn12 | |
| 2 | Encoder signal FreqIn67 | |
| 3 | Multi encoder | |
| 4 | Resolver | |
| 5 | Encoder signal FreqIn1267 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02260/1 | 0: No sensor | LS_WriteParamList: c495_val1 |
| C02260/2 | 0: No sensor | LS_WriteParamList: c495_val2 |
| C02260/3 | 0: No sensor | LS_WriteParamList: c495_val3 |
| C02260/4 | 0: No sensor | LS_WriteParamList: c495_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02261**

| Parameter \| Name: **C02261 \| LS_WriteParamList: SC: Field feedforward control** | | | Data type: UNSIGNED_16 Index: $22314_d = 572A_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | % | 600 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02261/1 | 200 % | | LS_WriteParamList: c576_val1 |
| C02261/2 | 200 % | | LS_WriteParamList: c576_val2 |
| C02261/3 | 200 % | | LS_WriteParamList: c576_val3 |
| C02261/4 | 200 % | | LS_WriteParamList: c576_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02262**

| Parameter \| Name: C02262 \| LS_WriteParamList: SC: Vp field weakening controller | | | Data type: UNSIGNED_16 Index: $22313_d = 5729_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0000 | | 2.0000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02262/1 | 0.0010 | | LS_WriteParamList: c577_val1 |
| C02262/2 | 0.0010 | | LS_WriteParamList: c577_val2 |
| C02262/3 | 0.0010 | | LS_WriteParamList: c577_val3 |
| C02262/4 | 0.0010 | | LS_WriteParamList: c577_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C02263**

| Parameter \| Name: C02263 \| LS_WriteParamList: SC: Tn field weakening controller | | | Data type: UNSIGNED_16 Index: $22312_d = 5728_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.1 | ms | 6200.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02263/1 | 20.0 ms | | LS_WriteParamList: c578_val1 |
| C02263/2 | 20.0 ms | | LS_WriteParamList: c578_val2 |
| C02263/3 | 20.0 ms | | LS_WriteParamList: c578_val3 |
| C02263/4 | 20.0 ms | | LS_WriteParamList: c578_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C02264**

| Parameter \| Name: C02264 \| LS_WriteParamList: Empfindlichkeit Sollwertvorsteuerung | | | Data type: UNSIGNED_8 Index: $22311_d = 5727_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Selection list** | | | |
| 0 | Inactive | | |
| 1 | 15 bits | | |
| 2 | 14 bits | | |
| 3 | 13 bits | | |
| 4 | 12 bits | | |
| 5 | 11 Bit | | |
| 6 | 10 Bit | | |
| 7 | 9 Bit | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02264/1 | 0: Inactive | | LS_WriteParamList: C653_s1_val1 |
| C02264/2 | 0: Inactive | | LS_WriteParamList: C653_s1_val2 |
| C02264/3 | 0: Inactive | | LS_WriteParamList: C653_s1_val3 |
| C02264/4 | 0: Inactive | | LS_WriteParamList: C653_s1_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02272**

| Parameter \| Name: **C02272 \| LS_WriteParamList: Motor phase direction of rotation** | | Data type: UNSIGNED_8 Index: $22303_d = 571F_h$ |
|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | |
| **Selection list** | | |
| 0 | not inverted | |
| 1 | inverted | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02272/1 | 0: Not inverted | LS_WriteParamList: c905_val1 |
| C02272/2 | 0: Not inverted | LS_WriteParamList: c905_val2 |
| C02272/3 | 0: Not inverted | LS_WriteParamList: c905_val3 |
| C02272/4 | 0: Not inverted | LS_WriteParamList: c905_val4 |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02273**

| Parameter \| Name: **C02273 \| LS_WriteParamList: Speed limitation** | | | Data type: INTEGER_16 Index: $22302_d = 571E_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 175.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02273/1 | 120.00 % | | LS_WriteParamList: c909_s1_val1 |
| C02273/2 | 120.00 % | | LS_WriteParamList: c909_s1_val2 |
| C02273/3 | 120.00 % | | LS_WriteParamList: c909_s1_val3 |
| C02273/4 | 120.00 % | | LS_WriteParamList: c909_s1_val4 |
| C02273/5 | 120.00 % | | LS_WriteParamList: c909_s2_val1 |
| C02273/6 | 120.00 % | | LS_WriteParamList: c909_s2_val2 |
| C02273/7 | 120.00 % | | LS_WriteParamList: c909_s2_val3 |
| C02273/8 | 120.00 % | | LS_WriteParamList: c909_s2_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02274**

| Parameter \| Name: **C02274 \| LS_WriteParamList: Frequency limitation** | | | Data type: UNSIGNED_16 Index: $22301_d = 571D_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Hz | 1300 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02274/1 | 1000 Hz | | LS_WriteParamList: c910_s1_val1 |
| C02274/2 | 1000 Hz | | LS_WriteParamList: c910_s1_val2 |
| C02274/3 | 1000 Hz | | LS_WriteParamList: c910_s1_val3 |
| C02274/4 | 1000 Hz | | LS_WriteParamList: c910_s1_val4 |
| C02274/5 | 1000 Hz | | LS_WriteParamList: c910_s2_val1 |
| C02274/6 | 1000 Hz | | LS_WriteParamList: c910_s2_val2 |
| C02274/7 | 1000 Hz | | LS_WriteParamList: c910_s2_val3 |
| C02274/8 | 1000 Hz | | LS_WriteParamList: c910_s2_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02275**

| Parameter \| Name: **C02275 \| LS_WriteParamList: Motor cable length** | | | Data type: UNSIGNED_16 Index: $22300_d = 571C_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | m | 1000.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02275/1 | 5.0 m | | LS_WriteParamList: c915_val1 |
| C02275/2 | 5.0 m | | LS_WriteParamList: c915_val2 |
| C02275/3 | 5.0 m | | LS_WriteParamList: c915_val3 |
| C02275/4 | 5.0 m | | LS_WriteParamList: c915_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02276**

| Parameter \| Name: **C02276 \| LS_WriteParamList: Motor cable cross-section** | | | Data type: UNSIGNED_16 Index: $22299_d = 571B_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.50 | mm^2 | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02276/1 | 6.00 mm^2 | | LS_WriteParamList: c916_val1 |
| C02276/2 | 6.00 mm^2 | | LS_WriteParamList: c916_val2 |
| C02276/3 | 6.00 mm^2 | | LS_WriteParamList: c916_val3 |
| C02276/4 | 6.00 mm^2 | | LS_WriteParamList: c916_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02278**

| Parameter \| Name: C02278 \| LS_WriteParamList: PSM: Maximum motor current field weakening | | | Data type: UNSIGNED_16 Index: $22297_d = 5719_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 500.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02278/1 | 30.00 % | | LS_WriteParamList: c938_val1 |
| C02278/2 | 30.00 % | | LS_WriteParamList: c938_val2 |
| C02278/3 | 30.00 % | | LS_WriteParamList: c938_val3 |
| C02278/4 | 30.00 % | | LS_WriteParamList: c938_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02279**

| Parameter \| Name: C02279 \| LS_WriteParamList: Ultimate motor current | | | Data type: UNSIGNED_16 Index: $22296_d = 5718_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | A | 3000.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02279/1 | 3000.0 A | | LS_WriteParamList: c939_val1 |
| C02279/2 | 3000.0 A | | LS_WriteParamList: c939_val2 |
| C02279/3 | 3000.0 A | | LS_WriteParamList: c939_val3 |
| C02279/4 | 3000.0 A | | LS_WriteParamList: c939_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02280**

| Parameter \| Name: C02280 \| LS_WriteParamList: Max. motor speed | | | Data type: UNSIGNED_16 Index: $22295_d = 5717_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 50 | rpm | 65000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02280/1 | 60000 rpm | | LS_WriteParamList: c965_val1 |
| C02280/2 | 60000 rpm | | LS_WriteParamList: c965_val2 |
| C02280/3 | 60000 rpm | | LS_WriteParamList: c965_val3 |
| C02280/4 | 60000 rpm | | LS_WriteParamList: c965_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02281**

| Parameter \| Name: **C02281 \| LS_WriteParamList: VFC: Time const. slip comp.** | | | Data type: UNSIGNED_16 Index: $22294_d = 5716_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | ms | 6000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02281/1 | 100 ms | | LS_WriteParamList: c966_val1 |
| C02281/2 | 100 ms | | LS_WriteParamList: c966_val2 |
| C02281/3 | 100 ms | | LS_WriteParamList: c966_val3 |
| C02281/4 | 100 ms | | LS_WriteParamList: c966_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02284**

| Parameter \| Name: **C02284 \| LS_WriteParamList: VFC: limitation V/f +encoder** | | | Data type: UNSIGNED_16 Index: $22291_d = 5713_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | Hz | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02284/1 | 10.00 Hz | | LS_WriteParamList: c971_s1_val1 |
| C02284/2 | 10.00 Hz | | LS_WriteParamList: c971_s1_val2 |
| C02284/3 | 10.00 Hz | | LS_WriteParamList: c971_s1_val3 |
| C02284/4 | 10.00 Hz | | LS_WriteParamList: c971_s1_val4 |
| C02284/5 | 100.00 Hz | | LS_WriteParamList: c971_s2_val1 |
| C02284/6 | 100.00 Hz | | LS_WriteParamList: c971_s2_val2 |
| C02284/7 | 100.00 Hz | | LS_WriteParamList: c971_s2_val3 |
| C02284/8 | 100.00 Hz | | LS_WriteParamList: c971_s2_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02285**

| Parameter \| Name: **C02285 \| LS_WriteParamList: VFC: Vp V/f +encoder** | | | Data type: UNSIGNED_16 Index: $22290_d = 5712_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | Hz/Hz | 64.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02285/1 | 0.100 Hz/Hz | | LS_WriteParamList: c972_val1 |
| C02285/2 | 0.100 Hz/Hz | | LS_WriteParamList: c972_val2 |
| C02285/3 | 0.100 Hz/Hz | | LS_WriteParamList: c972_val3 |
| C02285/4 | 0.100 Hz/Hz | | LS_WriteParamList: c972_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1000 | | | |

**C02286**

| Parameter \| Name: C02286 \| LS_WriteParamList: VFC: Ti V/f +encoder | | | Data type: UNSIGNED_16 Index: $22289_d = 5711_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | ms | 6000.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02286/1 | 100.0 ms | | LS_WriteParamList: c973_val1 |
| C02286/2 | 100.0 ms | | LS_WriteParamList: c973_val2 |
| C02286/3 | 100.0 ms | | LS_WriteParamList: c973_val3 |
| C02286/4 | 100.0 ms | | LS_WriteParamList: c973_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02287**

| Parameter \| Name: C02287 \| LS_WriteParamList: VFC-ECO: Vp CosPhi controller | | | Data type: UNSIGNED_16 Index: $22288_d = 5710_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | Hz/Hz | 64.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02287/1 | 0.500 Hz/Hz | | LS_WriteParamList: c975_val1 |
| C02287/2 | 0.500 Hz/Hz | | LS_WriteParamList: c975_val2 |
| C02287/3 | 0.500 Hz/Hz | | LS_WriteParamList: c975_val3 |
| C02287/4 | 0.500 Hz/Hz | | LS_WriteParamList: c975_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1000 | | | |

**C02288**

| Parameter \| Name: C02288 \| LS_WriteParamList: VFC-ECO: Ti CosPhi controller | | | Data type: UNSIGNED_16 Index: $22287_d = 570F_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | ms | 6000.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02288/1 | 200.0 ms | | LS_WriteParamList: c976_val1 |
| C02288/2 | 200.0 ms | | LS_WriteParamList: c976_val2 |
| C02288/3 | 200.0 ms | | LS_WriteParamList: c976_val3 |
| C02288/4 | 200.0 ms | | LS_WriteParamList: c976_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02289**

| Parameter \| Name: C02289 \| LS_WriteParamList: VFC-ECO: Minimum voltage U/f | | | Data type: INTEGER_16 Index: $22286_d = 570E_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 20.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02289/1 | 20.00 % | | LS_WriteParamList: c977_val1 |
| C02289/2 | 20.00 % | | LS_WriteParamList: c977_val2 |
| C02289/3 | 20.00 % | | LS_WriteParamList: c977_val3 |
| C02289/4 | 20.00 % | | LS_WriteParamList: c977_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02290**

| Parameter \| Name: C02290 \| LS_WriteParamList: VFC-ECO: Voltage reduction ramp | | | Data type: UNSIGNED_8 Index: $22285_d = 570D_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | s | 5.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02290/1 | 0.8 s | | LS_WriteParamList: c982_val1 |
| C02290/2 | 0.8 s | | LS_WriteParamList: c982_val2 |
| C02290/3 | 0.8 s | | LS_WriteParamList: c982_val3 |
| C02290/4 | 0.8 s | | LS_WriteParamList: c982_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02291**

| Parameter \| Name: C02291 \| LS_WriteParamList: SLVC: Field current controller gain | | | Data type: INTEGER_16 Index: $22284_d = 570C_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 20.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02291/1 | 0.50 % | | LS_WriteParamList: c985_val1 |
| C02291/2 | 0.50 % | | LS_WriteParamList: c985_val2 |
| C02291/3 | 0.50 % | | LS_WriteParamList: c985_val3 |
| C02291/4 | 0.50 % | | LS_WriteParamList: c985_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02292**

| Parameter \| Name: C02292 \| LS_WriteParamList: SLVC: Cross current controller gain | | | Data type: INTEGER_16 Index: $22283_d = 570B_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 20.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02292/1 | 0.00 % | | LS_WriteParamList: c986_val1 |
| C02292/2 | 0.00 % | | LS_WriteParamList: c986_val2 |
| C02292/3 | 0.00 % | | LS_WriteParamList: c986_val3 |
| C02292/4 | 0.00 % | | LS_WriteParamList: c986_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02293**

| Parameter \| Name: C02293 \| LS_WriteParamList: Inverter motor brake: nAdd | | | Data type: INTEGER_16 Index: $22282_d = 570A_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | rpm | 1000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02293/1 | 80 rpm | | LS_WriteParamList: c987_val1 |
| C02293/2 | 80 rpm | | LS_WriteParamList: c987_val2 |
| C02293/3 | 80 rpm | | LS_WriteParamList: c987_val3 |
| C02293/4 | 80 rpm | | LS_WriteParamList: c987_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02294**

| Parameter \| Name: C02294 \| LS_WriteParamList: Inverter motor brake: PT1 filter time | | | Data type: INTEGER_16 Index: $22281_d = 5709_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | ms | 100.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02294/1 | 0.0 ms | | LS_WriteParamList: c988_val1 |
| C02294/2 | 0.0 ms | | LS_WriteParamList: c988_val2 |
| C02294/3 | 0.0 ms | | LS_WriteParamList: c988_val3 |
| C02294/4 | 0.0 ms | | LS_WriteParamList: c988_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C02295**

| Parameter \| Name: **C02295 \| LS_WriteParamList: Flying restart fct.: Activation** | | Data type: UNSIGNED_8 Index: $22280_d = 5708_h$ |
|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | |
| **Selection list** | | |
| 0 | Off | |
| 1 | On | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02295/1 | 0: Off | LS_WriteParamList: c990_val1 |
| C02295/2 | 0: Off | LS_WriteParamList: c990_val2 |
| C02295/3 | 0: Off | LS_WriteParamList: c990_val3 |
| C02295/4 | 0: Off | LS_WriteParamList: c990_val4 |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02296**

| Parameter \| Name: **C02296 \| LS_WriteParamList: Flying restart fct.: process** | | Data type: UNSIGNED_16 Index: $22279_d = 5707_h$ |
|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | |
| **Selection list** | | |
| 0 | 0...+n \| Start: +10 Hz | |
| 1 | -n...0 \| Start: -10 Hz | |
| 2 | -n...+n \| Start: +10 Hz | |
| 3 | -n...+n \| Start: -10 Hz | |
| 4 | -n...+n \| Start: Cx992 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02296/1 | 2: -n...+n \| Start: +10 Hz | LS_WriteParamList: c991_val1 |
| C02296/2 | 2: -n...+n \| Start: +10 Hz | LS_WriteParamList: c991_val2 |
| C02296/3 | 2: -n...+n \| Start: +10 Hz | LS_WriteParamList: c991_val3 |
| C02296/4 | 2: -n...+n \| Start: +10 Hz | LS_WriteParamList: c991_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02297**

| Parameter \| Name: **C02297 \| LS_WriteParamList: Flying restart: Start frequency** | | | Data type: INTEGER_16 Index: $22278_d = 5706_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -200 | Hz | 200 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02297/1 | 10 Hz | | LS_WriteParamList: c992_val1 |
| C02297/2 | 10 Hz | | LS_WriteParamList: c992_val2 |
| C02297/3 | 10 Hz | | LS_WriteParamList: c992_val3 |
| C02297/4 | 10 Hz | | LS_WriteParamList: c992_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02298**

| Parameter \| Name: **C02298 \| LS_WriteParamList: Flying restart fct: Int. time** | | | Data type: UNSIGNED_16 Index: $22277_d = 5705_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | ms | 6000.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02298/1 | 300.0 ms | | LS_WriteParamList: c993_val1 |
| C02298/2 | 300.0 ms | | LS_WriteParamList: c993_val2 |
| C02298/3 | 300.0 ms | | LS_WriteParamList: c993_val3 |
| C02298/4 | 300.0 ms | | LS_WriteParamList: c993_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02299**

| Parameter \| Name: **C02299 \| LS_WriteParamList: Flying restart fct.: Current** | | | Data type: INTEGER_16 Index: $22276_d = 5704_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02299/1 | 25.00 % | | LS_WriteParamList: c994_val1 |
| C02299/2 | 25.00 % | | LS_WriteParamList: c994_val2 |
| C02299/3 | 25.00 % | | LS_WriteParamList: c994_val3 |
| C02299/4 | 25.00 % | | LS_WriteParamList: c994_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02300**

| Parameter \| Name: **C02300 \| LS_WriteParamList: SLPSM: Controlled current setpoint** | | | Data type: UNSIGNED_16 Index: $22275_d = 5703_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 5.00 | % | 400.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02300/1 | 100.00 % | | LS_WriteParamList: c995_s1_val1 |
| C02300/2 | 100.00 % | | LS_WriteParamList: c995_s1_val2 |
| C02300/3 | 100.00 % | | LS_WriteParamList: c995_s1_val3 |
| C02300/4 | 100.00 % | | LS_WriteParamList: c995_s1_val4 |
| C02300/5 | 20.00 % | | LS_WriteParamList: c995_s2_val1 |
| C02300/6 | 20.00 % | | LS_WriteParamList: c995_s2_val2 |
| C02300/7 | 20.00 % | | LS_WriteParamList: c995_s2_val3 |
| C02300/8 | 20.00 % | | LS_WriteParamList: c995_s2_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02301**

| Parameter \| Name: **C02301 \| LS_WriteParamList: SLPSM: Switching speed** | | | Data type: INTEGER_16 Index: $22274_d = 5702_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02301/1 | 13.00 % | | LS_WriteParamList: c996_s1_val1 |
| C02301/2 | 13.00 % | | LS_WriteParamList: c996_s1_val2 |
| C02301/3 | 13.00 % | | LS_WriteParamList: c996_s1_val3 |
| C02301/4 | 13.00 % | | LS_WriteParamList: c996_s1_val4 |
| C02301/5 | 8.00 % | | LS_WriteParamList: c996_s2_val1 |
| C02301/6 | 8.00 % | | LS_WriteParamList: c996_s2_val2 |
| C02301/7 | 8.00 % | | LS_WriteParamList: c996_s2_val3 |
| C02301/8 | 8.00 % | | LS_WriteParamList: c996_s2_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02302**

| Parameter \| Name: **C02302 \| LS_WriteParamList: SLPSM: Filter time - rotor position** | | | Data type: INTEGER_16 Index: $22273_d = 5701_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02302/1 | 5.00 % | | LS_WriteParamList: c997_val1 |
| C02302/2 | 5.00 % | | LS_WriteParamList: c997_val2 |
| C02302/3 | 5.00 % | | LS_WriteParamList: c997_val3 |
| C02302/4 | 5.00 % | | LS_WriteParamList: c997_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02303**

| Parameter \| Name: **C02303 \| LS_WriteParamList: SLPSM: Filter time rotor position** | | | Data type: INTEGER_16 Index: $22272_d = 5700_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.5 | ms | 20.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02303/1 | 3.0 ms | | LS_WriteParamList: c998_s1_val1 |
| C02303/2 | 3.0 ms | | LS_WriteParamList: c998_s1_val2 |
| C02303/3 | 3.0 ms | | LS_WriteParamList: c998_s1_val3 |
| C02303/4 | 3.0 ms | | LS_WriteParamList: c998_s1_val4 |
| C02303/5 | 5.0 ms | | LS_WriteParamList: c998_s2_val1 |
| C02303/6 | 5.0 ms | | LS_WriteParamList: c998_s2_val2 |
| C02303/7 | 5.0 ms | | LS_WriteParamList: c998_s2_val3 |
| C02303/8 | 5.0 ms | | LS_WriteParamList: c998_s2_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02304**

| Parameter \| Name: C02304 \| LS_WriteParamList: SLPSM: PLL gain | | | Data type: INTEGER_16 Index: $22271_d = 56FF_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | % | 1000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02304/1 | 400 % | | LS_WriteParamList: c999_val1 |
| C02304/2 | 400 % | | LS_WriteParamList: c999_val2 |
| C02304/3 | 400 % | | LS_WriteParamList: c999_val3 |
| C02304/4 | 400 % | | LS_WriteParamList: c999_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02305**

| Parameter \| Name: C02305 \| LS_WriteParamList: PSM: Ppp saturation characteristic | | | Data type: UNSIGNED_8 Index: $22270_d = 56FE_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | % | 255 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02305/1 | 100 % | | LS_WriteParamList: c2853_s1_val1 |
| C02305/2 | 100 % | | LS_WriteParamList: c2853_s1_val2 |
| C02305/3 | 100 % | | LS_WriteParamList: c2853_s1_val3 |
| C02305/4 | 100 % | | LS_WriteParamList: c2853_s1_val4 |
| C02305/5 | 100 % | | LS_WriteParamList: c2853_s2_val1 |
| C02305/6 | 100 % | | LS_WriteParamList: c2853_s2_val2 |
| C02305/7 | 100 % | | LS_WriteParamList: c2853_s2_val3 |
| C02305/8 | 100 % | | LS_WriteParamList: c2853_s2_val4 |
| C02305/9 | 100 % | | LS_WriteParamList: c2853_s3_val1 |
| C02305/10 | 100 % | | LS_WriteParamList: c2853_s3_val2 |
| C02305/11 | 100 % | | LS_WriteParamList: c2853_s3_val3 |
| C02305/12 | 100 % | | LS_WriteParamList: c2853_s3_val4 |
| C02305/13 | 100 % | | LS_WriteParamList: c2853_s4_val1 |
| C02305/14 | 100 % | | LS_WriteParamList: c2853_s4_val2 |
| C02305/15 | 100 % | | LS_WriteParamList: c2853_s4_val3 |
| C02305/16 | 100 % | | LS_WriteParamList: c2853_s4_val4 |
| C02305/17 | 100 % | | LS_WriteParamList: c2853_s5_val1 |
| C02305/18 | 100 % | | LS_WriteParamList: c2853_s5_val2 |
| C02305/19 | 100 % | | LS_WriteParamList: c2853_s5_val3 |
| C02305/20 | 100 % | | LS_WriteParamList: c2853_s5_val4 |
| C02305/21 | 100 % | | LS_WriteParamList: c2853_s6_val1 |
| C02305/22 | 100 % | | LS_WriteParamList: c2853_s6_val2 |
| C02305/23 | 100 % | | LS_WriteParamList: c2853_s6_val3 |
| C02305/24 | 100 % | | LS_WriteParamList: c2853_s6_val4 |
| C02305/25 | 100 % | | LS_WriteParamList: c2853_s7_val1 |
| C02305/26 | 100 % | | LS_WriteParamList: c2853_s7_val2 |

| Parameter \| Name:<br>**C02305 \| LS_WriteParamList: PSM: Ppp saturation characteristic** | | Data type: UNSIGNED_8<br>Index: $22270_d = 56FE_h$ |
|---|---|---|
| C02305/27 | 100 % | LS_WriteParamList: c2853_s7_val3 |
| C02305/28 | 100 % | LS_WriteParamList: c2853_s7_val4 |
| C02305/29 | 100 % | LS_WriteParamList: c2853_s8_val1 |
| C02305/30 | 100 % | LS_WriteParamList: c2853_s8_val2 |
| C02305/31 | 100 % | LS_WriteParamList: c2853_s8_val3 |
| C02305/32 | 100 % | LS_WriteParamList: c2853_s8_val4 |
| C02305/33 | 100 % | LS_WriteParamList: c2853_s9_val1 |
| C02305/34 | 100 % | LS_WriteParamList: c2853_s9_val2 |
| C02305/35 | 100 % | LS_WriteParamList: c2853_s9_val3 |
| C02305/36 | 100 % | LS_WriteParamList: c2853_s9_val4 |
| C02305/37 | 100 % | LS_WriteParamList: c2853_s10_val1 |
| C02305/38 | 100 % | LS_WriteParamList: c2853_s10_val2 |
| C02305/39 | 100 % | LS_WriteParamList: c2853_s10_val3 |
| C02305/40 | 100 % | LS_WriteParamList: c2853_s10_val4 |
| C02305/41 | 100 % | LS_WriteParamList: c2853_s11_val1 |
| C02305/42 | 100 % | LS_WriteParamList: c2853_s11_val2 |
| C02305/43 | 100 % | LS_WriteParamList: c2853_s11_val3 |
| C02305/44 | 100 % | LS_WriteParamList: c2853_s11_val4 |
| C02305/45 | 100 % | LS_WriteParamList: c2853_s12_val1 |
| C02305/46 | 100 % | LS_WriteParamList: c2853_s12_val2 |
| C02305/47 | 100 % | LS_WriteParamList: c2853_s12_val3 |
| C02305/48 | 100 % | LS_WriteParamList: c2853_s12_val4 |
| C02305/49 | 100 % | LS_WriteParamList: c2853_s13_val1 |
| C02305/50 | 100 % | LS_WriteParamList: c2853_s13_val2 |
| C02305/51 | 100 % | LS_WriteParamList: c2853_s13_val3 |
| C02305/52 | 100 % | LS_WriteParamList: c2853_s13_val4 |
| C02305/53 | 100 % | LS_WriteParamList: c2853_s14_val1 |
| C02305/54 | 100 % | LS_WriteParamList: c2853_s14_val2 |
| C02305/55 | 100 % | LS_WriteParamList: c2853_s14_val3 |
| C02305/56 | 100 % | LS_WriteParamList: c2853_s14_val4 |
| C02305/57 | 100 % | LS_WriteParamList: c2853_s15_val1 |
| C02305/58 | 100 % | LS_WriteParamList: c2853_s15_val2 |
| C02305/59 | 100 % | LS_WriteParamList: c2853_s15_val3 |
| C02305/60 | 100 % | LS_WriteParamList: c2853_s15_val4 |
| C02305/61 | 100 % | LS_WriteParamList: c2853_s16_val1 |
| C02305/62 | 100 % | LS_WriteParamList: c2853_s16_val2 |
| C02305/63 | 100 % | LS_WriteParamList: c2853_s16_val3 |
| C02305/64 | 100 % | LS_WriteParamList: c2853_s16_val4 |
| C02305/65 | 100 % | LS_WriteParamList: c2853_s17_val1 |
| C02305/66 | 100 % | LS_WriteParamList: c2853_s17_val2 |
| C02305/67 | 100 % | LS_WriteParamList: c2853_s17_val3 |
| C02305/68 | 100 % | LS_WriteParamList: c2853_s17_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02306**

| Parameter \| Name: **C02306 \| LS_WriteParamList: PSM: Imax Ppp saturation characteristic** | | | Data type: UNSIGNED_16 Index: $22269_d = 56FD_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | A | 3000.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02306/1 | 3000.0 A | | LS_WriteParamList: c2855_val1 |
| C02306/2 | 3000.0 A | | LS_WriteParamList: c2855_val2 |
| C02306/3 | 3000.0 A | | LS_WriteParamList: c2855_val3 |
| C02306/4 | 3000.0 A | | LS_WriteParamList: c2855_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02307**

| Parameter \| Name: **C02307 \| LS_WriteParamList: PSM: Activate Ppp saturation char.** | | Data type: UNSIGNED_8 Index: $22268_d = 56FC_h$ |
|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | |
| **Selection list** | | |
| 0 | Off | |
| 1 | On | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02307/1 | 0: Off | LS_WriteParamList: c2859_val1 |
| C02307/2 | 0: Off | LS_WriteParamList: c2859_val |
| C02307/3 | 0: Off | LS_WriteParamList: c2859_val |
| C02307/4 | 0: Off | LS_WriteParamList: c2859_val |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02311**

| Parameter \| Name: **C02311 \| LS_WriteParamList: PLI without motion: Adaptation of time duration** | | | Data type: INTEGER_8 Index: $22264_d = 56F8_h$ |
|---|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -10 | | 10 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02311/1 | 0 | | LS_WriteParamList: c2872_s1_val1 |
| C02311/2 | 0 | | LS_WriteParamList: c2872_s1_val2 |
| C02311/3 | 0 | | LS_WriteParamList: c2872_s1_val3 |
| C02311/4 | 0 | | LS_WriteParamList: c2872_s1_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02312**

| Parameter \| Name: **C02312 \| LS_WriteParamList: PLI without motion** | | Data type: UNSIGNED_16 Index: $22263_d = 56F7_h$ |
|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** | | |
| Bit 0 | for SLPSM with controller enable | |
| Bit 1 | for SC PSM with mains on | |
| Bit 2 | for SC PSM with controller enable | |
| Bit 3 | for SC PSM once after fault reset | |
| Bit 4 | Reserved | |
| Bit 5 | Reserved | |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02312/1 | 0x0001 | LS_WriteParamList: c2874_s1_val1 |
| C02312/2 | 0x0001 | LS_WriteParamList: c2874_s1_val2 |
| C02312/3 | 0x0001 | LS_WriteParamList: c2874_s1_val3 |
| C02312/4 | 0x0001 | LS_WriteParamList: c2874_s1_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT | | |

**C02313**

| Parameter \| Name: **C02313 \| LS_WriteParamList: PLI without motion: Adaptation of ident angle** | | Data type: INTEGER_8 Index: $22262_d = 56F6_h$ |
|---|---|---|
| From version 12.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -100 | ° | 100 |
| **Subcodes** | **Lenze setting** | **Info** |
| C02313/1 | 0 ° | LS_WriteParamList: c2875_s1_val1 |
| C02313/2 | 0 ° | LS_WriteParamList: c2875_s1_val2 |
| C02313/3 | 0 ° | LS_WriteParamList: c2875_s1_val3 |
| C02313/4 | 0 ° | LS_WriteParamList: c2875_s1_val4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02314**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C02314 \| Reserved** | Index: $22261_d = 56F5_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C02315**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C02315 \| LS_WriteParamList: Manual entry of motor type** | Index: $22260_d = 56F4_h$ |

From version 14.00.00 onwards: Parameter values (1st value ... 4th value) for optional motor data change-over

| Selection list | |
|---|---|
| 0 | Automatic |
| 1 | ASM |
| 2 | PSM |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C02315/1 | 0: Automatic | LS_WriteParamList: c1001_s1_val1 |
| C02315/2 | 0: Automatic | LS_WriteParamList: c1001_s1_val2 |
| C02315/3 | 0: Automatic | LS_WriteParamList: c1001_s1_val3 |
| C02315/4 | 0: Automatic | LS_WriteParamList: c1001_s1_val4 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C02430**

| Parameter \| Name: | Data type: UNSIGNED_8 |
|---|---|
| **C02430 \| Axis bus address and no. of nodes** | Index: $22145_d = 5681_h$ |

From version 02.00.00

▸ Data transfer axis bus

| Setting range (min. value \| unit \| max. value) | | |
|---|---|---|
| 0 | | 62 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C02430/1 | 0 | Axis bus address<br>• 0 ≡ Nodes disconnected from the axis bus<br>• 1 ≡ Master<br>• 2 ... 62 ≡ slave 1 ... slave 61 |
| C02430/2 | 2 | Axis bus no. of nodes<br>• Number of slaves connected to the axis bus.<br>• Setting only required for the master. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C02431**

| Parameter \| Name: C02431 \| Axis bus time settings | | | Data type: UNSIGNED_16 Index: $22144_d = 5680_h$ |
|---|---|---|---|
| From version 02.00.00 | | | ▸ Data transfer axis bus |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 65000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02431/1 | 3000 ms | | Axis bus decel. boot-up<br>• Delay during status change from "Boot-up" to "Operational". |
| C02431/2 | 1000 ms | | Axis bus decel. BusOff recovery<br>• Delay after which the "BusOff" status is reset automatically.<br>▸ Data error monitoring. |
| C02431/3 | 1000 ms | | Axis bus monitoring time<br>▸ Data error monitoring. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C02435**

| Parameter \| Name: C02435 \| Axis bus status | | Data type: UNSIGNED_8 Index: $22140_d = 567C_h$ |
|---|---|---|
| From version 02.00.00 | | ▸ Data transfer axis bus |
| **Selection list** (read only) | | |
| 0 | PDO active | |
| 1 | SDO active | |
| 2 | Reserved | |
| 3 | Reserved | |
| 4 | BootUp | |
| 5 | Stopped | |
| 6 | Reserved | |
| 7 | Off | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C02436**

| Parameter \| Name: C02436 \| Axis bus error status | | Data type: UNSIGNED_8 Index: $22139_d = 567B_h$ |
|---|---|---|
| From version 02.00.00 | | ▸ Data transfer axis bus |
| **Selection list** (read only) | | |
| 0 | No Error | |
| 1 | Warning ErrActive | |
| 2 | Warning ErrPassive | |
| 3 | Bus off | |
| 4 | Reserved | |
| 5 | Reserved | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02437**

| Parameter \| Name: C02437 \| Axis bus MessageError | | Data type: UNSIGNED_8 Index: $22138_d = 567A_h$ |
|---|---|---|
| From version 02.00.00 | | ▸ Data transfer axis bus |
| **Selection list** (read only) | | |
| 0 | No Error | |
| 1 | StuffError | |
| 2 | FormError | |
| 3 | AckError | |
| 4 | Bit1Error | |
| 5 | Bit0Error | |
| 6 | CRCError | |
| 7 | Reserved | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C02438**

| Parameter \| Name: C02438 \| CAB_Tx_Rx_diagnosis | | Data type: UNSIGNED_16 Index: $22137_d = 5679_h$ |
|---|---|---|
| From version 12.00.00 | | ▸ Data transfer axis bus |
| **Display area** (min. value \| unit \| max. value) | | |
| 0 | | 65535 |
| **Subcodes** | | **Info** |
| C02438/1 | | Axis bus Tx_Error |
| C02438/2 | | Axis bus Rx_Error |
| C02438/3 | | Axis bus transmitting meter |
| C02438/4 | | Axis bus receiving meter |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C02440**

| Parameter \| Name: C02440 \| AxisBusIO slave/master | | Data type: UNSIGNED_8 Index: $22135_d = 5677_h$ |
|---|---|---|
| From version 02.00.00 | | ▸ Axis bus |
| **Selection list** | | |
| 0 | Off | |
| 1 | Master | |
| 2 | slave | |
| 3 | IO | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02440/1 | 0: Off | Axis bus IO function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C02442**

| Parameter \| Name: **C02442 \| CAB decoupling inputs** | | Data type: UNSIGNED_16 Index: $22133_d = 5675_h$ |
|---|---|---|
| From version 12.00.00<br>Bit-coded selection of the events causing the decoupling of the receive data of the data transfer axis bus.<br>▸ Data transfer axis bus | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** | | |
| Bit 0 | BusOff_MsgErr | |
| Bit 1 | Warning | |
| Bit 2 | NodeStopped | |
| Bit 3 | Reserved | |
| Bit 4 | Reserved | |
| Bit 5 | Reserved | |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Trouble | |
| Bit 15 | Fault | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02442/1 | 0x0000 | Decoupling inputs from the axis bus |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C02443**

| Parameter \| Name: **C02443 \| CAB decoupling value** | | Data type: UNSIGNED_16 Index: $22132_d = 5674_h$ |
|---|---|---|
| From version 12.00.00<br>Definition of the value the receive data of the data transfer axis bus are to have in the decoupled state.<br>▸ Data transfer axis bus | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | | 65535 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C02443/1 | 0 | LS_AxisBusIn: wLine1 DiscVal |
| C02443/2 | 0 | LS_AxisBusIn: wLine2 DiscVal |
| C02443/3 | 0 | LS_AxisBusIn: wLine3 DiscVal |
| C02443/4 | 0 | LS_AxisBusIn: wCas1 DiscVal |
| C02443/5 | 0 | LS_AxisBusIn: wCas2 DiscVal |
| C02443/6 | 0 | LS_AxisBusIn: wCas3 DiscVal |
| C02443/7 | 0 | LS_AxisBusIn: wCas4 DiscVal |
| C02443/8 | 0 | LS_AxisBusAux: wAuxIn1 DiscVal |
| C02443/9 | 0 | LS_AxisBusAux: wAuxIn2 DiscVal |
| C02443/10 | 0 | LS_AxisBusAux: wAuxIn3 DiscVal |
| C02443/11 | 0 | LS_AxisBusAux: wAuxIn4 DiscVal |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C02444**

| Parameter \| Name: **C02444 \| CAB configuration** | | Data type: UNSIGNED_16 Index: $22131_d = 5673_h$ |
|---|---|---|
| From version 12.00.00<br>Bit-coded settings<br>▸ Data transfer axis bus | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** | | **Info** |
| Bit 0 | Edge detection SetFail | 0 ≡ input *bSetFail_DigOut* is level-sensitive.<br>1 ≡ input *bSetFail_DigOut* is edge-sensitive. |
| Bit 1 | Reserved | |
| Bit 2 | Reserved | |
| Bit 3 | Reserved | |
| Bit 4 | Reserved | |
| Bit 5 | Reserved | |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | same CobId for all Cas | |
| Bit 15 | 500 kbps | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02444/1 | 0x0000 | Axis bus configuration |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C02556**

| Parameter \| Name: **C02556 \| Position controller: Limitation** | | Data type: INTEGER_32 Index: $22019_d = 5603_h$ |
|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | Incr. | 2147483647 |
| **Subcodes** | **Lenze setting** | **Info** |
| C02556/1 | 65536 incr. | Following error limit<br>• Setting of the maximum permissible correcting variable or the maximum permissible following error for the position controller. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02580**

| Parameter \| Name: **C02580 \| Holding brake: Operating mode** | | Data type: UNSIGNED_8 Index: $21995_d = 55EB_h$ |
|---|---|---|
| Selection of the operating mode for holding brake control<br>▸ Holding brake control | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Brake control off** | No holding brake is used. Internal control is switched off. |
| 11 | Manually controlled | The holding brake is released and closed via a control bit in the MCK control word. |
| 12 | Autom. controlled | The holding brake is automatically released and closed via speed setpoint comparisons. |
| 13 | Semi-automat. controlled | The holding brake is released and closed via a control bit in the MCK control word.<br>• In contrast to the manual operation (mode 11)<br>  • the feedforward control is active in this mode, preventing a sagging e.g. in case of a hoist.<br>  • the brake in this mode also closes when the controller is inhibited in order to prevent the axis in a hoist from falling. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02581**

| Parameter \| Name: **C02581 \| Holding brake: Speed thresholds** | | | Data type: INTEGER_16 Index: $21994_d = 55EA_h$ |
|---|---|---|---|
| Speed setpoint threshold and hysteresis for automatic holding brake control<br>▸ Holding brake control | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02581/1 | 5.00 % | | Holding brake: Switching threshold<br>• Switching threshold of the speed setpoint from which on the holding brake is released/applied automatically. |
| C02581/2 | 1.00 % | | Holding brake: Hyst.release<br>• Hysteresis for holding brake release.<br>• Release threshold = switching threshold + release hysteresis |
| C02581/3 | 1.00 % | | Holding brake: Hyst. close<br>• Hysteresis for holding brake application.<br>• Application threshold = switching threshold - application hysteresis |
| C02581/4 | 0.00 % | | Holding brake: FF control starting value 1 |
| C02581/5 | 0.00 % | | Holding brake: FF control starting value 2 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02582**

| Parameter \| Name: **C02582 \| Holding brake: Setting** | | | Data type: UNSIGNED_8 Index: $21993_d = 55E9_h$ |
|---|---|---|---|
| Activation of functional holding brake control options<br>▸ Holding brake control | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x40** (decimal: 64) |

| Parameter \| Name:<br>**C02582 \| Holding brake: Setting** | | Data type: UNSIGNED_8<br>Index: $21993_d = 55E9_h$ |
|---|---|---|
| **Value is bit-coded: (☑ = bit set)** | | **Info** |
| Bit 0 ☐ | bMBrakeReleaseOut invert. | Activation of inverted control<br>• 1 ≡ Inverted logic of the control signal for the holding brake control switching element. |
| Bit 1 ☐ | Horizontal brake protection | Brake response in case of pulse inhibit<br>• 1 ≡ In the case of a pulse inhibit, the actual speed value is monitored which must reach the "Close" threshold value to cause the holding brake to be applied.<br>**Note:**<br>• This function is only active if bit 3 (horizontal/winding technology) is set as well. The function is used in order that, when the controller is inhibited, the holding brake of a drive with horizontal traverse path does not wear out during rotation.<br>• With vertical motion (bit 3 = 0), this function is not active. Especially with hoists and activated pulse inhibit of the inverter, an immediate application of the brake is essential for safety-related reasons! |
| Bit 2 ☐ | with hoist inv. feedfwd. control | Direction of feedforward control with vertical/hoist technology:<br>• 0 ≡ Positive direction<br>• 1 ≡ Negative direction<br>**Note:**<br>Reversal (Ccw) is then considered. |
| Bit 3 ☐ | Horizontal application | Direction of movement of the axis<br>• 0 ≡ The axis performs vertical movements. Gravitational acceleration causes movements.<br>• 1 ≡ The direction of the axis is horizontal or rotary. The gravitational acceleration does not cause any movement. |
| Bit 4 ☐ | Feedforward control C2581 | Selection of the feedforward control value<br>• 0 ≡ Automatic selection.<br>  • The torque saved at the last stop is used.<br>• 1 ≡ Manual selection.<br>  • *bMBrakeStartValue2* = FALSE: The feedforward control value 1 set in C02581/4 is used.<br>  • *bMBrakeStartValue2* = TRUE: The feedforward control value set in C02581/5 is used. |
| Bit 5 ☐ | Feedback monitoring | Activation of status monitoring<br>• 1 ≡ The *bMBrakeApplied* input for status detection of the brake (via a switching contact at the brake) is monitored after the waiting time set in C02589/3 has expired. |
| Bit 6 ☑ | SyncRampe L_NSet_1 | From version 02.00.00<br>Selection of the ramp time for the synchronisation process to setpoint speed after the brake opening time has elapsed<br>**Revised behaviour from version 02.00.00:**<br>• 1 ≡ The ramp time of the effective acceleration of the ramp function generator (L_NSet_1) is used (Lenze setting).<br>• 0 ≡ As before, the ramp time set in C02610/1 is used.<br>**Note:**<br>The changeover can be dynamically both via the ramp parameter and via bit 6. |
| Bit 7 ☐ | Reserved | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C02589**

| Parameter \| Name: **C02589 \| Holding brake: Time system** | | | Data type: UNSIGNED_16 Index: $21986_d = 55E2_h$ |
|---|---|---|---|
| Operating times of the holding brake<br>• The electromechanical delay times of the holding brake are specified in the data sheets or on the holding brake nameplate.<br>▸ Holding brake control | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02589/1 | 100 ms | | Holding brake: Application time<br>• Time in which the holding brake is completely applied from the beginning of control and in which the controller is inhibited. |
| C02589/2 | 100 ms | | Holding brake: Release time<br>• Time in which the holding brake is completely released from the beginning of control. |
| C02589/3 | 100 ms | | Holding brake: Waiting time status<br>• Time after which all transient reactions are completed and the switching status of the holding brake is stable. Beginning of monitoring the feedback signal for the switching status of the holding brake. |
| C02589/4 | 0 ms | | Holding brake: Ramp time FF ctrl |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02593**

| Parameter \| Name: **C02593 \| Holding brake: Activation time** | | | Data type: UNSIGNED_32 Index: $21982_d = 55DE_h$ |
|---|---|---|---|
| Time parameter for the delay of trigger signals of the holding brake control<br>▸ Holding brake control | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 3600.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02593/1 | 0.000 s | | Holding brake: Actual value monitoring<br>• Time in which the actual value is supposed to reach the threshold for closing the brake if the setpoint has already reached the threshold.<br>• Time > 0 s: If the actual speed value has not reached the threshold within the time for brake application, the holding brake is applied by control.<br>• Time = 0 s: The brake is only applied by control when the actual speed has reached the application threshold. |
| C02593/2 | 0.000 s | | Holding brake: Application delay<br>• Time by which the control process for holding brake application is delayed. The time expires when the speed setpoint has reached the switching threshold for application.<br>• With positioning processes, a continuous application and release of the holding brake can thus be suppressed for the set time. |
| C02593/3 | 0.000 s | | Holding brake: Reserved |
| C02593/4 | 0.000 s | | Holding brake: Reserved |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C02607**

| Parameter \| Name: **C02607 \| Holding brake: Status** | | | Data type: UNSIGNED_16 Index: $21968_d = 55D0_h$ |
|---|---|---|---|
| Switching status of the holding brake control ▸ Holding brake control | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Brake applied | | 1 ≡ Holding brake is completely applied |
| Bit 1 | Brake released | | 1 ≡ Holding brake is completely released |
| Bit 2 | Feedforward control active | | 1 ≡ Feedforward control for holding of the load via the motor is active before the holding brake releases. |
| Bit 3 | Closing active | | 1 ≡ The brake closing time (C02589/1) expires |
| Bit 4 | Forced release active | | 1 ≡ In case of automatic operation of the holding brake control, the brake is directly released via the MCK input *bMBrakeRelease* = TRUE |
| Bit 5 | Release active | | 1 ≡ The brake release time (C02589/2) expires |
| Bit 6 | Setpoint synchronisation active | | 1 ≡ A speed setpoint at the MCK is approached along a defined ramp after brake release |
| Bit 7 | Signalling contact error | | 1 ≡ The status monitoring has tripped |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C02610**

| Parameter \| Name: **C02610 \| MCK: Accel./decel. times** | | | Data type: UNSIGNED_32 Index: $21965_d$ = $55CD_h$ |
|---|---|---|---|
| Ramp times for speed setpoint synchronisation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02610/1 | 2.000 s | | Holding brake: ramp time synchr.<br>• Ramp time for the synchronisation process to setpoint speed after the brake opening time has elapsed<br>**Revised behaviour from version 02.00.00:**<br>• The setting made here is only effective if bit 6 "SyncRampe L_NSet_1" in C02582 is set to "0".<br>• In the Lenze setting of C02582 (Bit 6 = "1"), the ramp time of the effective acceleration of the ramp function generator (L_NSet_1) is used.<br>▸ Holding brake control |
| C02610/2 | 2.000 s | | MCK: Ramp time synchr. setpoint<br>• Time for synchronisation ramps between setpoint jumps occurring through the exceedance of minimum and maximum speed setpoint limit ranges. |
| C02610/3 | 2.000 s | | MCK: SM stopping ramp |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C02611**

| Parameter \| Name: **C02611 \| MCK: Limitations** | | | Data type: INTEGER_16 Index: $21964_d$ = $55CC_h$ |
|---|---|---|---|
| Speed setpoint limits for the determination of limited validity ranges<br><br>**Note:**<br>Traversing with setpoints through resulting blocking zones is executed with the ramp set in C02610/2.<br>▸ Speed Min/Max | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02611/1 | 199.99 % | | MCK: Pos. max. speed<br>• Upper limit of the speed setpoint limitation in positive direction of rotation. |
| C02611/2 | 0.00 % | | MCK: Pos. min. speed<br>• Lower limit of the speed setpoint limitation in positive direction of rotation. |
| C02611/3 | 0.00 % | | MCK: Neg. min. speed<br>• Lower limit of the speed setpoint limitation in negative direction of rotation. |
| C02611/4 | 199.99 % | | MCK: Neg. max. speed<br>• Upper limit of the speed setpoint limitation in negative direction of rotation. |
| C02611/5 | 0.50 % | | MCK: Max. speed<br>• From version 12.00.00<br>• Maximum value for transferring the current speed to the speed setpoint when the internal operating mode "StandBy" is exited. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02652**

| Parameter \| Name: **C02652 \| Settings of measuring system** | | | Data type: UNSIGNED_16 Index: $21923_d = 55A3_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | Actual MCTRL position received at mains switch-off | | ▸ Actual "MCTRL position received at mains switch-off" |
| Bit 1 ☐ | Delete reference when machine data is changed | | From version 12.00.00<br>▸ Status bit "HomPosAvailable" |
| Bit 2 ☐ | Adapt position encoder offset with homing [C1112/1] | | |
| Bit 3 ☐ | Reference reconstruction after encoder error removal | | |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C02810**

| Parameter \| Name: **C02810 \| TP: Edge selection** | | Data type: UNSIGNED_8 Index: $21765_d = 5505_h$ |
|---|---|---|
| ▸ Touch-probe detection | | |
| **Selection list** | | |
| 0 | Off | |
| 1 | Rising | |
| 2 | Falling | |
| 3 | Rising and falling | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02810/1 | 0: Off | Reserved |
| C02810/2 | 0: Off | Reserved |
| C02810/3 | 0: Off | TPDigIn3: Edge selection |
| C02810/4 | 0: Off | TPDigIn4: Edge selection |
| C02810/5 | 0: Off | TPDigIn5: Edge selection |
| C02810/6 | 0: Off | TPDigIn6: Edge selection |
| C02810/7 | 0: Off | TPDigIn7: Edge selection |
| C02810/8 | 0: Off | TPEncoderZtrack: Edge selection<br>• From version 12.00.00 |
| C02810/9 | 0: Off | TPResolverZtrack: Edge selection<br>• From version 12.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02811**

| Parameter \| Name: **C02811 \| TP: Sensor delay** | | | Data type: UNSIGNED_16 Index: $21764_d = 5504_h$ |
|---|---|---|---|
| ▸ Touch-probe detection | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | µs | 6999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02811/1 | 0 µs | | Reserved |
| C02811/2 | 0 µs | | Reserved |
| C02811/3 | 0 µs | | TPDigIn3: Sensor delay |
| C02811/4 | 0 µs | | TPDigIn4: Sensor delay |
| C02811/5 | 0 µs | | TPDigIn5: Sensor delay |
| C02811/6 | 0 µs | | TPDigIn6: Sensor delay |
| C02811/7 | 0 µs | | TPDigIn7: Sensor delay |
| C02811/8 | 0 µs | | TPEncoderZTrack: Sensor delay<br>• From version 12.00.00 |
| C02811/9 | 0 µs | | TPResolverZTrack: Sensor delay<br>• From version 12.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02812**

| Parameter \| Name: **C02812 \| TP: Position offset** | | | Data type: INTEGER_32 Index: $21763_d = 5503_h$ |
|---|---|---|---|
| | | | ▸ Touch-probe detection |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02812/1 | 0.0000 units | | Reserved |
| C02812/2 | 0.0000 units | | Reserved |
| C02812/3 | 0.0000 units | | TPDigIn3: Pos offset |
| C02812/4 | 0.0000 units | | TPDigIn4: Pos offset |
| C02812/5 | 0.0000 units | | TPDigIn5: Pos offset |
| C02812/6 | 0.0000 units | | TPDigIn6: Pos offset |
| C02812/7 | 0.0000 units | | TPDigIn7: Pos offset |
| C02812/8 | 0.0000 units | | TPEncoderZTrack: Pos offset<br>• From version 12.00.00 |
| C02812/9 | 0.0000 units | | TPResolverZTrack: Pos offset<br>• From version 12.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C02813**

| Parameter \| Name: **C02813 \| TP: Pos. window start** | | | Data type: INTEGER_32 Index: $21762_d = 5502_h$ |
|---|---|---|---|
| | | | ▸ Touch-probe detection |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02813/1 | -214748.3647 units | | TPDigIn3: Window start |
| C02813/2 | -214748.3647 units | | TPDigIn4: Window start |
| C02813/3 | -214748.3647 units | | TPDigIn5: Window start |
| C02813/4 | -214748.3647 units | | TPEncoderZTrack: Window start<br>• From version 12.00.00 |
| C02813/5 | -214748.3647 units | | TPResolverZTrack: Window start<br>• From version 12.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C02814**

| Parameter \| Name: **C02814 \| TP: Pos. window end** | | | Data type: INTEGER_32 Index: $21761_d = 5501_h$ |
|---|---|---|---|
| ▸ Touch-probe detection | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02814/1 | 214748.3647 units | | TPDigIn3: Window end |
| C02814/2 | 214748.3647 units | | TPDigIn4: Window end |
| C02814/3 | 214748.3647 units | | TPDigIn5: Window end |
| C02814/4 | 214748.3647 units | | TPEncoderZTrack: Window end<br>• From version 12.00.00 |
| C02814/5 | 214748.3647 units | | TPResolverZTrack: Window end<br>• From version 12.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C02815**

| Parameter \| Name: **C02815 \| TP: Position source** | | | Data type: UNSIGNED_8 Index: $21760_d = 5500_h$ |
|---|---|---|---|
| ▸ Touch-probe detection | | | |
| **Selection list** | | | |
| 0 | Position encoder actual value | | |
| 1 | Pos DigIn1/2 | | |
| 2 | Pos resolver | | |
| 3 | Pos MultiEncoder | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02815/1 | 0: Position encoder actual value | | Reserved |
| C02815/2 | 0: Position encoder actual value | | Reserved |
| C02815/3 | 0: Position encoder actual value | | TPDigIn3: Position source |
| C02815/4 | 0: Position encoder actual value | | TPDigIn4: Position source |
| C02815/5 | 0: Position encoder actual value | | TPDigIn5: Position source |
| C02815/6 | 0: Position encoder actual value | | TPDigIn6: Position source |
| C02815/7 | 0: Position encoder actual value | | TPDigIn7: Position source |
| C02815/8 | 0: Position encoder actual value | | TPEncoderZTrack: Position source<br>• From version 12.00.00 |
| C02815/9 | 0: Position encoder actual value | | TPResolverZTrack: Position source<br>• From version 12.00.00 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02816**

| Parameter \| Name: **C02816 \| TP: Signal counter** | | | Data type: UNSIGNED_16 Index: $21759_d = 54FF_h$ |
|---|---|---|---|
| ▸ Touch-probe detection | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | | | **Info** |
| C02816/1 | | | Reserved |
| C02816/2 | | | Reserved |
| C02816/3 | | | TPDigIn3: Signal counter |
| C02816/4 | | | TPDigIn4: Signal counter |
| C02816/5 | | | TPDigIn5: Signal counter |
| C02816/6 | | | TPDigIn6: Signal counter |
| C02816/7 | | | TPDigIn7: Signal counter |
| C02816/8 | | | TPEncoderZTrack: Signal counter<br>• From version 12.00.00 |
| C02816/9 | | | TPResolverZTrack: Signal counter<br>• From version 12.00.00 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02817**

| Parameter \| Name: **C02817 \| TP: TouchProbe position** | | | Data type: INTEGER_32 Index: $21758_d = 54FE_h$ |
|---|---|---|---|
| ▸ Touch-probe detection | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C02817/1 | | | Reserved |
| C02817/2 | | | Reserved |
| C02817/3 | | | TPDigIn3: TP position |
| C02817/4 | | | TPDigIn4: TP position |
| C02817/5 | | | TPDigIn5: TP position |
| C02817/6 | | | TPDigIn6: TP position |
| C02817/7 | | | TPDigIn7: TP position |
| C02817/8 | | | TPEncoderZTrack: TP position<br>• From version 12.00.00 |
| C02817/9 | | | TPResolverZTrack: TP position<br>• From version 12.00.00 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C02830**

| Parameter \| Name: **C02830 \| DIx: Debounce time** | | Data type: UNSIGNED_8 Index: $21745_d = 54F1_h$ |
|---|---|---|
| Debounce times for the digital inputs<br>▸ Digital input terminals | | |
| **Selection list** | | |
| 0 | 0.00 ms | |
| 1 | 0.25 ms | |
| 2 | 0.50 ms | |
| 3 | 0.75 ms | |
| 4 | 1.00 ms | |
| 5 | 1.25 ms | |
| 6 | 1.50 ms | |
| 7 | 1.75 ms | |
| 8 | 2.00 ms | |
| 10 | 2.50 ms | |
| 12 | 3.00 ms | |
| 14 | 3.50 ms | |
| 16 | 4.00 ms | |
| 18 | 4.50 ms | |
| 20 | 5.00 ms | |
| 22 | 5.50 ms | |
| 24 | 6.00 ms | |
| 28 | 7.00 ms | |
| 32 | 8.00 ms | |
| 36 | 9.00 ms | |
| 40 | 10.0 ms | |
| 44 | 11.0 ms | |
| 48 | 12.0 ms | |
| 52 | 13.0 ms | |
| 56 | 14.0 ms | |
| 64 | 16.0 ms | |
| 72 | 18.0 ms | |
| 80 | 20.0 ms | |
| 88 | 22.0 ms | |
| 96 | 24.0 ms | |
| 104 | 26.0 ms | |
| 112 | 28.0 ms | |
| 120 | 30.0 ms | |
| 128 | 32.0 ms | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02830/1 | 1: 0.25 ms | DI1 ... DI7: Debounce time |
| C02830/... | | |
| C02830/7 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02840**

| Parameter \| Name: **C02840 \| CountInx: Parameter** | | | Data type: UNSIGNED_32 Index: $21735_d = 54E7_h$ |
|---|---|---|---|
| Starting and comparison values for digital count inputs ▸ Use DI1(6) as counting input | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02840/1 | 0 incr. | | CountIn1: Starting value |
| C02840/2 | 65535 incr. | | CountIn1: Comparison value |
| C02840/3 | 0 incr. | | CountIn6: Starting value |
| C02840/4 | 65535 incr. | | CountIn6: Comparison value |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02841**

| Parameter \| Name: **C02841 \| CountInx: Counter content** | | | Data type: UNSIGNED_32 Index: $21734_d = 54E6_h$ |
|---|---|---|---|
| Display of the current counter content of the digital count inputs ▸ Use DI1(6) as counting input | | | |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 2147483647 | |
| **Subcodes** | | | **Info** |
| C02841/1 | | | CountIn1: Counter content |
| C02841/2 | | | CountIn6: Counter content |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02842**

| Parameter \| Name: **C02842 \| FreqInxx: Offset** | | | Data type: INTEGER_16 Index: $21733_d = 54E5_h$ |
|---|---|---|---|
| Offset for digital frequency inputs ▸ Using DI1(6) and DI2(7) as frequency inputs | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02842/1 | 0.00 % | | FreqIn12: Offset |
| C02842/2 | 0.00 % | | FreqIn67: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02843**

| Parameter \| Name: **C02843 \| FreqInxx: Gain** | | | Data type: INTEGER_16 Index: $21732_d = 54E4_h$ |
|---|---|---|---|
| Gain for digital frequency inputs ▸ Using DI1(6) and DI2(7) as frequency inputs | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02843/1 | 100.00 % | | FreqIn12: Gain |
| C02843/2 | 100.00 % | | FreqIn67: Gain |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02844**

| Parameter \| Name: **C02844 \| FreqIn12: Function** | | Data type: UNSIGNED_8 Index: $21731_d = 54E3_h$ |
|---|---|---|
| ▸ Output of the encoder position of the DI1/DI2 frequency input | | |
| **Selection list** | | |
| 0 | Loading with level | |
| 1 | Loading with edge | |
| 2 | Loading with level + reset | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02844/1 | 0: Loading with level | FreqIn12: PosIn function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02845**

| Parameter \| Name: **C02845 \| FreqIn12: PosIn comparison value** | | | Data type: INTEGER_32 Index: $21730_d = 54E2_h$ |
|---|---|---|---|
| ▸ Output of the encoder position of the DI1/DI2 frequency input | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | Incr. | 2147418112 | **0 incr.** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02853**

| Parameter \| Name: **C02853 \| PSM: Lss saturation characteristic** | | | Data type: UNSIGNED_8 Index: $21722_d = 54DA_h$ |
|---|---|---|---|
| | | | ▸ Current-dependent stator leakage inductance Lss(I) |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | % | 255 | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C02853/1 | 100 % | PSM: Lss saturation characteristic |
| C02853/2 | 100 % | PSM: Lss saturation characteristic |
| C02853/3 | 100 % | PSM: Lss saturation characteristic |
| C02853/4 | 100 % | PSM: Lss saturation characteristic |
| C02853/5 | 100 % | PSM: Lss saturation characteristic |
| C02853/6 | 100 % | PSM: Lss saturation characteristic |
| C02853/7 | 100 % | PSM: Lss saturation characteristic |
| C02853/8 | 100 % | PSM: Lss saturation characteristic |
| C02853/9 | 100 % | PSM: Lss saturation characteristic |
| C02853/10 | 100 % | PSM: Lss saturation characteristic |
| C02853/11 | 100 % | PSM: Lss saturation characteristic |
| C02853/12 | 100 % | PSM: Lss saturation characteristic |
| C02853/13 | 100 % | PSM: Lss saturation characteristic |
| C02853/14 | 100 % | PSM: Lss saturation characteristic |
| C02853/15 | 100 % | PSM: Lss saturation characteristic |
| C02853/16 | 100 % | PSM: Lss saturation characteristic |
| C02853/17 | 100 % | PSM: Lss saturation characteristic |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C02855**

| Parameter \| Name: **C02855 \| PSM: Imax Lss saturation characteristic** | | | Data type: UNSIGNED_16 Index: $21720_d = 54D8_h$ |
|---|---|---|---|
| | | | ▸ Current-dependent stator leakage inductance Lss(I) |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | A | 3000.0 | **3000.0 A** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10

**C02859**

| Parameter \| Name: **C02859 \| PSM: Activate Lss saturation char.** | Data type: UNSIGNED_8 Index: $21716_d = 54D4_h$ |
|---|---|
| ▸ Current-dependent stator leakage inductance Lss(I) | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **Off** |
| 1 | On |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C02862**

| Parameter \| Name: C02862 \| Resolver gain | | | Data type: UNSIGNED_16 Index: $21713_d = 54D1_h$ |
|---|---|---|---|
| ▸ Encoder/feedback system: Resolver | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02862/1 | 100.00 % | | Resolver: sine gain |
| C02862/2 | 100.00 % | | Resolver: cos gain |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02863**

| Parameter \| Name: C02863 \| Resolver: Phase error | | | Data type: INTEGER_16 Index: $21712_d = 54D0_h$ |
|---|---|---|---|
| ▸ Encoder/feedback system: Resolver | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02864**

| Parameter \| Name: C02864 \| MCTRL: Optimisations | | | Data type: UNSIGNED_16 Index: $21711_d = 54CF_h$ |
|---|---|---|---|
| From version 14.00.00<br>Activation of special functions of the internal motor control | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Motor temp. 90° for operation without KTY | | 1 ≡ Warm machine. In case the temperature tracking is not active, a motor temperature of 90 °C is assumed.<br>▸ Temperature compensation within the motor control |
| Bit 1 | No switch-off of the brake chopper after 4s | | 1 ≡ Limitation of the operating time of the brake chopper (brake transistor) is deactivated.<br>▸ Braking operation/brake energy management |
| Bit 2 | Brake chopper off in case of trip | | 0 ≡ No response of the brake chopper in the "Fault" device status.<br>1 ≡ In the "Fault" device status, the brake chopper is permanently switched off without any delay.<br>▸ Braking operation/brake energy management |
| Bit 3 | Reserved (do not change!) | | |
| Bit 4 | Motor ident: Switching frequency 8kHz | | Switching frequency for motor parameter identification:<br>0 ≡ 4 kHz<br>1 ≡ 8 kHz<br>▸ Preventing a decrease of the switching frequency |
| Bit 5 | Field/field weakening control in 500us | | Cycle time for field weakening control:<br>0 ≡ 1 ms<br>1 ≡ 500 µs<br>▸ Optimising the behaviour of the asynchronous motor in the field weakening range |

| Parameter \| Name:<br>**C02864 \| MCTRL: Optimisations** | | Data type: UNSIGNED_16<br>Index: $21711_d = 54CF_h$ |
|---|---|---|
| Bit 6 | Motor ident.: No calculation C577 C578 C2864 | From version 15.00.00: Following successful motor parameter identification, the field weakening controller parameters for ASM servo control are calculated as well. If these parameters are not to be calculated, bit 6 must be set to "1".<br>0 ≡ Calculate parameter<br>1 ≡ Do not calculate parameter<br>▸ Automatic motor data identification |
| Bit 7 | If C2879 Bit1=0: C087 is warm | From version 15.00.00, the too high slip with the activated option "Slip calculation from motor nameplate data" can be corrected for the ASM servo control by setting bit 7 to "1".<br>1 ≡ Rated motor speed (C00087) is assumed for a warm machine.<br>▸ Slip calculation for SC |
| Bit 8 | Suppress V/f emergency operation | From version 15.00.00, it is internally switched to the encoderless V/f characteristic control in case of a motor control type with feedback and an encoder open circuit in order to avoid impermissible motor movements.<br>• **Exception:** There will be no change-over in case of servo control for synchronous motors (PSM).<br>• In order that this "V/f emergency operation" works properly, the parameters relevant for the V/f characteristic control (base frequency, Vmin boost, slip compensation, etc.) have to be set correctly. As an alternative, a motor parameter identification can be executed as well.<br>• The change-over to "V/f emergency operation" is reported via bit 4 in C01000 and via the *bWirebreakUfLinearActive* status signal at the SB LS_DeviceMonitor.<br>• The setting in C00006 is not affected by the change-over.<br>The "V/f emergency operation" can be suppressed by setting bit 8 to "1".<br>0 ≡ V/f emergency operation in case of encoder open circuit<br>1 ≡ No V/f emergency operation in case of encoder open circuit |
| Bit 9 | Optimisation of field weakening control | |
| Bit 10 | V/f+encoder without adaptation of voltage | |
| Bit 11 | Latching of open circuit FreqIn12 or 67 | |
| Bit 12 | SLVC: No smoothing of nMotorTorqueAct_a | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02864/1 | 0x0000 | MCTRL: Optimisations |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT | | |

**C02865**

| Parameter \| Name: **C02865 \| MCTRL: Special settings** | | | Data type: UNSIGNED_16 Index: $21710_d = 54CE_h$ |
|---|---|---|---|
| From version 12.00.00<br>Activation of special functions of the internal motor control | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | No TorqueLimit if TorquemodeOn | | From version 12.00.00 onwards: If sensorless vector control (SLVC) or servo control (SC) are used, in the "torque control with speed limitation" mode the torque limitation via *nTorqueMotLimit_a* and *nTorqueGenLimit_a* is also active to limit the torque setpoint.<br>If you want to deactivate the torque limitation to retain the former function, set bit 0 to "1".<br>0 ≡ Torque limitation is active<br>1 ≡ Torque limitation is not active |
| Bit 1 | 250ms IMP before DCB | | |
| Bit 2 | 1000ms IMP before DCB | | |
| Bit 3 | DCB: Current controller and C036 app. to C088 | | |
| Bit 4 | Motorident.: No calculation C73 C75 C76 C77 C78 | | From version 12.00.00 onwards: Following successful motor parameter identification, the current controller and field controller parameters are calculated as well.<br>If these parameters are not to be calculated, bit 4 must be set to "1".<br>0 ≡ Calculate parameter<br>1 ≡ Do not calculate parameter<br>▸ Automatic motor data identification |
| Bit 5 | Motorident.: Calculation C70 C71 C72 | | From version 12.00.00 onwards: Following successful motor parameter identification, the speed controller parameters can automatically be calculated as well.<br>If these parameters are to be calculated, bit 5 must be set to "1".<br>0 ≡ Do not calculate parameter<br>1 ≡ Calculate parameter<br>▸ Automatic motor data identification |
| Bit 6 | Motorident.: Calculation C11 C22 C497 C966 C982 | | From version 12.00.00 onwards: Following successful motor parameter identification, further controller parameters can be calculated automatically.<br>If these parameters are to be calculated, bit 6 must be set to "1".<br>0 ≡ Do not calculate parameter<br>1 ≡ Calculate parameter<br>▸ Automatic motor data identification |
| Bit 7 | Brake chopper off in case of pulse inhibit | | From version 14.00.00<br>0 ≡ No response of the brake chopper in case of pulse inhibit.<br>1 ≡ In case of pulse inhibit, the brake chopper is permanently switched off without any delay.<br>▸ Braking operation/brake energy management |
| Bit 8 | DC-bus supply with DC charging connection | | |

| Parameter \| Name: **C02865 \| MCTRL: Special settings** | | Data type: UNSIGNED_16 Index: $21710_d = 54CE_h$ |
|---|---|---|
| Bit 9 | Inverter characteristics inactive | From version 12.00.00<br>In case of very low-inductance motors (e.g. synchronous motor with a rated frequency of 1000 Hz) the inverter error characteristic needs to be switched off as otherwise the current of the device might be switched off.<br>0 ≡ Inverter error characteristic is active<br>1 ≡ Inverter error characteristic is not active |
| Bit 10 | SCPSM: Extrapolation - actual speed value inactive | |
| Bit 11 | TopLine: Speed encoder signal cycle 500us | |
| Bit 12 | In case of QSP - nTorqueSetValue_a active | From version 13.00.00 |
| Bit 13 | In case of QSP - nTorquexxxLimit_a active | From version 13.00.00 |
| Bit 14 | Sensitivity of setpoint feedf. ctrl. 16 bits | From version 14.00.00 |
| Bit 15 | SC: Dead time of speed setpoint 500us | From version 14.00.00 |
| **Subcodes** | **Lenze setting** | **Info** |
| C02865/1 | 0x0000 | MCTRL: Special settings |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT | | |

**C02866**

| Parameter \| Name: **C02866 \| MCTRL: Special settings** | | Data type: UNSIGNED_8 Index: $21709_d = 54CD_h$ |
|---|---|---|
| Activation of special functions of the internal motor control | | |
| **Selection list** | | |
| 0 | No | |
| 1 | Yes | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02866/1 | 1: Yes | Motor ident.: Current controller par. from C075 C076<br>▸ Automatic motor data identification |
| C02866/2 | 0: No | Motor phase error monitoring before operation<br>• From version 02.00.00<br>▸ Motor phase error monitoring before operation |
| C02866/3 | 0: No | Brake chopper DC-bus slave<br>• From version 12.00.00<br>▸ Control of multiple internal brake choppers in the DC-bus system |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02867**

| Parameter \| Name: **C02867 - Identification procedure** | | Data type: UNSIGNED_8 Index: $21708_d = 54CC_h$ |
|---|---|---|
| Selection of the identification procedure for motor parameter identification<br>▸ Automatic motor parameter identification | | |
| **Selection list** | | **Info** |
| 0 | automatic | Automatic selection of the optimum identification procedure:<br>• For synchronous motors, the extended identification procedure is always used.<br>• For asynchronous motors with a rated motor power of up to 11 kW, the basic identification procedure is used.<br>• For asynchronous motors with a rated motor power of more than 11 kW, the extended identification procedure is used. |
| 1 | standard identification | • Only for asynchronous motors.<br>• Duration approx. 30 s |
| 2 | extended identification | • Stands out due to increased accuracy of the determined motor parameters.<br>• Also supports synchronous motors and asynchronous motors with a power of more than 11 kW.<br>• Duration approx. 80 s |
| **Subcodes** | **Lenze setting** | **Info** |
| C02867/1 | 0: automatic | Motor parameter identification: Process |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02868**

| Parameter \| Name: **C02868 \| Setting of compatibility** | | | Data type: UNSIGNED_32 Index: $21707_d = 54CB_h$ |
|---|---|---|---|
| From version 16.00.00<br>This parameter serves to activate corrections and optimisations in various functions of the inverter. The activation causes a deviation of the respective functionality from the previous firmware versions!<br>In each case, the changed functionality has to be checked after the activation took place.<br>For a detailed explanation, the version information in the AKB has to be used. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x00000000 | | 0xFFFFFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | xx.01: MCK stop ramp opt. | | 0 ≡ (Maloperation): Profile stop activation with corresponding machine and profile data causes a rotation in the opposite direction and setpoint oscillations around zero speed.<br>1 ≡ (Correction): Corrected traversing of stop ramp may reduce the stopping times. |
| Bit 1 | xx.02: MCK deceleration ramp opt. | | 0 ≡ (Maloperation): When corresponding profile parameters are entered, an impermissible jerk is caused during the braking phase when entering into the target position within a traversing profile.<br>1 ≡ (Correction): Depending on the profile data constellation, corrected traversing extends the respective traversing process towards the target position. |
| Bit 2 | xx.03: MCK Overchange opt. | | 0 ≡ (error CK10 occurs):<br>• Traversing profile with final speed ≠ 0 needs a reversing process after reaching the target position.<br>• Parameterised final speed in profile parameters cannot be reached at the target approach (final speed too high).<br>• Position target cannot be approached if<br>  • the initial speed of the traversing profile ± final speed<br>1 ≡ (Correction): Ck10 does not occur anymore in this setting.<br>• Travelling profiles with final speed is always executed in such a way that the target position will be reached if the parameterised final speed is neglected.<br>• Expected Ck10 errors do not occur anymore, traversing blocks are executed. |
| Bit 3 | xx.04: MCK PosFollower TargetPos | | 0 ≡ (Previous procedure):<br>• No information on target position *LS_MotionControlKernel.dnPosTarget_p* when a synchronisation process is started in the position follower mode. Previous display of the target position remains unchanged.<br>1 ≡ Display of the target position and thus the rejection of the previously saved values in *LS_MotionControlKernel.dnPosTarget_p*. |
| Bit 4 | xx.05: Reserved | | |
| Bit 5 | xx.06: Reserved | | |
| Bit 6 | xx.07: Reserved | | |
| Bit 7 | xx.08: Reserved | | |
| Bit 8 | xx.09: Reserved | | |
| Bit 9 | xx.10: Reserved | | |
| Bit 10 | xx.11: Reserved | | |
| Bit 11 | xx.12: Reserved | | |

| Parameter \| Name: **C02868 \| Setting of compatibility** | | Data type: UNSIGNED_32 Index: $21707_d = 54CB_h$ |
|---|---|---|
| Bit 12 | xx.13: Reserved | |
| Bit 13 | xx.14: Reserved | |
| Bit 14 | xx.15: Reserved | |
| Bit 15 | xx.16: Reserved | |
| Bit 16 | xx.17: Reserved | |
| Bit 17 | xx.18: Reserved | |
| Bit 18 | xx.19: Reserved | |
| Bit 19 | xx.20: Reserved | |
| Bit 20 | xx.21: Reserved | |
| Bit 21 | xx.22: Reserved | |
| Bit 22 | xx.23: Reserved | |
| Bit 23 | xx.24: Reserved | |
| Bit 24 | xx.25: Reserved | |
| Bit 25 | xx.26: Reserved | |
| Bit 26 | xx.27: Reserved | |
| Bit 27 | xx.28: Reserved | |
| Bit 28 | xx.29: Reserved | |
| Bit 29 | xx.30: Reserved | |
| Bit 30 | xx.31: Reserved | |
| Bit 31 | xx.32: Reserved | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02868/1 | 0x00000000 | List of modifications 01 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C02870**

| Parameter \| Name: **C02870 \| PLI without motion: Optimisation factor** | | | Data type: INTEGER_16 Index: $21705_d = 54C9_h$ |
|---|---|---|---|
| From version 02.00.00 | | | ▸ Pole position identification without motion |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 300.00 | |
| **Subcodes** | | | **Info** |
| C02870/1 | | | PLI without motion: Optimisation factor |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02871**

| Parameter \| Name: **C02871 \| PLI without motion: Running time** | | | Data type: INTEGER_16 Index: $21704_d = 54C8_h$ |
|---|---|---|---|
| From version 02.00.00 | | | ▸ Pole position identification without motion |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0.00 | ms | 300.00 | |
| **Subcodes** | | | **Info** |
| C02871/1 | | | PLI without motion: Running time |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02872**

| Parameter \| Name: **C02872 \| PLI without motion: Adaptation of time duration** | | | Data type: INTEGER_8 Index: $21703_d = 54C7_h$ |
|---|---|---|---|
| From version 02.00.00 | | | ▸ Pole position identification without motion |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -10 | | 10 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02872/1 | 0 | | PLI without motion: Adaptation of time duration |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02873**

| Parameter \| Name: **C02873 \| PLI without motion: Ident. el. rotor displ. angle** | | | Data type: INTEGER_16 Index: $21702_d = 54C6_h$ |
|---|---|---|---|
| From version 02.00.00 | | | ▸ Pole position identification without motion |
| **Display area** (min. value \| unit \| max. value) | | | |
| 0 | ° | 360 | |
| **Subcodes** | | | **Info** |
| C02873/1 | | | PLI without motion: Ident. el. rotor displ. angle |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02874**

| Parameter \| Name: **C02874 \| PLI without motion** | | Data type: UNSIGNED_16 Index: $21701_d = 54C5_h$ |
|---|---|---|
| From version 02.00.00 | | ▸ Pole position identification without motion |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** | | **Info** |
| Bit 0 | for SLPSM with controller enable | |
| Bit 1 | for SC PSM with mains on | |
| Bit 2 | for SC PSM with controller enable | |
| Bit 3 | for SC PSM once after fault reset | From version 14.00.00 |
| Bit 4 | Reserved | |
| Bit 5 | Reserved | |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02874/1 | 0x0001 | PLI without motion |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT | | |

**C02875**

| Parameter \| Name: **C02875 \| PLI without motion: Adaptation of ident angle** | | Data type: INTEGER_8 Index: $21700_d = 54C4_h$ |
|---|---|---|
| From version 02.00.00 | | ▸ Pole position identification without motion |
| **Setting range** (min. value \| unit \| max. value) | | |
| -100 | ° | 100 |
| **Subcodes** | **Lenze setting** | **Info** |
| C02875/1 | 0 ° | PLI without motion: Adaptation of ident angle |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02876**

| Parameter \| Name: **C02876 \| PSM: Max. motor temperature** | | | Data type: UNSIGNED_8 Index: $21699_d = 54C3_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 90 | °C | 200 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02876/1 | 150 °C | | PSM: Max. motor temperature |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02877**

| Parameter \| Name: **C02877 \| PSM temperature coefficient** | | | Data type: INTEGER_16 Index: $21698_d = 54C2_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -0.30 | %/°C | 0.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02877/1 | -0.11 %/°C | | PSM temperature coefficient |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C02878**

| Parameter \| Name: **C02878 \| KTY motor temperature compensation** | | | Data type: UNSIGNED_16 Index: $21697_d = 54C1_h$ |
|---|---|---|---|
| From version 02.00.00<br>A temperature compensation over the detected motor temperature (display in C00063/1) serves to compensate inaccuracies in the output torque within the motor control in case of temperature changes of the asynchronous or synchronous motor.<br>**Note!**<br>In the Lenze setting, the temperature compensation within the motor control is activated. The temperature compensation, however, is only active with speed encoder selection "3: Multi encoder" or "4: Resolver" in C00495 as well as error-free KTY temperature detection (display in C00063/1 ≠ 255 °C).<br>▸ Temperature compensation within the motor control | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | for SC PSM | | |
| Bit 1 | for SC ASM | | |
| Bit 2 | Reserved | | |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02878/1 | 0x0003 | | KTY motor temperature compensation |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT | | | |

**C02879**

| Parameter \| Name: C02879 \| Slip calculation from equivalent circuit diagram | | | Data type: UNSIGNED_16 Index: $21696_d = 54C0_h$ |
|---|---|---|---|
| From version 02.00.00<br>In order to achieve a better speed stability and torque accuracy, the slip calculation can be either derived from the motor nameplate data (e.g. rated motor speed) or the motor equivalent circuit diagram data (stator resistance, rotor resistance etc.). | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | SLVC | | ▸ Slip calculation for SLVC |
| Bit 1 | SC ASM | | ▸ Slip calculation for SC |
| Bit 2 | Reserved | | |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02879/1 | 0x0002 | | Slip calculation from equivalent circuit diagram |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT | | | |

**C02993**

| Parameter \| Name: C02993 \| FB xy position | Data type: UNSIGNED_32 Index: $21582_d = 544E_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C02994**

| Parameter \| Name: C02994 \| FB xy position | Data type: UNSIGNED_32 Index: $21581_d = 544D_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C02995**

| Parameter \| Name: C02995 \| FB display InputOutput | Data type: UNSIGNED_32 Index: $21580_d = 544C_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C02996**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C02996 \| FB display InputOutput2** | Index: $21579_d = 544B_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

**C02998**

| Parameter \| Name: | Data type: UNSIGNED_32 |
|---|---|
| **C02998 \| FB displayinputoutput3** | Index: $21577_d = 5449_h$ |
| **This code is for device-internal use only and must not be written to by the user!** | |

## 17.3 Overview of all parameters with power-related Lenze setting

| Parameters | Name | Unit | Mains voltage / device power | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 * 230 V | | | | | | | 3 * 400 V | | | | |
| | | | 0.25 kW | 0.37 kW | 0.55 kW | 0.75 kW | 1.1 kW | 1.5 kW | 2.2 kW | 0.37 kW | 0.55 kW | 0.75 kW | 1.1 kW | 1.5 kW |
| C00016 | VFC: Vmin boost | % | 5.51 | 4.32 | 4.02 | 3.79 | 4.32 | 2.93 | 2.25 | 4.32 | 4.02 | 3.79 | 4.32 | 2.93 |
| C00021 | Slip comp. | % | 8.67 | 6 | 6.33 | 6 | 7.33 | 7 | 4 | 6 | 6.33 | 6 | 7.33 | 7 |
| C00022 | Imax in motor mode | A | 2.97 | 4.2 | 5.25 | 7 | 9.62 | 12.25 | 16.62 | 2.27 | 3.15 | 4.2 | 5.6 | 6.82 |
| C00070/1 | SLVC: Vp speed controller | | 7.62 | 15.76 | 12.59 | 18.9 | 12.52 | 12.45 | 15.76 | 15.76 | 12.59 | 18.9 | 12.52 | 12.45 |
| C00070/2 | SC: Vp speed controller | | 4.35 | 9.01 | 7.19 | 10.8 | 7.15 | 7.11 | 9.01 | 9.01 | 7.19 | 10.8 | 7.15 | 7.11 |
| C00070/3 | SLPSM: Vp speed controller | | 1.09 | 2.25 | 1.79 | 2.7 | 1.79 | 1.78 | 2.25 | 2.25 | 1.8 | 2.7 | 1.79 | 1.78 |
| C00073/1 | VFC: Vp Imax controller | | 0.46 | 0.32 | 0.33 | 0.27 | 0.33 | 0.32 | 0.2 | 0.28 | 0.3 | 0.27 | 0.33 | 0.32 |
| C00074/1 | VFC: Ti Imax controller | ms | 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 |
| C00075 | Vp current controller | V/A | 41.88 | 34.59 | 23.43 | 14.97 | 9.53 | 7.23 | 4.73 | 103.76 | 70.29 | 44.9 | 28.54 | 21.7 |
| C00076 | Ti current controller | ms | 1.99 | 2.42 | 2.6 | 2.39 | 1.97 | 2.83 | 3.55 | 2.42 | 2.6 | 2.39 | 1.97 | 2.83 |
| C00077 | SC: Vp field controller | | 1.68 | 2.83 | 3.32 | 4.92 | 3.96 | 4.71 | 5.43 | 2.83 | 3.32 | 4.92 | 3.96 | 4.71 |
| C00078 | SC: Tn field controller | ms | 33.6 | 56.6 | 66.3 | 98.3 | 79.1 | 94.2 | 108.6 | 56.6 | 66.3 | 98.3 | 79.1 | 94.2 |
| C00081 | Rated motor power | kW | 0.25 | 0.37 | 0.55 | 0.75 | 1.1 | 1.5 | 2.2 | 0.37 | 0.55 | 0.75 | 1.1 | 1.5 |
| C00082 | Motor rotor resistance | mOhm | 11460 | 5480 | 3860 | 2820 | 2060 | 1438 | 740 | 16442 | 11580 | 8454 | 6150 | 4320 |
| C00084 | Motor stator resistance | mOhm | 10533 | 7133 | 4500 | 3127 | 2420 | 1277 | 667 | 21400 | 13500 | 9380 | 7260 | 3830 |
| C00085 | Motor stator leakage inductance | mH | 20.94 | 17.29 | 11.72 | 7.48 | 4.77 | 3.62 | 2.37 | 51.88 | 35.15 | 22.45 | 14.27 | 10.85 |
| C00087 | Rated motor speed | rpm | 1370 | 1410 | 1405 | 1410 | 1390 | 1395 | 1440 | 1410 | 1405 | 1410 | 1390 | 1395 |
| C00088 | Rated motor current | A | 1.4 | 1.6 | 2.4 | 3.3 | 4.8 | 6.3 | 9.2 | 0.95 | 1.4 | 1.9 | 2.8 | 3.6 |
| C00090 | Rated motor voltage | V | 230 | 230 | 230 | 230 | 230 | 230 | 230 | 400 | 400 | 400 | 400 | 400 |
| C00091 | Motor cosine phi | | 0.63 | 0.72 | 0.72 | 0.8 | 0.8 | 0.79 | 0.71 | 0.77 | 0.77 | 0.8 | 0.8 | 0.79 |
| C00092 | Motor magnetising inductance | mH | 333.3 | 386.7 | 286.7 | 300 | 186.7 | 116.7 | 70 | 1160 | 860 | 900 | 560 | 350 |
| C00093 | Power section ID | | 2512 | 3712 | 5512 | 7512 | 1122 | 1522 | 2222 | 3714 | 5514 | 7514 | 1124 | 1524 |
| C00095 | Motor magnetising current | A | 0.94 | 0.88 | 1.35 | 1.28 | 2.25 | 3.46 | 5.54 | 0.51 | 0.78 | 0.74 | 1.3 | 2 |
| C00098 | Rated device current | A | 1.7 | 2.4 | 3 | 4 | 5.5 | 7 | 9.5 | 1.3 | 1.8 | 2.4 | 3.2 | 3.9 |
| C00129 | Brake resistance value | Ohm | 180 | 180 | 100 | 100 | 33 | 33 | 33 | 390 | 390 | 390 | 180 | 180 |
| C00130 | Rated brake resistor power | W | 50 | 50 | 100 | 100 | 200 | 200 | 300 | 100 | 100 | 100 | 200 | 200 |
| C00131 | Rated heat amount of brake resist. | kWs | 7.5 | 7.5 | 15 | 15 | 30 | 30 | 45 | 15 | 15 | 15 | 30 | 30 |
| C00497 | Filter time constant | ms | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| C00916 | Motor cable cross-section | mm^2 | 1 | 1 | 1 | 1 | 1.5 | 1.5 | 1.5 | 1 | 1 | 1 | 1.5 | 1.5 |
| C00966 | VFC: Time const. slip comp. | ms | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 109 | 100 | 100 |
| C00982 | VFC-ECO: Voltage reduction ramp | s | 0.1 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 |
| C00987 | Inverter motor brake: nAdd | rpm | 260 | 180 | 190 | 180 | 220 | 210 | 120 | 180 | 190 | 180 | 220 | 210 |
| C00993 | Flying restart: Integration time | ms | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 |

| Parameters | Name | Unit | Mains voltage / device power | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3 * 400 V | | | | | | | | | | | |
| | | | 2.2 kW | 3.0 kW | 4.0 kW | 5.5 kW | 7.5 kW | 11 kW | 15 kW | 18 kW | 22 kW | 30 kW | 37 kW | 45 kW |
| C00016 | VFC: Vmin boost | % | 2.25 | 2.48 | 2.06 | 1.94 | 1.7 | 1.47 | 1.36 | 1.32 | 1.24 | 0.97 | 0.7 | 0.59 |
| C00021 | Slip comp. | % | 4 | 4.67 | 3.33 | 3.67 | 3 | 2.67 | 2.67 | 2 | 2.93 | 2.33 | 1.13 | 1.33 |
| C00022 | Imax in motor mode | A | 9.8 | 12.77 | 16.62 | 22.75 | 28.87 | 41.12 | 56 | 68.25 | 82.25 | 103.25 | 126 | 155.75 |
| C00070/1 | SLVC: Vp speed controller | | 15.76 | 11.4 | 15.42 | 14.05 | 26 | 32.41 | 29.22 | 43.23 | 40.95 | 36.28 | 75.12 | 72.06 |
| C00070/2 | SC: Vp speed controller | | 9.01 | 6.51 | 8.81 | 8.03 | 14.86 | 18.52 | 16.7 | 24.7 | 23.4 | 20.73 | 42.93 | 41.18 |
| C00070/3 | SLPSM: Vp speed controller | | 2.25 | 1.63 | 2.2 | 2.01 | 3.72 | 4.63 | 4.18 | 6.18 | 5.85 | 5.18 | 10.73 | 10.3 |
| C00073/1 | VFC: Vp Imax controller | | 0.2 | 0.23 | 0.17 | 0.17 | 0.14 | 0.11 | 0.11 | 0.08 | 0.12 | 0.1 | 0.09 | 0.09 |
| C00074/1 | VFC: Ti Imax controller | ms | 65 | 65 | 65 | 65 | 65 | 750 | 750 | 750 | 750 | 750 | 750 | 750 |
| C00075 | Vp current controller | V/A | 17 | 14.2 | 11.1 | 6.97 | 8.1 | 7 | 5.5 | 4 | 3.8 | 1.8 | 1.49 | 1.06 |
| C00076 | Ti current controller | ms | 4.25 | 4.38 | 5.34 | 4.77 | 8.62 | 10.61 | 11.96 | 10.53 | 12.67 | 10.59 | 14.57 | 15.17 |
| C00077 | SC: Vp field controller | | 5.46 | 5.4 | 8.44 | 7.29 | 7.67 | 12.8 | 14.84 | 19.74 | 21.47 | 31.45 | 35.17 | 35.59 |
| C00078 | SC: Tn field controller | ms | 109.2 | 108.1 | 168.8 | 145.9 | 153.3 | 256 | 296.7 | 394.7 | 429.3 | 500 | 997 | 859 |
| C00081 | Rated motor power | kW | 2.2 | 3 | 4 | 5.5 | 7.5 | 11 | 15 | 18.5 | 22 | 30 | 37 | 45 |
| C00082 | Motor rotor resistance | mOhm | 2220 | 1938 | 1262 | 842 | 642 | 276 | 186 | 168 | 130 | 111 | 36 | 29 |
| C00084 | Motor stator resistance | mOhm | 2000 | 1620 | 1040 | 730 | 470 | 330 | 230 | 190 | 150 | 85 | 51 | 35 |
| C00085 | Motor stator leakage inductance | mH | 8.5 | 7.1 | 5.55 | 3.49 | 4.05 | 3.5 | 2.75 | 2 | 1.9 | 0.9 | 0.74 | 0.53 |
| C00087 | Rated motor speed | rpm | 1440 | 1430 | 1450 | 1445 | 1455 | 1460 | 1460 | 1470 | 1456 | 1465 | 1483 | 1480 |
| C00088 | Rated motor current | A | 5.3 | 7.2 | 9.3 | 12.5 | 17 | 21 | 27.8 | 32.8 | 38.8 | 53.9 | 65 | 79 |
| C00090 | Rated motor voltage | V | 400 | 400 | 400 | 400 | 400 | 400 | 400 | 400 | 400 | 400 | 400 | 400 |
| C00091 | Motor cosine phi | | 0.73 | 0.75 | 0.73 | 0.77 | 0.76 | 0.85 | 0.87 | 0.9 | 0.9 | 0.87 | 0.87 | 0.88 |
| C00092 | Motor magnetising inductance | mH | 210 | 168 | 170 | 103 | 68 | 81 | 65.5 | 73 | 62.5 | 52.6 | 35.1 | 24.4 |
| C00093 | Power section ID | | 2224 | 3024 | 4024 | 5524 | 7524 | 1134 | 1534 | 1834 | 2234 | 3034 | 3734 | 4534 |
| C00095 | Motor magnetising current | A | 3.2 | 4 | 4.1 | 6.7 | 10.2 | 8.5 | 10.5 | 9.5 | 11.1 | 13.2 | 20.5 | 29.5 |
| C00098 | Rated device current | A | 5.6 | 7.3 | 9.5 | 13 | 16.5 | 23.5 | 32 | 39 | 47 | 59 | 72 | 89 |
| C00129 | Brake resistance value | Ohm | 180 | 82 | 47 | 47 | 27 | 27 | 18 | 15 | 15 | 7.5 | 7.5 | 7.5 |
| C00130 | Rated brake resistor power | W | 300 | 200 | 400 | 800 | 600 | 1200 | 1400 | 1200 | 1200 | 1900 | 1900 | 1900 |
| C00131 | Rated heat amount of brake resist. | kWs | 45 | 30 | 60 | 120 | 120 | 90 | 210 | 180 | 180 | 285 | 285 | 285 |
| C00497 | Filter time constant | ms | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| C00916 | Motor cable cross-section | mm^2 | 1.5 | 2.5 | 2.5 | 2.5 | 4 | 6 | 6 | 6 | 6 | 10 | 10 | 10 |
| C00966 | VFC: Time const. slip comp. | ms | 100 | 100 | 139 | 126 | 112 | 306 | 367 | 446 | 495 | 489 | 1016 | 877 |
| C00982 | VFC-ECO: Voltage reduction ramp | s | 0.3 | 0.3 | 0.4 | 0.8 | 0.8 | 0.8 | 1.1 | 1.3 | 1.5 | 1.5 | 1.5 | 1.5 |
| C00987 | Inverter motor brake: nAdd | rpm | 120 | 140 | 100 | 110 | 90 | 80 | 80 | 60 | 88 | 70 | 60 | 60 |
| C00993 | Flying restart: Integration time | ms | 300 | 300 | 417.3 | 379.4 | 336.7 | 1159.4 | 1237.3 | 1387.3 | 1265.1 | 1445 | 2989.5 | 2577.3 |

## 17.4 Selection list - analog signals

This selection list is relevant for the following configuration parameters:

| Parameters | |
|---|---|
| C00410 | L_SignalMonitor_a: Signal sources |
| C00620 | System connection list: 16-bit |
| C00700 | LA_NCtrl: Analog connection list |
| C00710 | LA_TabPos: Analog connection list |
| C00760 | LA_SwitchPos: Analog connection list |

| Selection list - analog signals | |
|---|---|
| 0 | Not connected |
| 1000 | LA_NCtrl: wDriveControlStatus |
| 1001 | LA_NCtrl: wFailNumber |
| 1002 | LA_NCtrl: nMotorCurrent_a |
| 1003 | LA_NCtrl: nMotorSpeedAct_a |
| 1006 | LA_NCtrl: nGPAnalogSwitchOut_a |
| 1007 | LA_NCtrl: nGPArithmetikOut_a |
| 1008 | LA_NCtrl: nGPMulDivOut_a |
| 1009 | LA_NCtrl: nGPSignalOut1_a |
| 1010 | LA_NCtrl: nGPSignalOut2_a |
| 1011 | LA_NCtrl: nGPSignalOut3_a |
| 1012 | LA_NCtrl: nGPSignalOut4_a |
| 1013 | LA_NCtrl: nMotorTorqueAct_a |
| 1014 | LA_NCtrl: nDCVoltage_a |
| 1015 | LA_NCtrl: nMotorVoltage_a |
| 1016 | LA_NCtrl: nMotorSpeedSet_a |
| 1017 | LA_NCtrl: wFailTypeDomain |
| 1023 | LA_NCtrl: wFreeOut1 |
| 1024 | LA_NCtrl: wFreeOut2 |
| 1025 | LA_NCtrl: wFreeOut3 |
| 1026 | LA_NCtrl: wFreeOut4 |
| 1100 | LA_TabPos: wDriveControlStatus |
| 1101 | LA_TabPos: wFailNoLow |
| 1102 | LA_TabPos: wFailNoHigh |
| 1103 | LA_TabPos: nMotorCurrent_a |
| 1104 | LA_TabPos: nMotorSpeedSet_a |
| 1105 | LA_TabPos: nMotorSpeedAct_a |
| 1106 | LA_TabPos: nMotorTorqueAct_a |
| 1107 | LA_TabPos: nDCVoltage_a |
| 1108 | LA_TabPos: nMotorVoltage_a |
| 1109 | LA_TabPos: wMckState1 |
| 1110 | LA_TabPos: wMckState2 |
| 1111 | LA_TabPos: wMckActOperationMode |
| 1112 | LA_TabPos: wActProfileNo |
| 1113 | LA_TabPos: wActPosMode |
| 1114 | LA_TabPos: nGPAnalogSwitchOut_a |
| 1115 | LA_TabPos: nGPArithmetikOut_a |
| 1116 | LA_TabPos: nGPMulDivOut_a |
| 1117 | LA_TabPos: nGPSignalOut1_a |
| 1118 | LA_TabPos: nGPSignalOut2_a |

| Selection list - analog signals | |
|---|---|
| 1119 | LA_TabPos: nGPSignalOut3_a |
| 1120 | LA_TabPos: nGPSignalOut4_a |
| 1121 | LA_TabPos: wGPCounter1Out |
| 1122 | LA_TabPos: wFreeOut1 |
| 1123 | LA_TabPos: wFreeOut2 |
| 1124 | LA_TabPos: wFreeOut3 |
| 1125 | LA_TabPos: wFreeOut4 |
| 1126 | LA_TabPos: nPosCtrlOutLimit_a |
| 1127 | LA_TabPos: nPosCtrlPAdapt_a |
| 1128 | LA_TabPos: wPosOutUnitsLW |
| 1129 | LA_TabPos: wPosOutUnitsHW |
| 1200 | LA_SwitchPos: wDriveControlStatus |
| 1201 | LA_SwitchPos: wFailNoLow |
| 1202 | LA_SwitchPos: wFailNoHigh |
| 1203 | LA_SwitchPos: nMotorCurrent_a |
| 1204 | LA_SwitchPos: nMotorSpeedSet_a |
| 1205 | LA_SwitchPos: nMotorSpeedAct_a |
| 1206 | LA_SwitchPos: nMotorTorqueAct_a |
| 1207 | LA_SwitchPos: nDCVoltage_a |
| 1208 | LA_SwitchPos: nMotorVoltage_a |
| 1209 | LA_SwitchPos: nGPAnalogSwitchOut_a |
| 1210 | LA_SwitchPos: nGPArithmetikOut_a |
| 1211 | LA_SwitchPos: nGPMulDivOut_a |
| 1212 | LA_SwitchPos: nGPSignalOut1_a |
| 1213 | LA_SwitchPos: nGPSignalOut2_a |
| 1214 | LA_SwitchPos: nGPSignalOut3_a |
| 1215 | LA_SwitchPos: nGPSignalOut4_a |
| 1221 | LA_SwitchPos: wFreeOut1 |
| 1222 | LA_SwitchPos: wFreeOut2 |
| 1223 | LA_SwitchPos: wFreeOut3 |
| 1224 | LA_SwitchPos: wFreeOut4 |
| 16000 | LS_AnalogInput: nIn1_a |
| 16001 | LS_AnalogInput: nIn2_a |
| 16002 | LP_CanIn1: wCtrl |
| 16003 | LP_CanIn1: wIn2 |
| 16004 | LP_CanIn1: wIn3 |
| 16005 | LP_CanIn1: wIn4 |
| 16006 | LP_CanIn2: wIn1 |
| 16007 | LP_CanIn2: wIn2 |
| 16008 | LP_CanIn2: wIn3 |
| 16009 | LP_CanIn2: wIn4 |
| 16010 | LP_CanIn3: wIn1 |
| 16011 | LP_CanIn3: wIn2 |
| 16012 | LP_CanIn3: wIn3 |
| 16013 | LP_CanIn3: wIn4 |
| 16014 | LS_DigitalInput: wCountIn1_LW |
| 16015 | LS_DigitalInput: wCountIn1_HW |
| 16016 | LS_DigitalInput: nFreqIn12_a |
| 16017 | LS_DigitalInput: nFreqIn12_v |
| 16018 | LS_DigitalInput: wCountIn6_LW |

| Selection list - analog signals | |
|---|---|
| 16019 | LS_DigitalInput: wCountIn6_HW |
| 16020 | LS_DigitalInput: nFreqIn67_a |
| 16021 | LS_DigitalInput: nFreqIn67_v |
| 16100 | LS_DataAccess: wOut1 |
| 16101 | LS_DataAccess: wOut2 |
| 16102 | LS_DataAccess: wOut3 |
| 16103 | LS_DataAccess: wOut4 |
| 16104 | LP_MciIn:wCtrl |
| 16105 | LP_MciIn: wIn2 |
| 16106 | LP_MciIn: wIn3 |
| 16107 | LP_MciIn: wIn4 |
| 16108 | LP_MciIn: wIn5 |
| 16109 | LP_MciIn: wIn6 |
| 16110 | LP_MciIn: wIn7 |
| 16111 | LP_MciIn: wIn8 |
| 16112 | LP_MciIn: wIn9 |
| 16113 | LP_MciIn: wIn10 |
| 16114 | LP_MciIn: wIn11 |
| 16115 | LP_MciIn: wIn12 |
| 16116 | LP_MciIn: wIn13 |
| 16117 | LP_MciIn: wIn14 |
| 16118 | LP_MciIn: wIn15 |
| 16119 | LP_MciIn: wIn16 |
| 16120 | LS_Keypad: nTorqueMotLim_a |
| 16121 | LS_Keypad: nTorqueGenLim_a |
| 16122 | LS_Keypad: nMainSetValue_a |
| 16123 | LS_CANManagement: wNodeID |
| 16130 | LS_ParReadWrite_1: wOutHWord |
| 16131 | LS_ParReadWrite_1: wOutLWord |
| 16132 | LS_ParReadWrite_2: wOutHWord |
| 16133 | LS_ParReadWrite_2: wOutLWord |
| 16134 | LS_ParReadWrite_3: wOutHWord |
| 16135 | LS_ParReadWrite_3: wOutLWord |
| 16136 | LS_ParReadWrite_4: wOutHWord |
| 16137 | LS_ParReadWrite_4: wOutLWord |
| 16138 | LS_ParReadWrite_5: wOutHWord |
| 16139 | LS_ParReadWrite_5: wOutLWord |
| 16140 | LS_ParReadWrite_6: wOutHWord |
| 16141 | LS_ParReadWrite_6: wOutLWord |
| 16170 | LS_AxisBusIn: wLine1 |
| 16171 | LS_AxisBusIn: wLine2 |
| 16172 | LS_AxisBusIn: wLine3 |
| 16173 | LS_AxisBusIn: wCas1 |
| 16174 | LS_AxisBusIn: wCas2 |
| 16175 | LS_AxisBusIn: wCas3 |
| 16176 | LS_AxisBusIn: wCas4 |
| 16177 | LS_AxisBusAux: wAuxIn1 |
| 16178 | LS_AxisBusAux: wAuxIn2 |
| 16179 | LS_AxisBusAux: wAuxIn3 |
| 16180 | LS_AxisBusAux: wAuxIn4 |
| 16181 | LS_AxisBusAux: wSlaveNo |
| 16300 | LS_MultiEncoder: nActSpeed_v |

| Selection list - analog signals | |
|---|---|
| 16301 | LS_MultiEncoder: nActSpeed_a |
| 16302 | LS_MultiEncoder: wHighWord |
| 16303 | LS_MultiEncoder: wLowWord |
| 16320 | LS_BusEncoder: nActSpeed_v |
| 16321 | LS_BusEncoder: nActSpeed_a |
| 16322 | LS_BusEncoder: wHighWord |
| 16323 | LS_BusEncoder: wLowWord |
| 16340 | LS_Resolver: nActSpeed_v |
| 16341 | LS_Resolver: nActSpeed_a |
| 16350 | LS_RetainData: wOut1 |
| 16351 | LS_RetainData: wOut2 |
| 16352 | LS_RetainData: wOut3 |
| 16353 | LS_RetainData: wOut4 |
| 16360 | LP_CanIn4: wIn1 |
| 16361 | LP_CanIn4: wIn2 |
| 16362 | LP_CanIn4: wIn3 |
| 16363 | LP_CanIn4: wIn4 |
| 20000 | LS_ParFix: nPos100_a |
| 20001 | LS_ParFix: nNeg100_a |
| 20002 | LS_ParFix: nPos199_99_a |
| 20003 | LS_ParFix: nNeg199_99_a |
| 20004 | LS_ParFix: w65535 |
| 20005 | LS_ParFix: wDriveCtrl |
| 20010 | LS_ParFree_a: nC472_1_a |
| 20011 | LS_ParFree_a: nC472_2_a |
| 20012 | LS_ParFree_a: nC472_3_a |
| 20013 | LS_ParFree_a: nC472_4_a |
| 20014 | LS_ParFree_a: nC472_5_a |
| 20015 | LS_ParFree_a: nC472_6_a |
| 20016 | LS_ParFree_a: nC472_7_a |
| 20017 | LS_ParFree_a: nC472_8_a |
| 20018 | LS_ParFree_v: nC473_1_v |
| 20019 | LS_ParFree_v: nC473_2_v |
| 20020 | LS_ParFree_v: nC473_3_v |
| 20021 | LS_ParFree_v: nC473_4_v |
| 20022 | LS_ParFree_v: nC473_5_v |
| 20023 | LS_ParFree_v: nC473_6_v |
| 20024 | LS_ParFree_v: nC473_7_v |
| 20025 | LS_ParFree_v: nC473_8_v |
| 20026 | LS_ParFree: wC471_1 |
| 20027 | LS_ParFree: wC471_2 |
| 20028 | LS_ParFree: wC471_3 |
| 20029 | LS_ParFree: wC471_4 |
| 20030 | LS_ParFree: wC471_5 |
| 20031 | LS_ParFree: wC471_6 |
| 20032 | LS_ParFree: wC471_7 |
| 20033 | LS_ParFree: wC471_8 |
| 20034 | LS_ParFree: wC471_9 |
| 20035 | LS_ParFree: wC471_10 |
| 20036 | LS_ParFree: wC471_11 |
| 20037 | LS_ParFree: wC471_12 |
| 20038 | LS_ParFree: wC471_13 |

| Selection list - analog signals | |
|---|---|
| 20039 | LS_ParFree: wC471_14 |
| 20040 | LS_ParFree: wC471_15 |
| 20041 | LS_ParFree: wC471_16 |
| 20042 | LS_ParFree: wC471_17 |
| 20043 | LS_ParFree: wC471_18 |
| 20044 | LS_ParFree: wC471_19 |
| 20045 | LS_ParFree: wC471_20 |
| 20046 | LS_ParFree: wC471_21 |
| 20047 | LS_ParFree: wC471_22 |
| 20048 | LS_ParFree: wC471_23 |
| 20049 | LS_ParFree: wC471_24 |
| 20050 | LS_ParFree: wC471_25 |
| 20051 | LS_ParFree: wC471_26 |
| 20052 | LS_ParFree: wC471_27 |
| 20053 | LS_ParFree: wC471_28 |
| 20054 | LS_ParFree: wC471_29 |
| 20055 | LS_ParFree: wC471_30 |
| 20056 | LS_ParFree: wC471_31 |
| 20057 | LS_ParFree: wC471_32 |
| 20058 | LS_ParFree_a: nC472_9_a |
| 20059 | LS_ParFree_a: nC472_10_a |
| 20060 | LS_ParFree_a: nC472_11_a |
| 20061 | LS_ParFree_a: nC472_12_a |
| 20062 | LS_ParFree_a: nC472_13_a |
| 20063 | LS_ParFree_a: nC472_14_a |
| 20064 | LS_ParFree_a: nC472_15_a |
| 20065 | LS_ParFree_a: nC472_16_a |
| 20066 | LS_ParFree_a_2: nC476_1_a |
| 20067 | LS_ParFree_a_2: nC476_2_a |
| 20068 | LS_ParFree_a_2: nC476_3_a |
| 20069 | LS_ParFree_a_2: nC476_4_a |
| 20070 | LS_ParFree_a_2: nC476_5_a |
| 20071 | LS_ParFree_a_2: nC476_6_a |
| 20072 | LS_ParFree_a_2: nC476_7_a |
| 20073 | LS_ParFree_a_2: nC476_8_a |
| 20074 | LS_ParFree_a_2: nC476_9_a |
| 20075 | LS_ParFree_a_2: nC476_10_a |
| 20076 | LS_ParFree_a_2: nC476_11_a |
| 20077 | LS_ParFree_a_2: nC476_12_a |
| 20078 | LS_ParFree_a_2: nC476_13_a |
| 20079 | LS_ParFree_a_2: nC476_14_a |
| 20080 | LS_ParFree_a_2: nC476_15_a |
| 20081 | LS_ParFree_a_2: nC476_16_a |
| 20082 | LS_ParFix_2: nPos100_a |
| 20083 | LS_ParFix_2: nNeg100_a |
| 20084 | LS_ParFix_2: nPos199_99_a |
| 20085 | LS_ParFix_2: nNeg199_99_a |
| 20086 | LS_ParFix_2: 65535 |
| 20087 | LS_ParFix_2: wDriveCtrl |
| 20088 | LS_ParFree_2: wC477_1 |
| 20089 | LS_ParFree_2: wC477_2 |
| 20090 | LS_ParFree_2: wC477_3 |

| Selection list - analog signals | |
|---|---|
| 20091 | LS_ParFree_2: wC477_4 |
| 20092 | LS_ParFree_2: wC477_5 |
| 20093 | LS_ParFree_2: wC477_6 |
| 20094 | LS_ParFree_2: wC477_7 |
| 20095 | LS_ParFree_2: wC477_8 |
| 20096 | LS_ParFree_2: wC477_9 |
| 20097 | LS_ParFree_2: wC477_10 |
| 20098 | LS_ParFree_2: wC477_11 |
| 20099 | LS_ParFree_2: wC477_12 |
| 20100 | LS_ParFree_2: wC477_13 |
| 20101 | LS_ParFree_2: wC477_14 |
| 20102 | LS_ParFree_2: wC477_15 |
| 20103 | LS_ParFree_2: wC477_16 |
| 20104 | LS_ParFree_2: wC477_17 |
| 20105 | LS_ParFree_2: wC477_18 |
| 20106 | LS_ParFree_2: wC477_19 |
| 20107 | LS_ParFree_2: wC477_20 |
| 20108 | LS_ParFree_2: wC477_21 |
| 20109 | LS_ParFree_2: wC477_22 |
| 20110 | LS_ParFree_2: wC477_23 |
| 20111 | LS_ParFree_2: wC477_24 |
| 20112 | LS_ParFree_2: wC477_25 |
| 20113 | LS_ParFree_2: wC477_26 |
| 20114 | LS_ParFree_2: wC477_27 |
| 20115 | LS_ParFree_2: wC477_28 |
| 20116 | LS_ParFree_2: wC477_29 |
| 20117 | LS_ParFree_2: wC477_30 |
| 20118 | LS_ParFree_2: wC477_31 |
| 20119 | LS_ParFree_2: wC477_32 |
| 20120 | LS_ParFree_v_2: nC478_1_v |
| 20121 | LS_ParFree_v_2: nC478_2_v |
| 20122 | LS_ParFree_v_2: nC478_3_v |
| 20123 | LS_ParFree_v_2: nC478_4_v |
| 20124 | LS_ParFree_v_2: nC478_5_v |
| 20125 | LS_ParFree_v_2: nC478_6_v |
| 20126 | LS_ParFree_v_2: nC478_7_v |
| 20127 | LS_ParFree_v_2: nC478_8_v |
| 20128 | LS_DFOut: nOutAct_v |
| 32000 | LS_MotorInterface: nMotorSpeedAct_a |
| 32001 | LS_MotorInterface: nOutputSpeedCtrl_a |
| 32002 | LS_MotorInterface: nInputJerkCtrl_a |
| 32003 | LS_MotorInterface: nInputTorqueCtrl_a |
| 32004 | LS_MotorInterface: nMotorTorqueAct_a |
| 32005 | LS_MotorInterface: nActualFluxx_a |
| 32006 | LS_MotorInterface: nDCVoltage_a |
| 32007 | LS_MotorInterface: nStatorCurrentIS_a |
| 32008 | LS_MotorInterface: nEffCurrentIq_a |
| 32009 | LS_MotorInterface: nReaktCurrentId_a |
| 32010 | LS_MotorInterface: wMaxMotorSpeed |
| 32011 | LS_MotorInterface: wMaxMotorTorque |
| 32012 | LS_MotorInterface: nMotorVoltage_a |
| 32013 | LS_MotorInterface: nMotorFreqAct_a |

| Selection list - analog signals | |
|---|---|
| 32014 | LS_MotorInterface: nEffSpeedSetValue_a |
| 32015 | LS_DeviceMonitor_MCTRL: nIxTRate_a |
| 32016 | LS_DeviceMonitor_MCTRL: nI2xTRate_a |
| 32017 | LS_MotorInterface: nOutputPosCtrl_a |
| 32018 | LS_MotorInterface: nHlgSetValue_a |
| 32019 | LS_MotorInterface: nMotorSpeedAct_v |
| 32020 | LS_MotorInterface: nSpeedCtrlIAct_a |
| 32021 | LS_MotorInterface: nVoltageAngleAct_a |
| 32022 | LS_DeviceMonitor_MCTRL: nActiveOuputPower_a |
| 32023 | LS_DeviceMonitor_MCTRL: nApparentOuputPower_a |
| 32024 | LS_DeviceMonitor_MCTRL: nMctrlCosinePhiAct_a |
| 32100 | LS_DriveInterface: wDeviceStateWord |
| 32101 | LS_DriveInterface: wDeviceAuxStateWord |
| 32102 | LS_DriveInterface: wDetermFailNoLow |
| 32103 | LS_DriveInterface: wDetermFailNoHigh |
| 32104 | LS_DriveInterface: wDetermFailNoShort |
| 32200 | LS_MotionControlKernel: nSpeedSet_v |
| 32201 | LS_MotionControlKernel: nSpeedCtrlI_a |
| 32202 | LS_MotionControlKernel: nSpeedSetValue_a |
| 32203 | LS_MotionControlKernel: nTorqueSetValue_a |
| 32204 | LS_MotionControlKernel: wActProfileNo |
| 32205 | LS_MotionControlKernel: wFollowProfileNo |
| 32206 | LS_MotionControlKernel: wMotionState1 |
| 32207 | LS_MotionControlKernel: wMotionState2 |
| 32208 | LS_MotionControlKernel: wAuxState |
| 32209 | LS_MotionControlKernel: nPWMAngleOffset |
| 32210 | LS_MotionControlKernel: nTorqueLimitAdapt_a |
| 32211 | Reserved: a32211 |
| 34900 | MCTRL: OszCh1 |
| 34901 | MCTRL: OszCh2 |
| 34902 | MCTRL: OszCh3 |
| 34903 | MCTRL: OszCh4 |
| 34904 | MCTRL: Status1 |
| 34905 | MCTRL: Status2 |
| 34906 | MCTRL: Status3 |
| 34907 | LS_DeviceMonitor: wUB_24V |
| 36000 | L_Absolut_1: nOut_a |
| 36001 | L_AddSub_1: nOut_a |
| 36002 | L_OffsetGain_1: nOut_a |
| 36003 | L_OffsetGain_2: nOut_a |
| 36004 | L_OffsetGainP_1: nOut_a |
| 36005 | L_OffsetGainP_2: nOut_a |
| 36006 | L_GainOffset_1: nOut_a |
| 36007 | L_GainOffset_2: nOut_a |
| 36008 | L_GainOffsetP_1: nOut_a |
| 36009 | L_GainOffsetP_2: nOut_a |
| 36010 | L_Negation_1: nOut_a |
| 36011 | L_Arithmetik_1: nOut_a |
| 36012 | L_Arithmetik_2: nOut_a |
| 36013 | L_AnalogSwitch_1: nOut_a |
| 36014 | L_AnalogSwitch_2: nOut_a |
| 36015 | L_AnalogSwitch_3: nOut_a |

| Selection list - analog signals | |
|---|---|
| 36016 | L_Limit_1: nOut_a |
| 36017 | L_Limit_2: nOut_a |
| 36018 | L_NSet_1: nOut |
| 36019 | L_MPot_1: nOut_a |
| 36020 | L_PCTRL_1: nOut_a |
| 36021 | L_SignalMonitor_a: nOut1_a |
| 36022 | L_SignalMonitor_a: nOut2_a |
| 36023 | L_NLim_1: nOut_a |
| 36024 | L_Counter_1: wOut |
| 36025 | L_OffsetGainP_3: nOut_a |
| 36026 | L_GainOffsetP_3: nOut_a |
| 36027 | L_SignalMonitor_a: nOut3_a |
| 36028 | L_SignalMonitor_a: nOut4_a |
| 36029 | L_MulDiv_1: nOut_a |
| 36030 | L_NLim_1: wState |
| 36031 | L_NSet_1: wState |
| 36032 | L_NSet_1: nSetValue |
| 36033 | L_PT1_1: nOut_a |
| 36034 | L_Absolut_2: nOut_a |
| 36035 | L_AnalogSwitch_4: nOut_a |
| 36036 | L_AnalogSwitch_5: nOut_a |
| 36037 | L_Arithmetik_3: nOut_a |
| 36038 | L_Arithmetik_4: nOut_a |
| 36039 | L_Arithmetik_5: nOut_a |
| 36045 | L_GainOffset_3: nOut_a |
| 36053 | L_MulDiv_2: nOut_a |
| 36054 | L_Negation_2: nOut_a |
| 36055 | L_NLim_2: nOut_a |
| 36056 | L_NLim_2: wState |
| 36057 | L_OffsetGain_3: nOut_a |
| 36058 | L_PT1_2: nOut_a |
| 36059 | L_PT1_3: nOut_a |
| 36064 | L_SampleHold_1: nOut_a |
| 36065 | L_SampleHold_2: nOut_a |
| 36068 | L_Counter_2: wOut |
| 36069 | L_Counter_3: wOut |
| 36073 | L_DT1_1: nOut_a |
| 36074 | L_ConvBitsToWord_1: wOut |
| 36075 | L_ConvBitsToWord_2: wOut |
| 36076 | L_ConvBitsToWord_3: wOut |
| 36077 | L_ConvDIntToWords_1: wOutLWord |
| 36078 | L_ConvDIntToWords_1: wOutHWord |
| 36079 | L_ConvDIntToWords_2: wOutLWord |
| 36080 | L_ConvDIntToWords_2: wOutHWord |
| 36081 | L_ConvDIntToWords_3: wOutLWord |
| 36082 | L_ConvDIntToWords_3: wOutHWord |
| 36083 | L_MckCtrlInterface_1: wOutMckPosCtrl_1 |
| 36084 | L_MckCtrlInterface_1: wOutMckPosCtrl_2 |
| 36085 | L_MckCtrlInterface_1: wFailState |
| 36086 | L_MckStateInterface_1: wOperationMode |
| 36087 | L_MckStateInterface_1: wActProfileNo |
| 36088 | L_MckStateInterface_1: wActPosMode |

| Selection list - analog signals | |
|---|---|
| 36089 | L_PosiShaftCtrlInterface_1: wOutMckPosCtrl_1 |
| 36090 | L_PosiShaftCtrlInterface_1: wOutMckPosCtrl_2 |
| 36091 | L_PCTRL_1: nPIDOut1_a |
| 36092 | L_PCTRL_1: nPIDOut2_a |
| 36093 | L_PCTRL_1: nInfluenceOut_a |
| 36094 | L_Curve_1: nOut_a |
| 36095 | L_Interpolator_1: nPhdOut_v |
| 36096 | L_Interpolator_1: nNOut_a |
| 36097 | L_ConvW_1: wOut |
| 36098 | L_ConvW_2: wOut |
| 36099 | L_ConvW_3: wOut |
| 36100 | L_ConvW_4: wOut |
| 36101 | L_SRFG_1: nOut_a |
| 36102 | L_SRFG_2: nOut_a |
| 36103 | L_MckStateInterface_1: wPosUnitsLW |
| 36104 | L_MckStateInterface_1: wPosUnitsHW |
| 36105 | L_SignalSwitch_1: wOut |
| 36106 | L_SignalSwitch_2: wOut |
| 36107 | L_SignalSwitch_3: wOut |
| 36108 | L_SignalSwitch_4: wOut |
| 36109 | L_SRFG_1: nDeltaOut_a |
| 36110 | L_SRFG_2: nDeltaOut_a |
| 36111 | L_Odometer_1: wLastMeasure |
| 36112 | L_FixSet_a_1: nOut_a |
| 36113 | L_FixSet_a_1: wSelect |
| 36114 | L_FixSet_w_1: wOut |
| 36115 | L_FixSet_w_1: wSelect |
| 36116 | L_FixSet_w_2: wOut |
| 36117 | L_FixSet_w_2: wSelect |
| 36120 | L_CalcDiameter_1: nDiameter_a |
| 36121 | L_CalcDiameter_1: nReziprDiameter_a |
| 36122 | L_CalcDiameter_1: nNReel_v |
| 36123 | L_CalcDiameter_1: nDMin_a |
| 36124 | L_ProcessCtrl_1: nOut_a |
| 36125 | L_ProcessCtrl_1: nOutRed_a |
| 36126 | L_ProcessCtrl_1: nDeviation_a |
| 36127 | L_ProcessCtrl_1: nDComponent |
| 36128 | L_PhiIntegrator_1: nOut16 |
| 36129 | L_PhiIntegrator_1: nSpeedGearAdd_v |
| 36130 | L_PosCtrlLin_1: nNOut_v |
| 36131 | L_PosCtrlLin_2: nNOut_v |
| 36132 | L_PhiIntegrator_1: nSpeedGear_v |
| 36133 | L_DFSET_1: nSetGain_v |
| 36134 | L_DFSET_1: nSetGearTrim_v |
| 36135 | L_DFSET_1: nSetGearTrim_a |
| 36136 | L_DFRFG_1: nOut_v |
| 36137 | L_ConvPA_1: nOut |
| 36138 | L_ConvPA_2: nOut |
| 36139 | L_ConvPA_3: nOut |
| 36140 | L_ConvX_1: nOut |
| 36141 | L_ConvX_2: nOut |
| 36142 | L_ConvX_3: nOut |

| Selection list - analog signals | |
|---|---|
| 36143 | LS_MotionControlKernel: wGearNum |
| 36144 | LS_MotionControlKernel: wGearDenom |
| 36145 | L_DFSET_1: nSpeedSetOut_v |
| 36146 | L_Curve_2: nOut_a |
| 36147 | L_Curve_3: nOut_a |
| 36148 | L_Curve_3: nCurveValue_a |
| 36150 | L_Sequencer_1: wMCKPosCtrl1 |
| 36151 | L_Sequencer_1: wMCKPosCtrl2 |
| 36152 | L_Sequencer_1: wAuxCtrl |
| 36153 | L_Sequencer_1: wState |
| 36154 | L_Sequencer_1: wActStep |
| 36155 | L_Sequencer_1: nSet_a |
| 36156 | L_Sequencer_1: wProfileNumber |
| 36157 | L_Sequencer_1: wDigitalOutputs |
| 36158 | L_ConvActPos_1: nPosOut_a |
| 36159 | L_ConvActPos_1: nVAdditive_a |
| 36160 | L_ConvActPos_1: nVOut_a |
| 36161 | L_ConvActPos_1: nMAdditive_a |
| 36162 | L_MFail_1: nNOut_a |
| 42000 | LA_NCtrl_In: wCANDriveControl |
| 42001 | LA_NCtrl_In: wCCMDriveControl |
| 42002 | LA_NCtrl_In: nTorqueMotLim |
| 42003 | LA_NCtrl_In: nTorqueGenLim |
| 42004 | LA_NCtrl_In: nPIDVpAdapt_a |
| 42005 | LA_NCtrl_In: nPIDActValue_a |
| 42006 | LA_NCtrl_In: nMainSetValue |
| 42007 | LA_NCtrl_In: nAuxSetValue |
| 42008 | LA_NCtrl_In: nGPAnalogSwitchIn1_a |
| 42009 | LA_NCtrl_In: nGPAnalogSwitchIn2_a |
| 42010 | LA_NCtrl_In: nGPArithmetikIn1_a |
| 42011 | LA_NCtrl_In: nGPArithmetikIn2_a |
| 42012 | LA_NCtrl_In: nGPMulDivIn_a |
| 42013 | LA_NCtrl_In: nGPCompareIn1_a |
| 42014 | LA_NCtrl_In: nGPCompareIn2_a |
| 42015 | LA_NCtrl_In: nVoltageAdd_a |
| 42016 | LA_NCtrl_In: nPIDInfluence_a |
| 42017 | LA_NCtrl_In: nPIDSetValue_a |
| 42018 | LA_NCtrl_In: nPWMAngleOffset |
| 42019 | LA_NCtrl_In: nBoost_a |
| 42020 | LA_NCtrl_In: wSMControl |
| 42025 | LA_NCtrl_In: wFreeIn1 |
| 42026 | LA_NCtrl_In: wFreeIn2 |
| 42027 | LA_NCtrl_In: wFreeIn3 |
| 42028 | LA_NCtrl_In: wFreeIn4 |
| 42100 | LA_TabPos_In: wCanDriveControl |
| 42101 | LA_TabPos_In: wMciDriveControl |
| 42102 | LA_TabPos_In: nTorqueMotLim_a |
| 42103 | LA_TabPos_In: nTorqueGenLim_a |
| 42104 | LA_TabPos_In: nMainSetValue_a |
| 42105 | LA_TabPos_In: nAuxSetValue_a |
| 42106 | LA_TabPos_In: wMckCtrl1 |
| 42107 | LA_TabPos_In: wMckCtrl2 |

| Selection list - analog signals | |
|---|---|
| 42108 | LA_TabPos_In: wMckOperationMode |
| 42109 | LA_TabPos_In: wPosProfileMode |
| 42110 | LA_TabPos_In: wPosProfileNo |
| 42111 | LA_TabPos_In: nGPAnalogSwitchIn1_a |
| 42112 | LA_TabPos_In: nGPAnalogSwitchIn2_a |
| 42113 | LA_TabPos_In: nGPArithmetikIn1_a |
| 42114 | LA_TabPos_In: nGPArithmetikIn2_a |
| 42115 | LA_TabPos_In: nGPMulDivIn_a |
| 42116 | LA_TabPos_In: nGPCompareIn1_a |
| 42117 | LA_TabPos_In: nGPCompareIn2_a |
| 42118 | LA_TabPos_In: wGPCounter1LdVal |
| 42119 | LA_TabPos_In: wGPCounter1CmpVal |
| 42120 | LA_TabPos_In: nSpeedOverride_a |
| 42121 | LA_TabPos_In: nAccOverride_a |
| 42122 | LA_TabPos_In: wFreeIn1 |
| 42123 | LA_TabPos_In: wFreeIn2 |
| 42124 | LA_TabPos_In: wFreeIn3 |
| 42125 | LA_TabPos_In: wFreeIn4 |
| 42126 | LA_TabPos_In: wSMControl |
| 42127 | LA_TabPos_In: wPosProfileUnitsLW |
| 42128 | LA_TabPos_In: wPosProfileUnitsHW |
| 42200 | LA_SwitchPos_In: wCANDriveControl |
| 42201 | LA_SwitchPos_In: wMCIDriveControl |
| 42202 | LA_SwitchPos_In: nVoltageAdd_a |
| 42203 | LA_SwitchPos_In: nBoost_a |
| 42204 | LA_SwitchPos_In: nPWMAngleOffset |
| 42205 | LA_SwitchPos_In: nTorqueMotLim_a |
| 42206 | LA_SwitchPos_In: nTorqueGenLim_a |
| 42207 | LA_SwitchPos_In: nMainSetValue_a |
| 42208 | LA_SwitchPos_In: nAuxSetValue_a |
| 42209 | LA_SwitchPos_In: nGPAnalogSwitchIn1_a |
| 42210 | LA_SwitchPos_In: nGPAnalogSwitchIn2_a |
| 42211 | LA_SwitchPos_In: nGPArithmetikIn1_a |
| 42212 | LA_SwitchPos_In: nGPArithmetikIn2_a |
| 42213 | LA_SwitchPos_In: nGPMulDivIn_a |
| 42214 | LA_SwitchPos_In: nGPCompareIn1_a |
| 42215 | LA_SwitchPos_In: nGPCompareIn2_a |
| 42216 | LA_SwitchPos_In: wSMControl |
| 42221 | LA_SwitchPos_In: wFreeIn1 |
| 42222 | LA_SwitchPos_In: wFreeIn2 |
| 42223 | LA_SwitchPos_In: wFreeIn3 |
| 42224 | LA_SwitchPos_In: wFreeIn4 |

## 17.5 Selection list - digital signals

This selection list is relevant for the following configuration parameters:

| Parameters | |
|---|---|
| C00411 | L_SignalMonitor_b: Signal sources |
| C00621 | System connection list: Bool |
| C00701 | LA_NCtrl: Digital connection list |
| C00711 | LA_TabPos: Digital connection list |
| C00761 | LA_SwitchPos: Digital connection list |

| Selection list - digital signals | |
|---|---|
| 0 | Not connected |
| 1000 | LA_NCtrl: bDriveReady |
| 1001 | LA_NCtrl: bDriveFail |
| 1002 | LA_NCtrl: bCInhActive |
| 1003 | LA_NCtrl: bQSPIsActive |
| 1004 | LA_NCtrl: bSpeedCcw |
| 1005 | LA_NCtrl: bSpeedActCompare |
| 1008 | LA_NCtrl: bGPDigitalDelayOut |
| 1009 | LA_NCtrl: bGPLogicOut |
| 1010 | LA_NCtrl: bGPSignalOut1 |
| 1011 | LA_NCtrl: bGPSignalOut2 |
| 1012 | LA_NCtrl: bGPSignalOut3 |
| 1013 | LA_NCtrl: bGPSignalOut4 |
| 1014 | LA_NCtrl: bOverLoadActive |
| 1015 | LA_NCtrl: bMBrakeReleaseOut |
| 1016 | LA_NCtrl: bMBrakeReleased |
| 1017 | LA_NCtrl: bGPCompareOut |
| 1018 | LA_NCtrl: bUnderLoadActive |
| 1019 | LA_NCtrl: bImaxActive |
| 1020 | LA_NCtrl: bSpeedSetReached |
| 1021 | LA_NCtrl: bSpeedActEqSet |
| 1022 | LA_NCtrl: bGPDFlipFlopOut |
| 1023 | LA_NCtrl: bGPDFlipFlopNegOut |
| 1029 | LA_NCtrl: bFreeOut1 |
| 1030 | LA_NCtrl: bFreeOut2 |
| 1031 | LA_NCtrl: bFreeOut3 |
| 1032 | LA_NCtrl: bFreeOut4 |
| 1033 | LA_NCtrl: bFreeOut5 |
| 1034 | LA_NCtrl: bFreeOut6 |
| 1035 | LA_NCtrl: bFreeOut7 |
| 1036 | LA_NCtrl: bFreeOut8 |
| 1100 | LA_TabPos: bDriveFail |
| 1101 | LA_TabPos: bDriveReady |
| 1102 | LA_TabPos: bCInhActive |
| 1103 | LA_TabPos: bQSPIsActive |
| 1104 | LA_TabPos: bSpeedCcw |
| 1105 | LA_TabPos: bSpeedActCompare |
| 1106 | LA_TabPos: bImaxActive |
| 1107 | LA_TabPos: bSpeedSetReached |
| 1108 | LA_TabPos: bMBrakeReleaseOut |

| Selection list - digital signals | |
|---|---|
| 1109 | LA_TabPos: bMBrakeReleased |
| 1110 | LA_TabPos: bHomeDone |
| 1111 | LA_TabPos: bHomePosAvailable |
| 1112 | LA_TabPos: bProfileDone |
| 1113 | LA_TabPos: bProfileBusy |
| 1114 | LA_TabPos: bAccelerating |
| 1115 | LA_TabPos: bConstantDuty |
| 1116 | LA_TabPos: bDecelerating |
| 1117 | LA_TabPos: bDwellTime |
| 1118 | LA_TabPos: bInTarget |
| 1119 | LA_TabPos: bGPDigitalDelayOut |
| 1120 | LA_TabPos: bGPLogicOut |
| 1121 | LA_TabPos: bGPCompareOut |
| 1122 | LA_TabPos: bGPSignalOut1 |
| 1123 | LA_TabPos: bGPSignalOut2 |
| 1124 | LA_TabPos: bGPSignalOut3 |
| 1125 | LA_TabPos: bGPSignalOut4 |
| 1126 | LA_TabPos: bGPDFlipFlop_Out |
| 1127 | LA_TabPos: bGPDFlipFlop_NegOut |
| 1128 | LA_TabPos: bGPCounter1Equal |
| 1129 | LA_TabPos: bFreeOut1 |
| 1130 | LA_TabPos: bFreeOut2 |
| 1131 | LA_TabPos: bFreeOut3 |
| 1132 | LA_TabPos: bFreeOut4 |
| 1133 | LA_TabPos: bFreeOut5 |
| 1134 | LA_TabPos: bFreeOut6 |
| 1135 | LA_TabPos: bFreeOut7 |
| 1136 | LA_TabPos: bFreeOut8 |
| 1200 | LA_SwitchPos: bDriveFail |
| 1201 | LA_SwitchPos: bWarningActive |
| 1202 | LA_SwitchPos: bSafeTorqueOff |
| 1203 | LA_SwitchPos: bDriveReady |
| 1204 | LA_SwitchPos: bCInhActive |
| 1205 | LA_SwitchPos: bImpIsActive |
| 1206 | LA_SwitchPos: bQSPIsActive |
| 1207 | LA_SwitchPos: bSpeedCcw |
| 1208 | LA_SwitchPos: bSpeedActCompare |
| 1209 | LA_SwitchPos: bImaxActive |
| 1210 | LA_SwitchPos: bSpeedSetReached |
| 1211 | LA_SwitchPos: bSpeedActEqSet |
| 1212 | LA_SwitchPos: bBrakeReleaseOut |
| 1213 | LA_SwitchPos: bBrakeReleased |
| 1214 | LA_SwitchPos: bGPDigitalDelayOut |
| 1215 | LA_SwitchPos: bGPLogicOut |
| 1216 | LA_SwitchPos: bGPCompareOut |
| 1217 | LA_SwitchPos: bGPDFlipFlop_Out |
| 1218 | LA_SwitchPos: bGPDFlipFlop_NegOut |
| 1219 | LA_SwitchPos: bGPSignalOut1 |
| 1220 | LA_SwitchPos: bGPSignalOut2 |
| 1221 | LA_SwitchPos: bGPSignalOut3 |

| Selection list - digital signals | |
|---|---|
| 1222 | LA_SwitchPos: bGPSignalOut4 |
| 1228 | LA_SwitchPos: bFreeOut1 |
| 1229 | LA_SwitchPos: bFreeOut2 |
| 1230 | LA_SwitchPos: bFreeOut3 |
| 1231 | LA_SwitchPos: bFreeOut4 |
| 1232 | LA_SwitchPos: bFreeOut5 |
| 1233 | LA_SwitchPos: bFreeOut6 |
| 1234 | LA_SwitchPos: bFreeOut7 |
| 1235 | LA_SwitchPos: bFreeOut8 |
| 16000 | LS_DigitalInput: bIn1 |
| 16001 | LS_DigitalInput: bIn2 |
| 16002 | LS_DigitalInput: bIn3 |
| 16003 | LS_DigitalInput: bIn4 |
| 16004 | LS_DigitalInput: bIn5 |
| 16005 | LS_DigitalInput: bIn6 |
| 16006 | LS_DigitalInput: bIn7 |
| 16008 | LS_DigitalInput: bCInh |
| 16009 | LS_DigitalInput: bCountIn1_Compare |
| 16010 | LS_DigitalInput: bCountIn6_Compare |
| 16011 | LS_AnalogInput: bCurrentErrorIn1 |
| 16012 | LS_AnalogInput: bCurrentErrorIn2 |
| 16013 | LP_CanIn1: bCtrl1_B0 |
| 16014 | LP_CanIn1: bCtrl1_B1 |
| 16015 | LP_CanIn1: bCtrl1_B2 |
| 16016 | LP_CanIn1: bCtrl1_B3 |
| 16017 | LP_CanIn1: bCtrl1_B4 |
| 16018 | LP_CanIn1: bCtrl1_B5 |
| 16019 | LP_CanIn1: bCtrl1_B6 |
| 16020 | LP_CanIn1: bCtrl1_B7 |
| 16021 | LP_CanIn1: bCtrl1_B8 |
| 16022 | LP_CanIn1: bCtrl1_B9 |
| 16023 | LP_CanIn1: bCtrl1_B10 |
| 16024 | LP_CanIn1: bCtrl1_B11 |
| 16025 | LP_CanIn1: bCtrl1_B12 |
| 16026 | LP_CanIn1: bCtrl1_B13 |
| 16027 | LP_CanIn1: bCtrl1_B14 |
| 16028 | LP_CanIn1: bCtrl1_B15 |
| 16029 | LP_CanIn2: bIn1_B0 |
| 16030 | LP_CanIn2: bIn1_B1 |
| 16031 | LP_CanIn2: bIn1_B2 |
| 16032 | LP_CanIn2: bIn1_B3 |
| 16033 | LP_CanIn2: bIn1_B4 |
| 16034 | LP_CanIn2: bIn1_B5 |
| 16035 | LP_CanIn2: bIn1_B6 |
| 16036 | LP_CanIn2: bIn1_B7 |
| 16037 | LP_CanIn2: bIn1_B8 |
| 16038 | LP_CanIn2: bIn1_B9 |
| 16039 | LP_CanIn2: bIn1_B10 |
| 16040 | LP_CanIn2: bIn1_B11 |
| 16041 | LP_CanIn2: bIn1_B12 |
| 16042 | LP_CanIn2: bIn1_B13 |
| 16043 | LP_CanIn2: bIn1_B14 |

| Selection list - digital signals | |
|---|---|
| 16044 | LP_CanIn2: bIn1_B15 |
| 16045 | LP_CanIn3: bIn1_B0 |
| 16046 | LP_CanIn3: bIn1_B1 |
| 16047 | LP_CanIn3: bIn1_B2 |
| 16048 | LP_CanIn3: bIn1_B3 |
| 16049 | LP_CanIn3: bIn1_B4 |
| 16050 | LP_CanIn3: bIn1_B5 |
| 16051 | LP_CanIn3: bIn1_B6 |
| 16052 | LP_CanIn3: bIn1_B7 |
| 16053 | LP_CanIn3: bIn1_B8 |
| 16054 | LP_CanIn3: bIn1_B9 |
| 16055 | LP_CanIn3: bIn1_B10 |
| 16056 | LP_CanIn3: bIn1_B11 |
| 16057 | LP_CanIn3: bIn1_B12 |
| 16058 | LP_CanIn3: bIn1_B13 |
| 16059 | LP_CanIn3: bIn1_B14 |
| 16060 | LP_CanIn3: bIn1_B15 |
| 16061 | LP_MciIn: bCtrl_B0 |
| 16062 | LP_MciIn: bCtrl_B1 |
| 16063 | LP_MciIn: bCtrl_B2 |
| 16064 | LP_MciIn: bCtrl_B3 |
| 16065 | LP_MciIn: bCtrl_B4 |
| 16066 | LP_MciIn: bCtrl_B5 |
| 16067 | LP_MciIn: bCtrl_B6 |
| 16068 | LP_MciIn: bCtrl_B7 |
| 16069 | LP_MciIn: bCtrl_B8 |
| 16070 | LP_MciIn: bCtrl_B9 |
| 16071 | LP_MciIn: bCtrl_B10 |
| 16072 | LP_MciIn: bCtrl_B11 |
| 16073 | LP_MciIn: bCtrl_B12 |
| 16074 | LP_MciIn: bCtrl_B13 |
| 16075 | LP_MciIn: bCtrl_B14 |
| 16076 | LP_MciIn: bCtrl_B15 |
| 16077 | LP_MciIn: bIn2_B0 |
| 16078 | LP_MciIn: bIn2_B1 |
| 16079 | LP_MciIn: bIn2_B2 |
| 16080 | LP_MciIn: bIn2_B3 |
| 16081 | LP_MciIn: bIn2_B4 |
| 16082 | LP_MciIn: bIn2_B5 |
| 16083 | LP_MciIn: bIn2_B6 |
| 16084 | LP_MciIn: bIn2_B7 |
| 16085 | LP_MciIn: bIn2_B8 |
| 16086 | LP_MciIn: bIn2_B9 |
| 16087 | LP_MciIn: bIn2_B10 |
| 16088 | LP_MciIn: bIn2_B11 |
| 16089 | LP_MciIn: bIn2_B12 |
| 16090 | LP_MciIn: bIn2_B13 |
| 16091 | LP_MciIn: bIn2_B14 |
| 16092 | LP_MciIn: bIn2_B15 |
| 16093 | LS_Keypad: bSetQuickstop |
| 16094 | LS_Keypad: bSetDCBrake |
| 16095 | LS_Keypad: bSetSpeedCcw |

| Selection list - digital signals | |
|---|---|
| 16096 | LS_Keypad: bJogSpeed1 |
| 16097 | LS_Keypad: bJogSpeed2 |
| 16098 | LS_Keypad: bMPotEnable |
| 16099 | LS_Keypad: bMPotUp |
| 16100 | LS_Keypad: bMPotDown |
| 16101 | LS_DigitalInput: bPosIn12_State |
| 16102 | LS_DigitalInput: bOutHC_BrakeApplied |
| 16110 | LS_ParReadWrite_1: bDone |
| 16111 | LS_ParReadWrite_1: bFail |
| 16112 | LS_ParReadWrite_2: bDone |
| 16113 | LS_ParReadWrite_2: bFail |
| 16114 | LS_ParReadWrite_3: bDone |
| 16115 | LS_ParReadWrite_3: bFail |
| 16116 | LS_ParReadWrite_4: bDone |
| 16117 | LS_ParReadWrite_4: bFail |
| 16118 | LS_ParReadWrite_5: bDone |
| 16119 | LS_ParReadWrite_5: bFail |
| 16120 | LS_ParReadWrite_6: bDone |
| 16121 | LS_ParReadWrite_6: bFail |
| 16122 | LS_WriteParamList: bDone |
| 16123 | LS_WriteParamList: bFail |
| 16161 | LS_CANManagement: bFail |
| 16162 | LS_CANManagement: bOperational |
| 16165 | LS_MultiEncoder: bFail |
| 16166 | LS_Resolver: bFail |
| 16200 | LS_SyncManagement: bSyncSignalOK |
| 16201 | LS_SyncManagement: bSyncPhaseOK |
| 16202 | LS_MultiEncoder: bState |
| 16205 | LS_MultiEncoder: bState2 |
| 16206 | LS_MultiEncoder: bState3 |
| 16303 | LS_TouchProbe: bTPDigIn3Received |
| 16304 | LS_TouchProbe: bTPDigIn4Received |
| 16305 | LS_TouchProbe: bTPDigIn5Received |
| 16306 | LS_TouchProbe: bTPDigIn6Received |
| 16307 | LS_TouchProbe: bTPDigIn7Received |
| 16308 | LS_AxisBusIO: bFail_DigIn |
| 16309 | LS_AxisBusIO: bResetFail_In |
| 16310 | LS_RetainData: bOut1 |
| 16311 | LS_RetainData: bOut2 |
| 16312 | LS_RetainData: bOut3 |
| 16313 | LS_RetainData: bOut4 |
| 16314 | LS_TouchProbe: bTPEncoderReceived |
| 16315 | LS_TouchProbe: bTPResolverReceived |
| 16320 | LS_AxisBusAux: bDone |
| 16321 | LS_AxisBusAux: bFail |
| 16400 | LP_CanIn4: bIn1_B0 |
| 16401 | LP_CanIn4: bIn1_B1 |
| 16402 | LP_CanIn4: bIn1_B2 |
| 16403 | LP_CanIn4: bIn1_B3 |
| 16404 | LP_CanIn4: bIn1_B4 |
| 16405 | LP_CanIn4: bIn1_B5 |
| 16406 | LP_CanIn4: bIn1_B6 |

| Selection list - digital signals | |
|---|---|
| 16407 | LP_CanIn4: bIn1_B7 |
| 16408 | LP_CanIn4: bIn1_B8 |
| 16409 | LP_CanIn4: bIn1_B9 |
| 16410 | LP_CanIn4: bIn1_B10 |
| 16411 | LP_CanIn4: bIn1_B11 |
| 16412 | LP_CanIn4: bIn1_B12 |
| 16413 | LP_CanIn4: bIn1_B13 |
| 16414 | LP_CanIn4: bIn1_B14 |
| 16415 | LP_CanIn4: bIn1_B15 |
| 20000 | LS_ParFix: bTrue |
| 20001 | LS_ParFree_b: bC470_1 |
| 20002 | LS_ParFree_b: bC470_2 |
| 20003 | LS_ParFree_b: bC470_3 |
| 20004 | LS_ParFree_b: bC470_4 |
| 20005 | LS_ParFree_b: bC470_5 |
| 20006 | LS_ParFree_b: bC470_6 |
| 20007 | LS_ParFree_b: bC470_7 |
| 20008 | LS_ParFree_b: bC470_8 |
| 20009 | LS_ParFree_b: bC470_9 |
| 20010 | LS_ParFree_b: bC470_10 |
| 20011 | LS_ParFree_b: bC470_11 |
| 20012 | LS_ParFree_b: bC470_12 |
| 20013 | LS_ParFree_b: bC470_13 |
| 20014 | LS_ParFree_b: bC470_14 |
| 20015 | LS_ParFree_b: bC470_15 |
| 20016 | LS_ParFree_b: bC470_16 |
| 20017 | LS_ParFree_b: bC470_17 |
| 20018 | LS_ParFree_b: bC470_18 |
| 20019 | LS_ParFree_b: bC470_19 |
| 20020 | LS_ParFree_b: bC470_20 |
| 20021 | LS_ParFree_b: bC470_21 |
| 20022 | LS_ParFree_b: bC470_22 |
| 20023 | LS_ParFree_b: bC470_23 |
| 20024 | LS_ParFree_b: bC470_24 |
| 20025 | LS_ParFree_b: bC470_25 |
| 20026 | LS_ParFree_b: bC470_26 |
| 20027 | LS_ParFree_b: bC470_27 |
| 20028 | LS_ParFree_b: bC470_28 |
| 20029 | LS_ParFree_b: bC470_29 |
| 20030 | LS_ParFree_b: bC470_30 |
| 20031 | LS_ParFree_b: bC470_31 |
| 20032 | LS_ParFree_b: bC470_32 |
| 20033 | LS_PulseGenerator: b100Hz |
| 20034 | LS_PulseGenerator: b10Hz |
| 20035 | LS_PulseGenerator: b2Hz |
| 20036 | LS_PulseGenerator: b1Hz |
| 20037 | LS_PulseGenerator: b1HzFlash |
| 20038 | LS_PulseGenerator: b2HzFlash |
| 20039 | LS_PulseGenerator: bSingleFlash1 |
| 20040 | LS_PulseGenerator: bSingleFlash2 |
| 20041 | LS_PulseGenerator: bDoubleFlash |
| 20042 | LS_PulseGenerator: bSquareWave |

| Selection list - digital signals | |
|---|---|
| 20043 | LS_PulseGenerator: bFirstCycle |
| 20044 | LS_ParFix_2: bTrue |
| 20045 | LS_ParFreeUnit_1: bDataValid |
| 20046 | LS_ParFreeUnit_2: bDataValid |
| 32000 | LS_DeviceMonitor_MCTRL: bFanFault |
| 32001 | LS_DeviceMonitor_MCTRL: bHeatSinkTemp |
| 32002 | LS_MotorInterface: bLimPosCtrlOut |
| 32003 | LS_MotorInterface: bLimSpeedCtrlOut |
| 32004 | LS_MotorInterface: bLimSpeedSetVal |
| 32005 | LS_MotorInterface: bLimTorqueSetVal |
| 32006 | LS_MotorInterface: bLimCurrentSetVal |
| 32007 | LS_DeviceMonitor_MCTRL: bUVDetected |
| 32008 | LS_DeviceMonitor_MCTRL: bOVDetected |
| 32009 | LS_DeviceMonitor_MCTRL: bMotorPhaseFault |
| 32010 | LS_DeviceMonitor_MCTRL: bEncoderComFault |
| 32011 | LS_DeviceMonitor_MCTRL: bIxtOverload |
| 32012 | LS_DeviceMonitor_MCTRL: bI2xtOverload |
| 32013 | LS_MotorInterface: bIdentificationActive |
| 32014 | LS_MotorInterface: bFlyingSyncActive |
| 32015 | LS_DeviceMonitor_MCTRL: bTorqueMax |
| 32016 | LS_DeviceMonitor_MCTRL: bNMax |
| 32017 | LS_DeviceMonitor_MCTRL: bFChopReduced |
| 32018 | LS_DeviceMonitor_MCTRL: bMotorPTC |
| 32019 | LS_DeviceMonitor_MCTRL: bMotorTemp |
| 32020 | LS_MotorInterface: bDcbActive |
| 32021 | LS_DeviceMonitor_MCTRL: bBrakeChopperFault |
| 32022 | LS_MotorInterface: bQspIsActive |
| 32023 | LS_MotorInterface: bHlgLoad |
| 32024 | LS_MotorInterface: bHlgStop |
| 32025 | LS_DeviceMonitor_MCTRL: bImpActive |
| 32026 | LS_DeviceMonitor_MCTRL: bClampActive |
| 32027 | LS_DeviceMonitor_MCTRL: bMainsFault |
| 32028 | LS_DeviceMonitor_MCTRL: bNmaxForFChop |
| 32029 | LS_DeviceMonitor_MCTRL: bShortCircuit |
| 32030 | LS_DeviceMonitor_MCTRL: bEarthFault |
| 32031 | LS_DeviceMonitor_MCTRL: bWirebreakUfLinearActive |
| 32032 | LS_DeviceMonitor_MCTRL: bCurrentMonitoringOverload |
| 32100 | LS_DriveInterface: bInit |
| 32101 | LS_DriveInterface: bReady |
| 32102 | LS_DriveInterface: bReadyToSwitchOn |
| 32103 | LS_DriveInterface: bOperationEnable |
| 32104 | LS_DriveInterface: bWarning |
| 32105 | LS_DriveInterface: bTrouble |
| 32106 | LS_DriveInterface: bFail |
| 32107 | LS_DriveInterface: bCollectedFail |
| 32108 | LS_DriveInterface: bSafeTorqueOff |
| 32109 | LS_DriveInterface: bIMPIsActive |
| 32110 | LS_DriveInterface: bCINHIsActive |
| 32111 | LS_DriveInterface: bSafetyIsActive |
| 32112 | LS_DriveInterface: bCwCcw |

| Selection list - digital signals | |
|---|---|
| 32113 | LS_DriveInterface: bNactCompare |
| 32200 | LS_MotionControlKernel: bPosCtrlOn |
| 32201 | LS_MotionControlKernel: bSpeedCtrlIOn |
| 32202 | LS_MotionControlKernel: bTorquemodeOn |
| 32203 | LS_MotionControlKernel: bDcBrakeOn |
| 32204 | LS_MotionControlKernel: bBrkReleaseOut |
| 32205 | LS_MotionControlKernel: bBrkReleased |
| 32206 | LS_MotionControlKernel: bDeltaPosOn |
| 32207 | LS_MotionControlKernel: bPosDerivativeOn |
| 32208 | LS_MotionControlKernel: bMotorRefOffsetOn |
| 32209 | LS_MotionControlKernel: bQspOn |
| 32210 | LS_MotionControlKernel: bPosBusy |
| 32211 | LS_MotionControlKernel: bPosDone |
| 32212 | LS_MotionControlKernel: bHomDone |
| 32213 | LS_MotionControlKernel: bHomAvailable |
| 32214 | LS_MotionControlKernel: bTorqueLimitAdaptOn |
| 32215 | Reserved: b32215 |
| 32216 | Reserved: b32216 |
| 32217 | Reserved: b32217 |
| 32218 | Reserved: b32218 |
| 32219 | LS_MotionControlKernel: bFollowErrLim1 |
| 32220 | LS_MotionControlKernel: bFollowErrLim2 |
| 32221 | LS_MotionControlKernel: bReadyToOperate |
| 36000 | L_And_1: bOut |
| 36001 | L_And_2: bOut |
| 36002 | L_And_3: bOut |
| 36003 | L_Or_1: bOut |
| 36004 | L_Or_2: bOut |
| 36005 | L_Or_3: bOut |
| 36006 | L_Not_1: bOut |
| 36007 | L_Not_2: bOut |
| 36008 | L_Not_3: bOut |
| 36009 | L_DFlipFlop_1: bOut |
| 36010 | L_RLQ_1: bQsp |
| 36011 | L_RLQ_1: CwCcw |
| 36012 | L_DigitalDelay_1: bOut |
| 36013 | L_Compare_1: bOut |
| 36014 | L_Compare_2: bOut |
| 36015 | L_Compare_3: bOut |
| 36016 | L_NSet_1: bRfgEq0 |
| 36017 | L_DigitalLogic_1: bOut |
| 36018 | L_Counter_1: bEqual |
| 36019 | L_SignalMonitor_b: bOut1 |
| 36020 | L_SignalMonitor_b: bOut2 |
| 36021 | L_SignalMonitor_b: bOut3 |
| 36022 | L_SignalMonitor_b: bOut4 |
| 36023 | L_PCTRL_1: bActEqSet |
| 36024 | L_NLim_1: bLimitActive |
| 36025 | L_DFlipFlop_1: bNegOut |
| 36026 | L_And5_1: bOut |
| 36027 | L_And5_2: bOut |
| 36028 | L_Compare_4: bOut |

| Selection list - digital signals | |
|---|---|
| 36029 | L_Compare_5: bOut |
| 36030 | L_ComparePhi_1: bOut |
| 36031 | L_ComparePhi_2: bOut |
| 36032 | L_ComparePhi_3: bOut |
| 36033 | L_ComparePhi_4: bOut |
| 36034 | L_ComparePhi_5: bOut |
| 36035 | L_DFlipFlop_2: bOut |
| 36036 | L_DFlipFlop_2: bNegOut |
| 36037 | L_DigitalDelay_2: bOut |
| 36038 | L_DigitalDelay_3: bOut |
| 36039 | L_Or_4: bOut |
| 36040 | L_DigitalLogic_3: bOut |
| 36041 | L_DigitalLogic_2: bOut |
| 36042 | L_DigitalLogic5_1: bOut |
| 36043 | L_DigitalLogic5_2: bOut |
| 36044 | L_NLim_2: bLimitActive |
| 36045 | L_Or5_1: bOut |
| 36046 | L_Or5_2: bOut |
| 36047 | L_Not_4: bOut |
| 36048 | L_Not_5: bOut |
| 36049 | L_Not_6: bOut |
| 36050 | L_Not_7: bOut |
| 36055 | L_PhaseIntK_1: bState |
| 36056 | L_PhaseIntK_2: bState |
| 36057 | L_Counter_2: bEqual |
| 36058 | L_Counter_3: bEqual |
| 36059 | L_RSFlipFlop_1: bOut |
| 36060 | L_RSFlipFlop_1: bNegOut |
| 36061 | L_RSFlipFlop_2: bOut |
| 36062 | L_RSFlipFlop_2: bNegOut |
| 36063 | L_Transient_5: bOut |
| 36064 | L_Transient_6: bOut |
| 36065 | L_Transient_7: bOut |
| 36066 | L_Transient_8: bOut |
| 36067 | L_ConvWordToBits_1: bBit0 |
| 36068 | L_ConvWordToBits_1: bBit1 |
| 36069 | L_ConvWordToBits_1: bBit2 |
| 36070 | L_ConvWordToBits_1: bBit3 |
| 36071 | L_ConvWordToBits_1: bBit4 |
| 36072 | L_ConvWordToBits_1: bBit5 |
| 36073 | L_ConvWordToBits_1: bBit6 |
| 36074 | L_ConvWordToBits_1: bBit7 |
| 36075 | L_ConvWordToBits_1: bBit8 |
| 36076 | L_ConvWordToBits_1: bBit9 |
| 36077 | L_ConvWordToBits_1: bBit10 |
| 36078 | L_ConvWordToBits_1: bBit11 |
| 36079 | L_ConvWordToBits_1: bBit12 |
| 36080 | L_ConvWordToBits_1: bBit13 |
| 36081 | L_ConvWordToBits_1: bBit14 |
| 36082 | L_ConvWordToBits_1: bBit15 |
| 36083 | L_ConvWordToBits_2: bBit0 |
| 36084 | L_ConvWordToBits_2: bBit1 |

| Selection list - digital signals | |
|---|---|
| 36085 | L_ConvWordToBits_2: bBit2 |
| 36086 | L_ConvWordToBits_2: bBit3 |
| 36087 | L_ConvWordToBits_2: bBit4 |
| 36088 | L_ConvWordToBits_2: bBit5 |
| 36089 | L_ConvWordToBits_2: bBit6 |
| 36090 | L_ConvWordToBits_2: bBit7 |
| 36091 | L_ConvWordToBits_2: bBit8 |
| 36092 | L_ConvWordToBits_2: bBit9 |
| 36093 | L_ConvWordToBits_2: bBit10 |
| 36094 | L_ConvWordToBits_2: bBit11 |
| 36095 | L_ConvWordToBits_2: bBit12 |
| 36096 | L_ConvWordToBits_2: bBit13 |
| 36097 | L_ConvWordToBits_2: bBit14 |
| 36098 | L_ConvWordToBits_2: bBit15 |
| 36099 | L_ConvWordToBits_3: bBit0 |
| 36100 | L_ConvWordToBits_3: bBit1 |
| 36101 | L_ConvWordToBits_3: bBit2 |
| 36102 | L_ConvWordToBits_3: bBit3 |
| 36103 | L_ConvWordToBits_3: bBit4 |
| 36104 | L_ConvWordToBits_3: bBit5 |
| 36105 | L_ConvWordToBits_3: bBit6 |
| 36106 | L_ConvWordToBits_3: bBit7 |
| 36107 | L_ConvWordToBits_3: bBit8 |
| 36108 | L_ConvWordToBits_3: bBit9 |
| 36109 | L_ConvWordToBits_3: bBit10 |
| 36110 | L_ConvWordToBits_3: bBit11 |
| 36111 | L_ConvWordToBits_3: bBit12 |
| 36112 | L_ConvWordToBits_3: bBit13 |
| 36113 | L_ConvWordToBits_3: bBit14 |
| 36114 | L_ConvWordToBits_3: bBit15 |
| 36115 | L_MckCtrlInterface_1: bFail |
| 36116 | L_MckStateInterface_1: bAccelerating |
| 36117 | L_MckStateInterface_1: bDecelerating |
| 36118 | L_MckStateInterface_1: bConstantSpeedDuty |
| 36119 | L_MckStateInterface_1: bSShaping |
| 36120 | L_MckStateInterface_1: bBusy |
| 36121 | L_MckStateInterface_1: bDone |
| 36122 | L_MckStateInterface_1: bHomingPosAvailable |
| 36123 | L_MckStateInterface_1: bHomingDone |
| 36124 | L_MckStateInterface_1: bDwellTime |
| 36125 | L_MckStateInterface_1: bInTarget |
| 36126 | L_MckStateInterface_1: bPosDone |
| 36127 | L_PosiShaftCtrlInterface_1: bQsp |
| 36128 | L_PosiShaftCtrlInterface_1: bRsp |
| 36129 | L_PosiShaftCtrlInterface_1: bTripSet |
| 36130 | L_PosiShaftCtrlInterface_1: bTripReset |
| 36131 | L_JogCtrlExtension_1: bRfgOut |
| 36132 | L_JogCtrlExtension_1: bJog1Out |
| 36133 | L_JogCtrlExtension_1: bJog2Out |
| 36134 | L_Curve_1: bLimit |
| 36135 | L_Interpolator_1: bIPulse |
| 36136 | L_Interpolator_1: bSignalError |

| Selection list - digital signals | |
|---|---|
| 36137 | L_MckCtrlInterface_1: bPosSetDataValid |
| 36138 | L_Transient_1: bOut |
| 36139 | L_Transient_2: bOut |
| 36140 | L_Transient_3: bOut |
| 36141 | L_Transient_4: bOut |
| 36142 | L_CalcDiameter_1: bDMaxLimit |
| 36143 | L_CalcDiameter_1: bDMinLimit |
| 36144 | L_CalcDiameter_1: bWebBreak |
| 36145 | L_PhiIntegrator_1: bOvfl32 |
| 36146 | L_PhiIntegrator_1: bOvfl16 |
| 36147 | L_PhiIntegrator_1: bInitDone |
| 36148 | L_PosCtrlLin_1: bInTarget |
| 36149 | L_PosCtrlLin_2: bInTarget |
| 36150 | L_SwitchPoint_1: bOut1 |
| 36151 | L_SwitchPoint_1: bOut2 |
| 36152 | L_SwitchPoint_1: bOut3 |
| 36153 | L_SwitchPoint_1: bOut4 |
| 36154 | L_DFSET_1: bAck |
| 36155 | L_DFSET_1: bFollowingErr |
| 36156 | L_DFSET_1: bPosOverflow |
| 36157 | L_DFRFG_1: bSync |
| 36158 | L_DFRFG_1: bFail |
| 36159 | L_Curve_2: bLimit |
| 36160 | L_Curve_3: bLimit |
| 36161 | L_Sequencer_1: bProgramBusy |
| 36162 | L_Sequencer_1: bStateReady |
| 36163 | L_Sequencer_1: bStateRun |
| 36164 | L_Sequencer_1: bStatePause |
| 36165 | L_Sequencer_1: bStateBreak |
| 36166 | L_Sequencer_1: bStateReset |
| 36167 | L_Sequencer_1: bStateDone |
| 36168 | L_Sequencer_1: bErrActive |
| 36169 | L_Sequencer_1: bWacthdogActive |
| 36170 | L_Sequencer_1: bErrBranch |
| 36171 | L_ConvUnitsToIncr_1: bDataValid |
| 36172 | L_ConvUnitsToIncr_2: bDataValid |
| 36173 | L_ConvUnitsToIncr_3: bDataValid |
| 36174 | L_ConvActPos_1: bPosOverrange |
| 36175 | L_ConvActPos_1: bInSetPosition |
| 36176 | L_ConvActPos_1: bNegOverrange |
| 36177 | L_MFail_1: bActive |
| 36178 | L_MFail_1: bnCtrlIReset |
| 36179 | L_Curve_3: bLimit |
| 36180 | L_SwitchPoint_1: bOut5 |
| 36181 | L_SwitchPoint_1: bOut6 |
| 36182 | L_SwitchPoint_1: bOut7 |
| 36183 | L_SwitchPoint_1: bOut8 |
| 36184 | L_SwitchPointPar_1: bOut1 |
| 36185 | L_SwitchPointPar_1: bOut2 |
| 36186 | L_SwitchPointPar_1: bOut3 |
| 36187 | L_SwitchPointPar_1: bOut4 |
| 36188 | L_SwitchPointPar_1: bOut5 |

| Selection list - digital signals | |
|---|---|
| 36189 | L_SwitchPointPar_1: bOut6 |
| 36190 | L_SwitchPointPar_1: bOut7 |
| 36191 | L_SwitchPointPar_1: bOut8 |
| 36284 | L_MckStateInterface_1: bReadyToOperate |
| 36285 | L_MckStateInterface_1: bOperationState1 |
| 36286 | L_MckStateInterface_1: bOperationState2 |
| 36287 | L_MckStateInterface_1: bOperationState3 |
| 36288 | L_MckStateInterface_1: bOperationState4 |
| 36289 | L_MckStateInterface_1: bOperationState5 |
| 36290 | L_MckStateInterface_1: bOperationState6 |
| 36291 | L_MckStateInterface_1: bOperationState7 |
| 36292 | L_MckStateInterface_1: bOperationState8 |
| 36293 | L_DigitalLogic_1: bOut2 |
| 36294 | L_DigitalLogic_1: bOut3 |
| 36295 | L_DigitalLogic_2: bOut2 |
| 36296 | L_DigitalLogic_2: bOut3 |
| 36297 | L_DigitalLogic_3: bOut2 |
| 36298 | L_DigitalLogic_3: bOut3 |
| 42000 | LA_NCtrl_In: bCInh |
| 42001 | LA_NCtrl_In: bFailReset |
| 42002 | LA_NCtrl_In: bSetQuickstop |
| 42003 | LA_NCtrl_In: bSetDCBrake |
| 42004 | LA_NCtrl_In: bRFG_Stop |
| 42005 | LA_NCtrl_In: bRFG_0 |
| 42007 | LA_NCtrl_In: bSetSpeedCcw |
| 42008 | LA_NCtrl_In: bJogSpeed1 |
| 42009 | LA_NCtrl_In: bJogSpeed2 |
| 42010 | LA_NCtrl_In: bJogSpeed4 |
| 42011 | LA_NCtrl_In: bJogSpeed8 |
| 42012 | LA_NCtrl_In: bJogRamp1 |
| 42013 | LA_NCtrl_In: bJogRamp2 |
| 42014 | LA_NCtrl_In: bJogRamp4 |
| 42015 | LA_NCtrl_In: bJogRamp8 |
| 42017 | LA_NCtrl_In: bMPOTInAct |
| 42018 | LA_NCtrl_In: bMPOTUp |
| 42019 | LA_NCtrl_In: bMPOTDown |
| 42020 | LA_NCtrl_In: bMBRKRelease |
| 42021 | LA_NCtrl_In: bGPFree1 |
| 42022 | LA_NCtrl_In: bGPFree2 |
| 42023 | LA_NCtrl_In: bGPAnalogSwitchSet |
| 42024 | LA_NCtrl_In: bGPDigitalDelayIn |
| 42025 | LA_NCtrl_In: bGPLogicIn1 |
| 42026 | LA_NCtrl_In: bGPLogicIn2 |
| 42027 | LA_NCtrl_In: bGPLogicIn3 |
| 42028 | LA_NCtrl_In: bGPDFlipFlopInD |
| 42029 | LA_NCtrl_In: bGPDFlipFlopInClk |
| 42030 | LA_NCtrl_In: bGPDFlipFlopInClr |
| 42031 | LA_NCtrl_In: bMPotEnable |
| 42032 | LA_NCtrl_In: bPIDEnableInfluenceRamp |
| 42033 | LA_NCtrl_In: bPIDIOff |
| 42034 | LA_NCtrl_In: bRLQCw |
| 42035 | LA_NCtrl_In: bRLQCcw |

| Selection list - digital signals | |
|---|---|
| 42041 | LA_NCtrl_In: bFreeIn1 |
| 42042 | LA_NCtrl_In: bFreeIn2 |
| 42043 | LA_NCtrl_In: bFreeIn3 |
| 42044 | LA_NCtrl_In: bFreeIn4 |
| 42045 | LA_NCtrl_In: bFreeIn5 |
| 42046 | LA_NCtrl_In: bFreeIn6 |
| 42047 | LA_NCtrl_In: bFreeIn7 |
| 42048 | LA_NCtrl_In: bFreeIn8 |
| 42100 | LA_TabPos_In: bCInh |
| 42101 | LA_TabPos_In: bFailReset |
| 42102 | LA_TabPos_In: bSetQuickstop |
| 42103 | LA_TabPos_In: bSetSpeedCcw |
| 42104 | LA_TabPos_In: bJogSpeed1 |
| 42105 | LA_TabPos_In: bJogSpeed2 |
| 42106 | LA_TabPos_In: bMPotEnable |
| 42107 | LA_TabPos_In: bMPotUp |
| 42108 | LA_TabPos_In: bMPotDown |
| 42109 | LA_TabPos_In: bMBrakeRelease |
| 42110 | LA_TabPos_In: bPosCtrlOn |
| 42111 | LA_TabPos_In: bLimitSwitchPos |
| 42112 | LA_TabPos_In: bLimitSwitchNeg |
| 42113 | LA_TabPos_In: bReleaseLimitSwitch |
| 42114 | LA_TabPos_In: bManJogPos |
| 42115 | LA_TabPos_In: bManJogNeg |
| 42116 | LA_TabPos_In: bManEnable2ndSpeed |
| 42117 | LA_TabPos_In: bEnableSpeedOverride |
| 42118 | LA_TabPos_In: bEnableAccOverride |
| 42119 | LA_TabPos_In: bHomeStartStop |
| 42120 | LA_TabPos_In: bHomeSetPosition |
| 42121 | LA_TabPos_In: bHomeResetPosition |
| 42122 | LA_TabPos_In: bHomeMark |
| 42123 | LA_TabPos_In: bPosSetProfilePosition |
| 42124 | LA_TabPos_In: bPosSetActualPosition |
| 42125 | LA_TabPos_In: bPosExecute |
| 42126 | LA_TabPos_In: bPosFinishTarget |
| 42127 | LA_TabPos_In: bPosDisableFollowProfile |
| 42128 | LA_TabPos_In: bPosStop |
| 42129 | LA_TabPos_In: bGPAnalogSwitchSet |
| 42130 | LA_TabPos_In: bGPDigitalDelayIn |
| 42131 | LA_TabPos_In: bGPLogicIn1 |
| 42132 | LA_TabPos_In: bGPLogicIn2 |
| 42133 | LA_TabPos_In: bGPLogicIn3 |
| 42134 | LA_TabPos_In: bGPDFlipFlop_InD |
| 42135 | LA_TabPos_In: bGPDFlipFlop_InClk |
| 42136 | LA_TabPos_In: bGPDFlipFlop_InClr |
| 42137 | LA_TabPos_In: bGPCounter1ClkUp |
| 42138 | LA_TabPos_In: bGPCounter1ClkDown |
| 42139 | LA_TabPos_In: bGPCounter1Load |
| 42140 | LA_TabPos_In: bMckOperationMode_1 |
| 42141 | LA_TabPos_In: bMckOperationMode_2 |
| 42142 | LA_TabPos_In: bMckOperationMode_4 |
| 42143 | LA_TabPos_In: bMckOperationMode_8 |

| Selection list - digital signals | |
|---|---|
| 42144 | LA_TabPos_In: bPosProfileNo_1 |
| 42145 | LA_TabPos_In: bPosProfileNo_2 |
| 42146 | LA_TabPos_In: bPosProfileNo_4 |
| 42147 | LA_TabPos_In: bPosProfileNo_8 |
| 42148 | LA_TabPos_In: bFreeIn1 |
| 42149 | LA_TabPos_In: bFreeIn2 |
| 42150 | LA_TabPos_In: bFreeIn3 |
| 42151 | LA_TabPos_In: bFreeIn4 |
| 42152 | LA_TabPos_In: bFreeIn5 |
| 42153 | LA_TabPos_In: bFreeIn6 |
| 42154 | LA_TabPos_In: bFreeIn7 |
| 42155 | LA_TabPos_In: bFreeIn8 |
| 42200 | LA_SwitchPos_In: bCInh |
| 42201 | LA_SwitchPos_In: bFailReset |
| 42202 | LA_SwitchPos_In: bSetQuickstop |
| 42203 | LA_SwitchPos_In: bSetDCBrake |
| 42204 | LA_SwitchPos_In: bRFG_Stop |
| 42205 | LA_SwitchPos_In: bSetSpeedCcw |
| 42206 | LA_SwitchPos_In: bRLQCw |
| 42207 | LA_SwitchPos_In: bRLQCcw |
| 42208 | LA_SwitchPos_In: bJogCtrlInputSel1 |
| 42209 | LA_SwitchPos_In: bJogCtrlInputSel2 |
| 42210 | LA_SwitchPos_In: bJogCtrlRfgIn |
| 42211 | LA_SwitchPos_In: bJogCtrlJog1 |
| 42212 | LA_SwitchPos_In: bJogCtrlJog2 |
| 42213 | LA_SwitchPos_In: bJogCtrlSlowDown1 |
| 42214 | LA_SwitchPos_In: bJogCtrlStop1 |
| 42215 | LA_SwitchPos_In: bJogCtrlSlowDown2 |
| 42216 | LA_SwitchPos_In: bJogCtrlStop2 |
| 42217 | LA_SwitchPos_In: bJogCtrlSlowDown3 |
| 42218 | LA_SwitchPos_In: bJogCtrlStop3 |
| 42219 | LA_SwitchPos_In: bJogSpeed4 |
| 42220 | LA_SwitchPos_In: bJogSpeed8 |
| 42221 | LA_SwitchPos_In: bJogRamp1 |
| 42222 | LA_SwitchPos_In: bJogRamp2 |
| 42223 | LA_SwitchPos_In: bJogRamp4 |
| 42224 | LA_SwitchPos_In: bJogRamp8 |
| 42225 | LA_SwitchPos_In: bMBrakeRelease |
| 42226 | LA_SwitchPos_In: bGPAnalogSwitchSet |
| 42227 | LA_SwitchPos_In: bGPDigitalDelayIn |
| 42228 | LA_SwitchPos_In: bGPLogicIn1 |
| 42229 | LA_SwitchPos_In: bGPLogicIn2 |
| 42230 | LA_SwitchPos_In: bGPLogicIn3 |
| 42231 | LA_SwitchPos_In: bGPDFlipFlop_InD |
| 42232 | LA_SwitchPos_In: bGPDFlipFlop_InClk |
| 42233 | LA_SwitchPos_In: bGPDFlipFlop_InClr |
| 42239 | LA_SwitchPos_In: bFreeIn1 |
| 42240 | LA_SwitchPos_In: bFreeIn2 |
| 42241 | LA_SwitchPos_In: bFreeIn3 |
| 42242 | LA_SwitchPos_In: bFreeIn4 |
| 42243 | LA_SwitchPos_In: bFreeIn5 |
| 42244 | LA_SwitchPos_In: bFreeIn6 |

| Selection list - digital signals | |
|---|---|
| 42245 | LA_SwitchPos_In: bFreeIn7 |
| 42246 | LA_SwitchPos_In: bFreeIn8 |
| 42247 | LS_MotorInterface: bBrakeChopperActive |
| 42248 | LS_MotorInterface: bLimSpeedTorquemodeOn |

## 17.6 Selection list - angle signals

This selection list is relevant for the following configuration parameters:

| Parameters | |
|---|---|
| C00622 | System connection list: Angle |
| C00712 | LA_TabPos: phi connection list |

| Selection list - angle signals | |
|---|---|
| 0 | Not connected |
| 1005 | LA_NCtrl: dnFreeOut1_p |
| 1006 | LA_NCtrl: dnFreeOut2_p |
| 1100 | LA_TabPos: dnTargetPos_p |
| 1101 | LA_TabPos: dnSetPos_p |
| 1102 | LA_TabPos: dnFreeOut1_p |
| 1103 | LA_TabPos: dnFreeOut2_p |
| 1104 | LA_TabPos: dnPosAct_p |
| 1105 | LA_TabPos: dnDeltaPosAct_p |
| 1205 | LA_SwitchPos: dnFreeOut1_p |
| 1206 | LA_SwitchPos: dnFreeOut2_p |
| 16000 | LP_CanIn1: dnIn34_p |
| 16001 | LP_CanIn2: dnIn34_p |
| 16002 | LP_CanIn3: dnIn34_p |
| 16003 | LP_MciIn: dnIn34_p |
| 16006 | LP_CanIn1: dnIn12_p |
| 16007 | LP_CanIn2: dnIn12_p |
| 16008 | LP_CanIn3: dnIn12_p |
| 16009 | LP_MciIn: dnIn56_p |
| 16010 | LP_MciIn: dnIn78_p |
| 16011 | LP_CanIn4: dnIn12_p |
| 16012 | LP_CanIn4: dnIn34_p |
| 17000 | LS_MultiEncoder: dnPosition_p |
| 17001 | LS_Resolver: dnPosition_p |
| 17010 | LS_BusEncoder: dnPosition_p |
| 17020 | LS_DigitalInput: dnPosIn12_p |
| 17101 | LS_TouchProbe: dnTPDigIn1_Position_p |
| 17102 | LS_TouchProbe: dnTPDigIn2_Position_p |
| 17103 | LS_TouchProbe: dnTPDigIn3_Position_p |
| 17104 | LS_TouchProbe: dnTPDigIn4_Position_p |
| 17105 | LS_TouchProbe: dnTPDigIn5_Position_p |
| 17106 | LS_TouchProbe: dnTPDigIn6_Position_p |
| 17107 | LS_TouchProbe: dnTPDigIn7_Position_p |
| 17108 | LS_TouchProbe: dnTPEncoder_Position_p |
| 17109 | LS_TouchProbe: dnTPResolver_Position_p |
| 17120 | LS_RetainData: dnOut1 |
| 17121 | LS_RetainData: dnOut2 |
| 17122 | LS_RetainData: dnOut3 |
| 17123 | LS_RetainData: dnOut4 |
| 17130 | LS_AxisBusIn: dnLine12 |
| 17131 | LS_AxisBusIn: dnCas12 |
| 17132 | LS_AxisBusIn: dnCas34 |
| 17133 | LS_AxisBusAux: dnAuxIn12 |

| Selection list - angle signals | |
|---|---|
| 17134 | LS_AxisBusAux: dnAuxIn34 |
| 20000 | LS_ParFree_p: dnC474_1_p |
| 20001 | LS_ParFree_p: dnC474_2_p |
| 20002 | LS_ParFree_p: dnC474_3_p |
| 20003 | LS_ParFree_p: dnC474_4_p |
| 20004 | LS_ParFree_p: dnC474_5_p |
| 20005 | LS_ParFree_p: dnC474_6_p |
| 20006 | LS_ParFree_p: dnC474_7_p |
| 20007 | LS_ParFree_p: dnC474_8_p |
| 20008 | LS_ParFree32: dnC479_1 |
| 20009 | LS_ParFree32: dnC479_2 |
| 20010 | LS_ParFree32: dnC479_3 |
| 20011 | LS_ParFree32: dnC479_4 |
| 20012 | LS_ParFree32: dnC479_5 |
| 20013 | LS_ParFree32: dnC479_6 |
| 20014 | LS_ParFree32: dnC479_7 |
| 20015 | LS_ParFree32: dnC479_8 |
| 20016 | LS_ParFreeUnit_1: dnC475_1 |
| 20017 | LS_ParFreeUnit_1: dnC475_2 |
| 20018 | LS_ParFreeUnit_1: dnC475_3 |
| 20019 | LS_ParFreeUnit_1: dnC475_4 |
| 20020 | LS_ParFreeUnit_1: dnC475_5 |
| 20021 | LS_ParFreeUnit_1: dnC475_6 |
| 20022 | LS_ParFreeUnit_1: dnC475_7 |
| 20023 | LS_ParFreeUnit_1: dnC475_8 |
| 20024 | LS_ParFreeUnit_1: dnC475_9 |
| 20025 | LS_ParFreeUnit_1: dnC475_10 |
| 20026 | LS_ParFreeUnit_1: dnC475_11 |
| 20027 | LS_ParFreeUnit_1: dnC475_12 |
| 20028 | LS_ParFreeUnit_1: dnC475_13 |
| 20029 | LS_ParFreeUnit_1: dnC475_14 |
| 20030 | LS_ParFreeUnit_1: dnC475_15 |
| 20031 | LS_ParFreeUnit_1: dnC475_16 |
| 20032 | LS_ParFreeUnit_2: dnC475_17 |
| 20033 | LS_ParFreeUnit_2: dnC475_18 |
| 20034 | LS_ParFreeUnit_2: dnC475_19 |
| 20035 | LS_ParFreeUnit_2: dnC475_20 |
| 20036 | LS_ParFreeUnit_2: dnC475_21 |
| 20037 | LS_ParFreeUnit_2: dnC475_22 |
| 20038 | LS_ParFreeUnit_2: dnC475_23 |
| 20039 | LS_ParFreeUnit_2: dnC475_24 |
| 20040 | LS_ParFreeUnit_2: dnC475_25 |
| 20041 | LS_ParFreeUnit_2: dnC475_26 |
| 20042 | LS_ParFreeUnit_2: dnC475_27 |
| 20043 | LS_ParFreeUnit_2: dnC475_28 |
| 20044 | LS_ParFreeUnit_2: dnC475_29 |
| 20045 | LS_ParFreeUnit_2: dnC475_30 |
| 20046 | LS_ParFreeUnit_2: dnC475_31 |
| 20047 | LS_ParFreeUnit_2: dnC475_32 |
| 32000 | LS_MotorInterface: dnMotorPosAct_p |

| Selection list - angle signals | |
|---|---|
| 32001 | LS_MotorInterface: dnMotorDeltaPosAct_p |
| 32200 | LS_MotionControlKernel: dnPosTarget_p |
| 32201 | LS_MotionControlKernel: dnPosSetValue_p |
| 32202 | LS_MotionControlKernel: dnDeltaPos_p |
| 32203 | LS_MotionControlKernel: dnMotorRefOffset_p |
| 32204 | LS_MotionControlKernel: dnPosSet_p |
| 32205 | LS_MotionControlKernel: dnPosSetRelative_p |
| 36040 | L_ArithmetikPhi_1: dnOut_p |
| 36046 | L_GainOffsetPhiP_1: dnOut_p |
| 36047 | L_GainOffsetPhiP_2: dnOut_p |
| 36050 | L_LimitPhi_1: dnOut_p |
| 36051 | L_LimitPhi_2: dnOut_p |
| 36052 | L_LimitPhi_3: dnOut_p |
| 36060 | L_OffsetGainPhiP_1: dnOut_p |
| 36061 | L_OffsetGainPhiP_2: dnOut_p |
| 36066 | L_PhaseIntK_1: dnOut_p |
| 36067 | L_PhaseIntK_2: dnOut_p |
| 36070 | L_ArithmetikPhi_2: dnOut_p |
| 36071 | L_ArithmetikPhi_3: dnOut_p |
| 36072 | L_SQrt_1: dnOut_p |
| 36073 | L_Mux_1: dnOut_p |
| 36074 | L_ConvWordsToDInt_1: dnOut_p |
| 36075 | L_ConvWordsToDInt_2: dnOut_p |
| 36076 | L_ConvWordsToDInt_3: dnOut_p |
| 36077 | L_ConvUnitsToIncr_1: dnOut_p |
| 36078 | L_ConvUnitsToIncr_2: dnOut_p |
| 36079 | L_ConvUnitsToIncr_3: dnOut_p |
| 36080 | L_Interpolator_1: dnPhiOut_p |
| 36081 | L_MckCtrlInterface_1: dnPosSetOut_p |
| 36082 | L_PhaseDiff_1: dnOut_p |
| 36083 | L_PhaseDiff_2: dnOut_p |
| 36086 | L_MckStateInterface_1: dnPosOut_p |
| 36087 | L_Odometer_1: dnPosOut_1_p |
| 36088 | L_Odometer_1: dnPosOut_2_p |
| 36089 | L_Odometer_1: dnPosOut_3_p |
| 36090 | L_Odometer_1: dnPosOut_4_p |
| 36091 | L_Odometer_1: dnPosOut_5_p |
| 36092 | L_Odometer_1: dnPosOut_6_p |
| 36093 | L_Odometer_1: dnPosOut_7_p |
| 36094 | L_Odometer_1: dnPosOut_8_p |
| 36095 | L_Odometer_1: dnDeltaPos_12_p |
| 36096 | L_Odometer_1: dnDeltaPos_23_p |
| 36097 | L_Odometer_1: dnDeltaPos_34_p |
| 36098 | L_Odometer_1: dnDeltaPos_45_p |
| 36099 | L_Odometer_1: dnDeltaPos_56_p |
| 36100 | L_Odometer_1: dnDeltaPos_67_p |
| 36101 | L_Odometer_1: dnDeltaPos_78_p |
| 36102 | L_Odometer_1: dnDeltaPos_18_p |
| 36103 | L_CalcDiameter_1: dnPos_p |
| 36104 | L_PhiIntegrator_1: dnOut32_p |
| 36105 | L_PosCtrlLin_1: dnPosOut_p |
| 36106 | L_PosCtrlLin_2: dnPosOut_p |

| Selection list - angle signals | |
|---|---|
| 36107 | L_DFSET_1: dnPosDiffOut_p |
| 36108 | L_DFSET_1: dnPosSetOut_p |
| 36109 | L_GearComp_1: dnOut_p |
| 36110 | L_ConvAP_1: dnOut_p |
| 36111 | L_ConvAP_2: dnOut_p |
| 36112 | L_ConvAP_3: dnOut_p |
| 36113 | L_ConvPP_1: dnOut_p |
| 36114 | L_ConvPP_2: dnOut_p |
| 36115 | L_ConvPP_3: dnOut_p |
| 36116 | L_SignalSwitch32_1: dnOut |
| 36117 | L_SignalSwitch32_2: dnOut |
| 36118 | L_SignalSwitch32_3: dnOut |
| 36119 | L_CalcDiameter_1: dwOutDiameter |
| 36120 | L_ArithmetikPhi_4: dnOut_p |
| 36121 | L_ArithmetikPhi_5: dnOut_p |
| 36122 | L_ArithmetikPhi_6: dnOut_p |
| 36123 | L_PhiDiv_1: dnOut_p |
| 36124 | L_PhiAdd_1: dnOut_p |
| 36125 | L_PhiAdd_1: dnOut2_p |
| 36126 | L_DFSET_1: dnDeltaPos_p |
| 42005 | LA_NCtrl_In: dnFreeIn1_p |
| 42006 | LA_NCtrl_In: dnFreeIn2_p |
| 42100 | LA_TabPos_In: dnPosProfilePosition |
| 42101 | LA_TabPos_In: dnFreeIn1_p |
| 42102 | LA_TabPos_In: dnFreeIn2_p |
| 42205 | LA_SwitchPos_In: dnFreeIn1_p |
| 42206 | LA_SwitchPos_In: dnFreeIn2_p |

## 17.7 Table of attributes

The table of attributes contains information required for a communication with the inverter via parameters.

### How to read the table of attributes:

| Column | | Meaning | Entry | |
|---|---|---|---|---|
| Code | | Parameter name | Cxxxxx | |
| Name | | Parameter short text (display text) | *Text* | |
| Type | | Parameter type | Selection list | Value from selection list |
| | | | Bit coded | Bit coded value |
| | | | Linear value | Value with setting range |
| | | | String | String |
| Index | dec | Index under which the parameter is addressed. The subindex for array variables corresponds to the Lenze subcode number. | 24575 - Lenze code number | Is only required for access via a bus system. |
| | hex | | $5FFF_h$ - Lenze code number | |
| Data | DS | Data structure | E | Single variable (only one parameter element) |
| | | | A | Array variable (several parameter elements) |
| | DA | Number of array elements (subcodes) | *Number* | |
| | DT | Data type | INTEGER_16 | 2 bytes with sign |
| | | | INTEGER_32 | 4 bytes with sign |
| | | | UNSIGNED_8 | 1 byte without sign |
| | | | UNSIGNED_16 | 2 bytes without sign |
| | | | UNSIGNED_32 | 4 bytes without sign |
| | | | VISIBLE_STRING [xx] | ASCII string (with character length xx) |
| | Factor | Factor for data transmission via a bus system, depending on the number of decimal positions | *Factor* | 1 ≡ no decimal positions<br>10 ≡ 1 decimal position<br>100 ≡ 2 decimal positions<br>1000 ≡ 3 decimal positions<br>10000 ≡ 4 decimal positions |
| | CINH | Writing is only possible if the controller is inhibited | CINH | |

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00001 | Keypad UserLevel | Selection list | 24574 | 5FFE | A | 1 | UNSIGNED_8 | 1 | |
| C00002 | Device commands | Selection list | 24573 | 5FFD | A | 34 | UNSIGNED_8 | 1 | |
| C00003 | Status of the last device command | Selection list | 24572 | 5FFC | E | 1 | UNSIGNED_8 | 1 | |
| C00005 | Application | Selection list | 24570 | 5FFA | E | 1 | UNSIGNED_16 | 1 | |
| C00006 | Motor control | Selection list | 24569 | 5FF9 | E | 1 | UNSIGNED_8 | 1 | |
| C00007 | Control mode | Selection list | 24568 | 5FF8 | E | 1 | UNSIGNED_16 | 1 | |
| C00008 | Original application\|control source | Selection list | 24567 | 5FF7 | E | 1 | UNSIGNED_16 | 1 | |
| C00010 | AIN1: Characteristic | Linear value | 24565 | 5FF5 | A | 8 | INTEGER_16 | 100 | |
| C00011 | Appl.: Reference speed | Linear value | 24564 | 5FF4 | E | 1 | UNSIGNED_16 | 1 | |
| C00012 | Accel. time - main setpoint | Linear value | 24563 | 5FF3 | E | 1 | UNSIGNED_32 | 1000 | |
| C00013 | Decel. time - main setpoint | Linear value | 24562 | 5FF2 | E | 1 | UNSIGNED_32 | 1000 | |
| C00015 | VFC: V/f base frequency | Linear value | 24560 | 5FF0 | E | 1 | UNSIGNED_16 | 10 | |
| C00016 | VFC: Vmin boost | Linear value | 24559 | 5FEF | E | 1 | UNSIGNED_16 | 100 | |
| C00018 | Switching frequency | Selection list | 24557 | 5FED | E | 1 | UNSIGNED_8 | 1 | |
| C00019 | Auto-DCB: Threshold | Linear value | 24556 | 5FEC | E | 1 | UNSIGNED_16 | 1 | |
| C00020 | AIN2: Characteristic | Linear value | 24555 | 5FEB | A | 8 | INTEGER_16 | 100 | |
| C00021 | Slip comp. | Linear value | 24554 | 5FEA | E | 1 | INTEGER_16 | 100 | |
| C00022 | Imax in motor mode | Linear value | 24553 | 5FE9 | E | 1 | UNSIGNED_16 | 100 | |
| Greyed out = display parameter (read only access) | | | | | | | | | |

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00023 | Imax in generator mode | Linear value | 24552 | 5FE8 | E | 1 | INTEGER_16 | 100 | |
| C00024 | LS_DriveInterface: bNActCompare | Linear value | 24551 | 5FE7 | E | 1 | INTEGER_16 | 100 | |
| C00025 | LS_DriveInterface: bNActHysterese | Linear value | 24550 | 5FE6 | A | 1 | INTEGER_16 | 100 | |
| C00026 | AINx: Offset | Linear value | 24549 | 5FE5 | A | 2 | INTEGER_16 | 100 | |
| C00027 | AINx: Gain | Linear value | 24548 | 5FE4 | A | 2 | INTEGER_32 | 10000 | |
| C00028 | AINx: Input voltage | Linear value | 24547 | 5FE3 | A | 2 | INTEGER_16 | 100 | |
| C00029 | AINx: Input current | Linear value | 24546 | 5FE2 | A | 2 | INTEGER_16 | 100 | |
| C00030 | LS_DFOut: Const. | Linear value | 24545 | 5FE1 | A | 1 | INTEGER_16 | 1 | |
| C00033 | AINx: Output value | Linear value | 24542 | 5FDE | A | 2 | INTEGER_16 | 100 | |
| C00034 | AINx: Configuration | Selection list | 24541 | 5FDD | A | 2 | UNSIGNED_8 | 1 | |
| C00036 | DC braking: Current | Linear value | 24539 | 5FDB | E | 1 | INTEGER_16 | 100 | |
| C00039 | Fixed setpoint x (L_NSet_1 n-Fix) | Linear value | 24536 | 5FD8 | A | 15 | INTEGER_16 | 100 | |
| C00050 | MCTRL: Speed setpoint | Linear value | 24525 | 5FCD | E | 1 | INTEGER_32 | 1 | |
| C00051 | MCTRL: Actual speed value | Linear value | 24524 | 5FCC | E | 1 | INTEGER_32 | 1 | |
| C00052 | Motor voltage | Linear value | 24523 | 5FCB | E | 1 | UNSIGNED_16 | 1 | |
| C00053 | DC-bus voltage | Linear value | 24522 | 5FCA | E | 1 | UNSIGNED_16 | 1 | |
| C00054 | Motor current | Linear value | 24521 | 5FC9 | E | 1 | UNSIGNED_16 | 100 | |
| C00055 | Actual values | Linear value | 24520 | 5FC8 | A | 4 | INTEGER_16 | 1 | |
| C00056 | Torque | Linear value | 24519 | 5FC7 | A | 2 | INTEGER_32 | 100 | |
| C00057 | Maximum torque | Linear value | 24518 | 5FC6 | E | 1 | UNSIGNED_32 | 100 | |
| C00058 | Output frequency | Linear value | 24517 | 5FC5 | E | 1 | INTEGER_32 | 100 | |
| C00059 | Appl.: Reference frequency C11 | Linear value | 24516 | 5FC4 | E | 1 | UNSIGNED_32 | 100 | |
| C00060 | Motor rotor position | Linear value | 24515 | 5FC3 | E | 1 | UNSIGNED_16 | 1 | |
| C00061 | Heatsink temperature | Linear value | 24514 | 5FC2 | E | 1 | INTEGER_16 | 1 | |
| C00062 | Interior temperature | Linear value | 24513 | 5FC1 | A | 1 | INTEGER_16 | 1 | |
| C00063 | Motor temperature | Linear value | 24512 | 5FC0 | A | 3 | INTEGER_16 | 1 | |
| C00064 | Device utilisation (Ixt) | Linear value | 24511 | 5FBF | A | 3 | INTEGER_16 | 100 | |
| C00065 | Supply voltage 24V | Linear value | 24510 | 5FBE | E | 1 | INTEGER_16 | 10 | |
| C00066 | Thermal motor load (I²xt) | Linear value | 24509 | 5FBD | E | 1 | INTEGER_16 | 100 | |
| C00070 | Vp speed controller | Linear value | 24505 | 5FB9 | A | 3 | UNSIGNED_16 | 100 | |
| C00071 | Ti speed controller | Linear value | 24504 | 5FB8 | A | 3 | UNSIGNED_16 | 10 | |
| C00072 | SC: Tdn speed controller | Linear value | 24503 | 5FB7 | E | 1 | UNSIGNED_16 | 100 | |
| C00073 | Imax/M controller gain | Linear value | 24502 | 5FB6 | A | 2 | UNSIGNED_16 | 100 | |
| C00074 | Reset time Imax/M controller | Linear value | 24501 | 5FB5 | A | 2 | UNSIGNED_16 | 1 | |
| C00075 | Vp current controller | Linear value | 24500 | 5FB4 | E | 1 | UNSIGNED_16 | 100 | |
| C00076 | Ti current controller | Linear value | 24499 | 5FB3 | E | 1 | UNSIGNED_16 | 100 | |
| C00077 | SC: Vp field controller | Linear value | 24498 | 5FB2 | E | 1 | UNSIGNED_16 | 100 | |
| C00078 | SC: Tn field controller | Linear value | 24497 | 5FB1 | E | 1 | UNSIGNED_16 | 10 | |
| C00079 | SC: Settings | Selection list | 24496 | 5FB0 | A | 4 | UNSIGNED_8 | 1 | |
| C00080 | Override point of field weakening | Linear value | 24495 | 5FAF | E | 1 | INTEGER_16 | 1 | |
| C00081 | Rated motor power | Linear value | 24494 | 5FAE | E | 1 | UNSIGNED_16 | 100 | |
| C00082 | Motor rotor resistance | Linear value | 24493 | 5FAD | E | 1 | UNSIGNED_32 | 1 | |
| C00083 | Motor rotor time constant | Linear value | 24492 | 5FAC | E | 1 | UNSIGNED_16 | 1 | |
| C00084 | Motor stator resistance | Linear value | 24491 | 5FAB | E | 1 | UNSIGNED_32 | 1 | |
| C00085 | Motor stator leakage inductance | Linear value | 24490 | 5FAA | E | 1 | UNSIGNED_16 | 100 | |
| C00087 | Rated motor speed | Linear value | 24488 | 5FA8 | E | 1 | UNSIGNED_16 | 1 | |
| C00088 | Rated motor current | Linear value | 24487 | 5FA7 | E | 1 | UNSIGNED_16 | 100 | |
| C00089 | Rated motor frequency | Linear value | 24486 | 5FA6 | E | 1 | UNSIGNED_16 | 1 | |
| C00090 | Rated motor voltage | Linear value | 24485 | 5FA5 | E | 1 | UNSIGNED_16 | 1 | |
| C00091 | Motor cosine phi | Linear value | 24484 | 5FA4 | E | 1 | UNSIGNED_8 | 100 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00092 | Motor magnetising inductance | Linear value | 24483 | 5FA3 | E | 1 | UNSIGNED_16 | 10 | |
| C00093 | Power section ID | Linear value | 24482 | 5FA2 | E | 1 | UNSIGNED_16 | 1 | |
| C00095 | Motor magnetising current | Linear value | 24480 | 5FA0 | E | 1 | UNSIGNED_16 | 100 | |
| C00097 | Rated motor torque | Linear value | 24478 | 5F9E | E | 1 | UNSIGNED_32 | 100 | |
| C00098 | Rated device current | Linear value | 24477 | 5F9D | E | 1 | UNSIGNED_16 | 10 | |
| C00099 | Firmware version | String | 24476 | 5F9C | E | 1 | VISIBLE_STRING [12] | | |
| C00100 | Firmware version | Linear value | 24475 | 5F9B | A | 4 | UNSIGNED_8 | 1 | |
| C00101 | Add. acceleration time x | Linear value | 24474 | 5F9A | A | 15 | UNSIGNED_32 | 1000 | |
| C00103 | Add. deceleration time x | Linear value | 24472 | 5F98 | A | 15 | UNSIGNED_32 | 1000 | |
| C00104 | Quick stop setting | Bit coded | 24471 | 5F97 | A | 1 | UNSIGNED_16 | 1 | |
| C00105 | Decel. time - quick stop | Linear value | 24470 | 5F96 | E | 1 | UNSIGNED_32 | 1000 | |
| C00106 | Auto-DCB: Hold time | Linear value | 24469 | 5F95 | E | 1 | UNSIGNED_32 | 1000 | |
| C00107 | DC braking: Hold time | Linear value | 24468 | 5F94 | E | 1 | UNSIGNED_32 | 1000 | |
| C00114 | DigInX: Inversion | Bit coded | 24461 | 5F8D | E | 1 | UNSIGNED_16 | 1 | |
| C00115 | DI 1/2 & 6/7: Function | Selection list | 24460 | 5F8C | A | 2 | UNSIGNED_8 | 1 | |
| C00117 | Status of brake output BD | Selection list | 24458 | 5F8A | E | 1 | UNSIGNED_8 | 1 | |
| C00118 | DigOutX: Inversion | Bit coded | 24457 | 5F89 | E | 1 | UNSIGNED_8 | 1 | |
| C00120 | Setting of motor overload (I²xt) | Linear value | 24455 | 5F87 | E | 1 | INTEGER_16 | 100 | |
| C00121 | Motor temp. warning threshold | Linear value | 24454 | 5F86 | A | 2 | UNSIGNED_16 | 1 | |
| C00122 | Initial value motor overload (I²xt) | Linear value | 24453 | 5F85 | A | 1 | UNSIGNED_16 | 100 | |
| C00123 | Device utilisat. threshold (Ixt) | Linear value | 24452 | 5F84 | E | 1 | INTEGER_16 | 100 | |
| C00124 | Current monitoring: Breaking current | Linear value | 24451 | 5F83 | A | 1 | UNSIGNED_16 | 100 | |
| C00129 | Brake resistance value | Linear value | 24446 | 5F7E | E | 1 | UNSIGNED_16 | 10 | |
| C00130 | Rated brake resistor power | Linear value | 24445 | 5F7D | E | 1 | UNSIGNED_16 | 1 | |
| C00131 | Rated heat amount of brake resist. | Linear value | 24444 | 5F7C | E | 1 | UNSIGNED_16 | 10 | |
| C00133 | Brake resistor utilisation | Linear value | 24442 | 5F7A | E | 1 | UNSIGNED_16 | 1 | |
| C00134 | L_NSet_1: Ramp smoothing | Selection list | 24441 | 5F79 | E | 1 | UNSIGNED_8 | 1 | |
| C00136 | Communication control words | Bit coded | 24439 | 5F77 | A | 2 | UNSIGNED_16 | 1 | |
| C00137 | Device status | Selection list | 24438 | 5F76 | E | 1 | UNSIGNED_16 | 1 | |
| C00138 | Internal control signals | Bit coded | 24437 | 5F75 | A | 3 | UNSIGNED_16 | 1 | |
| C00142 | Auto-start option | Bit coded | 24433 | 5F71 | E | 1 | UNSIGNED_8 | 1 | |
| C00144 | Thermal switching frequency reduction | Selection list | 24431 | 5F6F | E | 1 | UNSIGNED_8 | 1 | |
| C00148 | LS_DriveInterface: Error message config. | Bit coded | 24427 | 5F6B | E | 1 | UNSIGNED_16 | 1 | |
| C00150 | Status word | Bit coded | 24425 | 5F69 | E | 1 | UNSIGNED_16 | 1 | |
| C00155 | Extended status word | Bit coded | 24420 | 5F64 | E | 1 | UNSIGNED_16 | 1 | |
| C00158 | Cause of controller inhibit | Bit coded | 24417 | 5F61 | E | 1 | UNSIGNED_16 | 1 | |
| C00159 | Cause of quick stop QSP | Bit coded | 24416 | 5F60 | E | 1 | UNSIGNED_16 | 1 | |
| C00160 | Status determining error (16-bit) | Linear value | 24415 | 5F5F | A | 1 | UNSIGNED_16 | 1 | |
| C00161 | LS_SetError_x: Error number | Linear value | 24414 | 5F5E | A | 8 | UNSIGNED_16 | 1 | |
| C00162 | Error number masked | Linear value | 24413 | 5F5D | A | 1 | UNSIGNED_32 | 1 | |
| C00163 | Logbook - binary elements | Selection list | 24412 | 5F5C | A | 2 | UNSIGNED_16 | 1 | |
| C00164 | Logbook - analog elements | Selection list | 24411 | 5F5B | A | 1 | UNSIGNED_16 | 1 | |
| C00165 | Error information | String | 24410 | 5F5A | A | 2 | VISIBLE_STRING [14] | | |
| C00166 | Error information text | String | 24409 | 5F59 | A | 6 | VISIBLE_STRING [31] | | |
| C00168 | Status determining error | Linear value | 24407 | 5F57 | E | 1 | UNSIGNED_32 | 1 | |
| C00169 | Logbook setting | Bit coded | 24406 | 5F56 | E | 1 | UNSIGNED_16 | 1 | |
| C00170 | Current error | Linear value | 24405 | 5F55 | E | 1 | UNSIGNED_32 | 1 | |
| C00173 | Mains voltage | Selection list | 24402 | 5F52 | E | 1 | UNSIGNED_8 | 1 | CINH |
| C00174 | Reduced brake chopper threshold | Linear value | 24401 | 5F51 | E | 1 | UNSIGNED_8 | 1 | |
| C00175 | Brake energy management | Selection list | 24400 | 5F50 | E | 1 | UNSIGNED_8 | 1 | CINH |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00177 | Switching cycles | Linear value | 24398 | 5F4E | A | 8 | UNSIGNED_32 | 1 | |
| C00178 | Elapsed-hour meter | Linear value | 24397 | 5F4D | E | 1 | UNSIGNED_32 | 1 | |
| C00179 | Power-on time meter | Linear value | 24396 | 5F4C | E | 1 | UNSIGNED_32 | 1 | |
| C00180 | Running time | Linear value | 24395 | 5F4B | A | 3 | UNSIGNED_32 | 1 | |
| C00181 | Time settings | Linear value | 24394 | 5F4A | A | 1 | UNSIGNED_16 | 1 | |
| C00182 | L_NSet_1: S-ramp time PT1 | Linear value | 24393 | 5F49 | E | 1 | INTEGER_16 | 100 | |
| C00184 | AutoFailReset repetition time | Linear value | 24391 | 5F47 | E | 1 | UNSIGNED_16 | 1 | |
| C00185 | AutoFailReset residual runtime | Linear value | 24390 | 5F46 | E | 1 | UNSIGNED_16 | 1 | |
| C00186 | Max. number of AutoFailReset processes | Linear value | 24389 | 5F45 | E | 1 | UNSIGNED_8 | 1 | |
| C00187 | Current AutoFailReset processes | Linear value | 24388 | 5F44 | E | 1 | UNSIGNED_8 | 1 | |
| C00188 | AutoFailReset configuration | Selection list | 24387 | 5F43 | E | 1 | UNSIGNED_8 | 1 | |
| C00189 | Resp. to too frequent AutoFailReset | Selection list | 24386 | 5F42 | E | 1 | UNSIGNED_8 | 1 | |
| C00190 | L_NSet_1: Setpoint arithmetic | Selection list | 24385 | 5F41 | E | 1 | UNSIGNED_8 | 1 | |
| C00191 | Logbook access index User | Linear value | 24384 | 5F40 | A | 1 | UNSIGNED_8 | 1 | |
| C00192 | Logbook data User | String | 24383 | 5F3F | A | 1 | OCTET_STRING [32] | | |
| C00193 | Logbook element User | Linear value | 24382 | 5F3E | A | 6 | UNSIGNED_32 | 1 | |
| C00199 | Description data | String | 24376 | 5F38 | A | 5 | VISIBLE_STRING [32] | | |
| C00200 | Firmware product type | String | 24375 | 5F37 | E | 1 | VISIBLE_STRING [19] | | |
| C00201 | Firmware | String | 24374 | 5F36 | A | 9 | VISIBLE_STRING [22] | | |
| C00203 | Product type code | String | 24372 | 5F34 | A | 9 | VISIBLE_STRING [24] | | |
| C00204 | Serial number | String | 24371 | 5F33 | A | 9 | VISIBLE_STRING [24] | | |
| C00219 | Identity | Linear value | 24356 | 5F24 | A | 4 | UNSIGNED_32 | 1 | |
| C00220 | L_NSet_1: Acceleration time - add. setpoint | Linear value | 24355 | 5F23 | E | 1 | UNSIGNED_32 | 1000 | |
| C00221 | L_NSet_1: Deceleration time - add. setpoint | Linear value | 24354 | 5F22 | E | 1 | UNSIGNED_32 | 1000 | |
| C00222 | L_PCTRL_1: Vp | Linear value | 24353 | 5F21 | E | 1 | INTEGER_16 | 10 | |
| C00223 | L_PCTRL_1: Tn | Linear value | 24352 | 5F20 | E | 1 | UNSIGNED_16 | 1 | |
| C00224 | L_PCTRL_1: Kd | Linear value | 24351 | 5F1F | E | 1 | UNSIGNED_16 | 10 | |
| C00225 | L_PCTRL_1: MaxLimit | Linear value | 24350 | 5F1E | E | 1 | INTEGER_16 | 100 | |
| C00226 | L_PCTRL_1: MinLimit | Linear value | 24349 | 5F1D | E | 1 | INTEGER_16 | 100 | |
| C00227 | L_PCTRL_1: Acceleration time | Linear value | 24348 | 5F1C | E | 1 | UNSIGNED_32 | 1000 | |
| C00228 | L_PCTRL_1: Deceleration time | Linear value | 24347 | 5F1B | E | 1 | UNSIGNED_32 | 1000 | |
| C00231 | L_PCTRL_1: Operating range | Linear value | 24344 | 5F18 | A | 4 | INTEGER_16 | 100 | |
| C00233 | L_PCTRL_1: Root function | Selection list | 24342 | 5F16 | E | 1 | UNSIGNED_8 | 1 | |
| C00234 | Oscillation damping influence | Linear value | 24341 | 5F15 | E | 1 | UNSIGNED_16 | 100 | |
| C00235 | Oscillation damping filter time | Linear value | 24340 | 5F14 | E | 1 | UNSIGNED_8 | 1 | |
| C00236 | Oscillation damping field weakening | Linear value | 24339 | 5F13 | E | 1 | UNSIGNED_8 | 1 | |
| C00241 | L_NSet_1: Hyst. NSet reached | Linear value | 24334 | 5F0E | E | 1 | INTEGER_16 | 100 | |
| C00242 | L_PCTRL_1: Operating mode | Selection list | 24333 | 5F0D | E | 1 | UNSIGNED_8 | 1 | |
| C00243 | L_PCTRL_1: Accel. time influence | Linear value | 24332 | 5F0C | E | 1 | UNSIGNED_32 | 1000 | |
| C00244 | L_PCTRL_1: Deceleration time influence | Linear value | 24331 | 5F0B | E | 1 | UNSIGNED_32 | 1000 | |
| C00245 | L_PCTRL_1: PID output value | Linear value | 24330 | 5F0A | E | 1 | INTEGER_16 | 100 | |
| C00246 | L_PCTRL_1: nAct_a internal | Linear value | 24329 | 5F09 | E | 1 | INTEGER_16 | 100 | |
| C00247 | L_PCTRL_1: Window setpoint reached | Linear value | 24328 | 5F08 | E | 1 | INTEGER_16 | 100 | |
| C00249 | L_PT1_1: Time constant | Linear value | 24326 | 5F06 | E | 1 | UNSIGNED_16 | 1 | |
| C00250 | L_PT1 2-3: Time constant | Linear value | 24325 | 5F05 | A | 2 | INTEGER_16 | 1 | |
| C00251 | L_DT1_1: Time constant | Linear value | 24324 | 5F04 | E | 1 | INTEGER_16 | 1 | |
| C00252 | L_DT1_1: Gain | Linear value | 24323 | 5F03 | E | 1 | INTEGER_16 | 100 | |
| C00253 | L_DT1_1: Sensitivity | Selection list | 24322 | 5F02 | E | 1 | UNSIGNED_8 | 1 | |
| C00254 | Kp position controller | Linear value | 24321 | 5F01 | E | 1 | UNSIGNED_16 | 100 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00270 | SC: Freq. current setpoint filter | Linear value | 24305 | 5EF1 | E | 1 | UNSIGNED_16 | 10 | |
| C00271 | SC: Current setpoint filter width | Linear value | 24304 | 5EF0 | E | 1 | UNSIGNED_16 | 10 | |
| C00272 | SC: Current setpoint filter depth | Linear value | 24303 | 5EEF | E | 1 | UNSIGNED_16 | 1 | |
| C00273 | Motor moment of inertia | Linear value | 24302 | 5EEE | E | 1 | UNSIGNED_32 | 100 | |
| C00274 | SC: Max. change in acceleration | Linear value | 24301 | 5EED | E | 1 | UNSIGNED_16 | 10 | |
| C00275 | Setpoint feedforward control filtering | Linear value | 24300 | 5EEC | E | 1 | UNSIGNED_16 | 10 | |
| C00276 | SC: max. output voltage | Linear value | 24299 | 5EEB | E | 1 | UNSIGNED_8 | 1 | |
| C00280 | SC: Filter time const. DC detection | Linear value | 24295 | 5EE7 | E | 1 | UNSIGNED_16 | 1 | |
| C00312 | System runtimes | Linear value | 24263 | 5EC7 | A | 1 | UNSIGNED_32 | 1000 | |
| C00321 | Main program runtime | Linear value | 24254 | 5EBE | A | 2 | UNSIGNED_16 | 1 | |
| C00322 | Transmission mode CAN TxPDOs | Linear value | 24253 | 5EBD | A | 4 | UNSIGNED_8 | 1 | |
| C00323 | Transmission mode CAN Rx PDOs | Linear value | 24252 | 5EBC | A | 4 | UNSIGNED_8 | 1 | |
| C00324 | CAN transmit blocking time | Linear value | 24251 | 5EBB | A | 5 | UNSIGNED_16 | 1 | |
| C00338 | L_Arithmetik_1: Function | Selection list | 24237 | 5EAD | E | 1 | UNSIGNED_8 | 1 | |
| C00339 | L_Arithmetik_2: Function | Selection list | 24236 | 5EAC | E | 1 | UNSIGNED_8 | 1 | |
| C00341 | CAN management - error configuration | Bit coded | 24234 | 5EAA | E | 1 | UNSIGNED_16 | 1 | |
| C00342 | CAN decoupling PDOInOut | Bit coded | 24233 | 5EA9 | A | 2 | UNSIGNED_16 | 1 | |
| C00343 | LP_CanIn decoupling value | Linear value | 24232 | 5EA8 | A | 16 | UNSIGNED_16 | 1 | |
| C00344 | LP_CanOut decoupling value | Linear value | 24231 | 5EA7 | A | 16 | UNSIGNED_16 | 1 | |
| C00345 | CAN error status | Selection list | 24230 | 5EA6 | E | 1 | UNSIGNED_8 | 1 | |
| C00347 | CAN status HeartBeat producer | Selection list | 24228 | 5EA4 | A | 15 | UNSIGNED_8 | 1 | |
| C00349 | CAN setting - DIP switch | Bit coded | 24226 | 5EA2 | E | 1 | UNSIGNED_16 | 1 | |
| C00350 | CAN node address | Linear value | 24225 | 5EA1 | E | 1 | UNSIGNED_8 | 1 | |
| C00351 | CAN baud rate | Selection list | 24224 | 5EA0 | E | 1 | UNSIGNED_8 | 1 | |
| C00352 | CAN slave/master | Selection list | 24223 | 5E9F | E | 1 | UNSIGNED_8 | 1 | |
| C00353 | CAN IN/OUT COBID source | Selection list | 24222 | 5E9E | A | 4 | UNSIGNED_8 | 1 | |
| C00354 | COBID | Bit coded | 24221 | 5E9D | A | 8 | UNSIGNED_32 | 1 | |
| C00355 | Active COBID | Linear value | 24220 | 5E9C | A | 8 | UNSIGNED_16 | 1 | |
| C00356 | CAN time settings | Linear value | 24219 | 5E9B | A | 6 | UNSIGNED_16 | 1 | |
| C00357 | CAN monitoring times | Linear value | 24218 | 5E9A | A | 4 | UNSIGNED_16 | 1 | |
| C00358 | CANx_OUT data length | Linear value | 24217 | 5E99 | A | 4 | UNSIGNED_8 | 1 | |
| C00359 | CAN status | Selection list | 24216 | 5E98 | E | 1 | UNSIGNED_8 | 1 | |
| C00360 | CAN telegram counter | Linear value | 24215 | 5E97 | A | 14 | UNSIGNED_16 | 1 | |
| C00364 | CAN MessageError | Selection list | 24211 | 5E93 | E | 1 | UNSIGNED_8 | 1 | |
| C00366 | Number of CAN SDO channels | Selection list | 24209 | 5E91 | E | 1 | UNSIGNED_8 | 1 | |
| C00367 | CAN SYNC Rx identifier | Bit coded | 24208 | 5E90 | E | 1 | UNSIGNED_16 | 1 | |
| C00368 | CAN SYNC Tx identifier | Bit coded | 24207 | 5E8F | E | 1 | UNSIGNED_16 | 1 | |
| C00369 | CAN sync transmission cycle time | Linear value | 24206 | 5E8E | E | 1 | UNSIGNED_16 | 1 | |
| C00370 | SyncTxRxTimes | Linear value | 24205 | 5E8D | A | 2 | INTEGER_16 | 1 | |
| C00371 | CAN ErrorCode | Linear value | 24204 | 5E8C | A | 1 | UNSIGNED_16 | 1 | |
| C00372 | CAN_Tx_Rx_Error | Linear value | 24203 | 5E8B | A | 4 | UNSIGNED_8 | 1 | |
| C00381 | CAN Heartbeat producer time | Linear value | 24194 | 5E82 | E | 1 | UNSIGNED_16 | 1 | |
| C00385 | CAN node addr. HeartBeat producer | Linear value | 24190 | 5E7E | A | 15 | UNSIGNED_8 | 1 | |
| C00386 | CAN HeartBeat-ConsumerTime | Linear value | 24189 | 5E7D | A | 15 | UNSIGNED_16 | 1 | |
| C00387 | CAN-GatewayAddr | Linear value | 24188 | 5E7C | A | 1 | UNSIGNED_8 | 1 | |
| C00400 | LS_PulseGenerator | Linear value | 24175 | 5E6F | A | 3 | UNSIGNED_16 | 1 | |
| C00401 | CANxInOut: Inversion | Bit coded | 24174 | 5E6E | A | 8 | UNSIGNED_16 | 1 | |
| C00407 | LP_CanIn mapping | Linear value | 24168 | 5E68 | A | 8 | UNSIGNED_32 | 1 | |
| C00408 | LP_CanIn mapping selection | Selection list | 24167 | 5E67 | A | 4 | UNSIGNED_8 | 1 | |
| C00409 | LP_CanIn mapping | Linear value | 24166 | 5E66 | A | 16 | UNSIGNED_16 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00410 | L_SignalMonitor_a: Signal sources | Selection list | 24165 | 5E65 | A | 4 | UNSIGNED_16 | 1 | |
| C00411 | L_SignalMonitor_b: Signal sources | Selection list | 24164 | 5E64 | A | 4 | UNSIGNED_16 | 1 | |
| C00412 | L_SignalMonitor_b: Inversion | Bit coded | 24163 | 5E63 | E | 1 | UNSIGNED_8 | 1 | |
| C00413 | L_SignalMonitor_a: Offs./gain | Linear value | 24162 | 5E62 | A | 8 | INTEGER_16 | 100 | |
| C00417 | Deactivate resolver error comp. | Selection list | 24158 | 5E5E | E | 1 | UNSIGNED_8 | 1 | |
| C00420 | Number of encoder increments | Linear value | 24155 | 5E5B | A | 3 | UNSIGNED_16 | 1 | |
| C00421 | LS_MultiEncoder: Supply voltage | Linear value | 24154 | 5E5A | E | 1 | UNSIGNED_16 | 10 | |
| C00422 | LS_MultiEncoder: Encoder type | Selection list | 24153 | 5E59 | E | 1 | UNSIGNED_8 | 1 | CINH |
| C00423 | DOx: Delay times | Linear value | 24152 | 5E58 | A | 10 | UNSIGNED_16 | 1000 | |
| C00424 | Pulse form TTL encoder | Selection list | 24151 | 5E57 | E | 1 | UNSIGNED_8 | 1 | |
| C00425 | Encoder scanning time | Selection list | 24150 | 5E56 | A | 2 | UNSIGNED_8 | 1 | CINH |
| C00426 | SSI encoder: Data bits | Linear value | 24149 | 5E55 | A | 9 | UNSIGNED_8 | 1 | CINH |
| C00427 | SSI encoder: Bit rate | Selection list | 24148 | 5E54 | E | 1 | UNSIGNED_8 | 1 | CINH |
| C00428 | SSI encoder: Coding | Selection list | 24147 | 5E53 | E | 1 | UNSIGNED_8 | 1 | CINH |
| C00430 | LS_MultiEncoder: Max. initialization time | Linear value | 24145 | 5E51 | A | 1 | UNSIGNED_16 | 1 | CINH |
| C00431 | LS_MultiEncoder: bFail selection | Bit coded | 24144 | 5E50 | A | 1 | UNSIGNED_16 | 1 | |
| C00432 | LS_Resolver: bFail selection | Bit coded | 24143 | 5E4F | A | 1 | UNSIGNED_16 | 1 | |
| C00434 | OxU/I: Gain | Linear value | 24141 | 5E4D | A | 4 | INTEGER_16 | 100 | |
| C00435 | OxU/I: Offset | Linear value | 24140 | 5E4C | A | 4 | INTEGER_16 | 100 | |
| C00436 | OxU: Voltage | Linear value | 24139 | 5E4B | A | 2 | INTEGER_16 | 100 | |
| C00437 | OxI: Current | Linear value | 24138 | 5E4A | A | 2 | INTEGER_32 | 1000 | |
| C00439 | OxU/I: Input value | Linear value | 24136 | 5E48 | A | 4 | INTEGER_16 | 100 | |
| C00440 | LS_AnalogIn1: PT1 time constant | Linear value | 24135 | 5E47 | A | 2 | UNSIGNED_16 | 1 | |
| C00441 | Decoupling AnalogOut | Bit coded | 24134 | 5E46 | E | 1 | UNSIGNED_16 | 1 | |
| C00442 | AOutx: Decoupling value | Linear value | 24133 | 5E45 | A | 4 | INTEGER_16 | 100 | |
| C00443 | DIx: Level | Bit coded | 24132 | 5E44 | A | 2 | UNSIGNED_16 | 1 | |
| C00444 | DOx: Level | Bit coded | 24131 | 5E43 | A | 2 | UNSIGNED_16 | 1 | |
| C00445 | FreqInxx_nOut_v | Linear value | 24130 | 5E42 | A | 2 | INTEGER_16 | 1 | |
| C00446 | FreqInxx_nOut_a | Linear value | 24129 | 5E41 | A | 2 | INTEGER_16 | 100 | |
| C00447 | DigOut decoupling | Bit coded | 24128 | 5E40 | E | 1 | UNSIGNED_16 | 1 | |
| C00448 | DigOut decoupling value | Bit coded | 24127 | 5E3F | E | 1 | UNSIGNED_16 | 1 | |
| C00449 | FreqInxx_dnOut_p | Linear value | 24126 | 5E3E | A | 1 | INTEGER_32 | 1 | |
| C00451 | LS_MultiEncoder: Information bFail | Bit coded | 24124 | 5E3C | A | 1 | UNSIGNED_16 | 1 | |
| C00452 | LS_Resolver: Information bFail | Bit coded | 24123 | 5E3B | A | 1 | UNSIGNED_16 | 1 | |
| C00453 | Keypad: Default manual jog | Linear value | 24122 | 5E3A | A | 2 | UNSIGNED_32 | 1000 | |
| C00461 | Remote: Acceleration/deceleration time | Linear value | 24114 | 5E32 | A | 1 | UNSIGNED_32 | 1000 | |
| C00462 | Remote: Control | Linear value | 24113 | 5E31 | A | 2 | UNSIGNED_16 | 1 | |
| C00463 | Remote: MCK control | Bit coded | 24112 | 5E30 | A | 1 | UNSIGNED_32 | 1 | |
| C00464 | Remote: Monitoring timeout | Linear value | 24111 | 5E2F | A | 1 | UNSIGNED_16 | 1 | |
| C00465 | Keypad: Time-out welcome screen | Selection list | 24110 | 5E2E | E | 1 | INTEGER_32 | 1 | |
| C00466 | Keypad: Default parameter | Linear value | 24109 | 5E2D | E | 1 | INTEGER_32 | 1 | |
| C00467 | Keypad: Default welcome screen | Selection list | 24108 | 5E2C | E | 1 | INTEGER_32 | 1 | |
| C00469 | Keypad: Fct. STOP key | Selection list | 24106 | 5E2A | E | 1 | INTEGER_32 | 1 | CINH |
| C00470 | LS_ParFree_b | Selection list | 24105 | 5E29 | A | 32 | UNSIGNED_8 | 1 | |
| C00471 | LS_ParFree | Bit coded | 24104 | 5E28 | A | 32 | UNSIGNED_16 | 1 | |
| C00472 | LS_ParFree_a | Linear value | 24103 | 5E27 | A | 16 | INTEGER_16 | 100 | |
| C00473 | LS_ParFree_v | Linear value | 24102 | 5E26 | A | 8 | INTEGER_16 | 1 | |
| C00474 | LS_ParFree_p | Linear value | 24101 | 5E25 | A | 8 | INTEGER_32 | 1 | |
| C00475 | LS_ParFreeUnit_1_2 | Linear value | 24100 | 5E24 | A | 32 | INTEGER_32 | 10000 | |
| C00476 | LS_ParFree_a_2 | Linear value | 24099 | 5E23 | A | 16 | INTEGER_16 | 100 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00477 | LS_ParFree_2 | Bit coded | 24098 | 5E22 | A | 32 | UNSIGNED_16 | 1 | |
| C00478 | LS_ParFree_v_2 | Linear value | 24097 | 5E21 | A | 8 | INTEGER_16 | 1 | |
| C00479 | LS_ParFree32 | Linear value | 24096 | 5E20 | A | 8 | INTEGER_32 | 1 | |
| C00480 | LS_DisFree_b | Bit coded | 24095 | 5E1F | E | 1 | UNSIGNED_16 | 1 | |
| C00481 | LS_DisFree | Bit coded | 24094 | 5E1E | A | 8 | UNSIGNED_16 | 1 | |
| C00482 | LS_DisFree_a | Linear value | 24093 | 5E1D | A | 8 | INTEGER_16 | 100 | |
| C00483 | LS_DisFree_p | Linear value | 24092 | 5E1C | A | 8 | INTEGER_32 | 1 | |
| C00484 | Application units: Offset | Linear value | 24091 | 5E1B | A | 4 | INTEGER_16 | 100 | |
| C00485 | Application units: Display factor | Linear value | 24090 | 5E1A | A | 4 | INTEGER_32 | 10000 | |
| C00486 | Application units: Text | String | 24089 | 5E19 | A | 4 | VISIBLE_STRING [7] | | |
| C00487 | Application units | Linear value | 24088 | 5E18 | A | 4 | INTEGER_32 | 100 | |
| C00488 | L_JogCtrlEdgeDetect_1 | Selection list | 24087 | 5E17 | A | 6 | UNSIGNED_8 | 1 | |
| C00490 | Position encoder selection | Selection list | 24085 | 5E15 | E | 1 | UNSIGNED_8 | 1 | |
| C00491 | Hiperface messages | Linear value | 24084 | 5E14 | A | 4 | UNSIGNED_8 | 1 | |
| C00492 | Hiperface: Detected TypCode | Linear value | 24083 | 5E13 | E | 1 | UNSIGNED_8 | 1 | |
| C00493 | Hiperface: TypCode | Linear value | 24082 | 5E12 | E | 1 | UNSIGNED_8 | 1 | |
| C00494 | Hiperface: Resolutions | Linear value | 24081 | 5E11 | A | 2 | UNSIGNED_32 | 1 | |
| C00495 | Speed sensor selection | Selection list | 24080 | 5E10 | E | 1 | UNSIGNED_8 | 1 | |
| C00496 | Encoder evaluation method DigIn12 | Selection list | 24079 | 5E0F | E | 1 | UNSIGNED_8 | 1 | CINH |
| C00497 | Filter time constant | Linear value | 24078 | 5E0E | A | 4 | UNSIGNED_16 | 10 | |
| C00498 | Open-circuit monitoring | Selection list | 24077 | 5E0D | A | 1 | UNSIGNED_8 | 1 | |
| C00499 | Hiperface: Settings | Bit coded | 24076 | 5E0C | A | 1 | UNSIGNED_16 | 1 | |
| C00505 | Password data | String | 24070 | 5E06 | A | 3 | VISIBLE_STRING [16] | | |
| C00506 | PW protection internal config | Bit coded | 24069 | 5E05 | A | 1 | UNSIGNED_16 | 1 | |
| C00507 | Current password protection | Bit coded | 24068 | 5E04 | A | 1 | UNSIGNED_16 | 1 | |
| C00508 | PW protection config X6(DIAG) | Bit coded | 24067 | 5E03 | A | 1 | UNSIGNED_16 | 1 | |
| C00509 | PW protection config X1 (CAN) | Bit coded | 24066 | 5E02 | A | 1 | UNSIGNED_16 | 1 | |
| C00510 | PW protection config. MCI | Bit coded | 24065 | 5E01 | A | 1 | UNSIGNED_16 | 1 | |
| C00517 | User menu | Linear value | 24058 | 5DFA | A | 32 | INTEGER_32 | 1000 | |
| C00540 | LS_DFOut: Function | Selection list | 24035 | 5DE3 | E | 1 | UNSIGNED_8 | 1 | |
| C00545 | LS_DFOut: Angular offset | Linear value | 24030 | 5DDE | A | 1 | INTEGER_32 | 1 | |
| C00560 | Fan switching status | Selection list | 24015 | 5DCF | A | 2 | UNSIGNED_8 | 1 | |
| C00561 | Failure indication | Selection list | 24014 | 5DCE | A | 5 | UNSIGNED_8 | 1 | |
| C00563 | Current monitoring: Delay time | Linear value | 24012 | 5DCC | A | 1 | UNSIGNED_32 | 1000 | |
| C00565 | Resp. to mains phase failure | Selection list | 24010 | 5DCA | E | 1 | UNSIGNED_8 | 1 | |
| C00566 | Resp. to fan failure | Selection list | 24009 | 5DC9 | E | 1 | UNSIGNED_8 | 1 | |
| C00567 | Resp. to speed controller limited | Selection list | 24008 | 5DC8 | E | 1 | UNSIGNED_8 | 1 | |
| C00569 | Resp. to peak current | Selection list | 24006 | 5DC6 | A | 1 | UNSIGNED_8 | 1 | |
| C00570 | Resp. to controller limitations | Selection list | 24005 | 5DC5 | A | 4 | UNSIGNED_8 | 1 | |
| C00571 | Resp. to wrong controller setting | Selection list | 24004 | 5DC4 | A | 2 | UNSIGNED_8 | 1 | |
| C00572 | Brake resistor overload threshold | Linear value | 24003 | 5DC3 | E | 1 | UNSIGNED_8 | 1 | |
| C00574 | Resp. to brake resist. overtemp. | Selection list | 24001 | 5DC1 | E | 1 | UNSIGNED_8 | 1 | |
| C00576 | SC: Optimisation of field feedforward control | Linear value | 23999 | 5DBF | E | 1 | UNSIGNED_16 | 1 | |
| C00577 | SC: Vp field weakening controller | Linear value | 23998 | 5DBE | E | 1 | UNSIGNED_16 | 10000 | |
| C00578 | SC: Tn field weakening controller | Linear value | 23997 | 5DBD | E | 1 | UNSIGNED_16 | 10 | |
| C00579 | Resp. to max. speed/output freq. reached | Selection list | 23996 | 5DBC | E | 1 | UNSIGNED_8 | 1 | |
| C00580 | Resp. to operating system error | Selection list | 23995 | 5DBB | A | 1 | UNSIGNED_8 | 1 | |
| C00581 | Resp. to LS_SetError_x | Selection list | 23994 | 5DBA | A | 8 | UNSIGNED_8 | 1 | |
| C00582 | Resp. to heatsink temp.> shutdown temp. -5°C | Selection list | 23993 | 5DB9 | E | 1 | UNSIGNED_8 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00583 | Resp. to motor temperature KTY | Selection list | 23992 | 5DB8 | A | 6 | UNSIGNED_8 | 1 | |
| C00584 | Resp. to current monitoring | Selection list | 23991 | 5DB7 | A | 1 | UNSIGNED_8 | 1 | |
| C00585 | Resp. to motor overtemp. PTC | Selection list | 23990 | 5DB6 | E | 1 | UNSIGNED_8 | 1 | |
| C00586 | Resp. open circuit HTL encoder | Selection list | 23989 | 5DB5 | E | 1 | UNSIGNED_8 | 1 | |
| C00588 | Resp. to max. speed at switching freq. | Selection list | 23987 | 5DB3 | E | 1 | UNSIGNED_8 | 1 | |
| C00590 | Resp. to switch. frequency red. | Selection list | 23985 | 5DB1 | E | 1 | UNSIGNED_8 | 1 | |
| C00591 | Reakt. Achsbusfehler | Selection list | 23984 | 5DB0 | A | 2 | UNSIGNED_8 | 1 | |
| C00592 | Resp. to CAN bus connection | Selection list | 23983 | 5DAF | A | 5 | UNSIGNED_8 | 1 | |
| C00593 | Resp. to CANx_IN monitoring | Selection list | 23982 | 5DAE | A | 4 | UNSIGNED_8 | 1 | |
| C00594 | Resp. to control word error | Selection list | 23981 | 5DAD | A | 2 | UNSIGNED_8 | 1 | |
| C00595 | MCK: Resp. to MCK error | Selection list | 23980 | 5DAC | A | 15 | UNSIGNED_8 | 1 | |
| C00597 | Resp. to motor phase failure | Selection list | 23978 | 5DAA | E | 1 | UNSIGNED_8 | 1 | |
| C00598 | Resp. to open circuit AINx | Selection list | 23977 | 5DA9 | A | 2 | UNSIGNED_8 | 1 | |
| C00599 | Motor phase failure threshold | Linear value | 23976 | 5DA8 | E | 1 | INTEGER_16 | 100 | |
| C00600 | Resp. to DC bus voltage | Selection list | 23975 | 5DA7 | A | 1 | UNSIGNED_8 | 1 | |
| C00601 | Delayed resp. to fault: DC bus overvoltage | Linear value | 23974 | 5DA6 | A | 1 | UNSIGNED_16 | 1000 | |
| C00602 | Resp. to earth fault | Selection list | 23973 | 5DA5 | E | 1 | UNSIGNED_8 | 1 | |
| C00603 | Resp. to feedback | Selection list | 23972 | 5DA4 | A | 8 | UNSIGNED_8 | 1 | |
| C00604 | Resp. to device overload (Ixt) | Selection list | 23971 | 5DA3 | E | 1 | UNSIGNED_8 | 1 | |
| C00605 | Resp. to feedback | Selection list | 23970 | 5DA2 | A | 1 | UNSIGNED_8 | 1 | |
| C00606 | Resp. to motor overload (I²xt) | Selection list | 23969 | 5DA1 | E | 1 | UNSIGNED_8 | 1 | |
| C00607 | Resp. to max. freq. feedb. DIG12/67 | Selection list | 23968 | 5DA0 | E | 1 | UNSIGNED_8 | 1 | |
| C00608 | Resp. to maximum torque | Selection list | 23967 | 5D9F | E | 1 | UNSIGNED_8 | 1 | |
| C00609 | Resp. to maximum current | Selection list | 23966 | 5D9E | E | 1 | UNSIGNED_8 | 1 | |
| C00620 | System connection list: 16-bit | Selection list | 23955 | 5D93 | A | 111 | UNSIGNED_16 | 1 | |
| C00621 | System connection list: Bool | Selection list | 23954 | 5D92 | A | 199 | UNSIGNED_16 | 1 | |
| C00622 | System connection list: Angle | Selection list | 23953 | 5D91 | A | 25 | UNSIGNED_16 | 1 | |
| C00630 | L_Limit 1-2: Min/Max | Linear value | 23945 | 5D89 | A | 4 | INTEGER_16 | 100 | |
| C00631 | L_LimitPhi 1-3: Min/Max | Linear value | 23944 | 5D88 | A | 6 | INTEGER_32 | 1 | |
| C00632 | L_NSet_1: Max. skip freq. | Linear value | 23943 | 5D87 | A | 3 | INTEGER_16 | 100 | |
| C00633 | L_NSet_1: Min. skip freq. | Linear value | 23942 | 5D86 | A | 3 | INTEGER_16 | 100 | |
| C00634 | L_NSet_1: wState | Bit coded | 23941 | 5D85 | E | 1 | UNSIGNED_16 | 1 | |
| C00635 | L_NSet_1: nMaxLimit | Linear value | 23940 | 5D84 | E | 1 | INTEGER_16 | 100 | |
| C00636 | L_NSet_1: nMinLimit | Linear value | 23939 | 5D83 | E | 1 | INTEGER_16 | 100 | |
| C00637 | L_NSet_1: Output blocking zones | Linear value | 23938 | 5D82 | E | 1 | INTEGER_16 | 100 | |
| C00638 | L_NSet_1: Output ramp rounding | Linear value | 23937 | 5D81 | E | 1 | INTEGER_16 | 100 | |
| C00639 | L_NSet_1: Output add.value | Linear value | 23936 | 5D80 | E | 1 | INTEGER_16 | 100 | |
| C00640 | L_NSet_1: nNOut_a | Linear value | 23935 | 5D7F | E | 1 | INTEGER_16 | 100 | |
| C00643 | Resp. to PLI monitoring | Selection list | 23932 | 5D7C | A | 1 | UNSIGNED_8 | 1 | |
| C00644 | PLI traversing direction | Selection list | 23931 | 5D7B | A | 1 | UNSIGNED_8 | 1 | |
| C00645 | PLI max. permissible deflection | Linear value | 23930 | 5D7A | A | 2 | INTEGER_16 | 10 | |
| C00646 | PLI current amplitude | Linear value | 23929 | 5D79 | A | 2 | UNSIGNED_16 | 1 | |
| C00647 | PLI ramp time | Linear value | 23928 | 5D78 | A | 2 | UNSIGNED_16 | 1 | |
| C00650 | L_Arithmetik 3-5: Function | Selection list | 23925 | 5D75 | A | 3 | UNSIGNED_8 | 1 | |
| C00653 | Sensibility - Setpoint feedforward control | Selection list | 23922 | 5D72 | A | 1 | UNSIGNED_8 | 1 | |
| C00654 | Source of diff. setpoint feedforward control | Selection list | 23921 | 5D71 | A | 1 | UNSIGNED_8 | 1 | |
| C00660 | L_FixSet_a_1: Analog values | Linear value | 23915 | 5D6B | A | 16 | INTEGER_16 | 100 | |
| C00661 | L_FixSet_w_1: Fixed values | Linear value | 23914 | 5D6A | A | 16 | UNSIGNED_16 | 1 | |
| C00662 | L_FixSet_w_2: Fixed values | Linear value | 23913 | 5D69 | A | 16 | UNSIGNED_16 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00670 | L_OffsetGainP_1: Gain | Linear value | 23905 | 5D61 | E | 1 | INTEGER_32 | 10000 | |
| C00671 | L_OffsetGainP_2: Gain | Linear value | 23904 | 5D60 | E | 1 | INTEGER_32 | 10000 | |
| C00672 | L_OffsetGainP_3: Gain | Linear value | 23903 | 5D5F | E | 1 | INTEGER_32 | 10000 | |
| C00673 | L_OffsetGainPhiP 1-2: Offset | Linear value | 23902 | 5D5E | A | 2 | INTEGER_32 | 1 | |
| C00674 | L_OffsetGainPhiP 1-2: Gain | Linear value | 23901 | 5D5D | A | 2 | INTEGER_32 | 1 | |
| C00677 | L_GainOffsetP 1-3: Parameter | Linear value | 23898 | 5D5A | A | 6 | INTEGER_16 | 100 | |
| C00678 | L_GainOffsetPhiP 1-2: Parameter | Linear value | 23897 | 5D59 | A | 4 | INTEGER_32 | 1 | |
| C00679 | L_MulDiv_2: Parameter | Linear value | 23896 | 5D58 | A | 2 | INTEGER_16 | 1 | |
| C00680 | L_Compare_1: Fct. | Selection list | 23895 | 5D57 | E | 1 | UNSIGNED_8 | 1 | |
| C00681 | L_Compare_1: Hysteresis | Linear value | 23894 | 5D56 | E | 1 | INTEGER_16 | 100 | |
| C00682 | L_Compare_1: Window | Linear value | 23893 | 5D55 | E | 1 | INTEGER_16 | 100 | |
| C00685 | L_Compare_2: Fct. | Selection list | 23890 | 5D52 | E | 1 | UNSIGNED_8 | 1 | |
| C00686 | L_Compare_2: Hysteresis | Linear value | 23889 | 5D51 | E | 1 | INTEGER_16 | 100 | |
| C00687 | L_Compare_2: Window | Linear value | 23888 | 5D50 | E | 1 | INTEGER_16 | 100 | |
| C00690 | L_Compare_3: Fct. | Selection list | 23885 | 5D4D | E | 1 | UNSIGNED_8 | 1 | |
| C00691 | L_Compare_3: Hysteresis | Linear value | 23884 | 5D4C | E | 1 | INTEGER_16 | 100 | |
| C00692 | L_Compare_3: Window | Linear value | 23883 | 5D4B | E | 1 | INTEGER_16 | 100 | |
| C00693 | L_Compare 4-5: Fct. | Selection list | 23882 | 5D4A | A | 2 | UNSIGNED_8 | 1 | |
| C00694 | L_Compare 4-5: Hysteresis | Linear value | 23881 | 5D49 | A | 2 | INTEGER_16 | 100 | |
| C00695 | L_Compare 4-5: Window | Linear value | 23880 | 5D48 | A | 2 | INTEGER_16 | 100 | |
| C00696 | L_OffsetGainP_1: Offset | Linear value | 23879 | 5D47 | E | 1 | INTEGER_16 | 100 | |
| C00697 | L_OffsetGainP_2: Offset | Linear value | 23878 | 5D46 | E | 1 | INTEGER_16 | 100 | |
| C00698 | L_OffsetGainP_3: Offset | Linear value | 23877 | 5D45 | E | 1 | INTEGER_16 | 100 | |
| C00699 | L_MulDiv_1: Parameter | Linear value | 23876 | 5D44 | A | 2 | INTEGER_16 | 1 | |
| C00700 | LA_NCtrl: Analog connection list | Selection list | 23875 | 5D43 | A | 29 | UNSIGNED_16 | 1 | |
| C00701 | LA_NCtrl: Digital connection list | Selection list | 23874 | 5D42 | A | 48 | UNSIGNED_16 | 1 | |
| C00710 | LA_TabPos: Analog connection list | Selection list | 23865 | 5D39 | A | 31 | UNSIGNED_16 | 1 | |
| C00711 | LA_TabPos: Digital connection list | Selection list | 23864 | 5D38 | A | 56 | UNSIGNED_16 | 1 | |
| C00712 | LA_TabPos: phi connection list | Selection list | 23863 | 5D37 | A | 3 | UNSIGNED_16 | 1 | |
| C00720 | L_DigitalDelay_1: Delay | Linear value | 23855 | 5D2F | A | 2 | UNSIGNED_32 | 1000 | |
| C00721 | L_DigitalDelay 2..3: Delay | Linear value | 23854 | 5D2E | A | 4 | UNSIGNED_32 | 1000 | |
| C00725 | Current switching frequency | Selection list | 23850 | 5D2A | E | 1 | UNSIGNED_8 | 1 | |
| C00727 | LS_Keypad digital values | Linear value | 23848 | 5D28 | A | 8 | UNSIGNED_8 | 1 | |
| C00728 | Analog values - keypad | Linear value | 23847 | 5D27 | A | 3 | INTEGER_16 | 100 | |
| C00729 | Remote: Setpoint selection | Linear value | 23846 | 5D26 | A | 2 | INTEGER_16 | 100 | |
| C00760 | LA_SwitchPos: Analog connection list | Selection list | 23815 | 5D07 | A | 25 | UNSIGNED_16 | 1 | |
| C00761 | LA_SwitchPos: Digital connection list | Selection list | 23814 | 5D06 | A | 47 | UNSIGNED_16 | 1 | |
| C00800 | L_MPot_1: Upper limit | Linear value | 23775 | 5CDF | E | 1 | INTEGER_16 | 100 | |
| C00801 | L_MPot_1: Lower limit | Linear value | 23774 | 5CDE | E | 1 | INTEGER_16 | 100 | |
| C00802 | L_MPot_1: Acceleration time | Linear value | 23773 | 5CDD | E | 1 | UNSIGNED_16 | 10 | |
| C00803 | L_MPot_1: Deceleration time | Linear value | 23772 | 5CDC | E | 1 | UNSIGNED_16 | 10 | |
| C00804 | L_MPot_1: Inactive fct. | Selection list | 23771 | 5CDB | E | 1 | UNSIGNED_8 | 1 | |
| C00805 | L_MPot_1: Init fct. | Selection list | 23770 | 5CDA | E | 1 | UNSIGNED_8 | 1 | |
| C00806 | L_MPot_1: Use | Selection list | 23769 | 5CD9 | E | 1 | UNSIGNED_8 | 1 | |
| C00807 | L_NLim_1: Max.SkipFrq. | Linear value | 23768 | 5CD8 | A | 3 | INTEGER_16 | 100 | |
| C00808 | L_NLim_1: Min.SkipFrq. | Linear value | 23767 | 5CD7 | A | 3 | INTEGER_16 | 100 | |
| C00809 | L_NLim_2: Max.SkipFrq. | Linear value | 23766 | 5CD6 | A | 3 | INTEGER_16 | 100 | |
| C00810 | L_NLim_2: Min.SkipFrq. | Linear value | 23765 | 5CD5 | A | 3 | INTEGER_16 | 100 | |
| C00811 | L_NLim_1: Current output value | Linear value | 23764 | 5CD4 | A | 2 | INTEGER_16 | 100 | |
| C00812 | L_NLim: Current status | Bit coded | 23763 | 5CD3 | A | 2 | UNSIGNED_16 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00820 | L_DigitalLogic_1: Function | Selection list | 23755 | 5CCB | E | 1 | UNSIGNED_8 | 1 | |
| C00821 | L_DigitalLogic_1: Truth table | Selection list | 23754 | 5CCA | A | 24 | UNSIGNED_8 | 1 | |
| C00822 | L_DigitalLogic_2: Function | Selection list | 23753 | 5CC9 | E | 1 | UNSIGNED_8 | 1 | |
| C00823 | L_DigitalLogic_2: Truth table | Selection list | 23752 | 5CC8 | A | 24 | UNSIGNED_8 | 1 | |
| C00824 | L_DigitalLogic5_1: Function | Selection list | 23751 | 5CC7 | E | 1 | UNSIGNED_8 | 1 | |
| C00825 | L_DigitalLogic5_1: Truth table | Selection list | 23750 | 5CC6 | A | 32 | UNSIGNED_8 | 1 | |
| C00826 | L_DigitalLogic5_2: Function | Selection list | 23749 | 5CC5 | E | 1 | UNSIGNED_8 | 1 | |
| C00827 | L_DigitalLogic5_2: Truth table | Selection list | 23748 | 5CC4 | A | 32 | UNSIGNED_8 | 1 | |
| C00828 | L_DigitalLogic_3: Function | Selection list | 23747 | 5CC3 | E | 1 | UNSIGNED_8 | 1 | |
| C00829 | L_DigitalLogic_3: truth table | Selection list | 23746 | 5CC2 | A | 24 | UNSIGNED_8 | 1 | |
| C00830 | 16-bit inputs [%] | Linear value | 23745 | 5CC1 | A | 105 | INTEGER_16 | 100 | |
| C00831 | 16-bit inputs | Bit coded | 23744 | 5CC0 | A | 105 | UNSIGNED_16 | 1 | |
| C00832 | 16-bit inputs [incr./ms] | Linear value | 23743 | 5CBF | A | 105 | INTEGER_16 | 1 | |
| C00833 | Binary inputs | Selection list | 23742 | 5CBE | A | 131 | UNSIGNED_8 | 1 | |
| C00834 | 32-bit inputs [incr] | Linear value | 23741 | 5CBD | A | 12 | INTEGER_32 | 1 | |
| C00835 | 16-bit inputs [%] (Set2) | Linear value | 23740 | 5CBC | A | 162 | INTEGER_16 | 100 | |
| C00836 | 16-bit inputs (Set2) | Bit coded | 23739 | 5CBB | A | 162 | UNSIGNED_16 | 1 | |
| C00837 | 16-bit inputs [incr./ms] (Set2) | Linear value | 23738 | 5CBA | A | 162 | INTEGER_16 | 1 | |
| C00838 | Binary inputs (Set2) | Selection list | 23737 | 5CB9 | A | 213 | UNSIGNED_8 | 1 | |
| C00839 | 32-bit inputs [incr] (Set2) | Linear value | 23736 | 5CB8 | A | 96 | INTEGER_32 | 1 | |
| C00840 | 16-bit inputs I/O level [%] | Linear value | 23735 | 5CB7 | A | 111 | INTEGER_16 | 100 | |
| C00841 | 16-bit inputs I/O level | Bit coded | 23734 | 5CB6 | A | 111 | UNSIGNED_16 | 1 | |
| C00843 | Binary inputs I/O level | Selection list | 23732 | 5CB4 | A | 199 | UNSIGNED_8 | 1 | |
| C00844 | 32-bit inputs I/O level [incr] | Linear value | 23731 | 5CB3 | A | 25 | INTEGER_32 | 1 | |
| C00866 | CAN input words | Bit coded | 23709 | 5C9D | A | 16 | UNSIGNED_16 | 1 | |
| C00868 | CAN output words | Bit coded | 23707 | 5C9B | A | 16 | UNSIGNED_16 | 1 | |
| C00876 | MCI input words | Bit coded | 23699 | 5C93 | A | 16 | UNSIGNED_16 | 1 | |
| C00877 | MCI output words | Bit coded | 23698 | 5C92 | A | 16 | UNSIGNED_16 | 1 | |
| C00890 | MCI_InOut: Inversion | Bit coded | 23685 | 5C85 | A | 4 | UNSIGNED_16 | 1 | |
| C00905 | Motor phase direction of rotation | Selection list | 23670 | 5C76 | E | 1 | UNSIGNED_8 | 1 | CINH |
| C00909 | Speed limitation | Linear value | 23666 | 5C72 | A | 2 | INTEGER_16 | 100 | |
| C00910 | Frequency limitation | Linear value | 23665 | 5C71 | A | 2 | UNSIGNED_16 | 1 | |
| C00915 | Motor cable length | Linear value | 23660 | 5C6C | E | 1 | UNSIGNED_16 | 10 | |
| C00916 | Motor cable cross-section | Linear value | 23659 | 5C6B | E | 1 | UNSIGNED_16 | 100 | |
| C00917 | Motor cable resistance | Linear value | 23658 | 5C6A | E | 1 | UNSIGNED_16 | 1 | |
| C00918 | SC: Start motor magnetising current | Linear value | 23657 | 5C69 | E | 1 | UNSIGNED_8 | 1 | |
| C00919 | Load moment of inertia | Linear value | 23656 | 5C68 | A | 1 | UNSIGNED_32 | 100 | |
| C00920 | Rated device currents | Linear value | 23655 | 5C67 | A | 8 | UNSIGNED_16 | 10 | |
| C00925 | LS_Resolver: Number of pole pairs | Linear value | 23650 | 5C62 | E | 1 | UNSIGNED_8 | 1 | |
| C00926 | Pole position | Linear value | 23649 | 5C61 | A | 3 | INTEGER_16 | 10 | |
| C00927 | Motor rotor position | Linear value | 23648 | 5C60 | E | 1 | UNSIGNED_16 | 1 | |
| C00937 | Field-oriented motor currents | Linear value | 23638 | 5C56 | A | 2 | INTEGER_16 | 100 | |
| C00938 | PSM: Maximum motor current field weakening | Linear value | 23637 | 5C55 | E | 1 | UNSIGNED_16 | 100 | |
| C00939 | Ultimate motor current | Linear value | 23636 | 5C54 | E | 1 | UNSIGNED_16 | 10 | |
| C00940 | L_ConvW numerator | Linear value | 23635 | 5C53 | A | 4 | INTEGER_16 | 1 | |
| C00941 | L_ConvW denominator | Linear value | 23634 | 5C52 | A | 4 | INTEGER_16 | 1 | |
| C00942 | L_ConvW conversion method | Selection list | 23633 | 5C51 | A | 4 | UNSIGNED_8 | 1 | |
| C00950 | L_Interpolator_1: Activation FB functions | Selection list | 23625 | 5C49 | A | 3 | UNSIGNED_8 | 1 | |
| C00951 | L_Interpolator_1: No. of interpolation steps | Linear value | 23624 | 5C48 | E | 1 | UNSIGNED_16 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C00952 | L_Interpolator_1: Limit value - error cycles | Linear value | 23623 | 5C47 | E | 1 | UNSIGNED_16 | 1 | |
| C00953 | L_Interpolator_1: Speed-up | Linear value | 23622 | 5C46 | E | 1 | UNSIGNED_8 | 1 | |
| C00959 | L_Curve: Current output value | Linear value | 23616 | 5C40 | A | 3 | INTEGER_16 | 100 | |
| C00960 | L_Curve_1: Selected curve type | Selection list | 23615 | 5C3F | A | 1 | UNSIGNED_8 | 1 | |
| C00961 | L_Curve_1: Input limitation | Linear value | 23614 | 5C3E | A | 2 | INTEGER_16 | 100 | |
| C00963 | L_Curve_1: Table X-values | Linear value | 23612 | 5C3C | A | 32 | INTEGER_16 | 1 | |
| C00964 | L_Curve_1: Table Y-values | Linear value | 23611 | 5C3B | A | 32 | INTEGER_16 | 1 | |
| C00965 | Max. motor speed | Linear value | 23610 | 5C3A | E | 1 | UNSIGNED_16 | 1 | |
| C00966 | VFC: Time const. slip comp. | Linear value | 23609 | 5C39 | E | 1 | UNSIGNED_16 | 1 | |
| C00967 | VFC: Frequency interpol. point n | Linear value | 23608 | 5C38 | A | 11 | INTEGER_16 | 10 | |
| C00968 | VFC: Voltage interpol. point n | Linear value | 23607 | 5C37 | A | 11 | UNSIGNED_16 | 100 | |
| C00969 | Motor parameters | Linear value | 23606 | 5C36 | A | 1 | UNSIGNED_8 | 1 | |
| C00971 | VFC: V/f +encoder limitation | Linear value | 23604 | 5C34 | A | 2 | UNSIGNED_16 | 100 | |
| C00972 | VFC: Vp V/f +encoder | Linear value | 23603 | 5C33 | E | 1 | UNSIGNED_16 | 1000 | |
| C00973 | VFC: Ti V/f +encoder | Linear value | 23602 | 5C32 | E | 1 | UNSIGNED_16 | 10 | |
| C00975 | VFC-ECO: Vp CosPhi controller | Linear value | 23600 | 5C30 | E | 1 | UNSIGNED_16 | 1000 | |
| C00976 | VFC-ECO: Ti CosPhi controller | Linear value | 23599 | 5C2F | E | 1 | UNSIGNED_16 | 10 | |
| C00977 | VFC-ECO: Minimum voltage V/f | Linear value | 23598 | 5C2E | E | 1 | INTEGER_16 | 100 | |
| C00978 | VFC-ECO: Voltage reduction | Linear value | 23597 | 5C2D | E | 1 | INTEGER_16 | 1 | |
| C00979 | Cosine phi | Linear value | 23596 | 5C2C | A | 2 | INTEGER_16 | 100 | |
| C00980 | Output power | Linear value | 23595 | 5C2B | A | 2 | INTEGER_32 | 1000 | |
| C00981 | Energy display | Linear value | 23594 | 5C2A | A | 2 | INTEGER_32 | 100 | |
| C00982 | VFC-ECO: Voltage reduction ramp | Linear value | 23593 | 5C29 | E | 1 | UNSIGNED_8 | 10 | |
| C00983 | Delay | Linear value | 23592 | 5C28 | A | 2 | UNSIGNED_16 | 1 | |
| C00985 | SLVC: Gain of field current controller | Linear value | 23590 | 5C26 | E | 1 | INTEGER_16 | 100 | |
| C00986 | SLVC: Gain of cross current controller | Linear value | 23589 | 5C25 | E | 1 | INTEGER_16 | 100 | |
| C00987 | Inverter motor brake: nAdd | Linear value | 23588 | 5C24 | E | 1 | INTEGER_16 | 1 | |
| C00988 | Inverter motor brake: PT1 filter time | Linear value | 23587 | 5C23 | E | 1 | INTEGER_16 | 10 | |
| C00989 | Restart on the fly: Flying restart frequency fd_add | Linear value | 23586 | 5C22 | A | 1 | INTEGER_16 | 100 | |
| C00990 | Flying restart fct.: Activate | Selection list | 23585 | 5C21 | E | 1 | UNSIGNED_8 | 1 | CINH |
| C00991 | Flying restart fct.: Process | Selection list | 23584 | 5C20 | E | 1 | UNSIGNED_16 | 1 | |
| C00992 | Flying restart: Start frequency | Linear value | 23583 | 5C1F | E | 1 | INTEGER_16 | 1 | |
| C00993 | Flying restart: Integration time | Linear value | 23582 | 5C1E | E | 1 | UNSIGNED_16 | 10 | |
| C00994 | Flying restart: Current | Linear value | 23581 | 5C1D | E | 1 | INTEGER_16 | 100 | |
| C00995 | SLPSM: Controlled current setpoint | Linear value | 23580 | 5C1C | A | 2 | UNSIGNED_16 | 100 | |
| C00996 | SLPSM: Switching speed | Linear value | 23579 | 5C1B | A | 2 | INTEGER_16 | 100 | |
| C00997 | SLPSM: Filter cutoff frequency | Linear value | 23578 | 5C1A | E | 1 | INTEGER_16 | 100 | |
| C00998 | SLPSM: Filter time rotor position | Linear value | 23577 | 5C19 | A | 2 | INTEGER_16 | 10 | |
| C00999 | SLPSM: PLL gain | Linear value | 23576 | 5C18 | E | 1 | INTEGER_16 | 1 | |
| C01000 | MCTRL: Status | Bit coded | 23575 | 5C17 | E | 1 | UNSIGNED_16 | 1 | |
| C01001 | Manual entry of motor type | Selection list | 23574 | 5C16 | A | 1 | UNSIGNED_8 | 1 | |
| C01010 | L_ArithmetikPhi 1-3: Function | Selection list | 23565 | 5C0D | A | 3 | UNSIGNED_8 | 1 | |
| C01011 | L_ArithmetikPhi 4-6: Function | Selection list | 23564 | 5C0C | A | 3 | UNSIGNED_8 | 1 | |
| C01012 | L_PhiDiv: byDivision | Linear value | 23563 | 5C0B | A | 1 | INTEGER_8 | 1 | |
| C01020 | L_Odometer_1: Memory length | Selection list | 23555 | 5C03 | E | 1 | UNSIGNED_8 | 1 | |
| C01021 | L_Odometer_1: Memory type | Selection list | 23554 | 5C02 | E | 1 | UNSIGNED_8 | 1 | |
| C01022 | L_Odometer_1: Input selection | Selection list | 23553 | 5C01 | E | 1 | UNSIGNED_8 | 1 | |
| C01023 | L_Odometer_1: Edge selection | Selection list | 23552 | 5C00 | E | 1 | UNSIGNED_8 | 1 | |
| C01025 | L_Curve_2: Selected curve type | Selection list | 23550 | 5BFE | A | 1 | UNSIGNED_8 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C01026 | L_Curve_2: Input limitation | Linear value | 23549 | 5BFD | A | 2 | INTEGER_16 | 100 | |
| C01028 | L_Curve_2: Table X-values | Linear value | 23547 | 5BFB | A | 32 | INTEGER_16 | 1 | |
| C01029 | L_Curve_2: Table Y-values | Linear value | 23546 | 5BFA | A | 32 | INTEGER_16 | 1 | |
| C01030 | L_Curve_3: Selected curve type | Selection list | 23545 | 5BF9 | A | 1 | UNSIGNED_8 | 1 | |
| C01031 | L_Curve_3: Input limitation | Linear value | 23544 | 5BF8 | A | 2 | INTEGER_16 | 100 | |
| C01033 | L_Curve_3: Table X-values | Linear value | 23542 | 5BF6 | A | 32 | INTEGER_16 | 1 | |
| C01034 | L_Curve_3: Table Y-values | Linear value | 23541 | 5BF5 | A | 32 | INTEGER_16 | 1 | |
| C01035 | L_Curve_3: SelectCurve | Selection list | 23540 | 5BF4 | A | 1 | UNSIGNED_16 | 1 | |
| C01040 | L_SRFG_1..2 linear ramp time | Linear value | 23535 | 5BEF | A | 2 | UNSIGNED_32 | 1000 | |
| C01041 | L_SRFG_1..2 S-ramp time | Linear value | 23534 | 5BEE | A | 2 | UNSIGNED_32 | 1000 | |
| C01042 | L_SRFG_1..2 limitations of output values | Linear value | 23533 | 5BED | A | 4 | INTEGER_16 | 100 | |
| C01045 | L_ConvAP 1-3: numerator/denominator | Linear value | 23530 | 5BEA | A | 6 | INTEGER_16 | 1 | |
| C01046 | L_ConvPA 1-3: byDivision | Linear value | 23529 | 5BE9 | A | 3 | INTEGER_8 | 1 | |
| C01047 | L_GearComp_1: Offset | Linear value | 23528 | 5BE8 | A | 1 | INTEGER_16 | 1 | |
| C01048 | L_GearComp_1: Num_Denom | Linear value | 23527 | 5BE7 | A | 2 | INTEGER_16 | 1 | |
| C01049 | L_CalcDiameter_1: Status | Linear value | 23526 | 5BE6 | A | 1 | INTEGER_16 | 1 | |
| C01050 | L_CalcDiameter_1: Diameter recalculation | Linear value | 23525 | 5BE5 | A | 2 | UNSIGNED_32 | 1000 | |
| C01051 | L_CalcDiameter_1: Filter time constant | Linear value | 23524 | 5BE4 | A | 1 | UNSIGNED_16 | 1000 | |
| C01052 | L_CalcDiameter_1: Web break monitoring | Linear value | 23523 | 5BE3 | A | 1 | INTEGER_16 | 100 | |
| C01053 | L_ProcessCtrl_1: Controller times | Linear value | 23522 | 5BE2 | A | 5 | UNSIGNED_16 | 1000 | |
| C01054 | L_ProcessCtrl_1: System deviation | Linear value | 23521 | 5BE1 | A | 2 | INTEGER_16 | 100 | |
| C01055 | L_ProcessCtrl_1: Correcting variable limitation | Selection list | 23520 | 5BE0 | A | 1 | UNSIGNED_8 | 1 | |
| C01056 | L_ProcessCtrl_1: Controller gain | Linear value | 23519 | 5BDF | A | 1 | UNSIGNED_16 | 100 | |
| C01057 | L_CalcDiameter_1: Current diameter | Linear value | 23518 | 5BDE | A | 1 | UNSIGNED_32 | 1000 | |
| C01058 | L_PosCtrlLin_1-2: bDisable | Selection list | 23517 | 5BDD | A | 2 | UNSIGNED_8 | 1 | |
| C01059 | L_PosCtrlLin 1-2: Positioning behaviour | Selection list | 23516 | 5BDC | A | 2 | UNSIGNED_8 | 1 | |
| C01060 | L_PosCtrlLin 1-2: Ramps | Linear value | 23515 | 5BDB | A | 6 | INTEGER_32 | 1000 | |
| C01061 | L_PosCtrlLin 1-2: Traversing speed | Linear value | 23514 | 5BDA | A | 4 | INTEGER_16 | 1 | |
| C01062 | L_SwitchPoint_1: Dead time | Linear value | 23513 | 5BD9 | A | 8 | UNSIGNED_16 | 1 | |
| C01063 | L_SwitchPoint_1: Hysteresis | Linear value | 23512 | 5BD8 | A | 8 | UNSIGNED_16 | 1 | |
| C01064 | L_SwitchPoint_1: CenterMode | Selection list | 23511 | 5BD7 | A | 8 | UNSIGNED_8 | 1 | |
| C01065 | L_SwitchPoint_1: Running time | Linear value | 23510 | 5BD6 | A | 8 | UNSIGNED_16 | 1 | |
| C01066 | L_SwitchPoint_1: Status | Selection list | 23509 | 5BD5 | A | 8 | INTEGER_16 | 1 | |
| C01067 | Inversion of gearbox stages | Selection list | 23508 | 5BD4 | A | 3 | UNSIGNED_8 | 1 | |
| C01068 | L_SwitchPoint_1: Invert | Bit coded | 23507 | 5BD3 | A | 1 | UNSIGNED_16 | 1 | |
| C01069 | L_DFSET_1: Ramp settings | Linear value | 23506 | 5BD2 | A | 2 | UNSIGNED_16 | 1 | |
| C01070 | L_DFSET_1: Angular trimming | Linear value | 23505 | 5BD1 | A | 1 | INTEGER_32 | 1 | |
| C01071 | L_DFSET_1: Following error limit | Linear value | 23504 | 5BD0 | A | 1 | UNSIGNED_32 | 1 | |
| C01072 | L_DFSET_1: Multiplier - angular trimming | Linear value | 23503 | 5BCF | A | 1 | INTEGER_16 | 1 | |
| C01073 | L_DFSET_1: Adjustment | Bit coded | 23502 | 5BCE | A | 1 | UNSIGNED_8 | 1 | |
| C01074 | L_DFSET_1: Zero pulse divider | Linear value | 23501 | 5BCD | A | 2 | UNSIGNED_16 | 1 | |
| C01075 | L_DFSET_1: Synchronisation mode | Selection list | 23500 | 5BCC | A | 1 | UNSIGNED_8 | 1 | |
| C01076 | L_DFRFG_1: Times | Linear value | 23499 | 5BCB | A | 2 | UNSIGNED_32 | 1000 | |
| C01077 | L_DFRFG_1: Max. speed-up | Linear value | 23498 | 5BCA | A | 1 | INTEGER_16 | 1 | |
| C01078 | L_DFRFG_1: Following error | Linear value | 23497 | 5BC9 | A | 1 | UNSIGNED_32 | 1 | |
| C01079 | L_DFRFG_1: Synchronisation window | Linear value | 23496 | 5BC8 | A | 1 | UNSIGNED_16 | 1 | |
| C01080 | L_DFRFG_1: Offset | Linear value | 23495 | 5BC7 | A | 1 | INTEGER_32 | 1 | |
| C01081 | L_DFRFG_1: Sync. direction / TP function | Selection list | 23494 | 5BC6 | A | 1 | UNSIGNED_8 | 1 | |
| C01082 | LS_WriteParamList: Execute Mode | Selection list | 23493 | 5BC5 | E | 1 | UNSIGNED_8 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C01083 | LS_WriteParamList: Error status | Linear value | 23492 | 5BC4 | E | 1 | UNSIGNED_16 | 1 | |
| C01084 | LS_WriteParamList: Error line | Linear value | 23491 | 5BC3 | E | 1 | UNSIGNED_8 | 1 | |
| C01085 | LS_WriteParamList: Index | Linear value | 23490 | 5BC2 | A | 32 | INTEGER_32 | 1000 | |
| C01086 | LS_WriteParamList: WriteValue_1 | Linear value | 23489 | 5BC1 | A | 32 | INTEGER_32 | 1 | |
| C01087 | LS_WriteParamList: WriteValue_2 | Linear value | 23488 | 5BC0 | A | 32 | INTEGER_32 | 1 | |
| C01088 | LS_WriteParamList: WriteValue_3 | Linear value | 23487 | 5BBF | A | 32 | INTEGER_32 | 1 | |
| C01089 | LS_WriteParamList: WriteValue_4 | Linear value | 23486 | 5BBE | A | 32 | INTEGER_32 | 1 | |
| C01090 | LS_ParReadWrite 1-6: Index | Linear value | 23485 | 5BBD | A | 6 | INTEGER_32 | 1000 | |
| C01091 | LS_ParReadWrite 1-6: Cycle time | Selection list | 23484 | 5BBC | A | 6 | UNSIGNED_16 | 1 | |
| C01092 | LS_ParReadWrite 1-6: FailState | Linear value | 23483 | 5BBB | A | 6 | UNSIGNED_16 | 1 | |
| C01093 | LS_ParReadWrite 1-6: Arithmetic mode | Selection list | 23482 | 5BBA | A | 6 | UNSIGNED_8 | 1 | |
| C01094 | LS_ParReadWrite 1-6: Numerator | Linear value | 23481 | 5BB9 | A | 6 | INTEGER_16 | 1 | |
| C01095 | LS_ParReadWrite 1-6: Denominator | Linear value | 23480 | 5BB8 | A | 6 | INTEGER_16 | 1 | |
| C01098 | LS_ParReadWrite 1-6: Configuration | Bit coded | 23477 | 5BB5 | A | 6 | UNSIGNED_16 | 1 | |
| C01100 | Function L_Counter 1-3 | Selection list | 23475 | 5BB3 | A | 3 | UNSIGNED_8 | 1 | |
| C01101 | Comparison L_Counter 1-3 | Selection list | 23474 | 5BB2 | A | 3 | UNSIGNED_8 | 1 | |
| C01108 | L_SwitchPoint_1: Deadtime factor | Selection list | 23467 | 5BAB | A | 8 | UNSIGNED_8 | 1 | |
| C01109 | L_SwitchPointPar_1: Deadtime factor | Selection list | 23466 | 5BAA | A | 8 | UNSIGNED_8 | 1 | |
| C01110 | LS_MultiEncoder: Solid measure | Selection list | 23465 | 5BA9 | E | 1 | UNSIGNED_8 | 1 | |
| C01111 | LS_MultiEncoder: Encoder constant | Linear value | 23464 | 5BA8 | A | 3 | INTEGER_32 | 1 | |
| C01112 | LS_MultiEncoder: Position values | Linear value | 23463 | 5BA7 | A | 4 | INTEGER_32 | 10000 | |
| C01119 | LS_MultiEncoder: Current position | Linear value | 23456 | 5BA0 | A | 2 | INTEGER_32 | 10000 | |
| C01120 | Sync signal source | Selection list | 23455 | 5B9F | E | 1 | UNSIGNED_8 | 1 | |
| C01121 | Sync cycle time setpoint | Linear value | 23454 | 5B9E | E | 1 | UNSIGNED_16 | 1 | |
| C01122 | Sync phase position | Linear value | 23453 | 5B9D | E | 1 | UNSIGNED_16 | 1 | |
| C01123 | Sync window | Linear value | 23452 | 5B9C | E | 1 | UNSIGNED_16 | 1 | |
| C01124 | Sync correction width | Selection list | 23451 | 5B9B | E | 1 | UNSIGNED_8 | 1 | |
| C01130 | LS_RetainData: Selection | Bit coded | 23445 | 5B95 | A | 4 | UNSIGNED_16 | 1 | |
| C01131 | LS_RetainData: 16Bit data | Linear value | 23444 | 5B94 | A | 4 | UNSIGNED_16 | 1 | |
| C01132 | LS_RetainData: 32Bit data | Linear value | 23443 | 5B93 | A | 4 | INTEGER_32 | 1 | |
| C01133 | LS_RetainData: Bool data | Selection list | 23442 | 5B92 | A | 4 | UNSIGNED_8 | 1 | |
| C01134 | LS_RetainData: 16Bit data | Linear value | 23441 | 5B91 | A | 4 | UNSIGNED_16 | 1 | |
| C01135 | LS_RetainData: 32Bit data | Linear value | 23440 | 5B90 | A | 4 | INTEGER_32 | 1 | |
| C01136 | LS_RetainData: Bool data | Selection list | 23439 | 5B8F | A | 4 | UNSIGNED_8 | 1 | |
| C01138 | L_Transient 1-4: Function | Selection list | 23437 | 5B8D | A | 4 | UNSIGNED_8 | 1 | |
| C01139 | L_Transient 1-4: Pulse duration | Linear value | 23436 | 5B8C | A | 4 | UNSIGNED_16 | 1000 | |
| C01140 | L_Transient 5-8: Function | Selection list | 23435 | 5B8B | A | 4 | UNSIGNED_8 | 1 | |
| C01141 | L_Transient 5-8 pulse duration | Linear value | 23434 | 5B8A | A | 4 | UNSIGNED_16 | 1000 | |
| C01150 | L_PhaseIntK: Function | Selection list | 23425 | 5B81 | A | 2 | UNSIGNED_8 | 1 | |
| C01151 | L_PhaseIntK: Compare | Linear value | 23424 | 5B80 | A | 2 | INTEGER_32 | 1 | |
| C01152 | L_SwitchPointPar_1: Dead time | Linear value | 23423 | 5B7F | A | 8 | UNSIGNED_16 | 1 | |
| C01153 | L_SwitchPointPar_1: Hysteresis | Linear value | 23422 | 5B7E | A | 8 | UNSIGNED_16 | 1 | |
| C01154 | L_SwitchPointPar_1: CenterMode | Selection list | 23421 | 5B7D | A | 8 | UNSIGNED_8 | 1 | |
| C01155 | L_SwitchPointPar_1: Running time | Linear value | 23420 | 5B7C | A | 8 | UNSIGNED_16 | 1 | |
| C01156 | L_SwitchPointPar_1: Status | Selection list | 23419 | 5B7B | A | 8 | INTEGER_16 | 1 | |
| C01157 | L_SwitchPointPar_1:Position | Linear value | 23418 | 5B7A | A | 16 | INTEGER_32 | 10000 | |
| C01158 | L_SwitchPointPar_1: Invert | Bit coded | 23417 | 5B79 | A | 1 | UNSIGNED_16 | 1 | |
| C01190 | Motor thermal sensor | Selection list | 23385 | 5B59 | A | 2 | UNSIGNED_8 | 1 | |
| C01191 | PTC characteristic: Temperature 1/2 | Linear value | 23384 | 5B58 | A | 2 | UNSIGNED_8 | 1 | |
| C01192 | PTC characteristic: Resistance 1/2 | Linear value | 23383 | 5B57 | A | 2 | INTEGER_32 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C01193 | Motor temp. feedback system | Selection list | 23382 | 5B56 | E | 1 | UNSIGNED_8 | 1 | |
| C01201 | MCK: Cycle | Linear value | 23374 | 5B4E | A | 1 | INTEGER_32 | 10000 | |
| C01202 | MCK: iM motor/process | Linear value | 23373 | 5B4D | A | 2 | UNSIGNED_16 | 1 | |
| C01203 | MCK: iG motor/position encoder | Linear value | 23372 | 5B4C | A | 2 | UNSIGNED_16 | 1 | |
| C01204 | MCK: Feed constant | Linear value | 23371 | 5B4B | E | 1 | INTEGER_32 | 10000 | |
| C01205 | MCK: Position resolution | Linear value | 23370 | 5B4A | E | 1 | INTEGER_32 | 10000 | |
| C01206 | MCK: Mounting direction | Selection list | 23369 | 5B49 | A | 2 | UNSIGNED_8 | 1 | CINH |
| C01210 | MCK: Current positions | Linear value | 23365 | 5B45 | A | 9 | INTEGER_32 | 10000 | |
| C01211 | MCK: Speed | Linear value | 23364 | 5B44 | A | 3 | INTEGER_32 | 10000 | |
| C01213 | MCK: Max. traversing distance | Linear value | 23362 | 5B42 | A | 1 | INTEGER_32 | 1 | |
| C01215 | MCK: Following error | Linear value | 23360 | 5B40 | A | 3 | INTEGER_32 | 10000 | |
| C01216 | MCK: Positioning setting | Bit coded | 23359 | 5B3F | E | 1 | UNSIGNED_8 | 1 | |
| C01218 | MCK: Position follower setting | Bit coded | 23357 | 5B3D | E | 1 | UNSIGNED_8 | 1 | |
| C01219 | MCK: Speed follower setting | Bit coded | 23356 | 5B3C | E | 1 | UNSIGNED_8 | 1 | |
| C01220 | MCK: Ref. setting | Bit coded | 23355 | 5B3B | A | 1 | UNSIGNED_16 | 1 | |
| C01221 | MCK: Homing mode | Selection list | 23354 | 5B3A | E | 1 | UNSIGNED_8 | 1 | |
| C01222 | MCK: Ref. M limit mode 14/15 | Linear value | 23353 | 5B39 | E | 1 | INTEGER_16 | 100 | |
| C01223 | MCK: Ref. waiting time mode 14/15 | Linear value | 23352 | 5B38 | E | 1 | UNSIGNED_16 | 1 | |
| C01224 | MCK: Ref. speeds | Linear value | 23351 | 5B37 | A | 2 | INTEGER_32 | 10000 | |
| C01225 | MCK: Ref. accelerations | Linear value | 23350 | 5B36 | A | 2 | INTEGER_32 | 10000 | |
| C01226 | MCK: Ref. S-ramp time | Linear value | 23349 | 5B35 | A | 1 | UNSIGNED_16 | 1000 | |
| C01227 | MCK: Ref. positions | Linear value | 23348 | 5B34 | A | 2 | INTEGER_32 | 10000 | |
| C01228 | MCK: Ref. sequence profile | Linear value | 23347 | 5B33 | E | 1 | UNSIGNED_8 | 1 | |
| C01229 | MCK: Position limiting values | Linear value | 23346 | 5B32 | A | 2 | INTEGER_32 | 10000 | |
| C01230 | Manual jog: Setting | Bit coded | 23345 | 5B31 | E | 1 | UNSIGNED_8 | 1 | |
| C01231 | MCK: Manual jog speeds | Linear value | 23344 | 5B30 | A | 2 | INTEGER_32 | 10000 | |
| C01232 | MCK: Manual jog accelerations | Linear value | 23343 | 5B2F | A | 2 | INTEGER_32 | 10000 | |
| C01233 | MCK: Manual jog S-ramp time | Linear value | 23342 | 5B2E | A | 1 | UNSIGNED_16 | 1000 | |
| C01234 | MCK: Manual jog breakpoints | Linear value | 23341 | 5B2D | A | 4 | INTEGER_32 | 10000 | |
| C01235 | MCK: Manual jog waiting times | Linear value | 23340 | 5B2C | A | 1 | UNSIGNED_16 | 1000 | |
| C01236 | MCK: Speed follower | Linear value | 23339 | 5B2B | A | 1 | INTEGER_32 | 10000 | |
| C01237 | MCK: Acceleration follower | Linear value | 23338 | 5B2A | A | 2 | INTEGER_32 | 10000 | |
| C01238 | MCK: S-ramp time follower | Linear value | 23337 | 5B29 | A | 1 | UNSIGNED_16 | 1000 | |
| C01239 | Setpoint holding | Linear value | 23336 | 5B28 | E | 1 | UNSIGNED_32 | 1 | |
| C01240 | MCK: Control word | Bit coded | 23335 | 5B27 | E | 1 | UNSIGNED_32 | 1 | |
| C01241 | MCK: Status word | Bit coded | 23334 | 5B26 | E | 1 | UNSIGNED_32 | 1 | |
| C01242 | MCK: Current pos profile number | Linear value | 23333 | 5B25 | E | 1 | UNSIGNED_8 | 1 | |
| C01243 | MCK: Current operating mode | Selection list | 23332 | 5B24 | E | 1 | UNSIGNED_8 | 1 | |
| C01244 | MCK: Target detection - times | Linear value | 23331 | 5B23 | A | 3 | UNSIGNED_16 | 1 | |
| C01245 | MCK: Target detection - positions | Linear value | 23330 | 5B22 | A | 3 | INTEGER_32 | 10000 | |
| C01246 | MCK: Select signal source | Selection list | 23329 | 5B21 | A | 2 | UNSIGNED_8 | 1 | |
| C01247 | MCK: Diagnostic word | Bit coded | 23328 | 5B20 | E | 1 | UNSIGNED_32 | 1 | |
| C01248 | MCK: Ref. step | Selection list | 23327 | 5B1F | A | 1 | UNSIGNED_8 | 1 | |
| C01251 | MCK: Acceleration stop | Linear value | 23324 | 5B1C | A | 1 | INTEGER_32 | 10000 | |
| C01252 | MCK: S-ramp times stop | Linear value | 23323 | 5B1B | A | 1 | UNSIGNED_16 | 1000 | |
| C01293 | L_MckStateInterface_1: Status | Bit coded | 23282 | 5AF2 | A | 8 | UNSIGNED_8 | 1 | |
| C01294 | Mode: Position calculation | Selection list | 23281 | 5AF1 | A | 3 | UNSIGNED_8 | 1 | |
| C01295 | L_MCKStateInterface_1: Pos. selection | Selection list | 23280 | 5AF0 | E | 1 | UNSIGNED_8 | 1 | |
| C01296 | Mode: Position calculation | Selection list | 23279 | 5AEF | A | 2 | UNSIGNED_8 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C01297 | L_MckCtrlInterface_1: Alternative function | Bit coded | 23278 | 5AEE | E | 1 | UNSIGNED_8 | 1 | |
| C01298 | MCK: Operating mode change with profile no. | Selection list | 23277 | 5AED | A | 4 | UNSIGNED_8 | 1 | |
| C01299 | MCKI: Status MCKInterface | Bit coded | 23276 | 5AEC | E | 1 | UNSIGNED_8 | 1 | |
| C01300 | Profile data: Positioning mode | Selection list | 23275 | 5AEB | A | 15 | UNSIGNED_8 | 1 | |
| C01301 | Profile data: Position | Linear value | 23274 | 5AEA | A | 15 | INTEGER_32 | 10000 | |
| C01302 | Profile data: Speed | Linear value | 23273 | 5AE9 | A | 15 | INTEGER_32 | 10000 | |
| C01303 | Profile data: Acceleration | Linear value | 23272 | 5AE8 | A | 15 | INTEGER_32 | 10000 | |
| C01304 | Profile data: Deceleration | Linear value | 23271 | 5AE7 | A | 15 | INTEGER_32 | 10000 | |
| C01305 | Profile data: Final speed | Linear value | 23270 | 5AE6 | A | 15 | INTEGER_32 | 10000 | |
| C01306 | Profile data: S-ramp time | Linear value | 23269 | 5AE5 | A | 15 | UNSIGNED_16 | 1000 | |
| C01307 | Profile data: Sequence profile | Linear value | 23268 | 5AE4 | A | 15 | UNSIGNED_8 | 1 | |
| C01308 | Profile data: TP profile | Linear value | 23267 | 5AE3 | A | 15 | UNSIGNED_8 | 1 | |
| C01309 | Profile data: TP signal source | Selection list | 23266 | 5AE2 | A | 15 | UNSIGNED_8 | 1 | |
| C01310 | Profile data: PI position | Selection list | 23265 | 5AE1 | A | 15 | UNSIGNED_8 | 1 | |
| C01311 | Profile data: PI speed | Selection list | 23264 | 5AE0 | A | 15 | UNSIGNED_8 | 1 | |
| C01312 | Profile data: PI accel. | Selection list | 23263 | 5ADF | A | 15 | UNSIGNED_8 | 1 | |
| C01313 | Profile data: PI decel. | Selection list | 23262 | 5ADE | A | 15 | UNSIGNED_8 | 1 | |
| C01314 | Profile data: PI final speed | Selection list | 23261 | 5ADD | A | 15 | UNSIGNED_8 | 1 | |
| C01315 | Profile data: PI S-ramp time | Selection list | 23260 | 5ADC | A | 15 | UNSIGNED_8 | 1 | |
| C01320 | Profile data: Position | Linear value | 23255 | 5AD7 | A | 15 | INTEGER_32 | 10000 | |
| C01321 | Profile data: Speed | Linear value | 23254 | 5AD6 | A | 15 | INTEGER_32 | 10000 | |
| C01322 | Profile data: Acceleration | Linear value | 23253 | 5AD5 | A | 15 | INTEGER_32 | 10000 | |
| C01323 | Profile data: Deceleration | Linear value | 23252 | 5AD4 | A | 15 | INTEGER_32 | 10000 | |
| C01324 | Profile data: Final speed | Linear value | 23251 | 5AD3 | A | 15 | INTEGER_32 | 10000 | |
| C01325 | Profile data: S-ramp time | Linear value | 23250 | 5AD2 | A | 15 | UNSIGNED_16 | 1000 | |
| C01350 | ACDrive: Drive mode | Selection list | 23225 | 5AB9 | A | 1 | UNSIGNED_8 | 1 | |
| C01351 | ACDrive: Control word | Bit coded | 23224 | 5AB8 | A | 1 | UNSIGNED_16 | 1 | |
| C01352 | ACDrive: Status word | Bit coded | 23223 | 5AB7 | A | 1 | UNSIGNED_16 | 1 | |
| C01353 | ACDrive: Setpoint scaling | Linear value | 23222 | 5AB6 | A | 2 | INTEGER_8 | 1 | |
| C01400 | L_Sequencer_1: Sequence step | Linear value | 23175 | 5A87 | A | 100 | UNSIGNED_16 | 1 | |
| C01401 | L_Sequencer_1: Current step | Linear value | 23174 | 5A86 | A | 1 | UNSIGNED_16 | 1 | |
| C01402 | L_Sequencer_1: Sequence control: Status | Linear value | 23173 | 5A85 | A | 1 | UNSIGNED_16 | 1 | |
| C01403 | L_Sequencer_1: Sequence control: Control word | Linear value | 23172 | 5A84 | A | 1 | UNSIGNED_16 | 1 | |
| C01404 | L_Sequencer_1: Step for bCancel = TRUE | Linear value | 23171 | 5A83 | A | 1 | UNSIGNED_16 | 1 | |
| C01405 | L_Sequencer_1: PSInput | Selection list | 23170 | 5A82 | A | 50 | UNSIGNED_16 | 1 | |
| C01406 | L_Sequencer_1: PSLevel | Linear value | 23169 | 5A81 | A | 50 | UNSIGNED_8 | 1 | |
| C01407 | L_Sequencer_1: PSProfil | Linear value | 23168 | 5A80 | A | 50 | UNSIGNED_16 | 1 | |
| C01408 | L_Sequencer_1: PSStep | Linear value | 23167 | 5A7F | A | 50 | UNSIGNED_16 | 1 | |
| C01409 | L_Sequencer_1: PS_WD_Time | Linear value | 23166 | 5A7E | A | 50 | UNSIGNED_32 | 1000 | |
| C01410 | L_Sequencer_1: PS_WD_Step | Linear value | 23165 | 5A7D | A | 50 | UNSIGNED_16 | 1 | |
| C01411 | L_Sequencer_1: SwitchOut1 | Selection list | 23164 | 5A7C | A | 16 | UNSIGNED_16 | 1 | |
| C01412 | L_Sequencer_1: SwitchLevel1 | Linear value | 23163 | 5A7B | A | 16 | UNSIGNED_8 | 1 | |
| C01413 | L_Sequencer_1: SwitchOut2 | Selection list | 23162 | 5A7A | A | 16 | UNSIGNED_16 | 1 | |
| C01414 | L_Sequencer_1: SwitchLevel2 | Linear value | 23161 | 5A79 | A | 16 | UNSIGNED_8 | 1 | |
| C01415 | L_Sequencer_1: BranchIn | Selection list | 23160 | 5A78 | A | 16 | UNSIGNED_16 | 1 | |
| C01416 | L_Sequencer_1: BranchLevel | Linear value | 23159 | 5A77 | A | 16 | UNSIGNED_8 | 1 | |
| C01417 | L_Sequencer_1: BranchStep | Linear value | 23158 | 5A76 | A | 16 | UNSIGNED_16 | 1 | |
| C01418 | L_Sequencer_1: BranchStep1 | Linear value | 23157 | 5A75 | A | 2 | UNSIGNED_16 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C01419 | L_Sequencer_1: BranchStep2 | Linear value | 23156 | 5A74 | A | 2 | UNSIGNED_16 | 1 | |
| C01420 | L_Sequencer_1: BranchStep3 | Linear value | 23155 | 5A73 | A | 2 | UNSIGNED_16 | 1 | |
| C01421 | L_Sequencer_1: BranchStep4 | Linear value | 23154 | 5A72 | A | 2 | UNSIGNED_16 | 1 | |
| C01422 | L_Sequencer_1: BranchStep5 | Linear value | 23153 | 5A71 | A | 2 | UNSIGNED_16 | 1 | |
| C01423 | L_Sequencer_1: BranchStep6 | Linear value | 23152 | 5A70 | A | 2 | UNSIGNED_16 | 1 | |
| C01424 | L_Sequencer_1: BranchStep7 | Linear value | 23151 | 5A6F | A | 2 | UNSIGNED_16 | 1 | |
| C01425 | L_Sequencer_1: BranchStep8 | Linear value | 23150 | 5A6E | A | 2 | UNSIGNED_16 | 1 | |
| C01426 | L_Sequencer_1: BranchStep9 | Linear value | 23149 | 5A6D | A | 2 | UNSIGNED_16 | 1 | |
| C01427 | L_Sequencer_1: BranchStep10 | Linear value | 23148 | 5A6C | A | 2 | UNSIGNED_16 | 1 | |
| C01428 | L_Sequencer_1: BranchStep11 | Linear value | 23147 | 5A6B | A | 2 | UNSIGNED_16 | 1 | |
| C01429 | L_Sequencer_1: BranchStep12 | Linear value | 23146 | 5A6A | A | 2 | UNSIGNED_16 | 1 | |
| C01430 | L_Sequencer_1: BranchStep13 | Linear value | 23145 | 5A69 | A | 2 | UNSIGNED_16 | 1 | |
| C01431 | L_Sequencer_1: BranchStep14 | Linear value | 23144 | 5A68 | A | 2 | UNSIGNED_16 | 1 | |
| C01432 | L_Sequencer_1: BranchStep15 | Linear value | 23143 | 5A67 | A | 2 | UNSIGNED_16 | 1 | |
| C01433 | L_Sequencer_1: BranchStep16 | Linear value | 23142 | 5A66 | A | 2 | UNSIGNED_16 | 1 | |
| C01434 | L_Sequencer_1: BranchStep17 | Linear value | 23141 | 5A65 | A | 2 | UNSIGNED_16 | 1 | |
| C01435 | L_Sequencer_1: BranchStep18 | Linear value | 23140 | 5A64 | A | 2 | UNSIGNED_16 | 1 | |
| C01436 | L_Sequencer_1: BranchStep19 | Linear value | 23139 | 5A63 | A | 2 | UNSIGNED_16 | 1 | |
| C01437 | L_Sequencer_1: BranchStep20 | Linear value | 23138 | 5A62 | A | 2 | UNSIGNED_16 | 1 | |
| C01438 | L_Sequencer_1: Wartezeit | Linear value | 23137 | 5A61 | A | 8 | UNSIGNED_32 | 1000 | |
| C01439 | L_Sequencer_1: Wait Eingang f. Weiter | Selection list | 23136 | 5A60 | A | 8 | UNSIGNED_16 | 1 | |
| C01440 | L_Sequencer_1: WaitLevel | Linear value | 23135 | 5A5F | A | 8 | UNSIGNED_8 | 1 | |
| C01441 | L_Sequencer_1: SetZählernummer | Linear value | 23134 | 5A5E | A | 5 | UNSIGNED_16 | 1 | |
| C01442 | L_Sequencer_1: SetZählerStartwert | Linear value | 23133 | 5A5D | A | 5 | INTEGER_32 | 1 | |
| C01443 | L_Sequencer_1: Aktueller Zählerstand | Linear value | 23132 | 5A5C | A | 5 | INTEGER_32 | 1 | |
| C01444 | L_Sequencer_1: ZählerNummer | Linear value | 23131 | 5A5B | A | 8 | UNSIGNED_16 | 1 | |
| C01445 | L_Sequencer_1: SchrittwertZähler | Linear value | 23130 | 5A5A | A | 8 | INTEGER_32 | 1 | |
| C01446 | L_Sequencer_1: ZählerVergleichswert | Linear value | 23129 | 5A59 | A | 8 | INTEGER_32 | 1 | |
| C01447 | L_Sequencer_1: ZählerSprungziel | Linear value | 23128 | 5A58 | A | 8 | UNSIGNED_16 | 1 | |
| C01448 | L_Sequencer_1: ZählerVergleichsfunktion | Selection list | 23127 | 5A57 | A | 8 | UNSIGNED_16 | 1 | |
| C01449 | L_Sequencer_1: Standby Eingang f. Ende | Selection list | 23126 | 5A56 | A | 5 | UNSIGNED_16 | 1 | |
| C01450 | L_Sequencer_1: Standby PolaritätEingang | Linear value | 23125 | 5A55 | A | 5 | UNSIGNED_8 | 1 | |
| C01451 | L_Sequencer_1: Standby Modusfolger | Selection list | 23124 | 5A54 | A | 5 | UNSIGNED_16 | 1 | |
| C01452 | L_Sequencer_1: StandbySollwert | Linear value | 23123 | 5A53 | A | 5 | INTEGER_16 | 100 | |
| C01453 | L_Sequencer_1: Bool signal | Linear value | 23122 | 5A52 | A | 16 | UNSIGNED_8 | 1 | |
| C01454 | L_Sequencer_1: INT signal | Linear value | 23121 | 5A51 | A | 1 | INTEGER_16 | 100 | |
| C01455 | L_Sequencer_1: Word signal | Linear value | 23120 | 5A50 | A | 13 | UNSIGNED_16 | 1 | |
| C01460 | L_ConvActPos: TConst | Linear value | 23115 | 5A4B | A | 1 | UNSIGNED_16 | 1 | |
| C01461 | L_ConvActPos: MaxPos/MinPos | Linear value | 23114 | 5A4A | A | 2 | INTEGER_16 | 100 | |
| C01462 | L_ConvActPos: Length | Linear value | 23113 | 5A49 | A | 1 | UNSIGNED_32 | 1 | |
| C01463 | L_ConvActPos: UpperRef/LowerRef | Linear value | 23112 | 5A48 | A | 2 | INTEGER_16 | 100 | |
| C01464 | L_ConvActPos: WindowSetPos | Linear value | 23111 | 5A47 | A | 1 | INTEGER_16 | 100 | |
| C01465 | L_ConvActPos: DelayTime | Linear value | 23110 | 5A46 | A | 1 | UNSIGNED_16 | 1000 | |
| C01466 | L_ConvActPos: TConstVAdd | Linear value | 23109 | 5A45 | A | 1 | UNSIGNED_16 | 1 | |
| C01467 | L_ConvActPos: DeadBand | Linear value | 23108 | 5A44 | A | 1 | INTEGER_16 | 100 | |
| C01468 | L_ConvActPos: Damping | Linear value | 23107 | 5A43 | A | 1 | UNSIGNED_16 | 100 | |
| C01469 | L_MFail: Vp | Linear value | 23106 | 5A42 | A | 1 | UNSIGNED_16 | 1000 | |
| C01470 | L_MFail: Tn | Linear value | 23105 | 5A41 | A | 3 | UNSIGNED_16 | 1 | |
| C01501 | Resp. to communication error with MCI | Selection list | 23074 | 5A22 | A | 2 | UNSIGNED_8 | 1 | |
| C01670 | L_ComparePhi 1-5: Function | Selection list | 22905 | 5979 | A | 5 | UNSIGNED_8 | 1 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C01671 | L_ComparePhi 1-5: Hysteresis | Linear value | 22904 | 5978 | A | 5 | INTEGER_32 | 1 | |
| C01672 | L_ComparePhi 1-5: Window | Linear value | 22903 | 5977 | A | 5 | INTEGER_32 | 1 | |
| C01770 | Filter time - earth-fault detect. is running | Linear value | 22805 | 5915 | E | 1 | UNSIGNED_8 | 1 | |
| C01902 | Diagnostics X6: Max. baud rate | Selection list | 22673 | 5891 | E | 1 | UNSIGNED_16 | 1 | |
| C01903 | Diagnostics X6: Change baud rate | Selection list | 22672 | 5890 | E | 1 | UNSIGNED_8 | 1 | |
| C01905 | Diagnostics X6: Current baud rate | Linear value | 22670 | 588E | E | 1 | UNSIGNED_32 | 1 | |
| C02200 | LS_WriteParamList: Function | Selection list | 22375 | 5767 | A | 2 | UNSIGNED_8 | 1 | |
| C02210 | LS_WriteParamList: Motor control | Selection list | 22365 | 575D | A | 4 | UNSIGNED_8 | 1 | |
| C02212 | LS_WriteParamList: VFC: V/f base frequency | Linear value | 22363 | 575B | A | 4 | UNSIGNED_16 | 10 | |
| C02213 | LS_WriteParamList: VFC: Vmin boost | Linear value | 22362 | 575A | A | 4 | UNSIGNED_16 | 100 | |
| C02214 | LS_WriteParamList: Switching frequency | Selection list | 22361 | 5759 | A | 4 | UNSIGNED_8 | 1 | |
| C02215 | LS_WriteParamList: Auto-DCB: Threshold | Linear value | 22360 | 5758 | A | 4 | UNSIGNED_16 | 1 | |
| C02216 | LS_WriteParamList: Slip compensation | Linear value | 22359 | 5757 | A | 4 | INTEGER_16 | 100 | |
| C02217 | LS_WriteParamList: Imax in motor mode | Linear value | 22358 | 5756 | A | 4 | UNSIGNED_16 | 100 | |
| C02218 | LS_WriteParamList: Imax in generator mode | Linear value | 22357 | 5755 | A | 4 | INTEGER_16 | 100 | |
| C02219 | LS_WriteParamList: DC braking: Current | Linear value | 22356 | 5754 | A | 4 | INTEGER_16 | 100 | |
| C02220 | LS_WriteParamList: Vp speed controller | Linear value | 22355 | 5753 | A | 12 | UNSIGNED_16 | 100 | |
| C02221 | LS_WriteParamList: Ti speed controller | Linear value | 22354 | 5752 | A | 12 | UNSIGNED_16 | 10 | |
| C02222 | LS_WriteParamList: SC: Tdn speed controller | Linear value | 22353 | 5751 | A | 4 | UNSIGNED_16 | 100 | |
| C02223 | LS_WriteParamList: Imax/M controller gain | Linear value | 22352 | 5750 | A | 8 | UNSIGNED_16 | 100 | |
| C02224 | LS_WriteParamList: Imax/M controller reset time | Linear value | 22351 | 574F | A | 8 | UNSIGNED_16 | 1 | |
| C02225 | LS_WriteParamList: Vp current controller | Linear value | 22350 | 574E | A | 4 | UNSIGNED_16 | 100 | |
| C02226 | LS_WriteParamList: Ti current controller | Linear value | 22349 | 574D | A | 4 | UNSIGNED_16 | 100 | |
| C02227 | LS_WriteParamList: SC: Vp field controller | Linear value | 22348 | 574C | A | 4 | UNSIGNED_16 | 100 | |
| C02228 | LS_WriteParamList: SC: Tn field controller | Linear value | 22347 | 574B | A | 4 | UNSIGNED_16 | 10 | |
| C02229 | LS_WriteParamList: SC: settings | Selection list | 22346 | 574A | A | 16 | UNSIGNED_8 | 1 | |
| C02230 | LS_WriteParamList: Override point of field weakening | Linear value | 22345 | 5749 | A | 4 | INTEGER_16 | 1 | |
| C02231 | LS_WriteParamList: Rated motor power | Linear value | 22344 | 5748 | A | 4 | UNSIGNED_16 | 100 | |
| C02232 | LS_WriteParamList: Motor rotor resistance | Linear value | 22343 | 5747 | A | 4 | UNSIGNED_32 | 1 | |
| C02233 | LS_WriteParamList: Motor stator resistance | Linear value | 22342 | 5746 | A | 4 | UNSIGNED_32 | 1 | |
| C02234 | LS_WriteParamList: Motor stator leakage inductance | Linear value | 22341 | 5745 | A | 4 | UNSIGNED_16 | 100 | |
| C02236 | LS_WriteParamList: Rated motor speed | Linear value | 22339 | 5743 | A | 4 | UNSIGNED_16 | 1 | |
| C02237 | LS_WriteParamList: Rated motor current | Linear value | 22338 | 5742 | A | 4 | UNSIGNED_16 | 100 | |
| C02238 | LS_WriteParamList: Rated motor frequency | Linear value | 22337 | 5741 | A | 4 | UNSIGNED_16 | 1 | |
| C02239 | LS_WriteParamList: Rated motor voltage | Linear value | 22336 | 5740 | A | 4 | UNSIGNED_16 | 1 | |
| C02240 | LS_WriteParamList: Motor cosine phi | Linear value | 22335 | 573F | A | 4 | UNSIGNED_8 | 100 | |
| C02241 | LS_WriteParamList: Motor magnetizing inductance | Linear value | 22334 | 573E | A | 4 | UNSIGNED_16 | 10 | |
| C02242 | LS_WriteParamList: Motor magnetising current | Linear value | 22333 | 573D | A | 4 | UNSIGNED_16 | 100 | |
| C02244 | LS_WriteParamList: Auto-DCB: hold time | Linear value | 22331 | 573B | A | 4 | UNSIGNED_32 | 1000 | |
| C02245 | LS_WriteParamList: DC braking: Hold time | Linear value | 22330 | 573A | A | 4 | UNSIGNED_32 | 1000 | |
| C02246 | LS_WriteParamList: Setting of motor overload ($I^2xt$) | Linear value | 22329 | 5739 | A | 4 | INTEGER_16 | 100 | |
| C02249 | LS_WriteParamList: Oscillation damping influence | Linear value | 22326 | 5736 | A | 4 | UNSIGNED_16 | 100 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C02250 | LS_WriteParamList: Filter time - oscill. damping | Linear value | 22325 | 5735 | A | 4 | UNSIGNED_8 | 1 | |
| C02251 | LS_WriteParamList: Oscillation damping field weakening | Linear value | 22324 | 5734 | A | 4 | UNSIGNED_8 | 1 | |
| C02252 | LS_WriteParamList: Kp position controller | Linear value | 22323 | 5733 | A | 4 | UNSIGNED_16 | 100 | |
| C02256 | LS_WriteParamList: Moment of inertia | Linear value | 22319 | 572F | A | 4 | UNSIGNED_32 | 100 | |
| C02260 | LS_WriteParamList: Speed sensor selection | Selection list | 22315 | 572B | A | 4 | UNSIGNED_8 | 1 | |
| C02261 | LS_WriteParamList: SC: Field feedforward control | Linear value | 22314 | 572A | A | 4 | UNSIGNED_16 | 1 | |
| C02262 | LS_WriteParamList: SC: Vp field weakening controller | Linear value | 22313 | 5729 | A | 4 | UNSIGNED_16 | 10000 | |
| C02263 | LS_WriteParamList: SC: Tn field weakening controller | Linear value | 22312 | 5728 | A | 4 | UNSIGNED_16 | 10 | |
| C02264 | LS_WriteParamList: Sensibility - Setpoint feedforward control | Selection list | 22311 | 5727 | A | 4 | UNSIGNED_8 | 1 | |
| C02272 | LS_WriteParamList: Motor phase direction of rotation | Selection list | 22303 | 571F | A | 4 | UNSIGNED_8 | 1 | CINH |
| C02273 | LS_WriteParamList: Speed limitation | Linear value | 22302 | 571E | A | 8 | INTEGER_16 | 100 | |
| C02274 | LS_WriteParamList: Frequency limitation | Linear value | 22301 | 571D | A | 8 | UNSIGNED_16 | 1 | |
| C02275 | LS_WriteParamList: Motor cable length | Linear value | 22300 | 571C | A | 4 | UNSIGNED_16 | 10 | |
| C02276 | LS_WriteParamList: Motor cable cross-section | Linear value | 22299 | 571B | A | 4 | UNSIGNED_16 | 100 | |
| C02278 | LS_WriteParamList: PSM: Maximum motor current field weakening | Linear value | 22297 | 5719 | A | 4 | UNSIGNED_16 | 100 | |
| C02279 | LS_WriteParamList: Ultimate motor current | Linear value | 22296 | 5718 | A | 4 | UNSIGNED_16 | 10 | |
| C02280 | LS_WriteParamList: Max. motor speed | Linear value | 22295 | 5717 | A | 4 | UNSIGNED_16 | 1 | |
| C02281 | LS_WriteParamList: VFC: Time const. slip comp. | Linear value | 22294 | 5716 | A | 4 | UNSIGNED_16 | 1 | |
| C02284 | LS_WriteParamList: VFC: limitation V/f +encoder | Linear value | 22291 | 5713 | A | 8 | UNSIGNED_16 | 100 | |
| C02285 | LS_WriteParamList: VFC: Vp V/f +encoder | Linear value | 22290 | 5712 | A | 4 | UNSIGNED_16 | 1000 | |
| C02286 | LS_WriteParamList: VFC: Ti V/f +encoder | Linear value | 22289 | 5711 | A | 4 | UNSIGNED_16 | 10 | |
| C02287 | LS_WriteParamList: VFC-ECO: Vp CosPhi controller | Linear value | 22288 | 5710 | A | 4 | UNSIGNED_16 | 1000 | |
| C02288 | LS_WriteParamList: VFC-ECO: Ti CosPhi controller | Linear value | 22287 | 570F | A | 4 | UNSIGNED_16 | 10 | |
| C02289 | LS_WriteParamList: VFC-ECO: Minimum voltage U/f | Linear value | 22286 | 570E | A | 4 | INTEGER_16 | 100 | |
| C02290 | LS_WriteParamList: VFC-ECO: Voltage reduction ramp | Linear value | 22285 | 570D | A | 4 | UNSIGNED_8 | 10 | |
| C02291 | LS_WriteParamList: SLVC: Field current controller gain | Linear value | 22284 | 570C | A | 4 | INTEGER_16 | 100 | |
| C02292 | LS_WriteParamList: SLVC: Cross current controller gain | Linear value | 22283 | 570B | A | 4 | INTEGER_16 | 100 | |
| C02293 | LS_WriteParamList: Inverter motor brake: nAdd | Linear value | 22282 | 570A | A | 4 | INTEGER_16 | 1 | |
| C02294 | LS_WriteParamList: Inverter motor brake: PT1 filter time | Linear value | 22281 | 5709 | A | 4 | INTEGER_16 | 10 | |
| C02295 | LS_WriteParamList: Flying restart fct.: activation | Selection list | 22280 | 5708 | A | 4 | UNSIGNED_8 | 1 | CINH |
| C02296 | LS_WriteParamList: Flying restart fct.: process | Selection list | 22279 | 5707 | A | 4 | UNSIGNED_16 | 1 | |
| C02297 | LS_WriteParamList: Flying restart: start frequency | Linear value | 22278 | 5706 | A | 4 | INTEGER_16 | 1 | |
| C02298 | LS_WriteParamList: Flying restart fct: int. time | Linear value | 22277 | 5705 | A | 4 | UNSIGNED_16 | 10 | |
| C02299 | LS_WriteParamList: Flying restart fct.: current | Linear value | 22276 | 5704 | A | 4 | INTEGER_16 | 100 | |

Greyed out = display parameter (read only access)

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C02300 | LS_WriteParamList: SLPSM: Controlled current setpoint | Linear value | 22275 | 5703 | A | 8 | UNSIGNED_16 | 100 | |
| C02301 | LS_WriteParamList: SLPSM: Switching speed | Linear value | 22274 | 5702 | A | 8 | INTEGER_16 | 100 | |
| C02302 | LS_WriteParamList: SLPSM: Filter time - rotor position | Linear value | 22273 | 5701 | A | 4 | INTEGER_16 | 100 | |
| C02303 | LS_WriteParamList: SLPSM: Filter time rotor position | Linear value | 22272 | 5700 | A | 8 | INTEGER_16 | 10 | |
| C02304 | LS_WriteParamList: SLPSM: PLL gain | Linear value | 22271 | 56FF | A | 4 | INTEGER_16 | 1 | |
| C02305 | LS_WriteParamList: PSM: Ppp saturation characteristic | Linear value | 22270 | 56FE | A | 68 | UNSIGNED_8 | 1 | |
| C02306 | LS_WriteParamList: PSM: Imax Ppp saturation characteristic | Linear value | 22269 | 56FD | A | 4 | UNSIGNED_16 | 10 | |
| C02307 | LS_WriteParamList: PSM: Activate Ppp saturation char. | Selection list | 22268 | 56FC | A | 4 | UNSIGNED_8 | 1 | |
| C02311 | LS_WriteParamList: PLI without motion: Adaptation of time duration | Linear value | 22264 | 56F8 | A | 4 | INTEGER_8 | 1 | |
| C02312 | LS_WriteParamList: PLI without motion | Bit coded | 22263 | 56F7 | A | 4 | UNSIGNED_16 | 1 | |
| C02313 | LS_WriteParamList: PLI without motion: Adaptation of ident angle | Linear value | 22262 | 56F6 | A | 4 | INTEGER_8 | 1 | |
| C02315 | LS_WriteParamList: Manual entry of motor type | Selection list | 22260 | 56F4 | A | 4 | UNSIGNED_8 | 1 | |
| C02430 | Axis bus address and no. of nodes | Linear value | 22145 | 5681 | A | 2 | UNSIGNED_8 | 1 | |
| C02431 | Axis bus time settings | Linear value | 22144 | 5680 | A | 3 | UNSIGNED_16 | 1 | |
| C02435 | Axis bus status | Selection list | 22140 | 567C | E | 1 | UNSIGNED_8 | 1 | |
| C02436 | Axis bus error status | Selection list | 22139 | 567B | E | 1 | UNSIGNED_8 | 1 | |
| C02437 | Axis bus MessageError | Selection list | 22138 | 567A | E | 1 | UNSIGNED_8 | 1 | |
| C02438 | CAB_Tx_Rx_diagnosis | Linear value | 22137 | 5679 | A | 4 | UNSIGNED_16 | 1 | |
| C02440 | AxisBusIO slave/master | Selection list | 22135 | 5677 | A | 1 | UNSIGNED_8 | 1 | |
| C02442 | CAB decoupling inputs | Bit coded | 22133 | 5675 | A | 1 | UNSIGNED_16 | 1 | |
| C02443 | CAB decoupling value | Linear value | 22132 | 5674 | A | 11 | UNSIGNED_16 | 1 | |
| C02444 | CAB configuration | Bit coded | 22131 | 5673 | A | 1 | UNSIGNED_16 | 1 | |
| C02556 | Position controller: Limitation | Linear value | 22019 | 5603 | A | 1 | INTEGER_32 | 1 | |
| C02580 | Holding brake: Operating mode | Selection list | 21995 | 55EB | E | 1 | UNSIGNED_8 | 1 | |
| C02581 | Holding brake: Speed thresholds | Linear value | 21994 | 55EA | A | 5 | INTEGER_16 | 100 | |
| C02582 | Holding brake: Setting | Bit coded | 21993 | 55E9 | E | 1 | UNSIGNED_8 | 1 | |
| C02589 | Holding brake: Time system | Linear value | 21986 | 55E2 | A | 4 | UNSIGNED_16 | 1 | |
| C02593 | Holding brake: Activation time | Linear value | 21982 | 55DE | A | 4 | UNSIGNED_32 | 1000 | |
| C02607 | Holding brake: Status | Bit coded | 21968 | 55D0 | E | 1 | UNSIGNED_16 | 1 | |
| C02610 | MCK: Accel./decel. times | Linear value | 21965 | 55CD | A | 3 | UNSIGNED_32 | 1000 | |
| C02611 | MCK: Limitations | Linear value | 21964 | 55CC | A | 5 | INTEGER_16 | 100 | |
| C02652 | Settings of measuring system | Bit coded | 21923 | 55A3 | E | 1 | UNSIGNED_16 | 1 | |
| C02810 | TP: Edge selection | Selection list | 21765 | 5505 | A | 9 | UNSIGNED_8 | 1 | |
| C02811 | TP: Sensor delay | Linear value | 21764 | 5504 | A | 9 | UNSIGNED_16 | 1 | |
| C02812 | TP: Position offset | Linear value | 21763 | 5503 | A | 9 | INTEGER_32 | 10000 | |
| C02813 | TP: Pos window start | Linear value | 21762 | 5502 | A | 5 | INTEGER_32 | 10000 | |
| C02814 | TP: Pos window end | Linear value | 21761 | 5501 | A | 5 | INTEGER_32 | 10000 | |
| C02815 | TP: Position source | Selection list | 21760 | 5500 | A | 9 | UNSIGNED_8 | 1 | |
| C02816 | TP: Signal counter | Linear value | 21759 | 54FF | A | 9 | UNSIGNED_16 | 1 | |
| C02817 | TP: TouchProbe position | Linear value | 21758 | 54FE | A | 9 | INTEGER_32 | 10000 | |
| C02830 | DIx: Debounce time | Selection list | 21745 | 54F1 | A | 7 | UNSIGNED_8 | 1 | |
| C02840 | CountInx: Parameter | Linear value | 21735 | 54E7 | A | 4 | UNSIGNED_32 | 1 | |
| C02841 | CountInx: Counter content | Linear value | 21734 | 54E6 | A | 2 | UNSIGNED_32 | 1 | |
| C02842 | FreqInxx: Offset | Linear value | 21733 | 54E5 | A | 2 | INTEGER_16 | 100 | |
| Greyed out = display parameter (read only access) | | | | | | | | | |

| Code | Name | Type | Index | | Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | dec | hex | DS | DA | DT | Factor | CINH |
| C02843 | FreqInxx: Gain | Linear value | 21732 | 54E4 | A | 2 | INTEGER_16 | 100 | |
| C02844 | FreqIn12: Function | Selection list | 21731 | 54E3 | A | 1 | UNSIGNED_8 | 1 | |
| C02845 | FreqIn12: PosIn comparison value | Linear value | 21730 | 54E2 | E | 1 | INTEGER_32 | 1 | |
| C02853 | PSM: Lss saturation characteristic | Linear value | 21722 | 54DA | A | 17 | UNSIGNED_8 | 1 | |
| C02855 | PSM: Imax Lss saturation characteristic | Linear value | 21720 | 54D8 | E | 1 | UNSIGNED_16 | 10 | |
| C02859 | PSM: Activate Ppp saturation char. | Selection list | 21716 | 54D4 | E | 1 | UNSIGNED_8 | 1 | |
| C02862 | Resolver gain | Linear value | 21713 | 54D1 | A | 2 | UNSIGNED_16 | 100 | |
| C02863 | Resolver: Phase error | Linear value | 21712 | 54D0 | E | 1 | INTEGER_16 | 100 | |
| C02864 | MCTRL: Optimisations | Bit coded | 21711 | 54CF | A | 1 | UNSIGNED_16 | 1 | |
| C02865 | MCTRL: Special settings | Bit coded | 21710 | 54CE | A | 1 | UNSIGNED_16 | 1 | |
| C02866 | MCTRL: Special settings | Selection list | 21709 | 54CD | A | 3 | UNSIGNED_8 | 1 | CINH |
| C02867 | Identification process | Selection list | 21708 | 54CC | A | 1 | UNSIGNED_8 | 1 | CINH |
| C02868 | Setting of compatibility | Bit coded | 21707 | 54CB | A | 1 | UNSIGNED_32 | 1 | |
| C02870 | PLI without motion: Optimisation factor | Linear value | 21705 | 54C9 | A | 1 | INTEGER_16 | 100 | |
| C02871 | PLI without motion: Running time | Linear value | 21704 | 54C8 | A | 1 | INTEGER_16 | 100 | |
| C02872 | PLI without motion: Adaptation of time duration | Linear value | 21703 | 54C7 | A | 1 | INTEGER_8 | 1 | |
| C02873 | PLI without motion: Ident. el. rotor displ. angle | Linear value | 21702 | 54C6 | A | 1 | INTEGER_16 | 1 | |
| C02874 | PLI without motion | Bit coded | 21701 | 54C5 | A | 1 | UNSIGNED_16 | 1 | |
| C02875 | PLI without motion: Adaptation of ident angle | Linear value | 21700 | 54C4 | A | 1 | INTEGER_8 | 1 | |
| C02876 | PSM: Max. motor temperature | Linear value | 21699 | 54C3 | A | 1 | UNSIGNED_8 | 1 | |
| C02877 | PSM temperature coefficient | Linear value | 21698 | 54C2 | A | 1 | INTEGER_16 | 100 | |
| C02878 | KTY motor temperature compensation | Bit coded | 21697 | 54C1 | A | 1 | UNSIGNED_16 | 1 | |
| C02879 | Slip calculation from equivalent circuit diagram | Bit coded | 21696 | 54C0 | A | 1 | UNSIGNED_16 | 1 | |

Greyed out = display parameter (read only access)

# 18 Working with the FB Editor

The function block editor (in the following called "FB Editor") is available in the »Engineer« from the "StateLine" device version.

The FB Editor can be used to carry out an online monitoring of the technology application running in the device (e.g. for diagnostic purposes) and reconfigure the I/O interconnection of the technology application.

### Functional range

The options for processing function block interconnections depend on the device version:

| Function | Inverter Drives 8400 | | |
|---|---|---|---|
| | StateLine C | HighLine C | TopLine C |
| I/O interconnection can be reconfigured | ● | ● | ● |
| Application interconnection can be reconfigured | ● | ● | ● |
| Free interconnection | (●) | ● | ● |
| (●) Free interconnection for "StateLine C" is only available from version 12.00.00 and »Engineer« V2.17. | | | |

**Note!**

The illustrations of the FB Editor user interface and the dialog boxes in this documentation are based on the »Engineer« V2.10.

## 18.1 Basics

Using the function block interconnection, any signal interconnection can be implemented. Various FBs are available for digital signal processing, signal conversion and logic modules.

For special tasks it has proved of value to use the integrated technology applications as a basis for modifications or extensions of the available FB interconnections. Depending on the device version (see Functional range), the experienced user has the opportunity to implement own drive solutions independent of the predefined technology applications by using the "free interconnection".

For this purpose, the FB Editor provides the following functions:

- Copying & pasting of interconnection elements (also device-independent)
- Export & import of the interconnection
- Comparison of two interconnections (also online <-> offline comparison)
- Overview window and zoom functions
- Comments on the signal flow
- Online monitoring

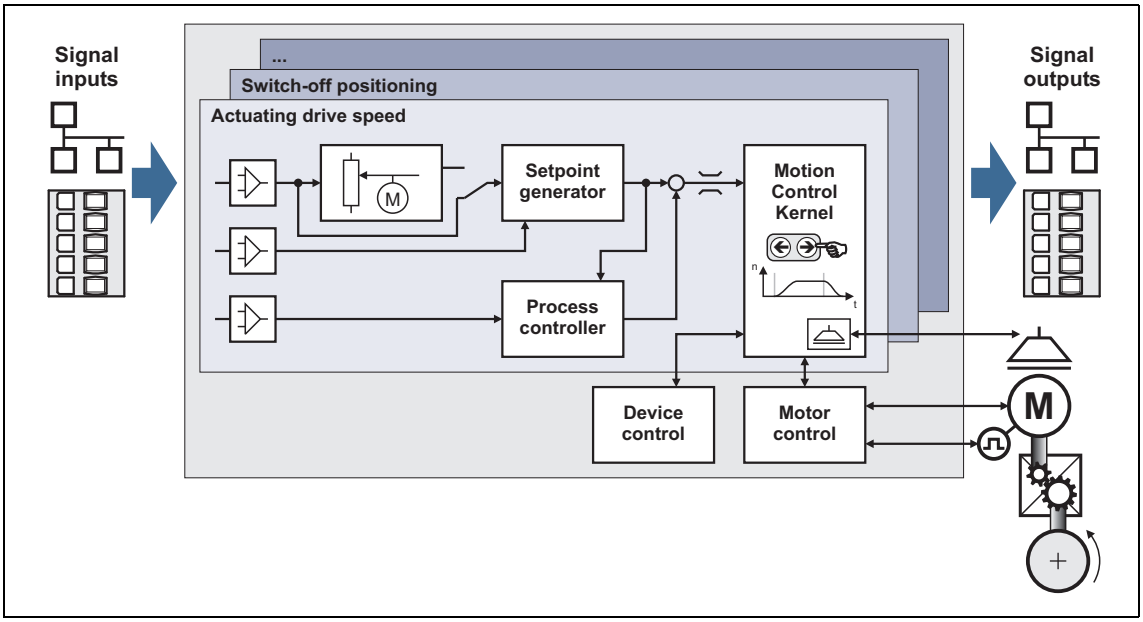
The option to mask out non-used inputs and outputs of modules is also sensible to minimise the complexity of the FB interconnection and to adapt the clarity of the interconnection to the customers need.

All graphical information of the FB interconnection view (positions of the FBs, line or flag presentation of the connection, visibility of the inputs/outputs) are saved with the parameter set in the memory module of the inverter and can be uploaded any time into the FB Editor of the »Engineer« even if the Engineer project is not available.

### 18.1.1 Basic components of a drive solution

A drive solution consists of the following basic components:

- Signal inputs (for control and setpoint signals)
- Signal flow of the technology application
- Signal outputs (for status and actual value signals)




[18-1] Basic components of a drive solution

Regarding the 8400 device series, these three components are available for the FB interconnection and classified as follows:

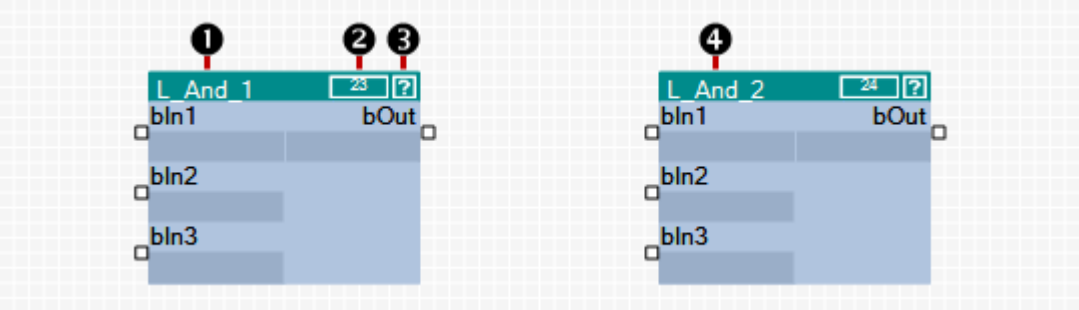
| Module type | Name | Task | Example |
|---|---|---|---|
| Function block | L_*name* | General function block for free interconnection (only HighLine) | L_Compare_1<br>L_PCTRL_1 |
| System block | LS_*name* | Signal interface to inverter-internal functions | LS_DigitalInput<br>LS_DriveInterface |
| Port block | LP_*name* | • Process data communication via a fieldbus using a communication module<br>• Process data communication via CAN on board | LP_CanIn1<br>LP_CanOut1<br>LP_MciIn<br>LP_MciOut |
| Application block | LA_*name* | Block for a technology application | LA_NCtrl<br>LA_SwitchPos |

Further information on the individual modules can be obtained from the following subchapters!

### 18.1.1.1 What is a function block?

A function block (FB) can be compared with an integrated circuit that contains a certain control logic and provides one or several values when being executed.

- The function blocks are classified alphabetically in a "function library".
- Each function block has a unique identifier and a processing number which defines the position at which the function block is calculated during runtime.




❶ Name of the function block
❷ Processing number
❸ Button for opening the online help for the function block
❹ Another available function block of the same type (numbering in ascending order)

[18-2] Information on a function block in the FB Editor

**Tip!**

A detailed description of all available function blocks can be found in the main chapter "Function library". (📖 1425)

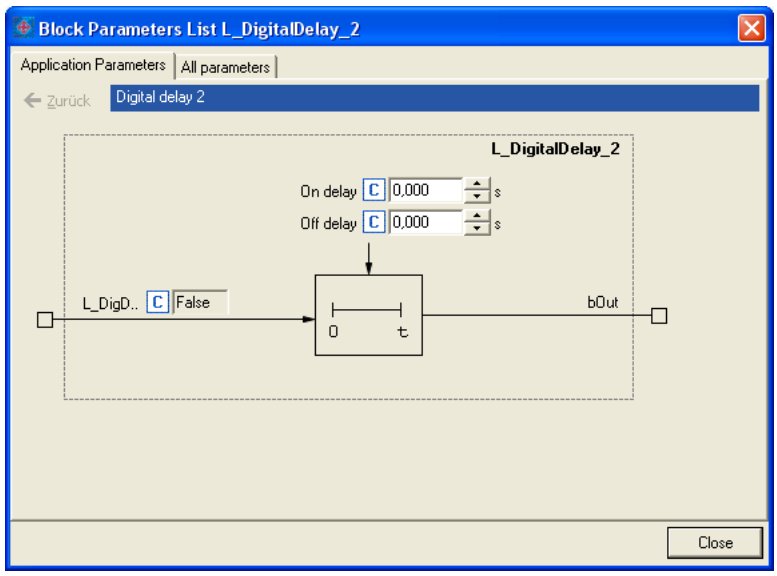
### 18.1.1.2 Parameterisable function blocks

Some function blocks have parameters which serve to change particular settings during operation, if required, or which display actual values & status information.

- The icon in the head of the module, a double-click on the module, or the **Parameter...** command in the *Context menu* of the module serve to open the parameterisation dialog or the parameter list for the module.

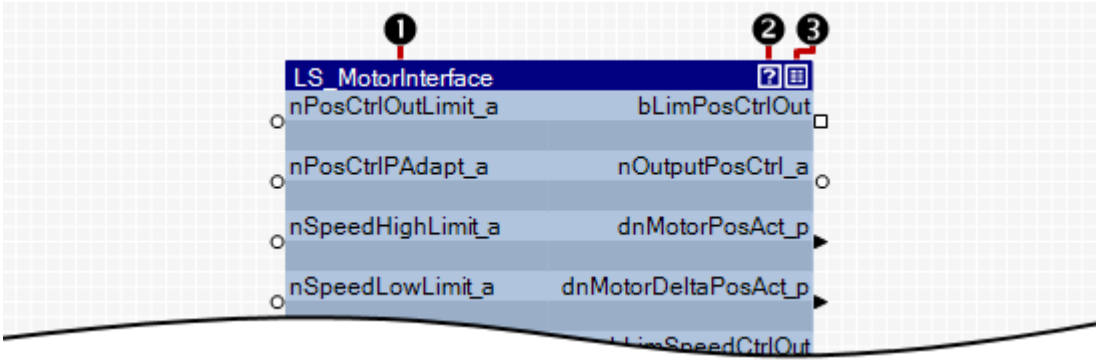
**Example**

Parameterisation dialog for the FB **L_DigitalDelay_2**:




### 18.1.1.3 What is a system block?

System blocks are a special variant of a function block. They partly activate real hardware, e. g. the digital and analog inputs/outputs and the motor control.




❶ Name of the system block
❷ Button for opening the online help for the system block
❸ Button for opening the parameterisation dialog or the parameter list for the system block

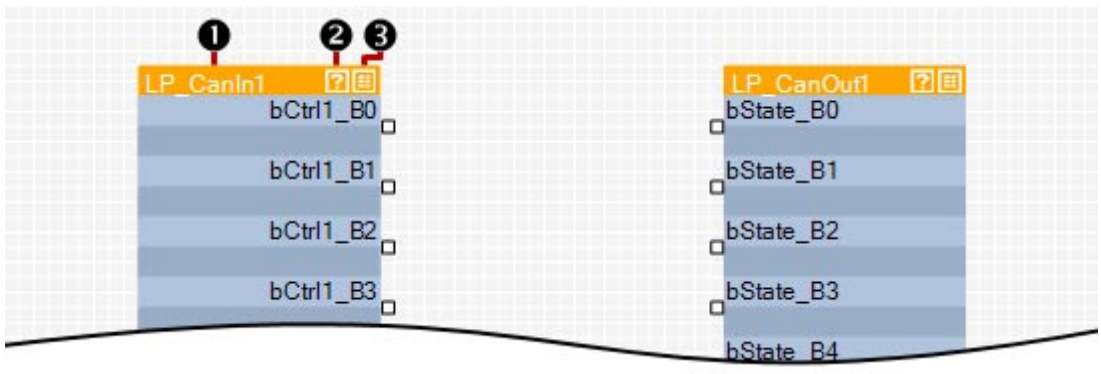
[18-3] Example: System block "LS_MotorInterface" for mapping the motor control

### 18.1.1.4 What is a port block?

A port block is a signal interface to a fieldbus. Input/output ports represent the input and output process data of the fieldbus.

- Port blocks LP_CanIn/LP_CanOut: Signal interface to the CAN bus
- Port blocks LP_MciIn/LP_MciOut: Signal interface to a plugged fieldbus module

If, for instance, the inverter is to be controlled via CAN bus or a fieldbus module, the input/output ports are connected to the application block (device-internal signal processing) in the I/O level of the FB Editor.




❶ Name of the port block
❷ Button for opening the online help for the port block
❸ Button for opening the parameterisation dialog or the parameter list for the port block

[18-4] Example: Input port "LP_CanIn1" and output port "LP_CanOut1"

### 18.1.1.5 What is an application block?

The application/technology function set in C00005 is shown as application block in the I/O level of the FB Editor.

The application block comprises the signal flow processing generated via function block interconnection for the selected application in each case (e.g. "actuating drive speed" or "switch-off positioning"). The function block interconnection is shown in detail on the application level.

**Tip!**

Every application block features so-called "free inputs and outputs" which you can use to transfer signals from the I/O level to the application level and vice versa.

- In the Lenze setting, these connectors are hidden in the function block editor.
- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

### 18.1.2 Conventions used for input/output identifiers

This chapter describes the conventions used for the identifiers of the inputs/outputs of the blocks. The conventions ensure a uniform and consistent terminology and make reading and comprehending the interconnection and application easier.

**Tip!**

The conventions used by Lenze are based on the "Hungarian Notation". This ensures that the most significant characteristics of the corresponding input/output (e.g. the data type) can be instantly recognised from its identifier.

An identifier consists of

- a data type entry
- an identifier (the "proper" name of the input/output)
- an (optional) signal type specification

#### Data type entry

The data type entry provides information about the data type of the corresponding input/output:

| Data type entry | Meaning | Resolution | Value range |
|---|---|---|---|
| b | BOOL | 1 bit | 0 ≡ FALSE / 1 ≡ TRUE |
| dn | DINT | 32 bits | -2147483647 ... 2147483647 |
| n | INT | 16 bits | -32767 ... 32767 |
| w | WORD | 16 bits | 0 ... 65535 |

#### Identifier

The identifier is the proper name of the input/output and should indicate the application or function.

- Identifiers always start with a capital letter.
- If an identifier consists of several "words", then each "word" must start with a capital letter.
- All other letters are written in lower case.

**Signal type entry**

In general, it is possible to assign a certain signal type to the inputs and outputs of the Lenze function blocks. There are e.g. digital, scaled, position, acceleration and speed signals.

- A corresponding ending (preceded by an underscore) is added to the identifier of the corresponding input/output to indicate the signal type.

| Signal type entry & port symbol in the FB Editor | | Meaning | Resolution | Value range |
|---|---|---|---|---|
| _a | ○ | Analog/scaled | 16 bits | ± 199.99 % |
| _v | ◁ / ▷ | Angular velocity | 16 bits | ± 30000.0 rpm |
| _p | ◀ / ▶ | Position | 32 bits | $-2^{31}$ ... $2^{31}$-1 increments |
| | □ | Digital (BOOL) | 8 bits | 0 ≡ FALSE; 1 ≡ TRUE |
| | ▬ | Other (WORD) | 16 bits | 0 ... 65535 |
| | ■ | Other (DINT) | 32 bits | -2147483647 ... 2147483647 |

## 18.1.3 Scaling of physical units

With regard to the parameter setting & configuration of the inverter it is very helpful to know the signal types and their scaling listed in the following table, which are used to process physical values (e.g. an angular velocity or position) in the function block interconnection.

| Signal type entry & port symbol in the FB Editor | | Meaning | Scaling | |
|---|---|---|---|---|
| | | | External value | ≡ internal value |
| _a | ○ | Analog/scaled | 100 % | ≡ $2^{14}$ ≡ 16384 |
| _v | ◁ / ▷ | Angular velocity | 15000 rpm | ≡ $2^{14}$ ≡ 16384 |
| _p | ◀ / ▶ | Position | 1 encoder revolution | ≡ $2^{16}$ increments |

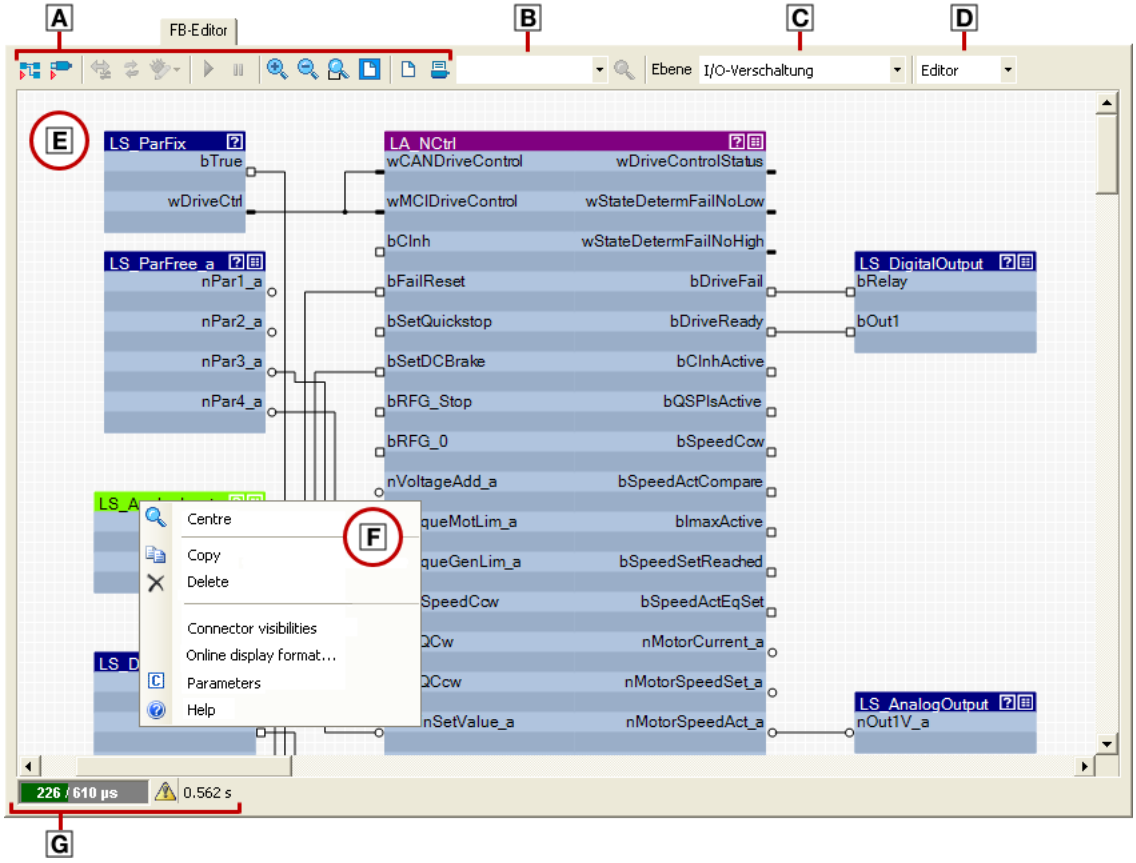
## 18.2 User interface

**How to access the FB Editor:**

1. Go to the *Project view* and select the 8400 inverter.
2. Go to *Workspace* and select the **FB Editor** tab.

   The FB Editor displays the wiring of the technology function selected in C00005. The interconnection of the I/Os of the inverter depends on the control mode selected in C00007.

The user interface of the FB Editor includes the following control and function elements:




A Toolbar | B Search function | C Level selection | D Editor view/overview

E Drawing area | F Context menu | G Status bar | Not shown: Overview

**Tip!**

Go to the *»Engineer« toolbar* and click the icon to hide the *Project View* and the *Message Window*. This increases the *Workspace* available for the FB Editor. A renewed click on the symbol shows the *Project View* and the *Message Window* again.

### 18.2.1 Toolbar

The FB Editor is provided with an individual toolbar in the upper position which in the following text is called *FB Editor toolbar*.

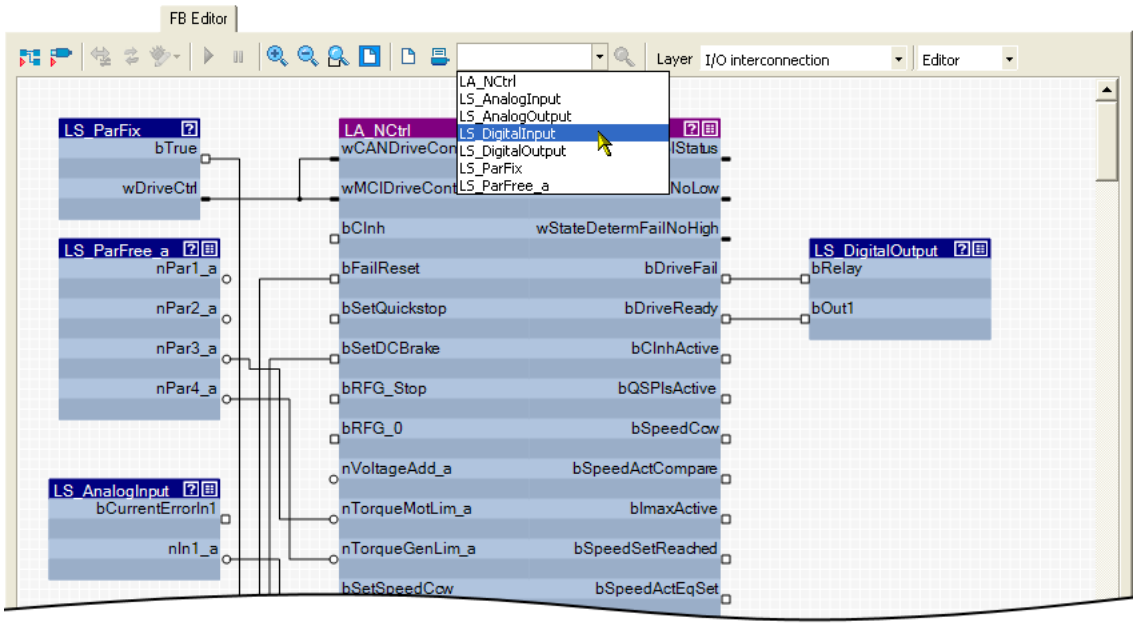
- Click on an icon to execute the corresponding function.

| Symbol | Function |
|---|---|
|  | Insert function block or system block<br>▸ Inserting a function block (📖 1395)<br>▸ Inserting a system block (📖 1397) |
|  | Insert port block (📖 1399) |
|  | Adjusting online and offline interconnection (📖 1418) |
|  | Acknowledge error in the interconnection / reload interconnection |
|  | Correct interconnection |
|  | Start online monitoring |
|  | Interrupt online monitoring |
|  | Close online monitoring |
|  | Enlarge view of interconnection |
|  | Reduce view of interconnection |
|  | Enlarge cutout of interconnection |
|  | Show total interconnection in the drawing area |
|  | Show print view |
|  | Printing the interconnection (📖 1419) |
|  | Search function (📖 1382) |

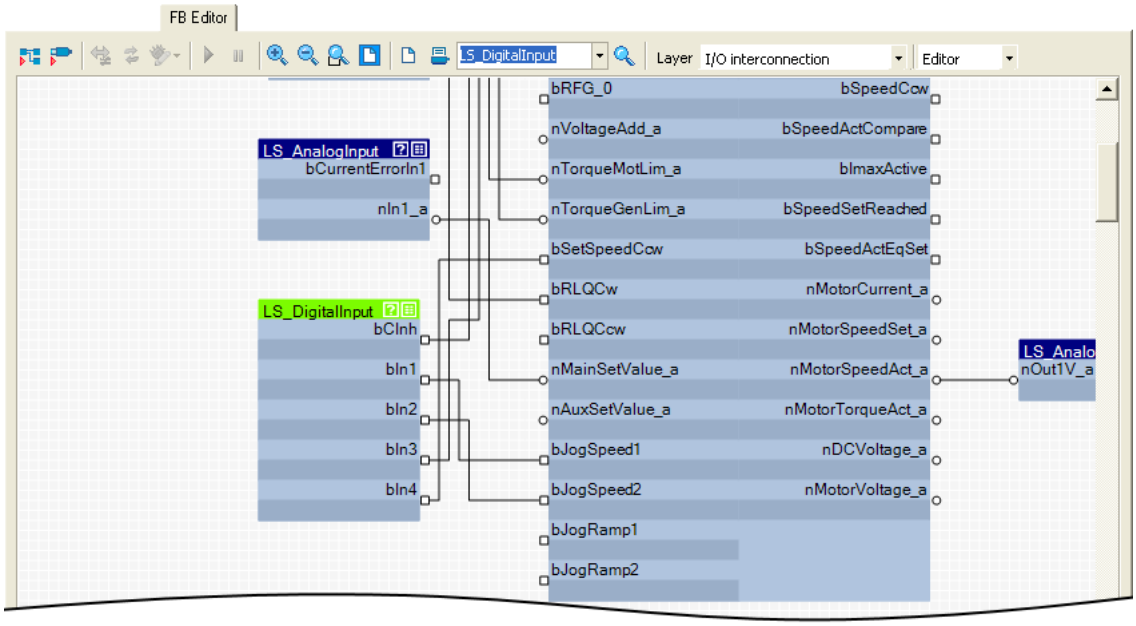
### 18.2.2 Search function

Use the search function to get quickly to a certain module of the interconnection.

- The list field of the search function contains all function blocks, system blocks, and port blocks of the interconnection:




- When you select a module in the list field, this module is zoomed in and selected at the same time (the following example shows the **LS_DigitalInput** system block):



**Tip!**

You can also enter any search text in the input field.

- If you click the icon, the cutout is moved to the object which contains this search text.
- Another click on the icon leads to a new search. Thus, you can navigate successively to all objects which contain the entered search text.
- The search text does not consider case sensitivity.

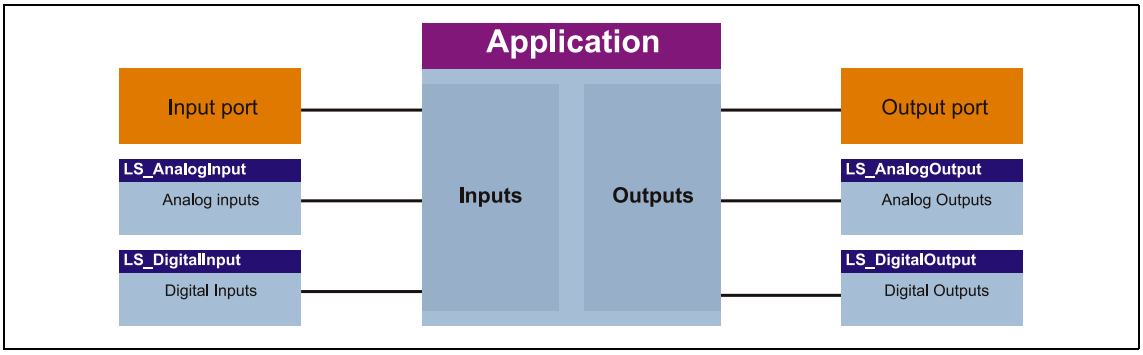
### 18.2.3 Level selection

Go to the **Level selection** list field and select the interconnection level to be displayed.

**"I/O interconnection" level**

This level displays only the I/O interconnection of the currently selected technology application for a better overview.

- Details of the application are masked out in this level.
- The interconnection of the I/Os of the inverter with the inputs and outputs of the application in detail depends on the control mode selected in C00007.
- The parameterisation dialogs on the **Application parameter** tab correspond to the application block displayed in this level.

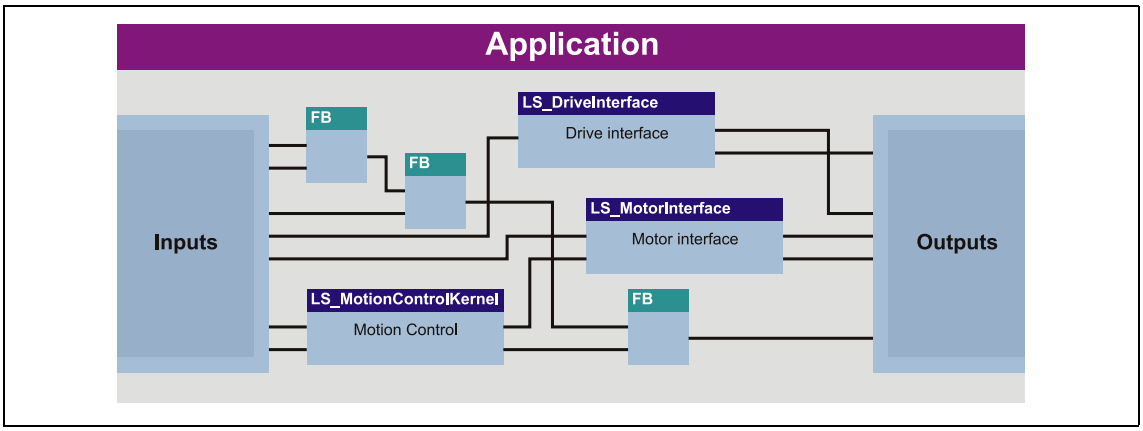



[18-5] Schematic diagram of "I/O interconnection"

### "Application interconnection" level

This level displays the interconnection of the application selected in C00005 in detail. All function blocks used in the application and the system blocks which provide the interfaces to the drive and motor interface and to the MotionControlKernel (MCK) are displayed with their connections.

- The interconnection of the I/Os of the inverter with the inputs and outputs of the application is masked out in this level.




[18-6] Schematic diagram of "Application interconnection"

**Tip!**

Every application block features so-called "free inputs and outputs" which you can use to transfer signals from the I/O level to the application level and vice versa.

- In the Lenze setting, these connectors are hidden in the function block editor.
- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

### "Free interconnection" level

This level serves to implement an individual drive solution for the device versions "StateLine C" (from version 12.00.00 and »Engineer« V2.17), "HighLine C" and "TopLine C".

**Note!**

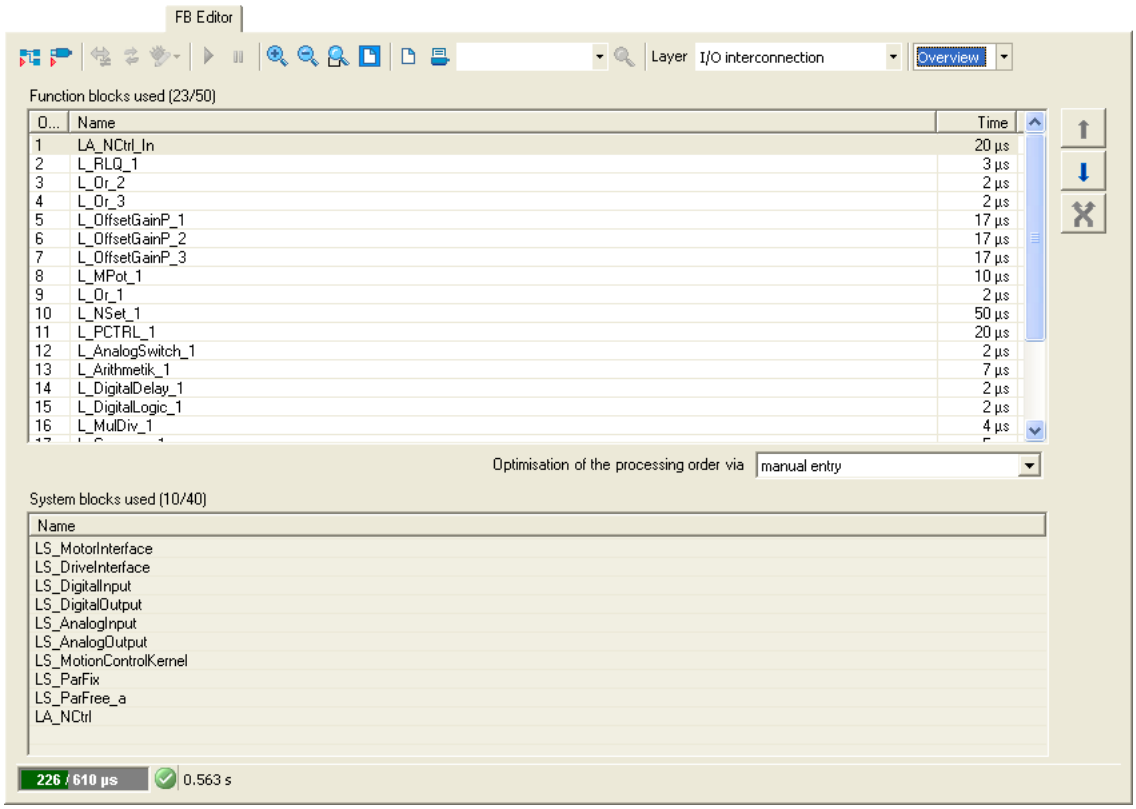
When you select the "Free interconnection" level for the first time, you are prompted to confirm whether the interconnection from the I/O level and the application level are to be combined and copied into this level.

When you confirm this confirmation prompt with **Yes**, the I/O level and the application level are not available anymore. This action can only be undone by resetting the application to a predefined Lenze application! ▸ Resetting changed interconnection (📖 1417)

### 18.2.4 Editor view/overview

Use the list field at the top right to change from the Editor to the overview and vice versa.

The overview shows all function blocks used of the interconnection in the upper list field in the order of their processing. The lower list field shows all used system blocks.

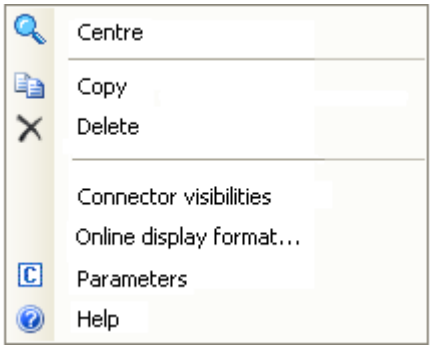



- The processing order of the function blocks can be optimised manually or according to an automatically generated selection. ▸ Changing the processing order (📖 1412)

### 18.2.5 Context menu

You can open a *context menu* via the right mouse button for each object (function block, system block, line, comment, etc.) and for the drawing area:

- The contents of the *context menu* depend on the type of object you click on.
- Example: *Context menu* for a function block:




### 18.2.6 Status bar

The status bar of the FB Editor shows, among other things, information about the system load and the error status of the interconnection:




| Symbol | Meaning |
|---|---|
| **A System load** | |
| 226 / 610 µs | Here: out of the available computing time of 610 µs, 226 µs are required by the application. |
| **B Error status of the interconnection** | |
| | The interconnection has no errors and no warnings |
| | The interconnection has errors and/or warnings |
| **C Communication status** | |
| | Offline |
| | Online |
| | Communication error |
| **D Adjustment status** | |
| = | Offline and online interconnection match |
| ≠ | Offline and online interconnection are different |
| **E Update rate for monitoring values** | |

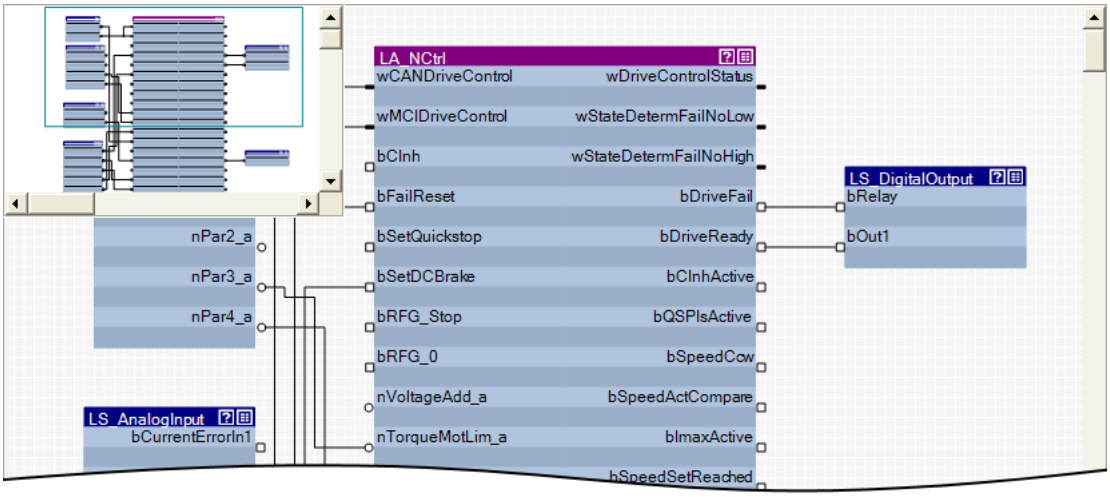
### 18.2.7 Overview

The overview window shows the drawing area in a reduced view. The overview window serves to e.g. move quickly through a more complex interconnection.

**How to show the monitor window:**

Go to the *Context Menu* of the drawing area and select the **Overview Window**.

- If you execute this command again, the overview window is hidden again.



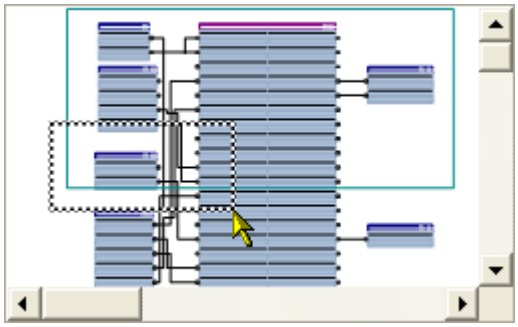

- The green frame in the overview window indicates the interconnection cutout that is currently displayed in the drawing area.
- Use the mouse pointer to shift and resize the cutout to be displayed.

**How to shift the cutout presented in the drawing area:**

1. Position the mouse pointer to the green frame in the overview window.
   - The mouse pointer symbol becomes a positioning cross.
2. Click left mouse button and shift the green frame to its new position by keeping the mouse button pressed, so that the desired cutout of the interconnection is displayed in the drawing area.

**How to redefine the cutout to be presented:**

In the overview window draw a frame around the area of the interconnection which is to be presented in the drawing window by keeping the left mouse button pressed:



- The aspect ratio of the frame is automatically adapted to the aspect ratio of the drawing area.
- According to the size of the frame that is drawn, also the presentation size of the objects in the drawing area changes.

**Tip!**

Go to the *FB Editor toolbar* and click the icon to adapt the view size so that all objects included in the interconnection are visible in the drawing area.

**Automatic scroll ("AutoScroll function")**

If you reach a window limitation in the drawing area when shifting an object or in the overview window when shifting the green frame, and if you then shortly hold the mouse pointer in this position, an automatic scrolling into the corresponding direction is carried out:

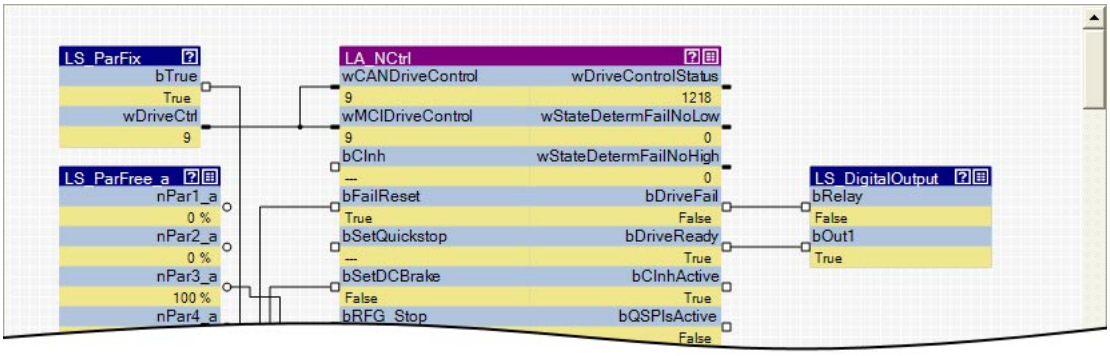
## 18.3 Using the FB Editor as "Viewer"

The main purpose of the FB Editor is the individual configuration of the selected technology application. However, you can also use the FB Editor to

- make a diagnosis of the application (when an online connection has been established),
- get a better understanding for the operating mode of the application,
- use the interconnection as an alternative parameterisation access.

### Diagnostics of the application

When an online connection to the inverter has been established, the current values are displayed at the inputs and outputs of the objects:




- Process-scaled signals can be scaled in a "user-defined" way for easy diagnostics in the FB Editor.
  ▸ Change online display format (📖 1392)

### Getting a better understanding for the operating mode of the application

Make yourself familiar with the signal flow of the interconnection to get a better understanding of the operating mode of the application or individual functional areas.

- The symbol in the head of the block or the **Help** command in the *context menu* for the block serve to open the online help for the block.

### Using the interconnection as an alternative parameterisation access

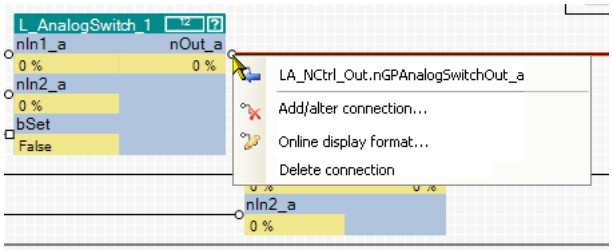
- The icon in the head of the module, a double-click on the module, or the **Parameter...** command in the *Context menu* of the module serve to open the parameterisation dialog or the parameter list for the module.

### 18.3.1 Following connections of inputs and outputs

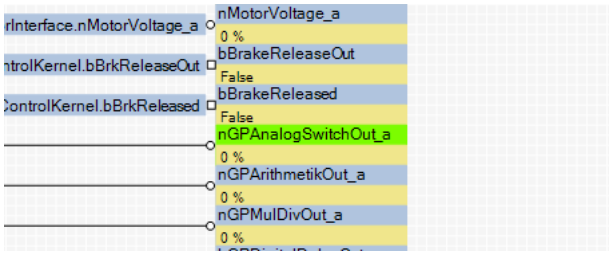
In addition to the Search function you can use the *context menu* of inputs and outputs to follow connections and quickly reach certain signals.

**How to navigate from one output to another connected input:**

1. Open the *context menu* (right mouse button) of the port symbol at the output.
   - The *context menu* for the port symbol contains all inputs which are connected to the output:



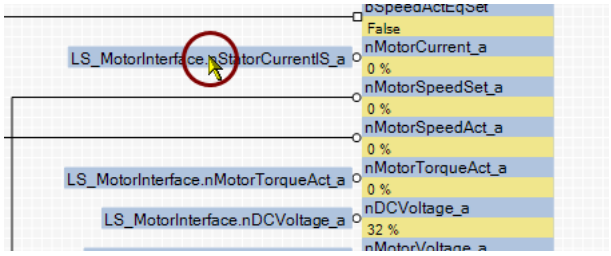

2. Select input in the *context menu* to which you want to navigate.
   - As a result, the selected input is displayed in the centre of the drawing area (in this example: nGPAnalogSwitchOut_a):



**How to navigate from one input to another connected output:**

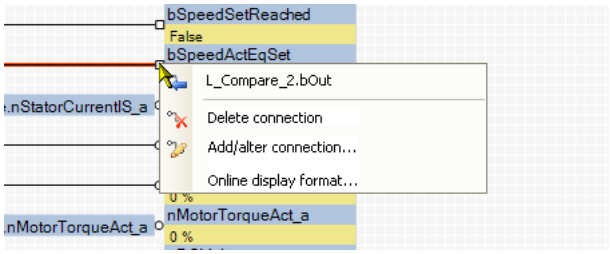
If the input is connected to a flag:

- Double-click the flag:




If the input is connected to a line:

1. Open the *context menu* (right mouse button) of the port symbol at the output:




2. Select output in the *context menu*.
   - Since an output can only be connected to an input, the *context menu* contains only an output.

The output is displayed in the centre of the drawing area.

## 18.3.2 Keyboard commands for navigation

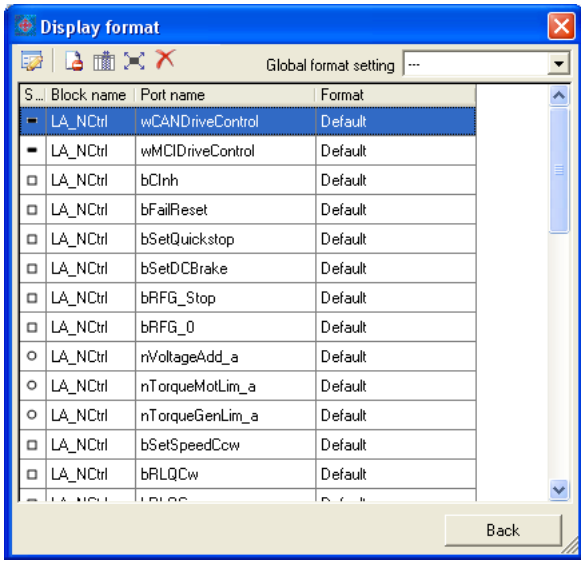
| Keyboard command | Function |
|---|---|
| **<Picture ▲ >** | Scroll up |
| **<Picture ▼ >** | Scroll down |
| **<Shift> + <picture ▲ >** | Scroll to the left |
| **<Shift> + <picture ▼ >** | Scroll to the right |
| **<POS1>** | Scroll to the left edge of the interconnection |
| **<END>** | Scroll to the right edge of the interconnection |
| **<Ctrl> + <Pos1>** | Scroll to the left upper corner of the interconnection |
| **<Ctrl> + <End>** | Scroll to the right lower corner of the interconnection |

### 18.3.3 Change online display format

For online monitoring in the FB Editor the display format of the input and output data of a block can be adapted individually. Process-scaled signals can be scaled in a "user-defined" way for easy diagnostics in the FB Editor. Thus, the display of these signals gets a process reference.

**How to change the data display format of block inputs/outputs:**

1. Go to the *context menu* of the block and select the **Online display format** command.
   - **Tip**: You can call the *context menu* of a block by clicking with the right mouse button on the header of the block.
   - The *Display format* dialog box is displayed:




2. Select the inputs/outputs from the list the display format of which is to be changed.
   - **Note**: In the **Global format setting** list field the "---" entry must be selected so that the display format can be changed.
   - If you click further inputs/outputs while pressing **<Ctrl>** they are added to an already existing selection (multi-selection).
   - The **<Shift>** key serves to select a related area of inputs/outputs.
   - More functions:

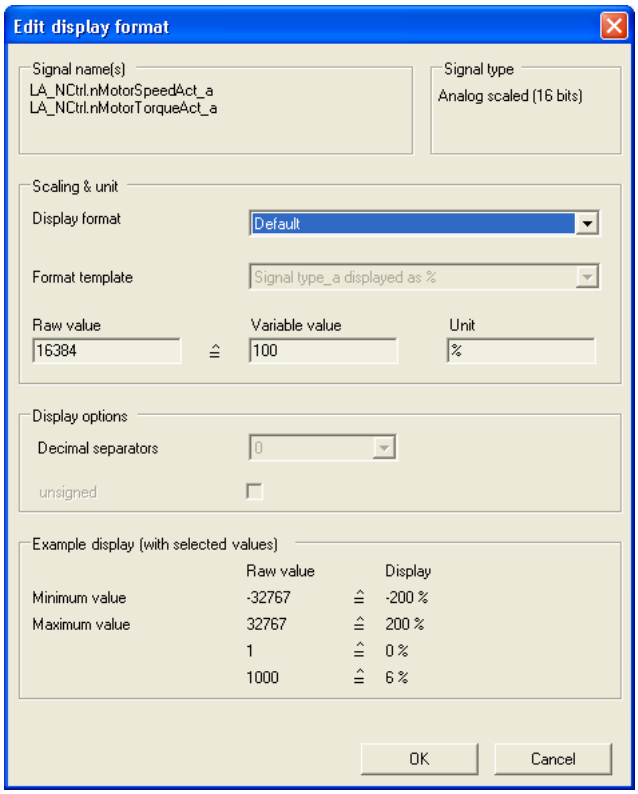
     Display masked out connections

     Display additional information

     Select all inputs/outputs

     Reset all format information

3. Click the symbol to edit the display format of the selected inputs/outputs.
   - The *Edit display format* dialog box is displayed:




4. Go to the **Display format** list field and select the "User-defined" entry.
5. Go to the **Format template** list field and select "No template".
6. Select the required scaling, unit, number of decimal positions, and sign handling.
7. Click **OK** to accept the settings and close the *Edit display format* dialog box.
   - The *Display format* dialog box now displays the text "User-defined" for the changed inputs/outputs in the **Format** column.

   After all required formats have been changed:
8. Click **Back** to close the *Display format* dialog box.
   - For online monitoring, the changed format is used.

## 18.4 Reconfiguring the predefined interconnection

How to proceed:

1. Insert additionally required objects into the interconnection.
2. Hide unneeded inputs/outputs of function blocks and system blocks to obtain a clearly arranged interconnection.
3. Arrange the objects in the drawing area in a reasonable manner.
4. Establish the connections required for the desired function.
5. If required, change (optimise) the processing order of the function blocks.

**Tip!**

Detailed information on the individual steps can be obtained from the following subchapters!

**Note!**

With the "StateLine" version, the interconnection shown in the application level cannot be edited.

### 18.4.1 Inserting/Deleting objects

Objects can be inserted in the interconnection via the *FB Editor toolbar* and the *context menu* of the drawing area. The following subchapters provide detailed information on how to insert/delete the different objects.

| Symbol | Function |
|---|---|
|  | Inserting a function block (📖 1395) |
| | Inserting a system block (📖 1397) |
|  | Insert port block (📖 1399) |
| | Inserting a comment (📖 1401) |

**Tip!**

Use the *context menu* of the drawing area to insert a function block, system block, port block or comment directly to the current position of the mouse pointer in the drawing area.

If you insert an object via the corresponding icon in the *FB Editor toolbar*, the object is always placed at the top left corner in the drawing area.

Interconnection elements cannot only be copied within the same interconnection but also across all devices within the same project, as long as the devices stem from the same product family. ▸ Copying interconnection elements (across all devices) (📖 1414)

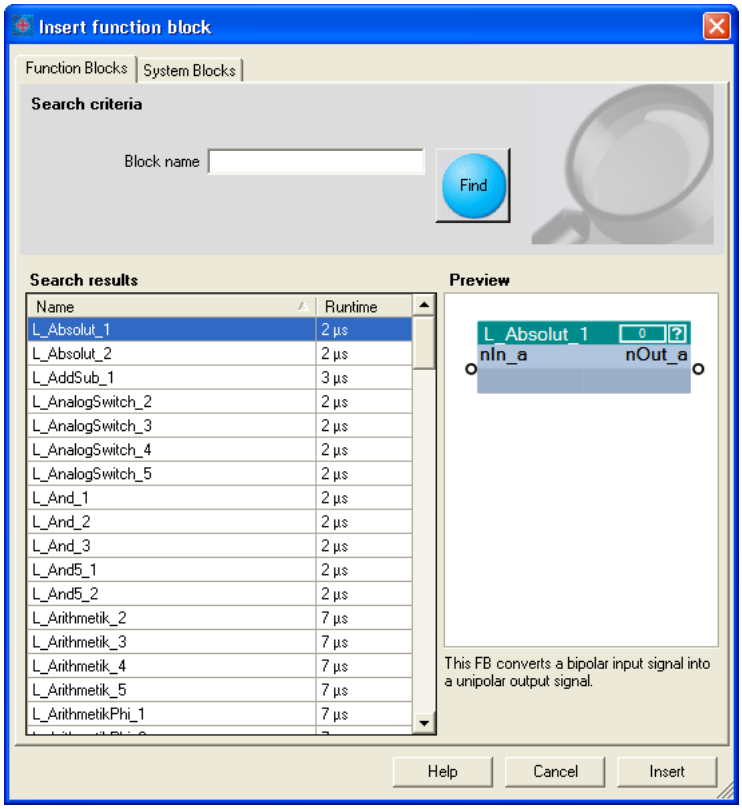
### 18.4.1.1 Inserting a function block

**Note!**

In the FB Editor, function blocks are only available in the "Application interconnection" level!

**How to insert a function block into the interconnection:**

1. In the *FB Editor toolbar*, click the icon.
   - The *Insert Function Block* dialog box appears:
2. Unless it is already displayed, select the **Function Blocks** tab.
   - All function blocks available are displayed in the **Search results** list field.




   - A preview of the selected function block is displayed.
   - A detailed description of all available function blocks can be found in the main chapter "Function library". (📖 1425)
3. If required, define **Search criteria** to narrow down the available function blocks:
   - **Block name**:
     String which must be contained in the name of the function block.
4. After changing the search criteria, press the **Find** button to update the selection.
   - Then, only the function blocks complying with the features set in the search criteria are shown in the **Search Results** list field.
   - If no search criteria are set, all function blocks available are shown.
5. Select the function block to be inserted in the **Search results** list field.

6. Click the **Insert** button.
   - The dialog box is closed and the selected function block is inserted into the interconnection.

### Context menu for the function block

If you right-click on the header of a function block, a *context menu* opens via which you can execute the following functions in addition to the general processing functions (Copy, Insert, Delete):

| Command | | Function |
|---|---|---|
|  | **Centre** | Move the visible cutout of the drawing area so that the block is centred. |
| | **Connector visibilities...** | Define visible inputs and outputs of the block.<br>▸ Changing connector visibilities (📖 1404) |
| | **Online display format...** | Adapt the display format of the input and output data of the block individually for online monitoring.<br>▸ Change online display format (📖 1392) |
|  | **Parameters...** | Open the parameter list/parameterisation dialog for the block.<br>• Only if function block is parameterisable. |
|  | **Help** | Show online help for the block. |

### Related topics

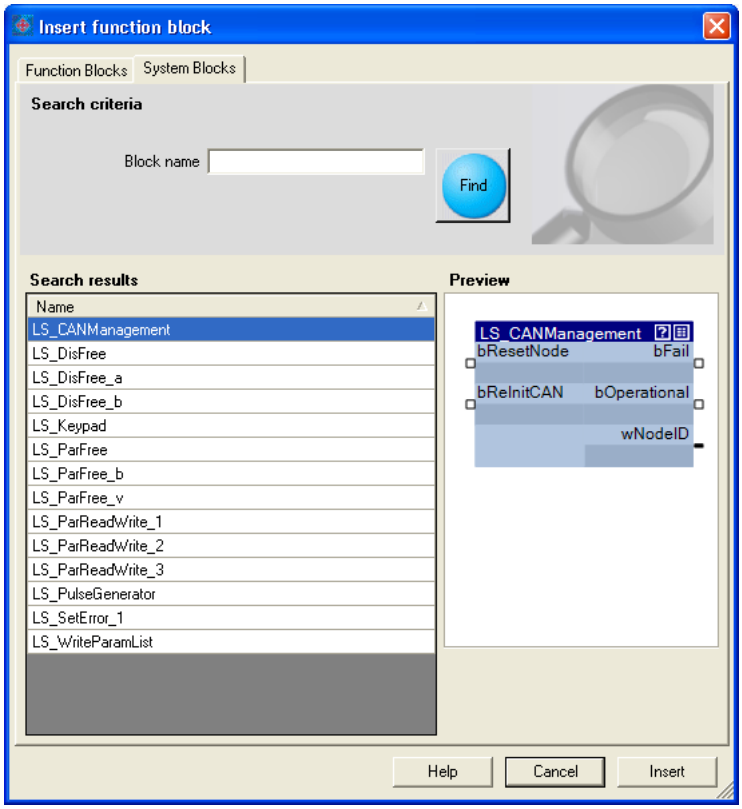
- Deleting objects that are no longer required (📖 1403)
- Changing connector visibilities (📖 1404)
- Arranging objects in the drawing area (📖 1405)
- Creating/deleting connections (📖 1406)
- Changing the processing order (📖 1412)

### 18.4.1.2 Inserting a system block

A system block is inserted similarly to the way a function block is inserted.

**How to insert a system block into the interconnection:**

1. In the *FB Editor toolbar*, click the icon.
   - The *Insert Function Block* dialog box appears:
2. Unless it is already displayed, select the **System Blocks** tab.
   - All system blocks available are displayed in the **Search results** list field.




   - A preview of the selected function block is displayed.
3. If required, define **Search criteria** to accordingly narrow down the system blocks available:
   - **Block name**:
     String which must be contained in the name of the system block.
4. After changing the search criteria, press the **Find** button to update the selection.
   - Then, only the system blocks complying with the features set in the search criteria are shown in the **Search Results** list field.
   - If no search criteria are set, all system blocks available are shown.

5. Select the system block to be inserted in the **Search results** list field.
6. Click the **Insert** button.
   - The dialog box is closed and the selected system block is inserted into the interconnection.

### Context menu for the system block

If you right-click on the header of a system block, a *context menu* opens via which you can execute the following functions in addition to the general processing functions (Copy, Insert, Delete):

| Command | | Function |
|---|---|---|
|  | **Centre** | Move the visible cutout of the drawing area so that the block is centred. |
| | **Connector visibilities...** | Define visible inputs and outputs of the block.<br>▸ Changing connector visibilities (📖 1404) |
| | **Online display format...** | Adapt the display format of the input and output data of the block individually for online monitoring.<br>▸ Change online display format (📖 1392) |
|  | **Parameters...** | Open the parameter list/parameterisation dialog for the block. |
|  | **Help** | Show online help for the block. |

### Related topics

- Deleting objects that are no longer required (📖 1403)
- Changing connector visibilities (📖 1404)
- Arranging objects in the drawing area (📖 1405)
- Creating/deleting connections (📖 1406)

### 18.4.1.3 Insert port block

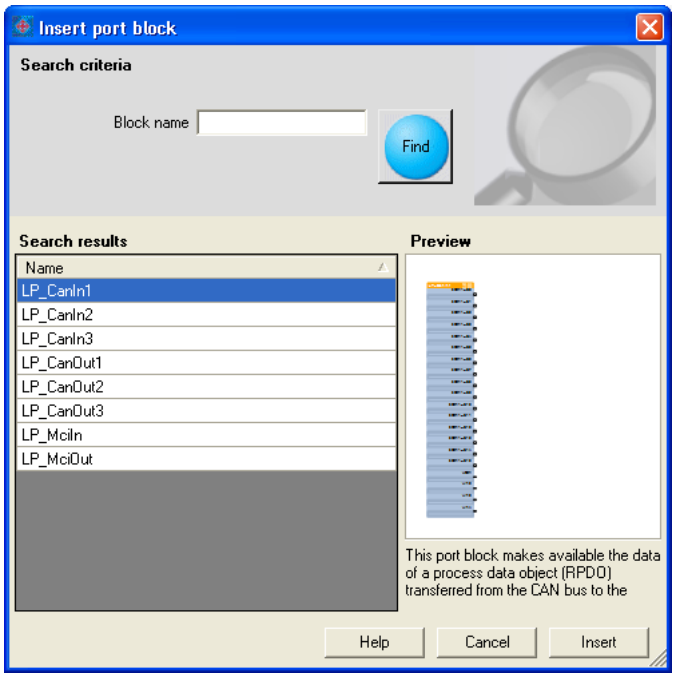
All input/output ports defined for the application on the **Ports** tab can be inserted into the interconnection in the form of port blocks in order to get access to the associated element variables.

**Tip!**

You can change between the **Ports** and **FB Editor** tabs at any time to define new ports and afterwards insert them into the interconnection.

**How to insert a port block into the interconnection:**

1. In the *FB Editor toolbar*, click the icon.
   - The *Insert port block* dialog box appears.
   - All port blocks available are displayed in the **Search results** list field.




   - A preview of the selected port block is displayed.
2. If required, define **search criteria** to accordingly narrow down the port blocks available:
   - **Block name**:
     String which must be contained in the name of the port block.
3. After changing the search criteria, press the **Find** button to update the selection.
   - Then, only the port blocks complying with the features set in the search criteria are shown in the **Search Results** list field.
   - If no search criteria are set, all port blocks available are shown.
4. Select the port block to be inserted in the **Search results** list field.
5. Click the **Insert** button.
   - The dialog box is closed and the selected port block is inserted into the interconnection.

### Context menu for the port block

If you right-click on the header of a port block, a *context menu* opens via which you can execute the following functions in addition to the general processing functions (Copy, Insert, Delete):

| Command | | Function |
|---|---|---|
|  | **Centre** | Move the visible cutout of the drawing area so that the block is centred. |
| | **Connector visibilities...** | Define visible inputs and outputs of the block.<br>▸ Changing connector visibilities (📖 1404) |
| | **Online display format...** | Adapt the display format of the input and output data of the block individually for online monitoring.<br>▸ Change online display format (📖 1392) |
|  | **Parameters...** | Open the parameter list/parameterisation dialog for the block. |
|  | **Help** | Show online help for the block. |

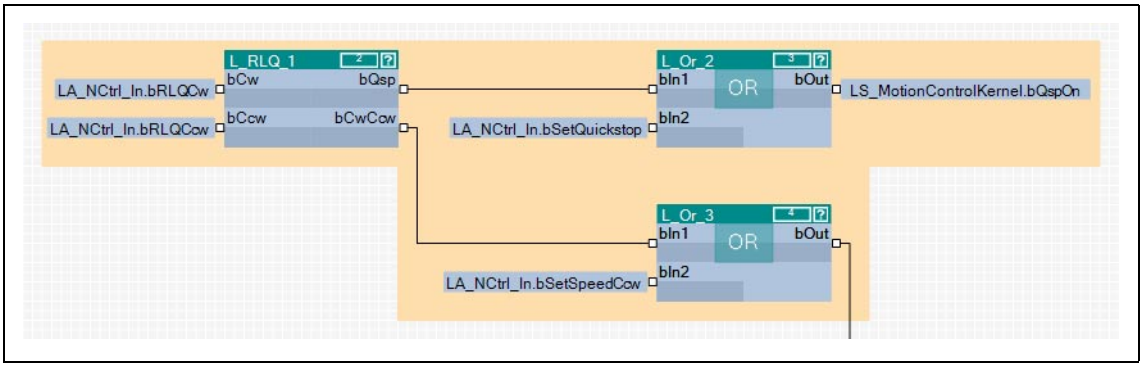
### Related topics

- Deleting objects that are no longer required (📖 1403)
- Changing connector visibilities (📖 1404)
- Arranging objects in the drawing area (📖 1405)
- Creating/deleting connections (📖 1406)

### 18.4.1.4 Inserting a comment

Comments can be inserted at any position in the drawing area.

As of the »Engineer« V2.10, the interior colour and text alignment of a comment can be changed via a properties dialog. Now the sizes of comments can also be changed using the mouse pointer. When using different interior colours you can use comments to graphically arrange areas that belong together in terms of function or separate them from other areas:



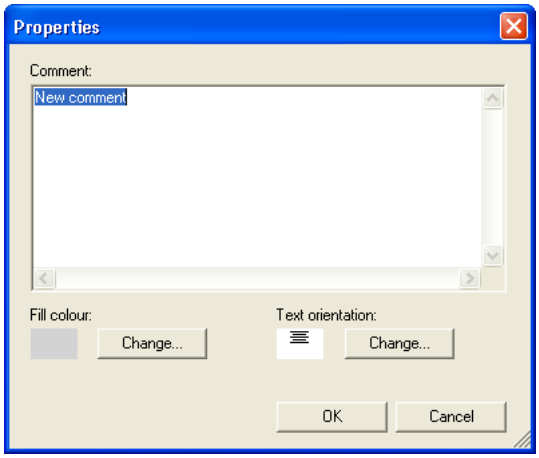

[18-7] Example: Graphical arrangement of FBs by means of two comments that overlap.

**Note!**

The term "Arrangement" does not mean a logical arrangement of the function blocks. The comments are only graphical presentation elements of the FB Editor.

**How to insert a new comment into the interconnection:**

1. Move the mouse pointer to the (free) position in the drawing area where the comment is to be inserted.
2. Go to the *Context menu* (right mouse key) and select the **New comment** command.
   - The *Properties* dialog box is displayed:




3. Enter the required comment into the text field.
4. Optional: Change preset interior colour.
   - For this purpose, click the left **Change...** button to open the *Colour* dialog box to select another interior colour.

5. Optional: Change preset text alignment.
   - For this purpose, click the right **Change...** button to open the *Text alignment* dialog box to select another text alignment.
6. Press **OK** to close the *Properties* dialog box and insert the comment.
   - After being inserted, the corner points of the comment are shown:




7. Optional: Change size of the comment.
   - For this purpose click one of the corner points with the left mouse button and enlarge the comment to the required size with the mouse button pressed.




8. Optional: Drag comment.
   - For this purpose click the comment with the left mouse button and move the comment to the required position with the mouse button pressed.

**Tip!**

The *Properties* dialog box for a comment already available can be opened by double-clicking the comment.

**Related topics**

- Deleting objects that are no longer required (📖 1403)
- Arranging objects in the drawing area (📖 1405)
- Creating/deleting connections (📖 1406)

### 18.4.1.5 Deleting objects that are no longer required

Objects that are no longer required can be easily deleted again. "Delete" only means that the object is removed from the drawing area. If you have deleted an object from the drawing area, you can reinsert it any time into the interconnection.

**Note!**

Deleting an object cannot be undone.

Together with the object, all available connections to this object are deleted.

**How to delete objects that are no longer required:**

1. Select objects to be deleted.
   - You can select a single object by clicking the header of the object.
   - You can select objects that are placed together by drawing a frame around these objects while keeping the mouse button pressed.
   - If you click the header of further objects while pressing **<Ctrl>**, these will be added to an already existing selection (multi-selection).
   - All selected objects are highlighted by a light green header.
2. Press **<Del>**.

**Related topics**

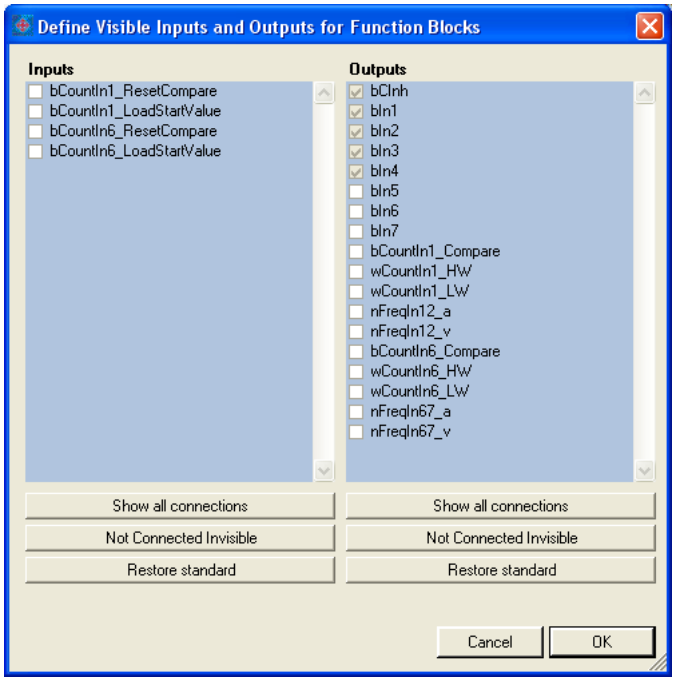
- Deleting connections that are no longer required (📖 1411)

### 18.4.2 Changing connector visibilities

Inputs and outputs that are not connected can be hidden for each block. This serves to reduce the dimension of the block. The interconnection becomes clearer.

**How to define the visible inputs and outputs:**

1. Go to the context menu of the block and select the **Connector visibilities** command.
   - The *Define Visible Inputs and Outputs for Function Blocks* is displayed:




   - All visible connections have a checkmark.
   - In case of a block that is inserted anew, all inputs and outputs are visible at first.
   - Inputs and outputs with a light grey checkbox are already connected and thus cannot be hidden.
2. By setting/removing the checkmarks or via the buttons you can define the visible inputs and outputs.
3. Press **OK** to accept the selected definition and close the dialog box.

### 18.4.3 Arranging objects in the drawing area

All objects can be freely arranged in the drawing area by dragging with the mouse.

We recommend to make an arrangement in which the required connections between the inputs and outputs can be created easily. A division into functional areas may also be sensible to get a better understanding of the application.

Objects which are already connected, can also be dragged to another (free) position in the drawing area. The available connections will be automatically re-routed after dragging.



**How to drag an object:**

1. Click the header of the object (and keep the button pressed).
2. Keep the button pressed and drag the object to the required position in the drawing area.
   - Via **<Esc>** you can cancel this action.



**How to drag several objects at the same time:**

1. Select the objects to be dragged.
   - You can select a single object by clicking the header of the object.
   - If you click the header of further objects while pressing **<Ctrl>**, these will be added to an already existing selection (multi-selection).
   - You can easily select objects that are placed together by drawing a frame around these objects while keeping the mouse button pressed.
   - All selected objects are highlighted by a light green header.
2. Keep the mouse button pressed on the header of one of the selected objects and drag it to the required position in the drawing area.
   - Via **<Esc>** you can cancel this action.



**Note!**

A red header indicates that the object overlaps with other objects in the drawing area!

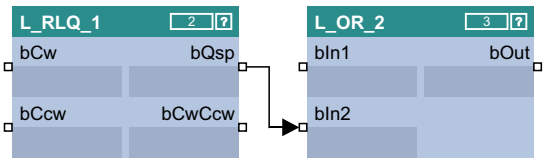
Arrange the objects so that no overlap occurs.

### 18.4.4 Creating/deleting connections

After adding objects and arranging them in a reasonable manner within the drawing area, you can create the connections between the available objects which are required for the desired function.

A connection always has a direction and therefore always has a source and a target.

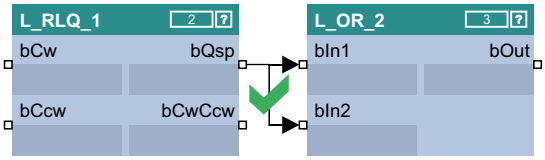
- An output represents a possible source in the interconnection.
- An input represents a possible target in the interconnection.




#### Permissible/impermissible connections

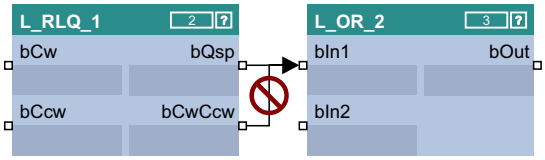
Several connections can lead from one output.

- Therefore it is always possible to start a new connection from an output.



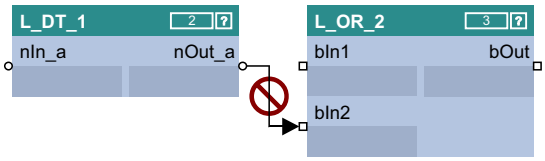

However, maximally one connection may end in an input.

- Therefore it is only possible to start a new connection from an input if there is no connection already ending in this input.



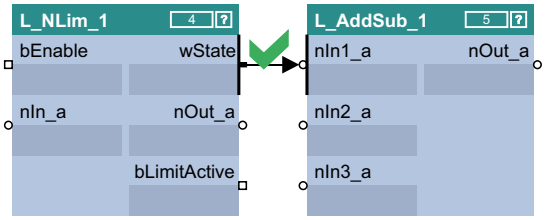

Only inputs/outputs of the same signal type can be connected.

- Thus, a connection between different port symbol cannot be established.




From the »Engineer« V2.12 "Analog/scaled" (_a) and "Miscellaneous (WORD)" signal types can also be interconnected.

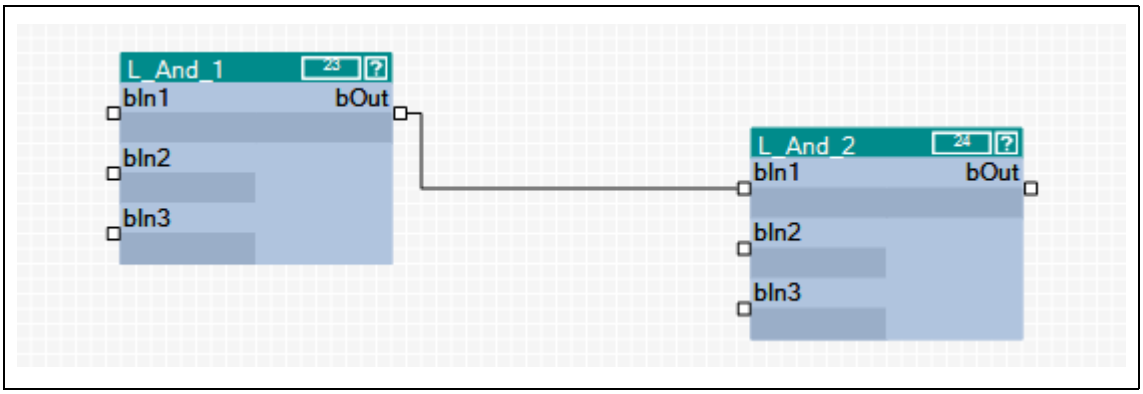
- The implicit type conversion is indicated by a vertical black bar at the port symbol.



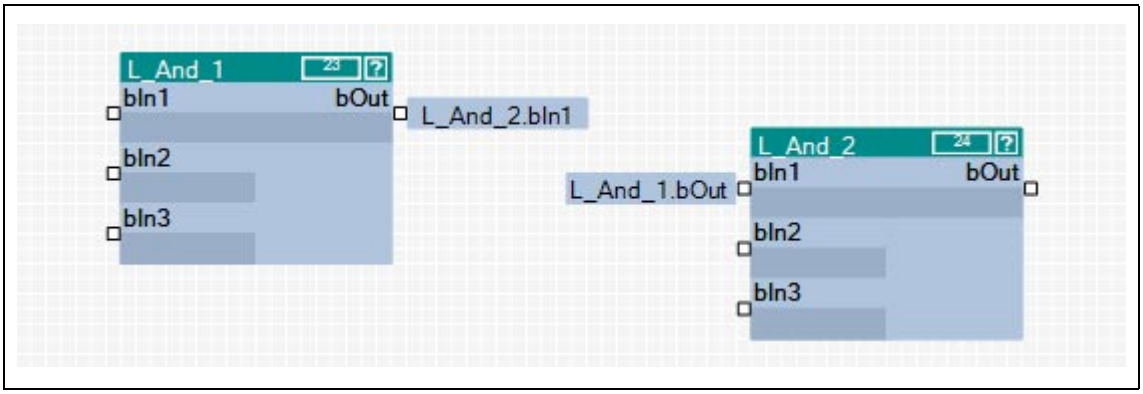

From the »Engineer« V2.13 "Analog/scaled" (_a) and "Angular velocity" (_v) signal types can also be interconnected.

### Connection types

Connections can either be created by means of connection lines or port identifiers ("flags")




[18-8] Example 1: Connection via connection line




[18-9] Example 2: Connection via flags

**Tip!**

The commands **Show as flag** or **Show as line** in the *context menu* of a connection serve to change the representation of the connection at any time.

When an output is connected to several inputs via flags, three points are displayed ("...") at the output instead of the concrete input identifier. The *context menu* of the port symbol shows all inputs which are connected to the output.

### 18.4.4.1 Creating a connection using the connection line

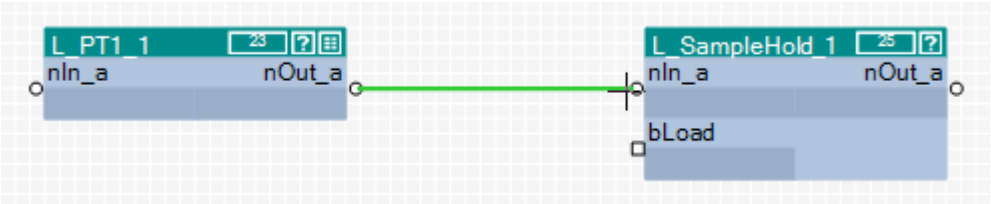
**How to create a connection using the connection line:**

1. Click the port symbol from which the new connection is to be started.
   - It is only possible to start a new connection from an input if there is no connection already ending in this input.
   - If you then move the mouse pointer away from the port symbol, a new connection is "drawn" from this port symbol.
   - Via **<Esc>** you can cancel this action.
2. Click the port symbol where the connection is to end.
   - Thereupon the corresponding connection is routed automatically if the connection is permissible.

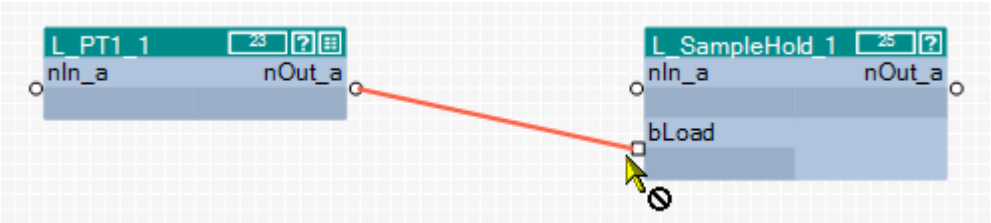
**Tip!**

If you move the mouse pointer across the port symbol while drawing a new connection, you can see whether the connection is permissible or not from the colour of the drawn line and from the mouse pointer symbol.

- Permissible connection:




- Impermissible connection (different port symbol):




The command **Show as flag** in the *context menu* of a line serves to change the representation of the connection at any time.

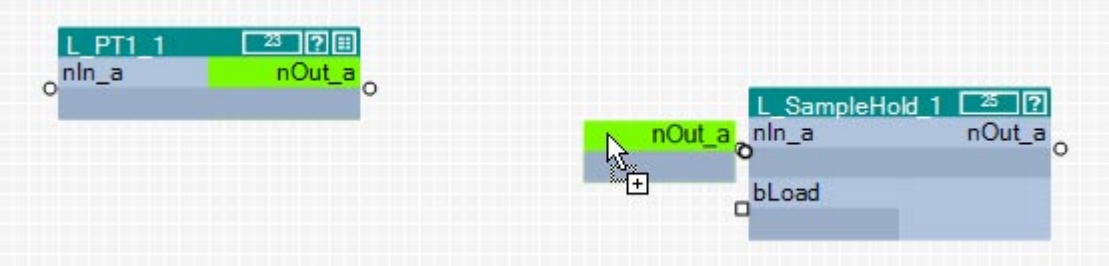
### 18.4.4.2 Creating a connection using port identifiers

**How to create a connection with port identifiers:**

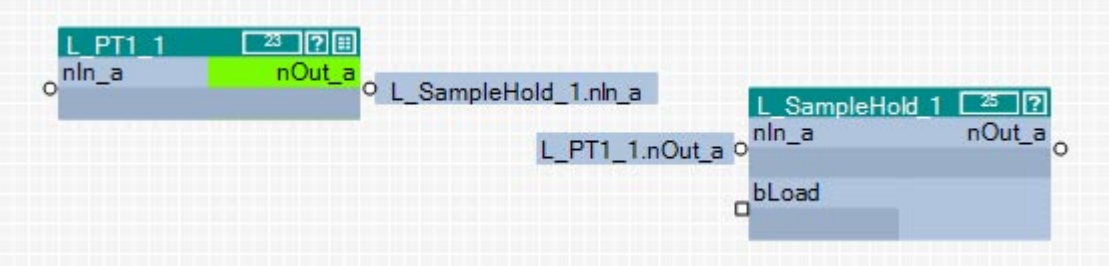
1. Click the port identifier.
   - The selected port is highlighted in light green:




2. Drag the port segment to the required port while keeping the left mouse button pressed:




After releasing the mouse button, the connection via port identifiers (flags) is created. The corresponding port identifier consists of the block name and the name of the input/output:




**Tip!**

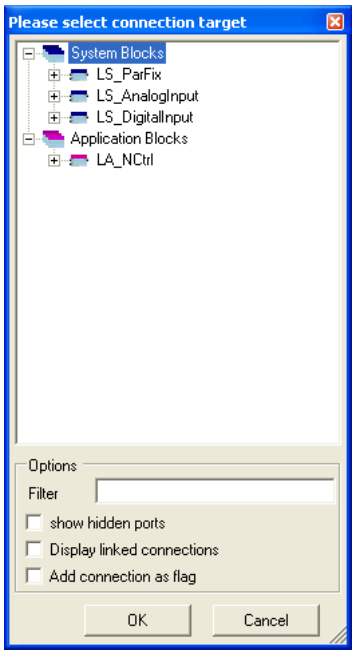
The command **Show as line** in the *context menu* of a flag serves to change the representation of the connection at any time.

### 18.4.4.3 Creating a connection via connection dialog

You can also create connections by means of a selection dialog instead of dragging by mouse. This especially makes sense if there is a great distance between the ports to be connected in the drawing area.

**How to create a connection using the selection dialog:**

1. Right-click the port identifier or click the port symbol from which the connection is to start.
   - The *context menu* for the port is displayed.
2. Go to the *context menu* for the port and select the **Add/change connection...** command.
   - The *Add/change connection* dialog box is displayed:




   - In a tree structure all inputs and outputs of the application are shown to which a connection is permissible.
   - You can enter an optional text into the **Filter** input field to reduce the selection to the blocks or ports which contain the entered text.
   - If you activate the **Show hidden ports** control field, the hidden ports for system and function blocks are shown as well.
3. Select the port where the connection is to end from the tree structure.
4. Activate the **Add connection as flag** control field if a port identifier (flag) is to be inserted instead of a connection line.
5. Press **OK** to create the connection to the selected port and close the dialog box.

### 18.4.4.4 Deleting connections that are no longer required



**How to delete connection lines:**

1. Select connection lines to be deleted.
   - Select a single connection line by directly clicking on the connection line with the right mouse button.
   - If you click further connection lines while pressing **<Ctrl>** they are added to an already existing selection (multi-selection).
   - All connection lines are highlighted in red.
2. Press **<Del>**.



**How to delete port identifiers/flags:**

1. Select the port identifiers to be deleted.
   - Select a single port identifier by directly clicking on the port identifier with the left mouse button.
   - If you click further port identifiers while pressing **<Ctrl>** they are added to an already existing selection (multi-selection).
   - All selected port identifiers are highlighted by a light green header.
2. Press **<Del>**.

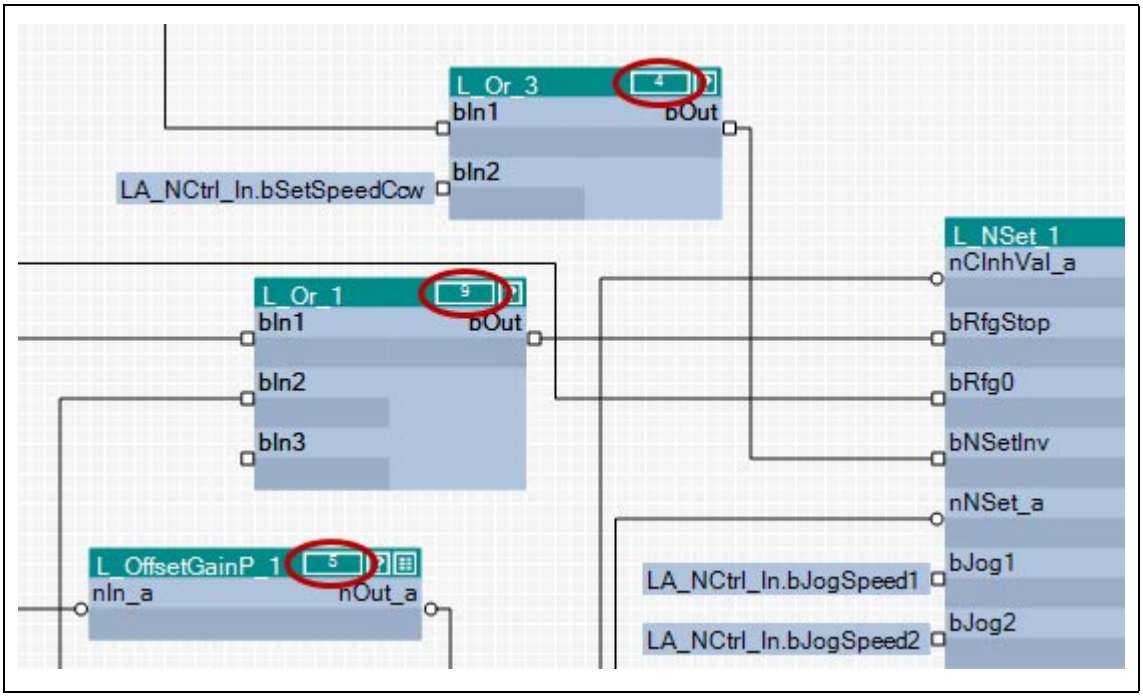
**Related topics**

▸ Deleting objects that are no longer required (📖 1403)

### 18.4.5 Changing the processing order

If you insert a function block into the interconnection, an order index is automatically assigned to this function block. By means of this order index it is defined in which order the individual function blocks are calculated at runtime.

- The first function block inserted contains the order index "1", the next function block inserted contains the order index "2", etc.
- The respective order index is displayed in the header of the function block in the rectangle after the block name.




[18-10] Example: Function blocks with order index

**Note!**

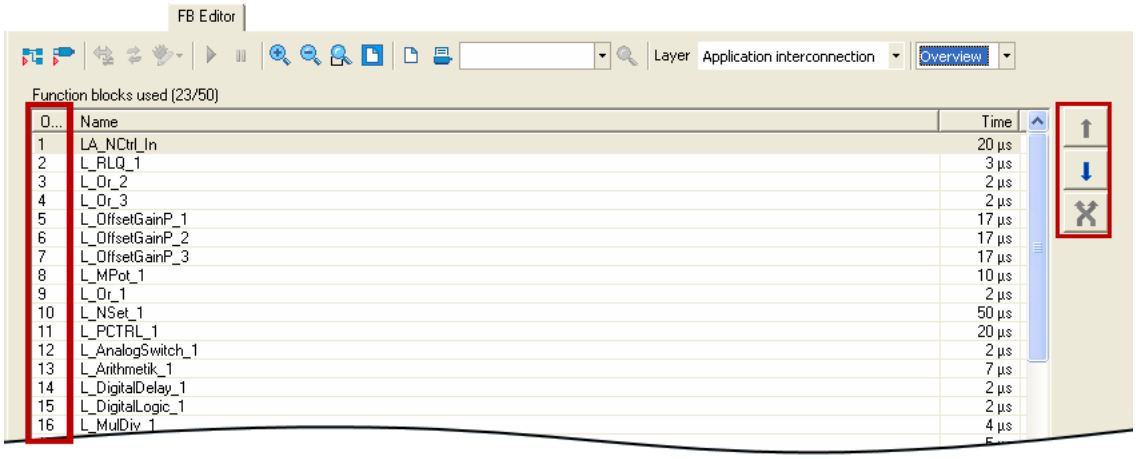
When a function block is shifted, its order index is maintained.

The processing order influences the result!

- In certain cases it may be sensible to change the processing order, but if you select an unfavourable processing order, errors may arise!

**How to change the processing order manually:**

1. Use the list field at the top right to change from the Editor to the overview.
   - The overview displays all function blocks of the interconnection in the order of their processing
   - In the first "Order" column the order index of each function block is listed.
2. Unless already selected, select the entry "Manual selection" in the **Optimisation...** list field.




3. Select the function block which is to receive a different position within the processing order.
   - If you click further function blocks while pressing **<Ctrl>** they are added to an already existing selection (multi-selection).
   - The **<Shift>** key serves to select a related area of function blocks.
4. Move the function block(s) to the desired position using the ↑ and ↓ buttons.
   - The ✕ button serves to exchange two selected function blocks with regard to their order.
5. Repeat steps 3 and 4 until the required processing order has been established.

**Changing the processing order according to an automatically generated selection**

In addition to the manual selection, the **Optimisation...** list field also offers two options for an automatic adaptation of the processing order:

- **Signal flow**: The processing order is optimised according to the signal flow.
- **Topology**: The processing order is optimised according to the x/y arrangement of the function blocks in the FB Editor.

As long as an automatic adaptation has been selected, a manual change of the processing order is not possible.

### 18.4.6 Copying interconnection elements (across all devices)

Interconnection elements can be copied across the devices within the project if the devices belong to the same product family (e.g. Inverter Drives 8400).

All types of blocks and comments can be copied to the clipboard via the **Copy** command or the **<Ctrl>+<c>** shortcut and then be inserted into the FB interconnection of the same or another project device of the same product family using the **Paste** command or the **<Ctrl>+<v>** shortcut.

- During the copy process into the clipboard, existing connections between copied blocks are copied as well, and the layout is kept too. Moreover, the separate technical objects (e.g port definition) are copied. Selected connections cannot be copied on their own.
- The **Paste** command is available if the clipboard is not empty and if it was copied from a device of the same product family. Within this product family, all device types (e.g. 8400 xxxxLine Vxx.xx) are permitted.
- After the **Paste** command has been selected, a dialog box is displayed which serves to select which elements are to be inserted from the clipboard and how to solve name conflicts, if any.
- After inserting the elements, they are marked in the target interconnection in order to be repositioned or deleted again to undo the insertion.
- Inserting from the clipboard can be repeated. The originally copied contents of the clipboard remains unchanged when it is inserted.



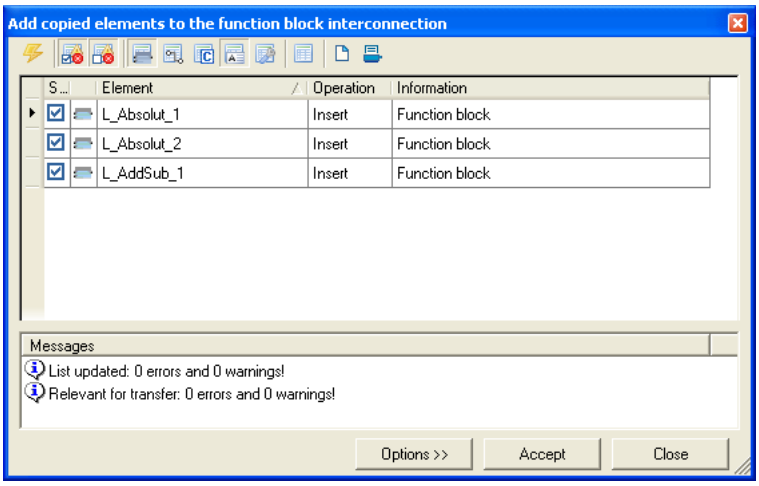
**How to copy one or several interconnection elements:**

1. Select the objects to be copied.
   - You can select a single object by clicking the header of the object.
   - If you click the header of further objects while pressing **<Ctrl>**, these will be added to an already existing selection (multi-selection).
   - You can easily select elements that are placed together by drawing a frame around these elements while keeping the mouse button pressed.
   - All selected objects are highlighted by a light green header.
2. Go to the *context menu* and select the **Copy** command (or **<Ctrl>+<c>**).
   - The selected elements are copied into the clipboard of the FB Editor.
3. If the elements are to be copied into a function block interconnection of another project device, change to the corresponding interconnection via the *project view*.
4. Go to the *context menu* and select the **Paste** command (or **<Ctrl>+<v>**).
5. Go to the *Insert FB interconnection* dialog box and select the elements to be inserted from the clipboard.
   - Detailed information on this dialog box can be obtained from the following subchapter "Insert options for copied elements". (📖 1416)

6. Click **Insert** to insert the selected elements into the target interconnection as defined.
   - Only possible if at least one element in the list has been selected for insertion.
   - Insertion is also possible via the **<Enter>** button if at least one element is selected from the list for insertion.
   - The original layout and the relative position of the inserted blocks to each other are maintained.
   - When copying across the devices, you also insert the corresponding separate technical objects (e.g. port definition).
   - The inserted elements are deleted from the list. If the list is empty, the dialog box is closed and the connections are inserted depending on the selected option.
7. If there are still elements to be entered in the list, repeat steps 5 and 6 until all elements are inserted as intended.
8. Press **Close** to stop the insertion and close the dialog box.
   - You can also use **<Esc>** or **<Enter>** to close the dialog box if "Insert" is not active.
   - The elements inserted into the target interconnection so far are maintained.
   - The connections for the blocks inserted so far are inserted depending on the selected option.

### 18.4.6.1 Insert options for copied elements

If interconnection elements have been copied to the clipboard, the »Engineer« will display a list of all elements contained in the clipboard when selecting the command **Insert** in the *Insert FB interconnection* dialog box:




The list shows the elements which can be added to the target interconnection, and the elements which cannot be added.

- In the "Selection" column, you can check/uncheck the elements to be added.
- Connections are only inserted when the dialog box is closed, which applies to all modules inserted so far. They are displayed as lines or flags, like in the original, but re-routed.
- The symbols in the *Toolbar* serve to execute the following functions:

| Symbol | Function |
|---|---|
|  | Add the selected elements to the interconnection |
|  | Show the elements to be added but are marked with an error or warning. |
|  | Show the elements not to be added and marked with an error or warning. |
|  | Show blocks |
|  | Show connections |
|  | Show parameters |
|  | Show comments |
|  | Show system elements |
|  | Show all |
|  | Print Preview |
|  | Print list |

- The buttons serve to execute the following functions:

| Button | Function |
|---|---|
| **Paste** | Add elements selected in the list to the target interconnection<br>• Only possible if at least one element in the list has been selected for insertion.<br>• Insertion is also possible via the **<Enter>** button if at least one element is selected from the list for insertion.<br>• The original layout and the relative position of the inserted blocks to each other are maintained.<br>• When copying across the devices, you also insert the corresponding separate technical objects (e.g. port definition).<br>• The added elements are simultaneously deleted from the list. The connections are added depending on the selected option. |
| **Close** | Close dialog box.<br>• You can also use **<Esc>** or **<Enter>** to close the dialog box if "Insert" is not active.<br>• The elements inserted into the target interconnection so far are maintained.<br>• The connections for the blocks inserted so far are inserted depending on the selected option. |

### 18.4.7 Resetting changed interconnection

If you only made changes on the I/O level, you can reset them by selecting a predefined control scheme in C00007. If you have also made changes on the application level, you must first reset the changed application to a predefined application in C00005.

**How to reset the application interconnection to a predefined application:**

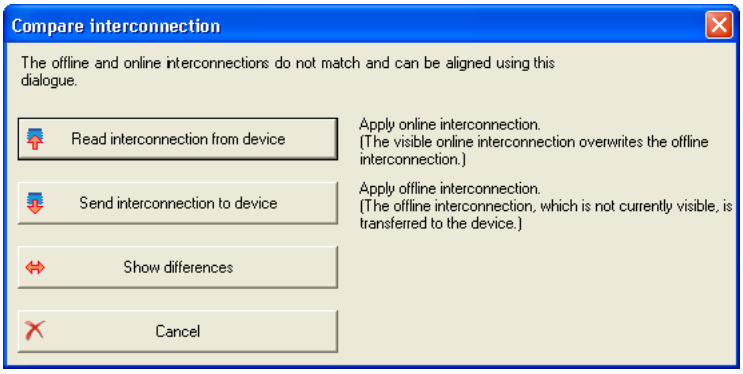
1. Go to the **Application parameters** tab.
2. Select the required application in the **Application** list field.

**How to reset the I/O interconnection to a predefined control scheme:**

1. Go to the **Application parameters** tab.
2. Select the required control scheme in the **Control source** list field.

## 18.5 Adjusting online and offline interconnection

If the »Engineer« detects that online and offline interconnection differ from each other, the *Compare interconnection* dialog box is displayed with various options for the adjustment:




**Tip!**

The dialog box can also be opened via the symbol in the *FB Editor toolbar*.

| Button | Function |
|---|---|
| **Read interconnection from device** | Add the interconnection in the device to the FB Editor. The interconnection existing in the FB Editor will be overwritten by this action. |
| **Send interconnection to device** | Transfer the offline interconnection which is currently not visible in the FB Editor to the device. The interconnection existing in the device will be overwritten by this action. |
| **Show differences** | Showing differences between online and offline interconnection. |
| **Cancel** | Close the *Adjust interconnection* dialog box without making an adjustment. |

## 18.6 Printing the interconnection

The interconnection can be printed for documentation purposes, optionally on one page, on four pages, or not scaled.

**Tip!**

By clicking the icon in the *FB Editor toolbar*, you can get a print view before printing.

**How to print the interconnection:**

1. In the *FB Editor toolbar*, click the icon.
   - The *Circuit print size* dialog box is displayed.
2. Select the desired size and press **OK**.
   - The standard dialog box *Print* appears.
3. Press **OK** to start the printing process.

## 18.7 Comparing interconnections

The comparison operation serves to compare FB interconnections of 8400 devices within the project. An offline<>online comparison and the comparison of two online devices are possible.

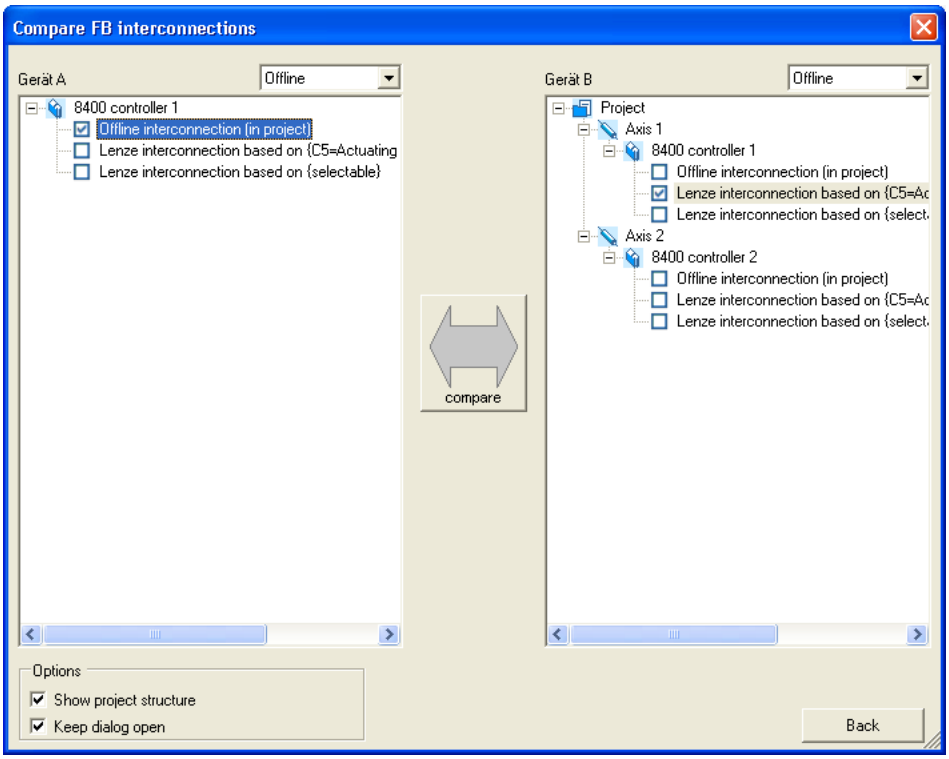
**Note!**

Only applications can be compared which have been enabled in the FB Editor!

Block positions, line representations, and connector visibilities are not compared.

**How to compare two FB interconnections:**

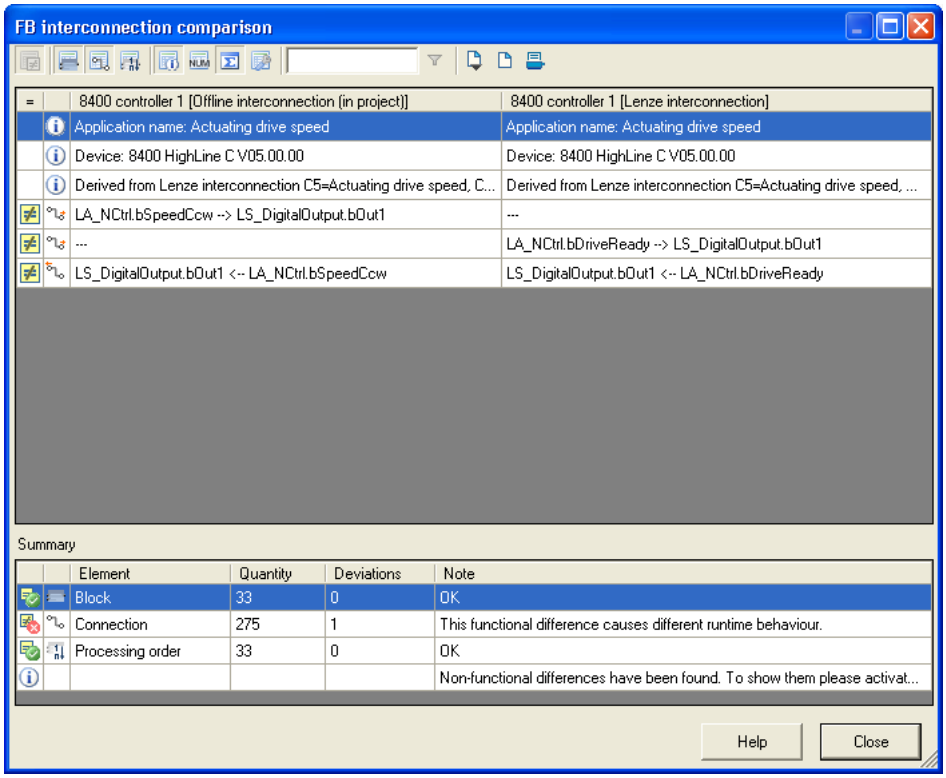
1. Select the command **Application data→Compare FB interconnections...**.
   - The *Compare FB interconnections* dialog box is displayed:




2. Select the interconnections to be compared in the project view represented on the left and right.
   - In order to execute a comparison with an online device, select "Online" in one of the two upper list fields. Then all available online devices are displayed for selection.
   - If you select "Online" in one of the two upper list fields, you can also compare the interconnections of two available online devices.
3. Click **Compare**.
   - If the comparison was executed successfully, the comparison result is displayed as a list (see the following section).
   - If a comparison of the selected interconnections is not possible, a corresponding message is displayed.
4. In order to stop the comparison operation and close the dialog box: Press **Back**.

### Representation of the comparison result

The comparison result is displayed in the form of a list in the *FB interconnection comparison* dialog box:




- The symbols in the *Toolbar* serve to show or hide different details and export and print the shown list.

| Symbol | Function |
|---|---|
|  | Only show differences<br>• Button can only be activated in expert mode. |
|  | Show blocks |
|  | Show connections |
|  | Show processing sequence |
|  | Show properties for blocks and connections<br>• Function is only available in expert mode. |
|  | Show comments<br>• Function is only available in expert mode. |
|  | Show block parameters<br>• Function is only available in expert mode. |
|  | Show application parameters<br>• Function is only available in expert mode. |
|  | Show all<br>• Function is only available in expert mode. |
|  | Show general information |
|  | Show summary |

| Symbol | Function |
|---|---|
|  | Activate expert mode<br>• In the expert mode, also non-functional differences are shown. |
|  | Use filter<br>• Only show list entries which contain the text entered in the input field.<br>• Function is only available in expert mode. |
|  | Export shown list as comma-separated list (*.csv) |
|  | Print Preview |
|  | Print list |

## 18.8 Copying an interconnection

In contrast to copying/inserting selected interconnection elements via the clipboard, the function described in this chapter serves to replace the current FB interconnection of a device completely by the FB interconnection of another project device.

**Note!**

The complete FB interconnection can only be copied between devices of the same device type and version (e.g. 8400 HighLine C V1.0).

A complete interconnection comprises:

- Function blocks (use and parameter values)
- System blocks (application and parameter values)
- Port blocks (use and parameter values)
- Connections
- Comments
- Interconnection layout (arrangement of the modules)
- Port definition of the ports used in the FB interconnection

**How to copy the complete interconnection into another project device:**

1. Select the application with the FB interconnection to be copied in the *project view*.
2. Select the command **Application data→Copy FB interconnections...**.
3. Go to *project view* and select the application which is to be inserted into the copied FB interconnection.
4. Select the command **Application data→Add FB interconnection...**.
   - The command can only be activated if an FB interconnection has been copied from a device of the same device type and version.
   - After the command has been executed, the module assembly is compared. If there are relevant deviations, the insertion is refused and a corresponding message is displayed.
   - If an insertion is possible, you are asked if the FB interconnection is to be inserted.
5. Confirm the question if the copied FB interconnection is to be inserted with **Yes**.
   - After the insertion, an update of the project is required.

## 18.9 Exporting/Importing an interconnection

The interconnection existing in the project can be exported to a file for reuse/transfer to other devices.

**Note!**

The file can only be imported to devices of the same device type and version (e.g. 8400 HighLine C V1.0).

**How to export the interconnection from the project to a file:**

1. Go to the *Project view* in the *context menu* of the inverter and select the **Export FB interconnection...** command.
2. Enter the memory location and the file name for the interconnection to be exported in the *Export FB interconnection* dialog box.
3. Click **Save** to export the interconnection and close the dialog box.

**How to import the interconnection from a file to the project:**

1. Go to the *Project view* in the *context menu* of the inverter and select the **Import FB interconnection...** command.
2. Select the file with the interconnection to be imported in the *Import FB interconnection* dialog box.
3. Click **Open** to import the interconnection and close the dialog box.

# 19 Function library

## 19.1 Function blocks

This chapter describes the function blocks which are available for the inverter in the FB Editor.



The system blocks are described in the chapter "System blocks". (📖 1742)



**Note!**

A maximum of 75 function blocks can be used for a function block interconnection. The maximum calculating time is approx. 300 µs.

### Overview of function blocks available

| Function block | Runtime | Function |
|---|---|---|
| L_Absolute_1<br>L_Absolute_2 | 2 µs | ... converts a bipolar input signal into a unipolar output signal. |
| L_AddSub_1 | 3 µs | ... adds / subtracts analog input signals. |
| L_AnalogSwitch_1<br>L_AnalogSwitch_2<br>L_AnalogSwitch_3<br>L_AnalogSwitch_4<br>L_AnalogSwitch_5 | 2 µs | ... switches between two analog input signals. |
| L_And_1<br>L_And_2<br>L_And_3 | 2 µs | ... ANDs three binary signals. |
| L_And5_1<br>L_And5_2 | 2 µs | ... ANDs five binary signals. |
| L_Arithmetik_1<br>L_Arithmetik_2<br>L_Arithmetik_3<br>L_Arithmetik_4<br>L_Arithmetik_5 | 7 µs | ... combines two analog signals arithmetically. |
| L_ArithmetikPhi_1<br>L_ArithmetikPhi_2<br>L_ArithmetikPhi_3<br>L_ArithmetikPhi_4<br>L_ArithmetikPhi_5<br>L_ArithmetikPhi_6 | 7 µs | ... combines two angle signals arithmetically.<br>• L_ArithmetikPhi_4 ... L_ArithmetikPhi_6 are available from version 12.00.00. |
| L_CalcDiameter_1 | 2 µs | ... can calculate the reel diameter from line speed and winding speed.<br>• This FB is available from version 02.00.00. |
| L_Compare_1<br>L_Compare_2<br>L_Compare_3<br>L_Compare_4<br>L_Compare_5 | 5 µs | ... compares two analog signals and can be used e.g. to implement a trigger. |

| Function block | Runtime | Function |
|---|---|---|
| L_ComparePhi_1<br>L_ComparePhi_2<br>L_ComparePhi_3<br>L_ComparePhi_4<br>L_ComparePhi_5 | 5 µs | ... compares two angle signals. |
| L_ConvActPos_1 | 0 µs | ... filters and scales a dancer position value to an available setting range.<br>• This FB is available from version 12.00.00. |
| L_ConvAP_1<br>L_ConvAP_2<br>L_ConvAP_3 | 0 µs | ... converts an analog value into a position.<br>• These FBs are available from version 02.00.00. |
| L_ConvBitsToWord_1<br>L_ConvBitsToWord_2<br>L_ConvBitsToWord_3 | 3 µs | ... converts 16 bit input values of the type "BOOL" into an output value of the type "WORD". |
| L_ConvDIntToWords_1<br>L_ConvDIntToWords_2<br>L_ConvDIntToWords_3 | 3 µs | ... converts an input value of the type "DINT" into two output values of the type "WORD". |
| L_ConvPA_1<br>L_ConvPA_2<br>L_ConvPA_3 | 3 µs | ... converts a position into an analog value.<br>• These FBs are available from version 02.00.00. |
| L_ConvPP_1<br>L_ConvPP_2<br>L_ConvPP_3 | 3 µs | ... converts a position with dynamic fraction.<br>• These FBs are available from version 02.00.00. |
| L_ConvUnitsToIncr_1<br>L_ConvUnitsToIncr_2<br>L_ConvUnitsToIncr_3 | 3 µs | ... converts a position value provided in the real unit of the machine into an internal 32-bit position value.<br>• These FBs are available from version 12.00.00. |
| L_ConvW_1<br>L_ConvW_2<br>L_ConvW_3<br>L_ConvW_4 | 2 µs | ... enables parameterisable conversion between analog signal forms.<br>• Division is remainder considered. |
| L_ConvWordsToDInt_1<br>L_ConvWordsToDInt_2<br>L_ConvWordsToDInt_3 | 3 µs | ... converts two inputs values of the type "WORD" into an output value of the type "DINT". |
| L_ConvWordToBits_1<br>L_ConvWordToBits_2<br>L_ConvWordToBits_3 | 3 µs | ... converts an input value of "WORD" type into 16 individual binary signals. |
| L_ConvX_1<br>L_ConvX_2<br>L_ConvX_3 | 0 µs | ... scales an analog value.<br>• These FBs are available from version 02.00.00. |
| L_Counter_1<br>L_Counter_2<br>L_Counter_3 | 3 µs | ... is a digital upcounter and downcounter. |
| L_Curve_1<br>L_Curve_2<br>L_Curve_3 | 4 µs | ... can optionally display a characteristic function or a curve function y=f(x), the X axis being the input signal and the Y axis being the output signal.<br>From version 12.00.00, the FB **L_Curve_3** for winding applications can also be used for creating a tensile force profile depending on the diameter.<br>• L_Curve_2 and L_Curve_3 are available from version 02.00.00. |
| L_DFlipFlop_1<br>L_DFlipFlop_2 | 3 µs | ... provides two stable states depending on the input signals. |
| L_DFRFG_1 | 3 µs | ... synchronises a drive (slave) to the master value of a master drive and then executes a angular synchronism with regard to the digital frequency.<br>• This FB is available from version 02.00.00. |

| Function block | Runtime | Function |
|---|---|---|
| L_DFSET_1 | 3 µs | ... prepares the master value for a slave drive. This FB enables the inverter to follow the master drive true to speed and angle.<br>• This FB is available from version 02.00.00. |
| L_DigitalDelay_1<br>L_DigitalDelay_2<br>L_DigitalDelay_3 | 2 µs | ... delays binary signals. |
| L_DigitalLogic_1<br>L_DigitalLogic_2<br>L_DigitalLogic_3 | 2 µs | ... provides a binary output signal which is generated by the logic combination of three input signals.<br>• L_DigitalLogic_3 is available from version 02.00.00. |
| L_DigitalLogic5_1<br>L_DigitalLogic5_2 | 2 µs | ... provides a binary output signal which is generated by the logic combination of five input signals. |
| L_DT1_1 | 3 µs | ...differentiates signals. The function block can, for instance, be used to apply an acceleration (dv/dt). |
| L_FixSet_a_1 | 2 µs | ... outputs one of 16 parameterisable analog signals. |
| L_FixSet_w_1<br>L_FixSet_w_2 | 2 µs | ... outputs one of 16 parameterisable data words. |
| L_GainOffset_1<br>L_GainOffset_2<br>L_GainOffset_3 | 3 µs | ... can amplify an analog input signal and then add an offset to it.<br>• Gain and offset can be set via FB inputs. |
| L_GainOffsetP_1<br>L_GainOffsetP_2<br>L_GainOffsetP_3 | 3 µs | ... can amplify an analog input signal and then add an offset to it.<br>• Gain and offset can be set via parameters. |
| L_GainOffsetPhiP_1<br>L_GainOffsetPhiP_2 | 3 µs | ... can amplify an angle signal and then add an offset to it.<br>• Gain and offset can be set via parameters. |
| L_GearComp_1 | 3 µs | ... compensates elasticities in the drive train (gearbox compensation).<br>• This FB is available from version 02.00.00. |
| L_Interpolator_1 | 5 µs | ... can interpolate a position setpoint and/or an analog value e.g. to compensate for larger bus transmission cycles or to continue signal characteristics if data telegrams are missing. |
| L_JogCtrlExtension_1 | 5 µs | ... can be connected upstream to the L_NSet ramp generator to implement a switch-off positioning at limit switch. |
| L_Limit_1<br>L_Limit_2 | 3 µs | ... limits an analog input signal to an adjustable value range. |
| L_LimitPhi_1<br>L_LimitPhi_2<br>L_LimitPhi_3 | 3 µs | ... limits an angle signal to an adjustable value range. |
| L_MckCtrlInterface_1 | 5 µs | ... provides the application with process inputs for controlling various basic functions of the Motion Control Kernel. ▸ MCKInterface (📖 581) |
| L_MckStateInterface_1 | 5 µs | ... provides the application with various items of status information of the Motion Control via process outputs. ▸ MCKInterface (📖 581) |
| L_MFail_1 | 0 µs | ... can be used for implementing a mains failure control.<br>• This FB is available from version 12.00.00. |
| L_MPot_1 | 10 µs | ... replaces a hardware motor potentiometer as setpoint source. |
| L_MulDiv_1<br>L_MulDiv_2 | 4 µs | ... multiplies the analog input signal with a factor.<br>• Not remainder considered. |
| L_Mux_1 | 3 µs | ... switches one of eight selectable input signals to the output. |
| L_Negation_1<br>L_Negation_2 | 2 µs | ... negates an analog input signal. |
| L_NLim_1<br>L_NLim_2 | 3 µs | ... can suppress up to three parameterisable blocking zones within a continuous signal characteristic of an analog input signal. |

| Function block | Runtime | Function |
|---|---|---|
| L_Not_1<br>L_Not_2<br>L_Not_3<br>L_Not_4<br>L_Not_5<br>L_Not_6<br>L_Not_7 | 2 µs | ... inverts a digital input signal. |
| L_NSet_1 | 50 µs | ... contains a ramp generator with comprehensive parameterisation and control options to condition a setpoint signal. |
| L_Odometer_1 | 2 µs | ... detects positions and calculates distances. |
| L_OffsetGain_1<br>L_OffsetGain_2<br>L_OffsetGain_3 | 4 µs | ... can add an offset to an analog input signal and amplify it afterwards.<br>• Offset and gain can be set via FB inputs. |
| L_OffsetGainP_1<br>L_OffsetGainP_2<br>L_OffsetGainP_3 | 17 µs | ... can add an offset to an analog input signal and amplify it afterwards.<br>• Offset and gain can be set via parameters. |
| L_OffsetGainPhiP_1<br>L_OffsetGainPhiP_2 | 17 µs | ... can add an offset to an angle signal and amplify it afterwards.<br>• Offset and gain can be set via parameters. |
| L_Or_1<br>L_Or_2<br>L_Or_3<br>L_Or_4 | 2 µs | ... ORs three binary signals.<br>• L_Or_4 is available from version 02.00.00. |
| L_Or5_1<br>L_Or5_2 | 2 µs | ... ORs five binary signals. |
| L_PCTRL_1 | 20 µs | ... is a PID controller and can be used for various control tasks. |
| L_PhaseDiff_1<br>L_PhaseDiff_2 | 2 µs | ... generates a position difference for the defined position setpoint from a position value and a speed signal. |
| L_PhaseIntK_1<br>L_PhaseIntK_2 | 5 µs | ... integrates a speed to an angle. |
| L_PhiAdd_1 | 2 µs | ... adds or subtracts angle signals.<br>• This FB is available from version 12.00.00. |
| L_PhiDiv_1 | 2 µs | ... divides or multiplies angle signals in the power of two format.<br>• This FB is available from version 12.00.00. |
| L_PhiIntegrator_1 | 2 µs | ... evaluates a speed with a gearbox factor and adds them in an integrator.<br>• This FB is available from version 02.00.00. |
| L_PosCtrlLin_1<br>L_PosCtrlLin_2 | 2 µs | ... is a simple linear profile generator.<br>• These FBs are available from version 02.00.00. |
| L_PosiShaftCtrlInterface_1 | 5 µs | FB in preparation! |
| L_ProcessCtrl_1 | 2 µs | ... can realise a dancer position or tension control.<br>• This FB is available from version 02.00.00. |
| L_PT1_1<br>L_PT1_2<br>L_PT1_3 | 3 µs | ... filters and delays analog signals. |
| L_RLQ_1 | 3 µs | ... links a selected direction of rotation to the QSP function with wire-break protection. |
| L_RSFlipFlop_1<br>L_RSFlipFlop_2 | 3 µs | ... saves a binary input information element and resets it on command.. |
| L_SampleHold_1<br>L_SampleHold_2 | 3 µs | ... saves a value. |
| L_Sequencer_1 | 10 µs | ... processes a positioning program based on a sequence table.<br>• This FB is available from version 12.00.00. |
| L_SignalMonitor_a | 15 µs | ... serves to output analog output signals of other FBs, SBs or LAs. |

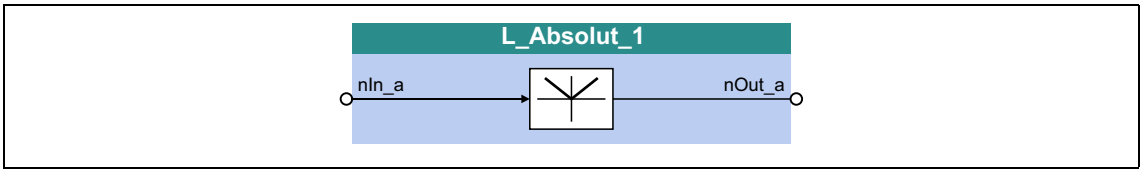
| Function block | Runtime | Function |
|---|---|---|
| L_SignalMonitor_b | 3 µs | ... serves to output binary output signals of other FBs, SBs or LAs. |
| L_SignalSwitch_1<br>L_SignalSwitch_2<br>L_SignalSwitch_3<br>L_SignalSwitch_4 | 2 µs | ... switches between two input signals of the "WORD" data type. |
| L_SignalSwitch32_1<br>L_SignalSwitch32_2<br>L_SignalSwitch32_3 | 2 µs | ... .switches between two input signals of "DINT" data type.<br>• These FBs are available from version 02.00.00. |
| L_SQrt_1 | 3 µs | ... outputs the square root for a DINT input value. |
| L_SRFG_1<br>L_SRFG_2 | 2 µs | ... is a ramp function generator with S-shaped ramps for limiting the temporal rise of analog signals. |
| L_SwitchPoint_1 | 3 µs | ... provides position switch points, i.e. digital switches the binary statuses (FALSE/TRUE) of which depend on the actual position.<br>• The position switch point positions can be set via FB inputs.<br>• This FB is available from version 02.00.00. |
| L_SwitchPointPar_1 | 3 µs | ... provides position switch points, i.e. digital switches the binary statuses (FALSE/TRUE) of which depend on the actual position.<br>• The position switch point positions can be set via parameters.<br>• This FB is available from version 12.00.00. |
| L_Transient_1<br>L_Transient_2<br>L_Transient_3<br>L_Transient_4<br>L_Transient_5<br>L_Transient_6<br>L_Transient_7<br>L_Transient_8 | 3 µs | ... evaluates digital signal edges and converts them into timed pulses. |

**Related topics:**

- Overview of system blocks available (📖 1742)
- Working with the FB Editor (📖 1373)

### 19.1.1 L_Absolut_1

This FB converts a bipolar input signal into a unipolar output signal.




#### Inputs

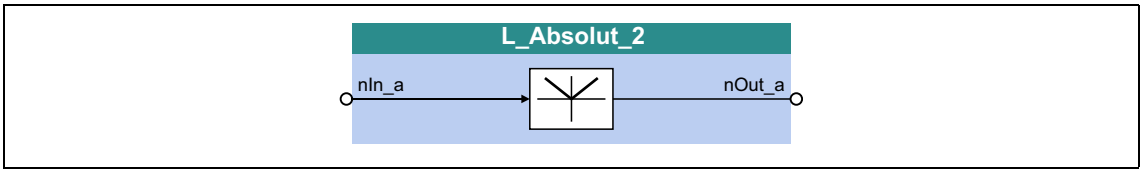
| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 19.1.2 L_Absolut_2

This FB converts a bipolar input signal into a unipolar output signal.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

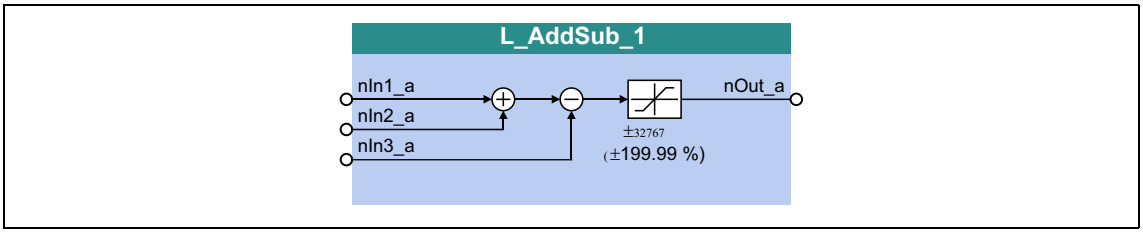
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 19.1.3 L_AddSub_1

This FB has two adding inputs and one subtracting input.

- The value provided at the *nOut_a* output is internally limited to ±32767.




#### Inputs

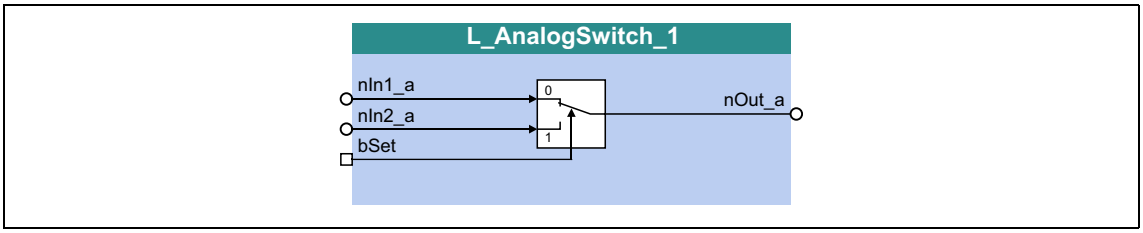
| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1<br>• This input is added |
| nIn2_a<br>INT | Input signal 2<br>• This input is added |
| nIn3_a<br>INT | Input signal 3<br>• This input is subtracted |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• *nOut_a = nIn1_a + nIn2_a - nIn3_a*<br>• Internal limitation to ±32767 (±199.99 %) |

### 19.1.4 L_AnalogSwitch_1

This FB changes over between two analog input signals. The change-over is controlled via a boolean input signal.




#### Inputs

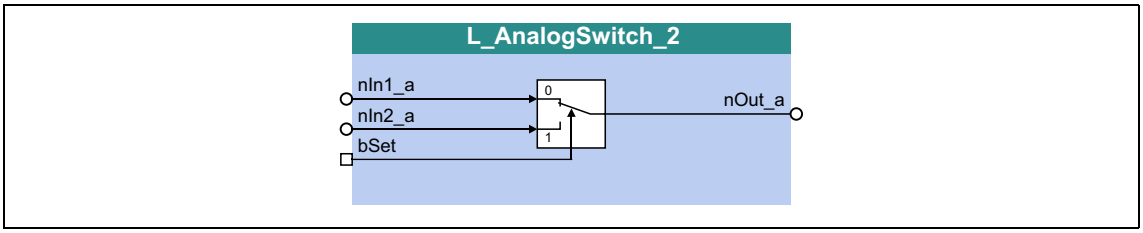
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn1_a<br>INT | Input signal 1 | |
| nIn2_a<br>INT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output to *nOut_a* | |
| | FALSE | *nIn1_a* |
| | TRUE | *nIn2_a* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 19.1.5 L_AnalogSwitch_2

This FB changes over between two analog input signals. The change-over is controlled via a boolean input signal.




#### Inputs

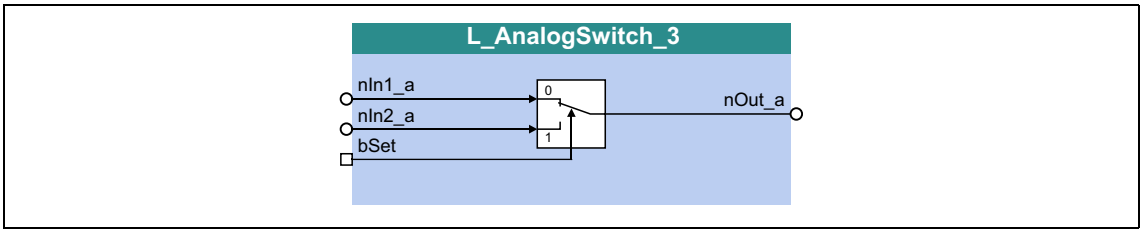
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn1_a<br>INT | Input signal 1 | |
| nIn2_a<br>INT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output to *nOut_a* | |
| | FALSE | *nIn1_a* |
| | TRUE | *nIn2_a* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 19.1.6 L_AnalogSwitch_3

This FB changes over between two analog input signals. The change-over is controlled via a boolean input signal.




#### Inputs

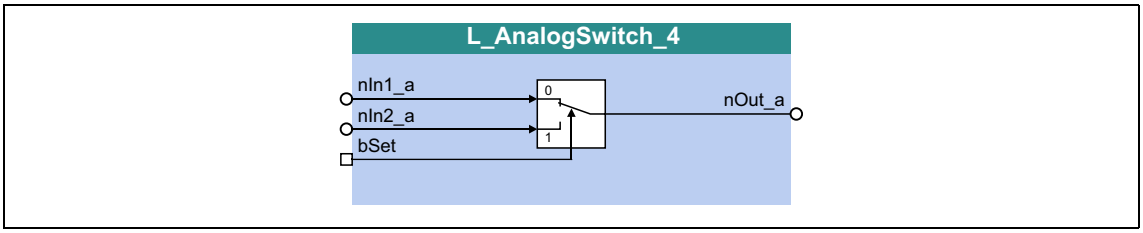
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn1_a<br>INT | Input signal 1 | |
| nIn2_a<br>INT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output to *nOut_a* | |
| | FALSE | *nIn1_a* |
| | TRUE | *nIn2_a* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 19.1.7 L_AnalogSwitch_4

This FB changes over between two analog input signals. The change-over is controlled via a boolean input signal.




#### Inputs

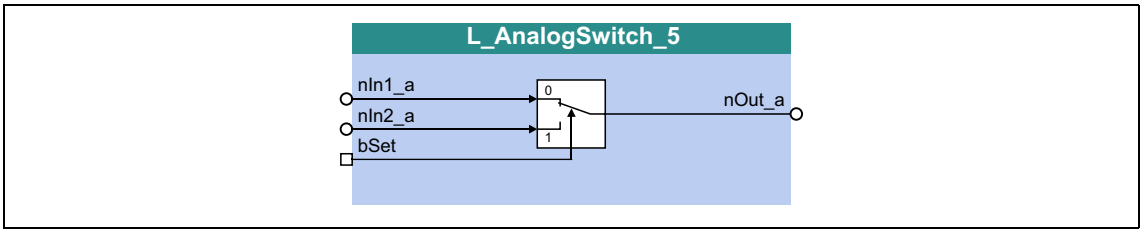
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn1_a<br>INT | Input signal 1 | |
| nIn2_a<br>INT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output to *nOut_a* | |
| | FALSE | *nIn1_a* |
| | TRUE | *nIn2_a* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 19.1.8 L_AnalogSwitch_5

This FB changes over between two analog input signals. The change-over is controlled via a boolean input signal.




#### Inputs

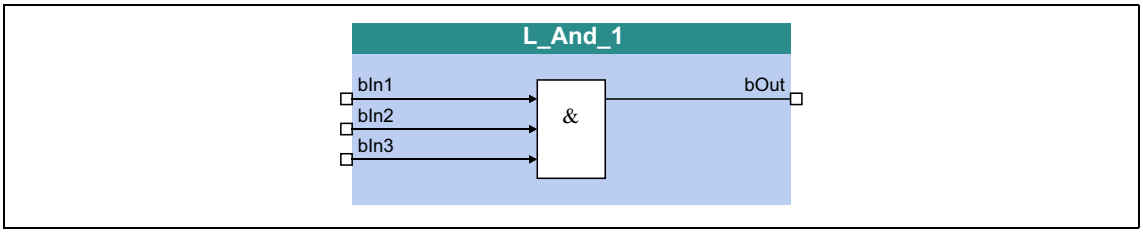
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn1_a<br>INT | Input signal 1 | |
| nIn2_a<br>INT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output to *nOut_a* | |
| | FALSE | *nIn1_a* |
| | TRUE | *nIn2_a* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 19.1.9 L_And_1

This FB implements AND operations for input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

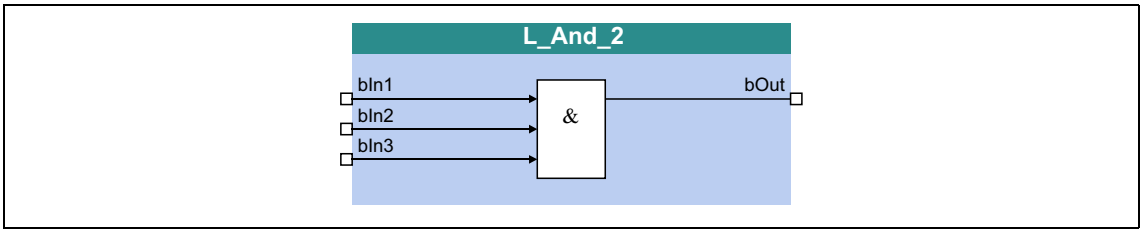
#### Function

| Inputs | | | Out |
|---|---|---|---|
| bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE |

[19-1] Truth table of the FB L_And_1

### 19.1.10 L_And_2

This FB implements AND operations for input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

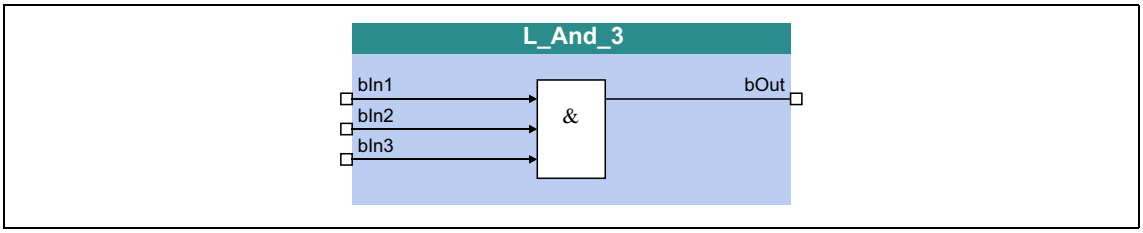
#### Function

| Inputs | | | Out |
|---|---|---|---|
| bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE |

[19-2] Truth table of the FB L_And_2

### 19.1.11 L_And_3

This FB implements AND operations for input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

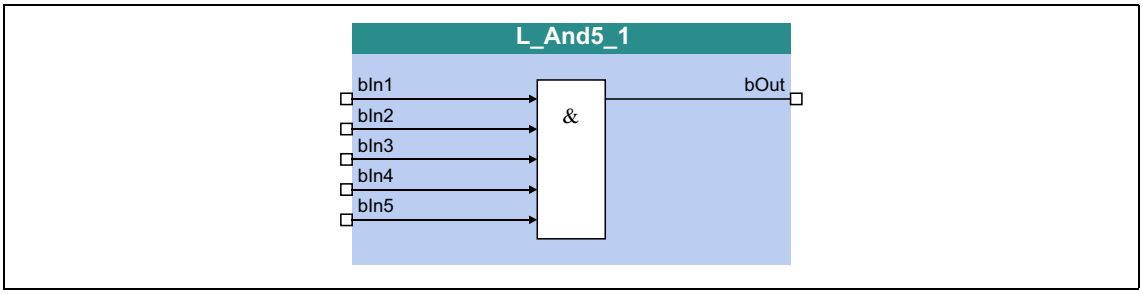
#### Function

| Inputs | | | Out |
|---|---|---|---|
| bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE |

[19-3] Truth table of the FB L_And_3

### 19.1.12 L_And5_1

This FB implements AND operations for input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

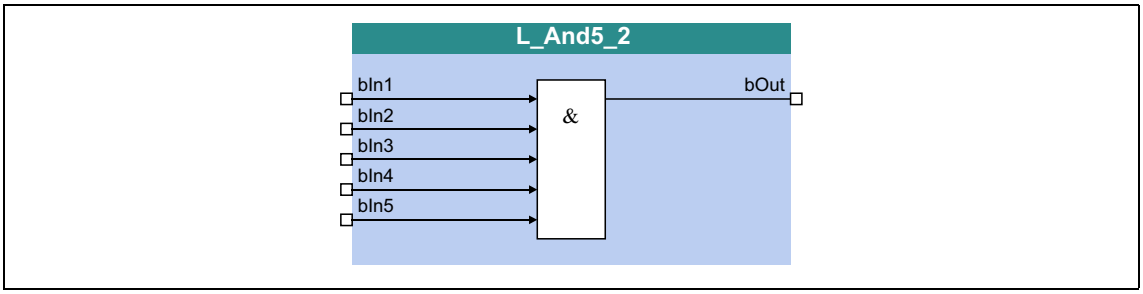
#### Function

| Inputs | | | | | Out |
|---|---|---|---|---|---|
| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | FALSE | FALSE | TRUE | |
| FALSE | FALSE | FALSE | TRUE | FALSE | |
| FALSE | FALSE | FALSE | TRUE | TRUE | |
| FALSE | FALSE | TRUE | FALSE | FALSE | |
| ... | | | | | |
| TRUE | TRUE | TRUE | FALSE | TRUE | |
| TRUE | TRUE | TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE | TRUE | TRUE |

[19-4] Truth table of the FB L_And5_1

### 19.1.13 L_And5_2

This FB implements AND operations for input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

#### Function

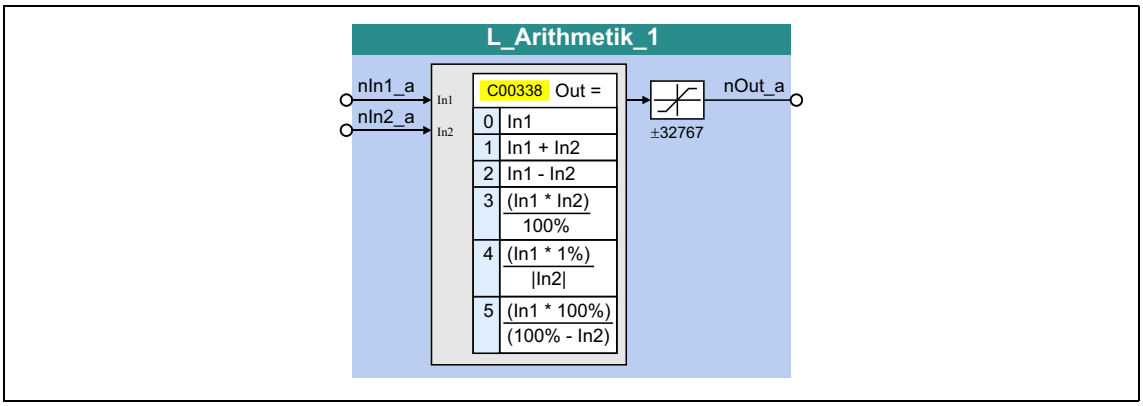
| Inputs | | | | | Out |
|---|---|---|---|---|---|
| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | FALSE | FALSE | TRUE | |
| FALSE | FALSE | FALSE | TRUE | FALSE | |
| FALSE | FALSE | FALSE | TRUE | TRUE | |
| FALSE | FALSE | TRUE | FALSE | FALSE | |
| ... | | | | | |
| TRUE | TRUE | TRUE | FALSE | TRUE | |
| TRUE | TRUE | TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE | TRUE | TRUE |

[19-5] Truth table of the FB L_And5_2

### 19.1.14 L_Arithmetik_1

This FB can implement an arithmetic link between two analog signals.

- The arithmetic function is selected in C00338.
- All internal intermediate results and the value output at the *nOut_a* output are internally limited to ±32767.
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %) |

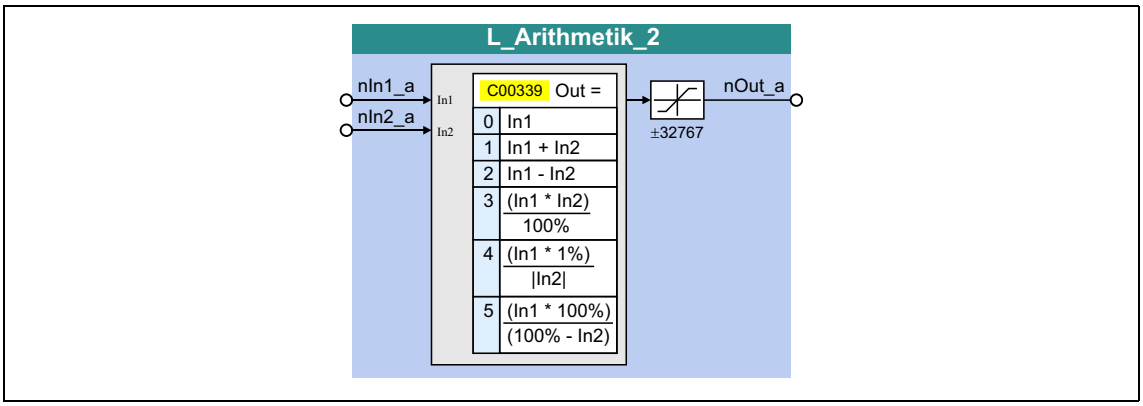
#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00338 | | | Function selection |
| | 0 | $nOut_a = nIn1_a$ | |
| | 1 | $nOut_a = nIn1_a + nIn2_a$ | |
| | 2 | $nOut_a = nIn1_a - nIn2_a$ | |
| | 3 | $nOut_a = \frac{nIn1_a \cdot nIn2_a}{16384}$ | |
| | 4 | $nOut_a = \frac{nIn1_a}{\lvert nIn2_a \rvert} \cdot 164$ | When the denominator has the value "0", it will be set to "1". |
| | 5 | $nOut_a = \frac{nIn1_a}{16384 - nIn2_a} \cdot 16384$ | |

### 19.1.15 L_Arithmetik_2

This FB can implement an arithmetic link between two analog signals.

- The arithmetic function is selected in C00339.
- All internal intermediate results and the value output at the *nOut_a* output are internally limited to ±32767.
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %) |

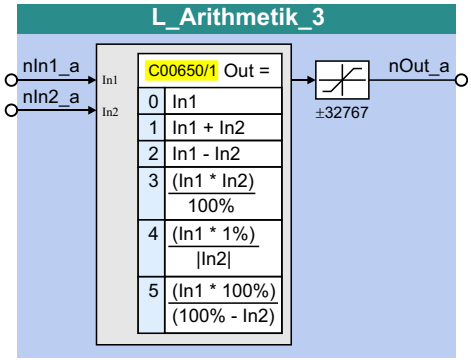
#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00339 | | | Function selection |
| | 0 | $nOut_a = nIn1_a$ | |
| | 1 | $nOut_a = nIn1_a + nIn2_a$ | |
| | 2 | $nOut_a = nIn1_a - nIn2_a$ | |
| | 3 | $nOut_a = \frac{nIn1_a \cdot nIn2_a}{16384}$ | |
| | 4 | $nOut_a = \frac{nIn1_a}{\lvert nIn2_a \rvert} \cdot 164$ | When the denominator has the value "0", it will be set to "1". |
| | 5 | $nOut_a = \frac{nIn1_a}{16384 - nIn2_a} \cdot 16384$ | |

### 19.1.16 L_Arithmetik_3

This FB can implement an arithmetic link between two analog signals.

- The arithmetic function is selected in C00650/1.
- All internal intermediate results and the value output at the *nOut_a* output are internally limited to ±32767.
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %) |

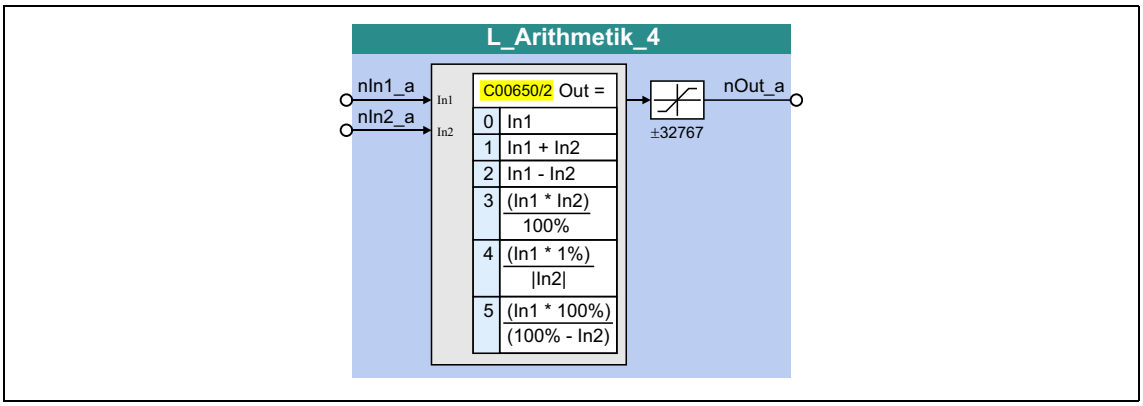
#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00650/1 | | | Function selection |
| | 0 | $nOut_a = nIn1_a$ | |
| | 1 | $nOut_a = nIn1_a + nIn2_a$ | |
| | 2 | $nOut_a = nIn1_a - nIn2_a$ | |
| | 3 | $nOut_a = \frac{nIn1_a \cdot nIn2_a}{16384}$ | |
| | 4 | $nOut_a = \frac{nIn1_a}{\lvert nIn2_a \rvert} \cdot 164$ | When the denominator has the value "0", it will be set to "1". |
| | 5 | $nOut_a = \frac{nIn1_a}{16384 - nIn2_a} \cdot 16384$ | |

### 19.1.17 L_Arithmetik_4

This FB can implement an arithmetic link between two analog signals.

- The arithmetic function is selected in C00650/2.
- All internal intermediate results and the value output at the *nOut_a* output are internally limited to ±32767.
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %) |

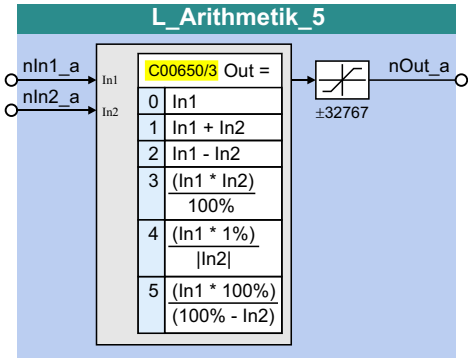
#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00650/2 | | | Function selection |
| | 0 | $nOut_a = nIn1_a$ | |
| | 1 | $nOut_a = nIn1_a + nIn2_a$ | |
| | 2 | $nOut_a = nIn1_a - nIn2_a$ | |
| | 3 | $nOut_a = \frac{nIn1_a \cdot nIn2_a}{16384}$ | |
| | 4 | $nOut_a = \frac{nIn1_a}{\lvert nIn2_a \rvert} \cdot 164$ | When the denominator has the value "0", it will be set to "1". |
| | 5 | $nOut_a = \frac{nIn1_a}{16384 - nIn2_a} \cdot 16384$ | |

### 19.1.18 L_Arithmetik_5

This FB can implement an arithmetic link between two analog signals.

- The arithmetic function is selected in C00650/3.
- All internal intermediate results and the value output at the *nOut_a* output are internally limited to ±32767.
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %) |

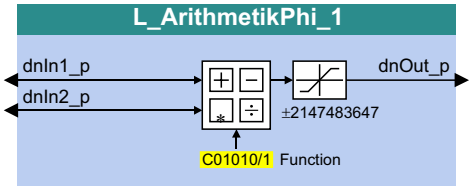
#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00650/3 | | | Function selection |
| | 0 | $nOut_a = nIn1_a$ | |
| | 1 | $nOut_a = nIn1_a + nIn2_a$ | |
| | 2 | $nOut_a = nIn1_a - nIn2_a$ | |
| | 3 | $nOut_a = \frac{nIn1_a \cdot nIn2_a}{16384}$ | |
| | 4 | $nOut_a = \frac{nIn1_a}{\lvert nIn2_a \rvert} \cdot 164$ | When the denominator has the value "0", it will be set to "1". |
| | 5 | $nOut_a = \frac{nIn1_a}{16384 - nIn2_a} \cdot 16384$ | |

### 19.1.19 L_ArithmetikPhi_1

This FB can implement an arithmetic link between two angle signals.

- The arithmetic function is selected in C01010/1.
- All internal intermediate results and the value provided at the *dnOut_p* output are internally limited to ±2147483647 ($\pm 2^{31} - 1$).
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |

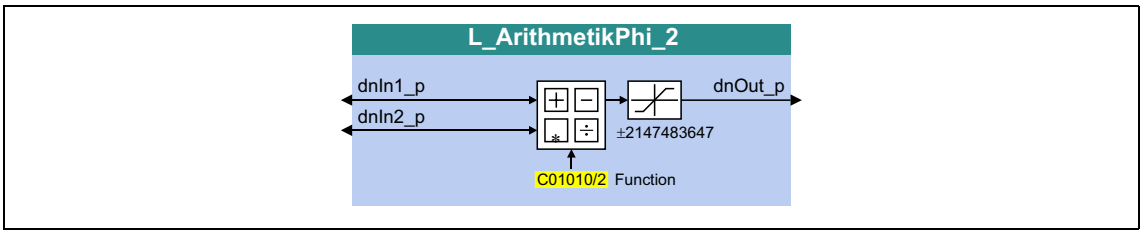
#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01010/1 | | | Function selection |
| | 0 | dnOut_p = dnIn1_p | |
| | 1 | dnOut_p = dnIn1_p + dnIn2_p | |
| | 2 | dnOut_p = dnIn1_p - dnIn2_p | |
| | 3 | dnOut_p = dnIn1_p * dnIn2_p | |
| | 4 | dnOut_p = dnIn1_p / dnIn2_p | |

### 19.1.20 L_ArithmetikPhi_2

This FB can implement an arithmetic link between two angle signals.

- The arithmetic function is selected in C01010/2.
- All internal intermediate results and the value provided at the *dnOut_p* output are internally limited to ±2147483647 ($\pm 2^{31}$ - 1).
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |

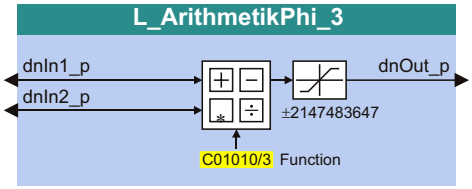
#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01010/2 | | | Function selection |
| | 0 | dnOut_p = dnIn1_p | |
| | 1 | dnOut_p = dnIn1_p + dnIn2_p | |
| | 2 | dnOut_p = dnIn1_p - dnIn2_p | |
| | 3 | dnOut_p = dnIn1_p * dnIn2_p | |
| | 4 | dnOut_p = dnIn1_p / dnIn2_p | |

### 19.1.21 L_ArithmetikPhi_3

This FB can implement an arithmetic link between two angle signals.

- The arithmetic function is selected in C01010/3.
- All internal intermediate results and the value provided at the *dnOut_p* output are internally limited to ±2147483647 (±$2^{31}$ - 1).
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |

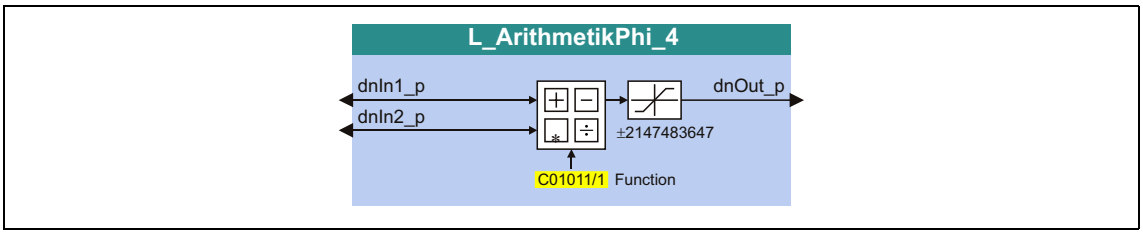
#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01010/3 | | | Function selection |
| | 0 | dnOut_p = dnIn1_p | |
| | 1 | dnOut_p = dnIn1_p + dnIn2_p | |
| | 2 | dnOut_p = dnIn1_p - dnIn2_p | |
| | 3 | dnOut_p = dnIn1_p * dnIn2_p | |
| | 4 | dnOut_p = dnIn1_p / dnIn2_p | |

### 19.1.22 L_ArithmetikPhi_4

This FB can implement an arithmetic link between two angle signals.

- The arithmetic function is selected in C01011/1.
- All internal intermediate results and the value provided at the *dnOut_p* output are internally limited to ±2147483647 ($\pm 2^{31}$ - 1).
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |

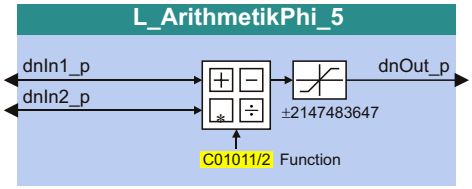
#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01011/1 | | | Function selection |
| | 0 | dnOut_p = dnIn1_p | |
| | 1 | dnOut_p = dnIn1_p + dnIn2_p | |
| | 2 | dnOut_p = dnIn1_p - dnIn2_p | |
| | 3 | dnOut_p = dnIn1_p * dnIn2_p | |
| | 4 | dnOut_p = dnIn1_p / dnIn2_p | |

### 19.1.23 L_ArithmetikPhi_5

This FB can implement an arithmetic link between two angle signals.

- The arithmetic function is selected in C01011/2.
- All internal intermediate results and the value provided at the *dnOut_p* output are internally limited to ±2147483647 ($\pm 2^{31}$ - 1).
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |

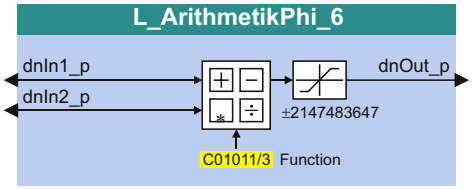
#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01011/2 | | | Function selection |
| | 0 | dnOut_p = dnIn1_p | |
| | 1 | dnOut_p = dnIn1_p + dnIn2_p | |
| | 2 | dnOut_p = dnIn1_p - dnIn2_p | |
| | 3 | dnOut_p = dnIn1_p * dnIn2_p | |
| | 4 | dnOut_p = dnIn1_p / dnIn2_p | |

### 19.1.24 L_ArithmetikPhi_6

This FB can implement an arithmetic link between two angle signals.

- The arithmetic function is selected in C01011/3.
- All internal intermediate results and the value provided at the *dnOut_p* output are internally limited to ±2147483647 ($\pm 2^{31}$ - 1).
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

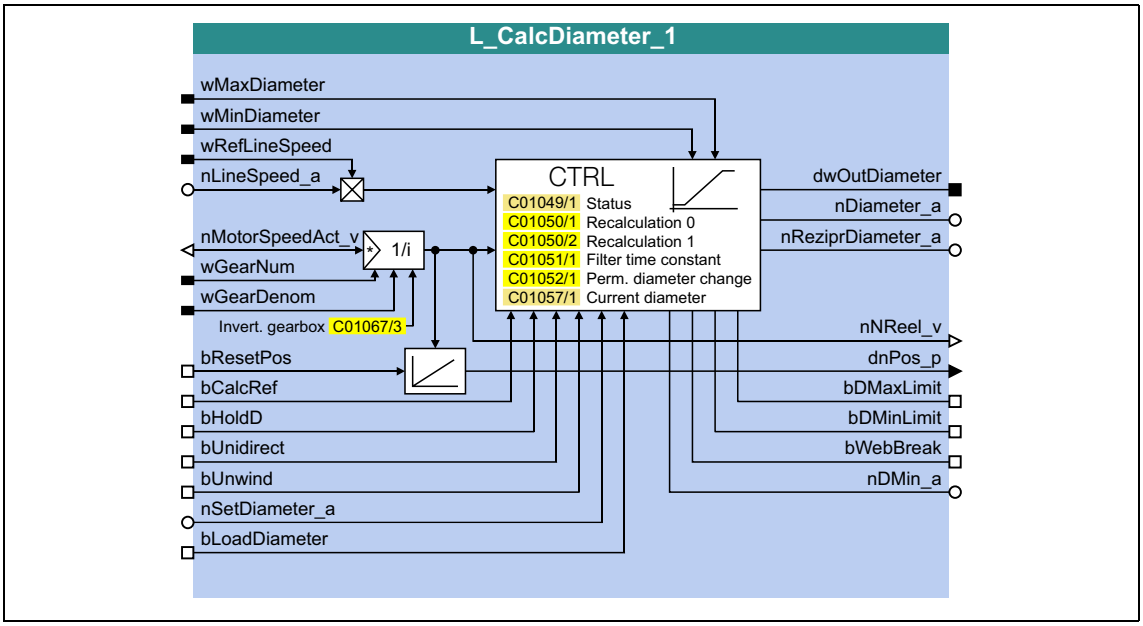
| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |

#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01011/3 | | | Function selection |
| | 0 | dnOut_p = dnIn1_p | |
| | 1 | dnOut_p = dnIn1_p + dnIn2_p | |
| | 2 | dnOut_p = dnIn1_p - dnIn2_p | |
| | 3 | dnOut_p = dnIn1_p * dnIn2_p | |
| | 4 | dnOut_p = dnIn1_p / dnIn2_p | |

## 19.1.25 L_CalcDiameter_1

This FB serves to calculate the reel diameter from line speed and winding speed.




### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wMaxDiameter<br>WORD | Maximum diameter in [mm]<br>• Internal limitation to 10000 mm (≡ 10 m) | |
| wMinDiameter<br>WORD | Minimum diameter in [mm]<br>• Internal limitation to 10000 mm (≡ 10 m) | |
| wRefLineSpeed<br>WORD | Maximum material speed in [0.1 m/min]<br>• Scaling: 2500 ≡ 250.0 m/min<br>• Internal limitation to 3000 m/min | |
| nLineSpeed_a<br>INT | Material speed setpoint<br>• Scaling: 16384 ≡ maximum material speed (*wRefLineSpeed*) | |
| nMotorSpeedAct_v<br>INT | Current motor speed in [increments/ms]<br>• Scaling: 16384 ≡ 15000 rpm | |
| wGearNum<br>WORD | Gearbox factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wGearDenom<br>WORD | Gearbox factor (denominator)<br>• Internal limitation to 1 ... 32767 | |
| bResetPos<br>BOOL | Reset angle of rotation *dnPos_p* | |
| | TRUE | The angle of rotation *dnPos_p* is reset. |
| bCalcRef<br>BOOL | Selection of the calculation cycle | |
| | FALSE | Use diameter recalculation 0 (C01050/1). |
| | TRUE | Use diameter recalculation 1 (C01050/2). |
| bHoldD<br>BOOL | Hold last diameter value | |
| | TRUE | The diameter value output last is not overwritten by the new values. |
| bUnidirect<br>BOOL | Enable of only one change of direction and activation of the web break monitoring | |
| | TRUE | Only the change of direction defined via *bUnwind* is enabled. |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bUnwind<br>BOOL | Selection of the enabled change or direction<br>• Only when *bUnidirect* = TRUE | |
| | FALSE | Clockwise rotation (Cw) |
| | TRUE | Counter-clockwise rotation (Ccw) |
| nSetDiameter_a<br>DINT | Selection of an initial value/external diameter signal<br>• Scaling: 16384 ≡ maximum diameter (*wMaxDiameter*)<br>• The value is accepted by setting *bLoadDiameter* to TRUE. | |
| bLoadDiameter<br>BOOL | Accept initial value *nSetDiameter_a*<br>• This input has a higher priority than the *bHold*Dinput. | |
| | TRUE | Accept the value at *nSetDiameter_a*. |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dwOutDiameter<br>DWORD | Current diameter in [µm]<br>• Internal limitation to 10 m<br>• Unfiltered |
| nDiameter_a<br>INT | Current diameter in [%]<br>• 100 % ≡ Maximum diameter (*wMaxDiameter*)<br>• filtered via PT1 element (filter time constant can be set in C01051/1) |
| nReziprDiameter_a<br>INT | Reciprocal value of the current diameter in [%]<br>• 100 % ≡ Minimum diameter (*wMinDiameter*) |
| nNReel_v<br>INT | Current winding speed in [increments/ms]<br>• Scaling: 16384 ≡ 15000 rpm |
| bDMaxLimit<br>BOOL | Limit value monitoring |
| | TRUE: Upper limit value *wMaxDiameter* reached. |
| bDMinLimit<br>BOOL | Limit value monitoring |
| | TRUE: Lower limit value *wMinDiameter* reached. |
| bWebBreak<br>BOOL | Web break monitoring |
| | TRUE: Web break after passing the monitor window set in C01052/1. |
| nDMin_a<br>INT | Minimum diameter in [%]<br>• 100 % ≡ Maximum diameter (*wMaxDiameter*) |

### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01049/1 | | | Status |
| | -10 | Line speed overflow | |
| | -1 | *wMinDiameter* > *wMaxDiameter* | |
| | 0 | OK - diameter has been recalculated | |
| | 10 | Diameter has been initialised with minimum diameter (*wMinDiameter)* | |
| | 20 | Diameter is loaded (*bLoadDiameter* active) | |
| | 30 | Diameter is held (*bHoldD* active) | |

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01050/1 | 0.001 | Rev. | 2.000 | Diameter recalculation 0<br>• Number of revolutions after which a diameter calculation is executed.<br>• Setting is effective when *bCalcRef* = FALSE.<br>• Initialisation: 1.000 |
| C01050/2 | 0.001 | Rev. | 2.000 | Diameter recalculation 1<br>• Number of revolutions after which a diameter calculation is executed.<br>• Setting is effective when *bCalcRef* = TRUE.<br>• Initialisation: 0.100 |
| C01051/1 | 0.010 | s | 3.000 | Filter time constant for calculated diameter values<br>• Initialisation: 1.000 s |
| C01052/1 | 0.00 | % | 100.00 | Permissible diameter change in opposite direction (web break monitoring)<br>• Only relevant when *bUnidirect* = TRUE.<br>• Referring to the maximum diameter (*wMaxDiameter*).<br>• Initialisation: 10.00 % |
| C01057/1 | 0.000 | mm | 10000.000 | Current diameter<br>• Read only |
| C01067/3 | | | | Invert. gearbox nMotorSpeedAct_v |
| | 0 | not inverted | | |
| | 1 | inverted | | |
| | 2 | Automatically from MCK | | |

### 19.1.25.1 Set initial value

An initial value or an external diameter signal can be defined at the *nSetDiameter_a* input.

- this value is accepted if *bLoadDiameter* is set to TRUE.
- In case of acceptance, the diameter calculation will be reset and the filter for diameter calculation is loaded with the defined initial value.

### 19.1.25.2 Calculate diameter

For reel diameter calculation, the material speed and the current winding speed are integrated cyclically.

- The material speed is defined via *nLineSpeed_a*.
- The winding speed is calculated from the *nMotorSpeedAct_v* motor speed and the current *wGearNum*/*wGearDenom* gearbox factor.
- C01050/1...2 can be used to define two different calculation cycles. The setting to the used is selected via *bCalcRef*.
- At the end of each integration interval, a new diameter value results from the division of the integrator values.
- An absolute value generation and smoothing via a first order low pass is carried out. The time constant of this filter can be set in C01051/1. The *dwOutDiameter* output is not filtered.

### 19.1.25.3 Select change direction, web break monitoring

By setting *bUnidirect* to TRUE, you only enable one change direction for diameter calculation and simultaneously activate the web break monitoring.

- The permissible change direction can be defined via *bUnwind*.
- A recalculated value is only accepted if it exceeds the value saved last in the permissible direction.

After a web break, the diameter values calculated successively mostly run very quickly in the opposite direction.

- C01052/1 serves to define the maximally permissible deviation in the opposite direction. If it is exceeded, the *bWebBreak* output is set to TRUE.
- If *bUnidirect* is set to FALSE, both change directions are enabled and the internal memory is always overwritten with the recalculated value which switches off the web break monitoring.

### 19.1.25.4 Holding the current value

By setting *bHoldD* to TRUE, you fix the diameter value *nDiameter_a* output last.

- The diameter calculation is reset and the filter for diameter calculation is loaded with the internally held diameter value.

### 19.1.25.5 Limit value monitoring

*wMinDiameter* and *wMaxDiameter* serve to define the limit values for the given diameter values. If they are reached/exceeded, the corresponding ouput (*bDMaxLimit*/*bDminLimit*) is set to TRUE.

- The unfiltered diameter value *dwOutDiameter* and the filtered diameter value *nDiameter_a* are limited.
- The limit values are entered in [mm].
- The hysteresis for resetting *bDMaxLimit*/*bDminLimit* is permanently set to 1 % of *wMaxDiameter*.
- For parameter setting of further FBs, the value in *wMinDiameter* with regard to *wMaxDiameter* is provided via the *nDMin_a* output.

### 19.1.25.6 Converting diameter in 1/D

For evaluating the material speed with 1/D (speed-controlled winder), the reciprocal value of the reel diameter is provided via *nReziprDiameter_a*.

- This value refers to the limitation value *wMinDiameter*.

### 19.1.25.7 Physical state variables

The *dwOutDiameter* output provides the current diameter as physical quantity (1 LSB ≡ 1 m).

*nNReel_v* outputs the winding speed which results from the *nMotorSpeedAct_v* motor speed and the current (*wGearNum*/*wGearDenom*) gearbox factor.

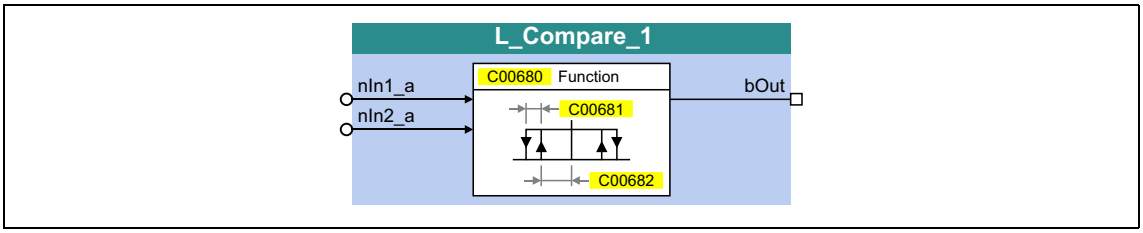
For supporting further functions (e.g. a traversing control), the winding speed is integrated additionally.

- The angle of rotation is output via *dnPos_p*.
- By setting *bResetPos* to TRUE, the angle of rotation is reset.

### 19.1.26 L_Compare_1

This FB compares two analog signals and can be used e.g. to implement a trigger.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

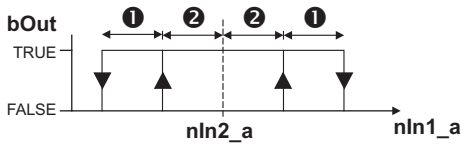
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00680 | | | | Function selection |
| | 1 | nIn1 = nIn2 | | |
| | 2 | nIn1 > nIn2 | | |
| | 3 | nIn1 < nIn2 | | |
| | 4 | \|nIn1\| = \|nIn2\| | | |
| | 5 | \|nIn1\| > \|nIn2\| | | |
| | 6 | \|nIn1\| < \|nIn2\| | | |
| C00681 | 0.00 | % | 100.00 | Hysteresis<br>• Lenze setting: 0.50 % |
| C00682 | 0.00 | % | 100.00 | Window<br>• Lenze setting: 2.00 % |

### 19.1.26.1 Function 1: nIn1 = nIn2

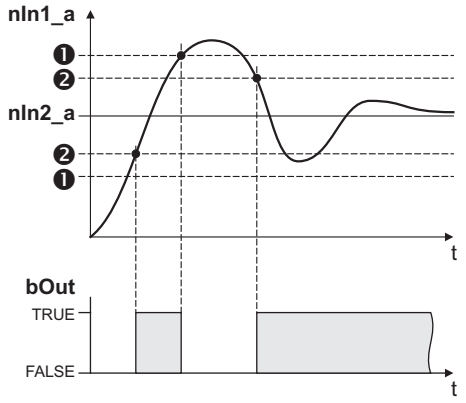
This function compares two signals with regard to equality. It can, for instance, provide the comparison "actual speed equals setpoint speed " ($n_{act} = n_{set}$).

- Use C00682 to set the window within which the equality is to apply.
- Use C00681 to set a hysteresis if the input signals are not stable and the output oscillates.




❶ C00681: Hysteresis
❷ C00682: Window

[19-6] Function 1: Switching performance

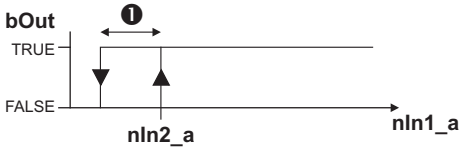



❶ C00681: Hysteresis
❷ C00682: Window

[19-7] Function 1: Example

### 19.1.26.2 Function 2: nIn1 > nIn2

This function serves, for instance, to implement the comparison "actual speed is higher than a limit value" ($n_{act} > n_x$) for one direction of rotation.

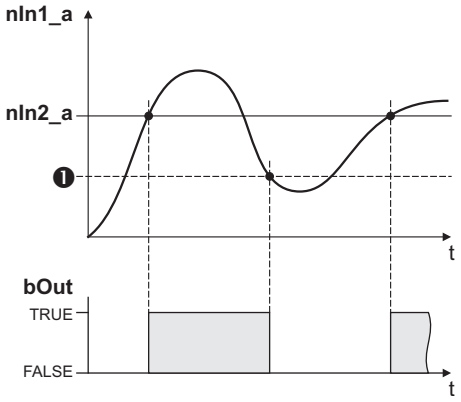



❶ C00681: Hysteresis

[19-8] Function 2: Switching performance

**Functional sequence**

1. If the value at *nIn1_a* exceeds the value *nIn2_a*, *bOut* changes from FALSE to TRUE.
2. Only if the signal at *nIn1_a* falls below the value of *nIn2_a - hysteresis* again, *bOut* changes back from TRUE to FALSE.

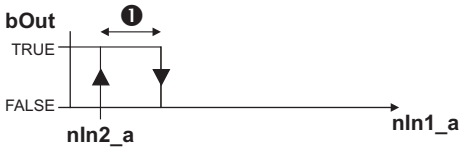



❶ C00681: Hysteresis

[19-9] Function 2: Example

### 19.1.26.3 Function 3: nIn1 < nIn2

This function serves, for instance, to implement the comparison "actual speed is lower than a limit value" ($n_{act} < n_x$) for one direction of rotation.

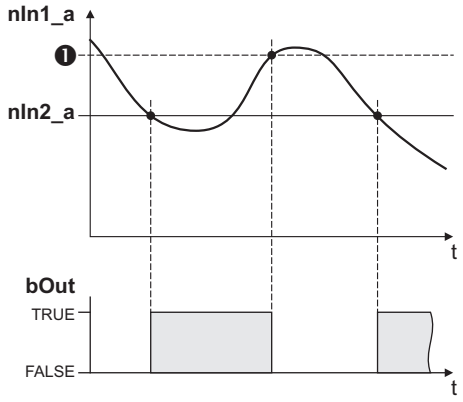



❶ C00681: Hysteresis

[19-10] Function 3: Switching performance

#### Functional sequence

1. If the value at *nIn1_a* falls below the value at *nIn2_a*, *bOut* changes from FALSE to TRUE.
2. Only if the signal at *nIn1_a* exceeds the value of *nIn2_a - hysteresis* again, *bOut* changes back from TRUE to FALSE.




❶ C00681: Hysteresis

[19-11] Function 3: Example

### 19.1.26.4 Function 4: |nIn1| = |nIn2|

This function serves to implement e.g. the comparison "$n_{act}$ = 0". This function is similar to function 1. However, the amount is generated by the input signals before signal processing (without sign).

▸ Function 1: nIn1 = nIn2

### 19.1.26.5 Function 5: |nIn1| > |nIn2|

This function serves to implement e.g. the comparison "$|n_{act}| > |n_x|$" irrespective of the direction of rotation. This function is similar to function 2. However, the amount is generated by the input signals before signal processing (without sign).

▸ Function 2: nIn1 > nIn2

### 19.1.26.6 Function 6: |nIn1| < |nIn2|

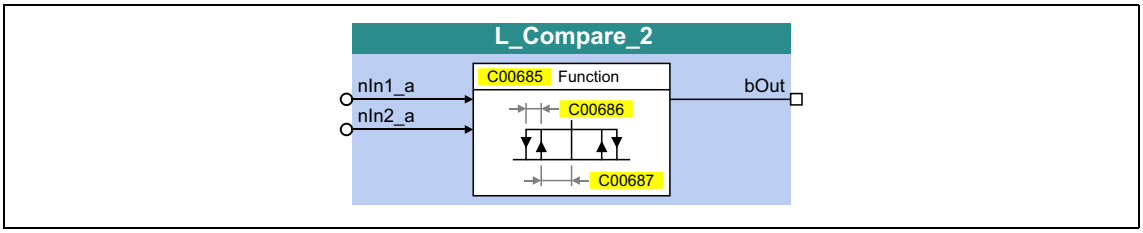
This function serves to implement the comparison "$|n_{act}| < |n_x|$" independent of the direction of rotation. This function is similar to function 3. However, the amount is generated by the input signals before signal processing (without sign).

▸ Function 3: nIn1 < nIn2

### 19.1.27 L_Compare_2

This FB compares two analog signals and can be used e.g. to implement a trigger.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

#### Parameters

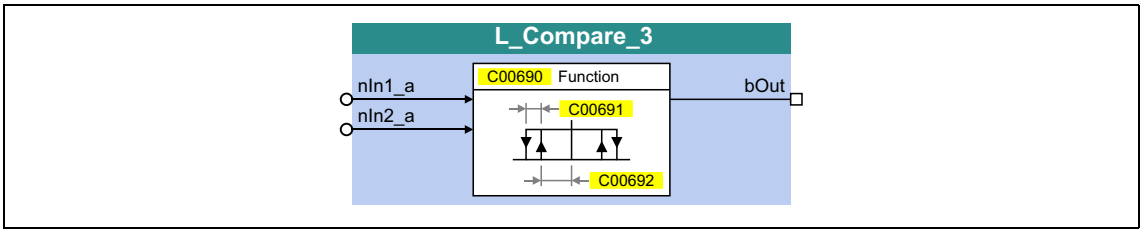
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00685 | | | | Function selection |
| | 1 | nIn1 = nIn2 | | |
| | 2 | nIn1 > nIn2 | | |
| | 3 | nIn1 < nIn2 | | |
| | 4 | \|nIn1\| = \|nIn2\| | | |
| | 5 | \|nIn1\| > \|nIn2\| | | |
| | 6 | \|nIn1\| < \|nIn2\| | | |
| C00686 | 0.00 | % | 100.00 | Hysteresis<br>• Lenze setting: 0.50 % |
| C00687 | 0.00 | % | 100.00 | Window<br>• Lenze setting: 2.00 % |

For a detailed functional description see L_Compare_1.

### 19.1.28 L_Compare_3

This FB compares two analog signals and can be used e.g. to implement a trigger.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

#### Parameters

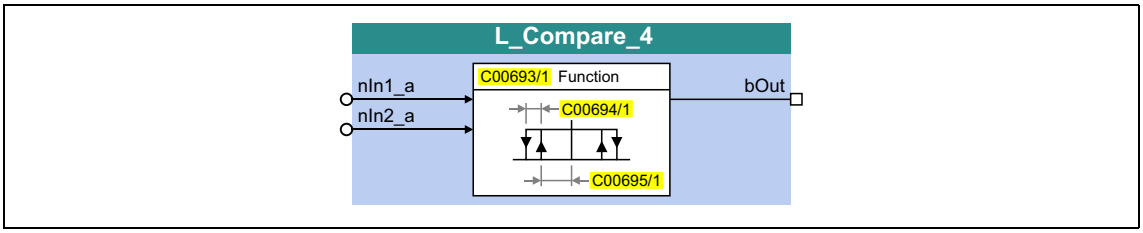
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00690 | | | | Function selection |
| | 1 | nIn1 = nIn2 | | |
| | 2 | nIn1 > nIn2 | | |
| | 3 | nIn1 < nIn2 | | |
| | 4 | \|nIn1\| = \|nIn2\| | | |
| | 5 | \|nIn1\| > \|nIn2\| | | |
| | 6 | \|nIn1\| < \|nIn2\| | | |
| C00691 | 0.00 | % | 100.00 | Hysteresis<br>• Lenze setting: 0.50 % |
| C00692 | 0.00 | % | 100.00 | Window<br>• Lenze setting: 2.00 % |

For a detailed functional description see L_Compare_1.

### 19.1.29 L_Compare_4

This FB compares two analog signals and can be used e.g. to implement a trigger.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

#### Parameters

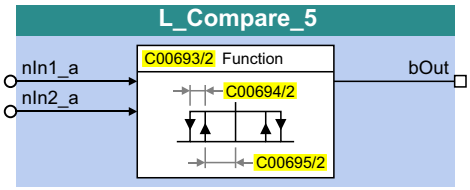
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00693/1 | | | | Function selection |
| | 1 | nIn1 = nIn2 | | |
| | 2 | nIn1 > nIn2 | | |
| | 3 | nIn1 < nIn2 | | |
| | 4 | \|nIn1\| = \|nIn2\| | | |
| | 5 | \|nIn1\| > \|nIn2\| | | |
| | 6 | \|nIn1\| < \|nIn2\| | | |
| C00694/1 | 0.00 | % | 100.00 | Hysteresis<br>• Lenze setting: 0.00 % |
| C00695/1 | 0.00 | % | 100.00 | Window<br>• Lenze setting: 0.00 % |

For a detailed functional description see L_Compare_1.

### 19.1.30 L_Compare_5

This FB compares two analog signals and can be used e.g. to implement a trigger.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

#### Parameters

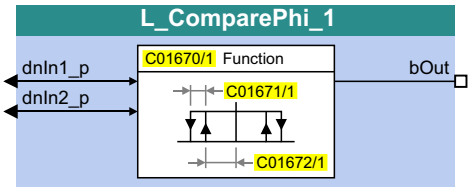
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00693/2 | | | | Function selection |
| | 1 | nIn1 = nIn2 | | |
| | 2 | nIn1 > nIn2 | | |
| | 3 | nIn1 < nIn2 | | |
| | 4 | \|nIn1\| = \|nIn2\| | | |
| | 5 | \|nIn1\| > \|nIn2\| | | |
| | 6 | \|nIn1\| < \|nIn2\| | | |
| C00694/2 | 0.00 | % | 100.00 | Hysteresis<br>• Lenze setting: 0.00 % |
| C00695/2 | 0.00 | % | 100.00 | Window<br>• Lenze setting: 0.00 % |

For a detailed functional description see L_Compare_1.

### 19.1.31 L_ComparePhi_1

This FB compares two angle signals.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

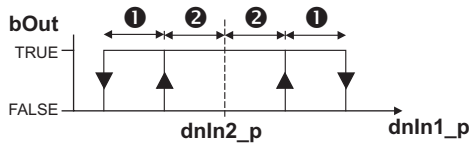
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01670/1 | | | | Function selection |
| | 1 | dnIn1 = dnIn2 | | |
| | 2 | dnIn1 > dnIn2 | | |
| | 3 | dnIn1 < dnIn2 | | |
| | 4 | \|dnIn1\| = \|dnIn2\| | | |
| | 5 | \|dnIn1\| > \|dnIn2\| | | |
| | 6 | \|dnIn1\| < \|dnIn2\| | | |
| C01671/1 | 0 | Incr. | 1073741824 | Hysteresis<br>• Lenze setting: 0 incr. |
| C01672/1 | 0 | Incr. | 1073741824 | Window<br>• Lenze setting: 0 incr. |

### 19.1.31.1 Function 1: dnIn1 = dnIn2

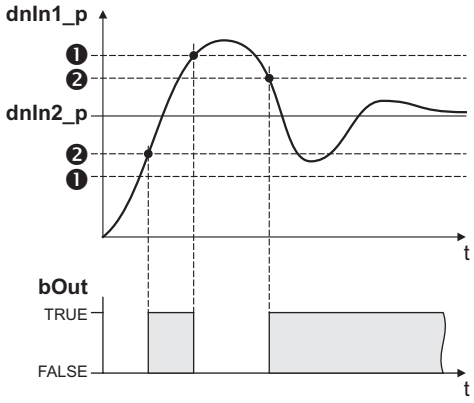
This function compares two signals with regard to equality. It can, for instance, provide the comparison "actual speed equals setpoint speed " ($n_{act} = n_{set}$).

- Use C01672 to set the window within which the equality is to apply.
- Use C01671 to set a hysteresis if the input signals are not stable and the output oscillates.




❶ C001671: Hysteresis
❷ C001672: Window

[19-12] Function 1: Switching performance

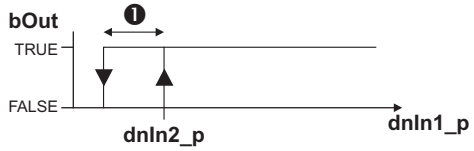



❶ C001671: Hysteresis
❷ C001672: Window

[19-13] Function 1: Example

### 19.1.31.2 Function 2: dnIn1 > dnIn2

This function serves, for instance, to implement the comparison "actual speed is higher than a limit value" ($n_{act} > n_x$) for one direction of rotation.

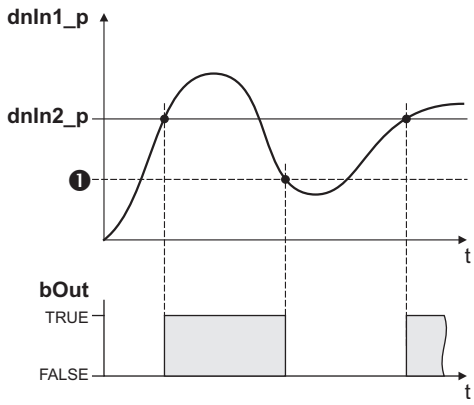



❶ C001671: Hysteresis

[19-14] Function 2: Switching performance

#### Functional sequence

1. If the value at *dnIn1_p* exceeds the value at *dnIn2_p*, *bOut* changes from FALSE to TRUE.
2. Only if the signal at *dnIn1_p* falls below the value of *dnIn2_p* - *hysteresis* again, *bOut* changes back from TRUE to FALSE.

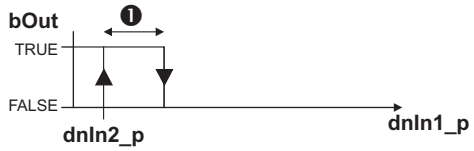



❶ C001671: Hysteresis

[19-15] Function 2: Example

### 19.1.31.3 Function 3: dnIn1 < dnIn2

This function serves, for instance, to implement the comparison "actual speed is lower than a limit value" ($n_{act} < n_x$) for one direction of rotation.

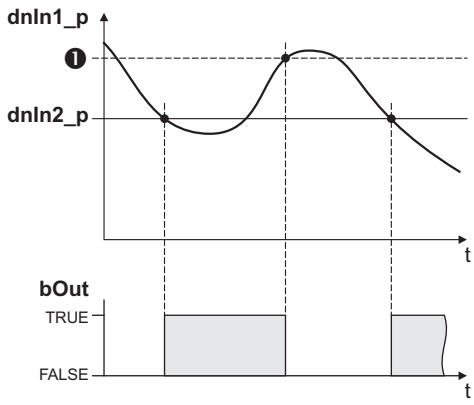



❶ C001671: Hysteresis

[19-16] Function 3: Switching performance

#### Functional sequence

1. If the value at *dnIn1_p* falls below the value at *dnIn2_p*, *bOut* changes from FALSE to TRUE.
2. Only if the signal at *dnIn1_p* exceeds the value of *dnIn2_p - hysteresis* again, *bOut* changes back from TRUE to FALSE.




❶ C001671: Hysteresis

[19-17] Function 3: Example

### 19.1.31.4 Function 4: |dnIn1| = |dnIn2|

This function serves to implement e.g. the comparison "$n_{act} = 0$". This function is similar to function 1. However, the amount is generated by the input signals before signal processing (without sign).

▸ Function 1: dnIn1 = dnIn2

### 19.1.31.5 Function 5: |dnIn1| > |dnIn2|

This function serves to implement e.g. the comparison "$|n_{act}| > |n_x|$" irrespective of the direction of rotation. This function is similar to function 2. However, the amount is generated by the input signals before signal processing (without sign).

▸ Function 2: dnIn1 > dnIn2

### 19.1.31.6 Function 6: |dnIn1| < |dnIn2|

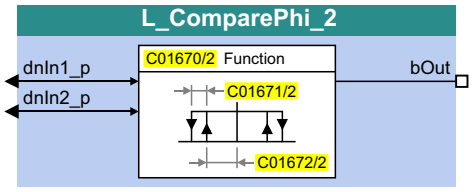
This function serves to implement the comparison "$|n_{act}| < |n_x|$" independent of the direction of rotation. This function is similar to function 3. However, the amount is generated by the input signals before signal processing (without sign).

▸ Function 3: dnIn1 < dnIn2

### 19.1.32 L_ComparePhi_2

This FB compares two angle signals.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

#### Parameters

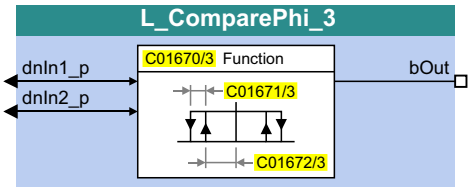
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01670/2 | | | | Function selection |
| | 1 | dnIn1 = dnIn2 | | |
| | 2 | dnIn1 > dnIn2 | | |
| | 3 | dnIn1 < dnIn2 | | |
| | 4 | \|dnIn1\| = \|dnIn2\| | | |
| | 5 | \|dnIn1\| > \|dnIn2\| | | |
| | 6 | \|dnIn1\| < \|dnIn2\| | | |
| C01671/2 | 0 | Incr. | 1073741824 | Hysteresis<br>• Lenze setting: 0 incr. |
| C01672/2 | 0 | Incr. | 1073741824 | Window<br>• Lenze setting: 0 incr. |

For a detailed functional description see L_ComparePhi_1.

### 19.1.33 L_ComparePhi_3

This FB compares two angle signals.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

#### Parameters

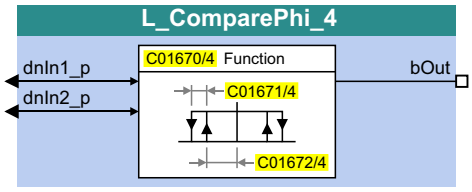
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01670/3 | | | | Function selection |
| | 1 | dnIn1 = dnIn2 | | |
| | 2 | dnIn1 > dnIn2 | | |
| | 3 | dnIn1 < dnIn2 | | |
| | 4 | \|dnIn1\| = \|dnIn2\| | | |
| | 5 | \|dnIn1\| > \|dnIn2\| | | |
| | 6 | \|dnIn1\| < \|dnIn2\| | | |
| C01671/3 | 0 | Incr. | 1073741824 | Hysteresis<br>• Lenze setting: 0 incr. |
| C01672/3 | 0 | Incr. | 1073741824 | Window<br>• Lenze setting: 0 incr. |

For a detailed functional description see L_ComparePhi_1.

### 19.1.34 L_ComparePhi_4

This FB compares two angle signals.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

#### Parameters

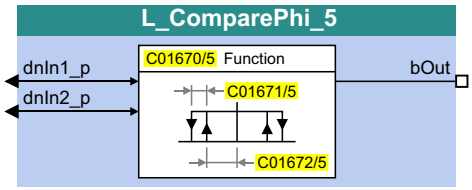
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01670/4 | | | | Function selection |
| | 1 | dnIn1 = dnIn2 | | |
| | 2 | dnIn1 > dnIn2 | | |
| | 3 | dnIn1 < dnIn2 | | |
| | 4 | \|dnIn1\| = \|dnIn2\| | | |
| | 5 | \|dnIn1\| > \|dnIn2\| | | |
| | 6 | \|dnIn1\| < \|dnIn2\| | | |
| C01671/4 | 0 | Incr. | 1073741824 | Hysteresis<br>• Lenze setting: 0 incr. |
| C01672/4 | 0 | Incr. | 1073741824 | Window<br>• Lenze setting: 0 incr. |

For a detailed functional description see L_ComparePhi_1.

### 19.1.35 L_ComparePhi_5

This FB compares two angle signals.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

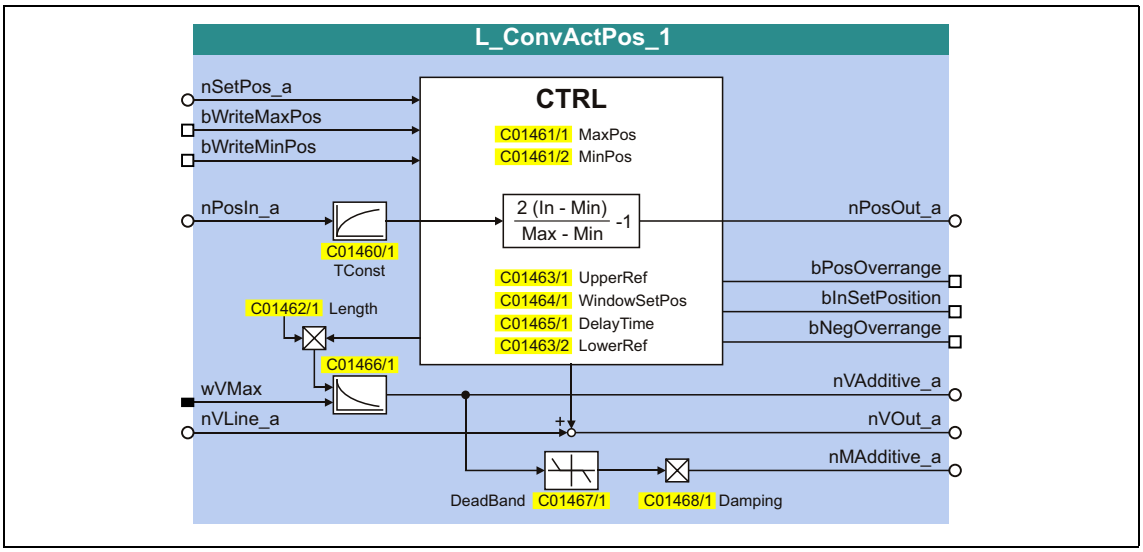
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01670/5 | | | | Function selection |
| | 1 | dnIn1 = dnIn2 | | |
| | 2 | dnIn1 > dnIn2 | | |
| | 3 | dnIn1 < dnIn2 | | |
| | 4 | \|dnIn1\| = \|dnIn2\| | | |
| | 5 | \|dnIn1\| > \|dnIn2\| | | |
| | 6 | \|dnIn1\| < \|dnIn2\| | | |
| C01671/5 | 0 | Incr. | 1073741824 | Hysteresis<br>• Lenze setting: 0 incr. |
| C01672/5 | 0 | Incr. | 1073741824 | Window<br>• Lenze setting: 0 incr. |

For a detailed functional description see L_ComparePhi_1.

### 19.1.36 L_ConvActPos_1

This FB serves to filter an actual dancer position value and scale it to the available setting range. The setting range is defined by parameterisable limit positions which can optionally be accepted automatically by a "teach-in" function.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nSetPos_a<br>INT | Current position setpoint<br>• Scaling: 16384 ≡ 100 % |
| bWriteMaxPos<br>BOOL | "Teach-In" function for the upper limit position |
| | TRUE: Actual position is saved as upper limit position in C01461/1. |
| bWriteMinPos<br>BOOL | "Teach-In" function for the lower limit position |
| | TRUE: Actual position is saved as lower limit position in C01461/2. |
| nPosIn_a<br>INT | Actual position<br>• Scaling: 16384 ≡ 100 % |
| wVMax<br>WORD | Maximum material speed in [0.1 m/min]<br>• Scaling: 2500 ≡ 250.0 m/min<br>• Internal limitation to 3000 m/min |
| nVLine_a<br>INT | Current material speed<br>• Scaling: 16384 ≡ Maximum material speed (*wVMax*) |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nPosOut_a<br>INT | Converted actual position value<br>• Scaling:<br>16384 ≡ 100 % ≡ upper limit position (C01461/1)<br>-16384 ≡ -100 % ≡ lower limit position (C01461/2) |
| bPosOverrange<br>BOOL | Monitoring of upper home position |
| | TRUE: Upper home position (C01463/1) reached. |
| bInSetPosition<br>BOOL | Monitoring of setpoint |
| | TRUE: Setpoint *nSetPos_a* reached. |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bNegOverrange<br>BOOL | Monitoring of lower home position | |
| | TRUE | Lower home position (C01463/2) reached. |
| nVAdditive_a<br>INT | Material speed caused by the dancer motion<br>• Scaling: 16384 ≡ Maximum material speed (*wVMax*) | |
| nVOut_a<br>INT | Current circumferential speed of the reel when the dancer hub is entered<br>• Scaling: 16384 ≡ Maximum material speed (*wVMax*) | |
| nMAdditive_a<br>INT | Additional torque for damping the dancer position control<br>• Scaling: 16384 ≡ 100 % | |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01460/1 | 0 | ms | 1000 | TConst<br>• Time constant for filtering the actual position value.<br>• Lenze setting: 10 ms |
| C01461/1 | | % | | Upper limit position<br>• Lenze setting: 100.00 % |
| C01461/2 | | % | | Lower limit position<br>• Lenze setting: -100.00 % |
| C01462/1 | | mm | | dwLength<br>• Storage volumen of the dancer (dancer hub).<br>• Lenze setting: 0 mm |
| C01463/1 | | % | | Upper home position<br>• Lenze setting: 90.00 % |
| C01463/2 | | % | | Lower home position<br>• Lenze setting: -90.00 % |
| C01464/1 | | % | | nWindowSetPos<br>• Tolerance zone for monitoring the dancer position.<br>• Lenze setting: 10.00 % |
| C01465/1 | | s | | wDelayTime<br>• Delay time for monitoring the dancer position.<br>• Lenze setting: 0.100 s |
| C01466/1 | | ms | | wTConstVAdd<br>• Time constant for filtering the additional speed.<br>• Lenze setting: 10 ms |
| C01467/1 | | % | | nDeadBand<br>• Dead band area for feedforward control torque (*nMAdditive_a*).<br>• Lenze setting: 1.00 % |
| C01468/1 | | | | wDamping<br>• Damping factor for feedforward control torque (*nMAdditive_a*).<br>• Lenze setting: 0.00 |

### 19.1.36.1 Determining limit positions

**Option 1: Parameterising the limit positions**

- Enter the actual value where the material stored in the dancer reaches its minimum length as upper limit position in C01461/1.
- Enterh the actual value where the material stored in the dancer reaches its maximum length as lower limit position in C01461/2.
- If the dancer detection provides an inverse signal, the lower limit position can be greater than the upper limit position.

**Option 2: Defining the limit positions using the "teach-in" function**

Alternatively to the direct setting of the limit values in the two parameters mentioned above, it is possible to bring the dancer into the respective limit position and then transfer the current value into the corresponding parameter by setting *bWriteMaxPos* bzw. *bWriteMinPos* to TRUE ("teach-in" function).

### 19.1.36.2 Monitoring of the dancer position

When the actual position value *nPosIn_a* reaches the current position setpoint *nSetPos_a*, the *bInSetPosition* output is set to TRUE.

- A tolerance zone can be set in C01464/1 for reaching the setpoint. when the actual dancer value leaves this tolerance zone, the *bInSetPosition* output will only be reset to FALSE after the delay time set in C01465/1 has elapsed (switch off delayed).
- In C01463/1 and C01463/2, reference values for the limit position monitoring can be set. If the actual dancer value reaches the set reference value towards the limit position, the corresponding (*bPosOverrange* output or *bNegOverrange*) output is set to TRUE (switch-off delayed) after the delay time set in C01465/1 has elapsed.

### 19.1.36.3 Circumferential speed of the reel

If the storage of the dancer is set in C01462/1, the additional speed of the material resulting from the motion of the dancer is provided at the *nVAdditive_a* output. When being added to the *nVLine_a* material speed, it results in the current circumferential speed of the reel via the *nVOut_a* output.

- The storage results from e.g. twice the distance between the two limit positions multiplied by the number of material wraps.
- The line speed is determined via *wVMax* and *nVLine_a*.
- In order to suppress the natural "noise" of the actual value signal with regard to the effect on the additional speed, the additional speed passes through a filter. The time constant of the filter can be set in C01466/1.

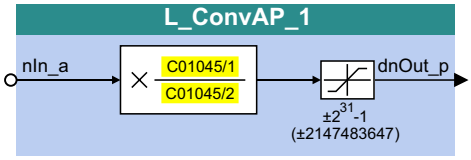
### 19.1.36.4 Additional torque for damping the dancer control

If the storage of the dancer is set in C01462/1, an additional feedforward control torque *nMAdditive_a* can be generated for damping the dancer control.

- The feedforward control torque is derived from the negated additional speed (double dancer speed) evaluated with the damping factor set in C01468/1.
- In order to suppress the natural "noise" of the actual value signal with regard to the effect on the additional torque, the additional speed passes through a dead band component in addition to the filter. The dead band area can be set in C01467/1.

### 19.1.37 L_ConvAP_1

This FB converts an analog value into a position.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value<br>• Scaling: 16384 ≡ 100 % |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±$2^{31}$-1 (±2147483647) |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01045/1 | -32767 | | 32767 | Counter<br>• Lenze setting: 1 |
| C01045/2 | -32767 | | 32767 | Denominator<br>• Lenze setting: 1 |

#### Function

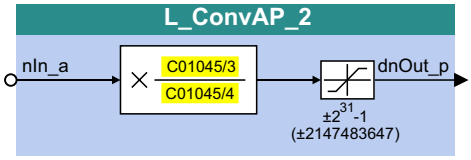
**Note!**

Conversion is remainder considered.

$$dnOut_p = nIn_a \cdot \frac{C01045/1}{C01045/2}$$

### 19.1.38 L_ConvAP_2

This FB converts an analog value into a position.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value<br>• Scaling: 16384 ≡ 100 % |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±$2^{31}$-1 (±2147483647) |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01045/3 | -32767 | | 32767 | Counter<br>• Lenze setting: 1 |
| C01045/4 | -32767 | | 32767 | Denominator<br>• Lenze setting: 1 |

#### Function

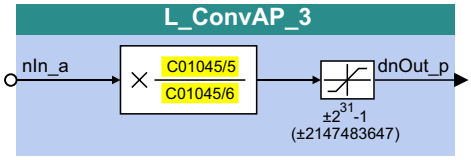
**Note!**

Conversion is remainder considered.

$$dnOut_p = nIn_a \cdot \frac{C01045/3}{C01045/4}$$

### 19.1.39 L_ConvAP_3

This FB converts an analog value into a position.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value<br>• Scaling: 16384 ≡ 100 % |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±$2^{31}$-1 (±2147483647) |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01045/5 | -32767 | | 32767 | Counter<br>• Lenze setting: 1 |
| C01045/6 | -32767 | | 32767 | Denominator<br>• Lenze setting: 1 |

#### Function

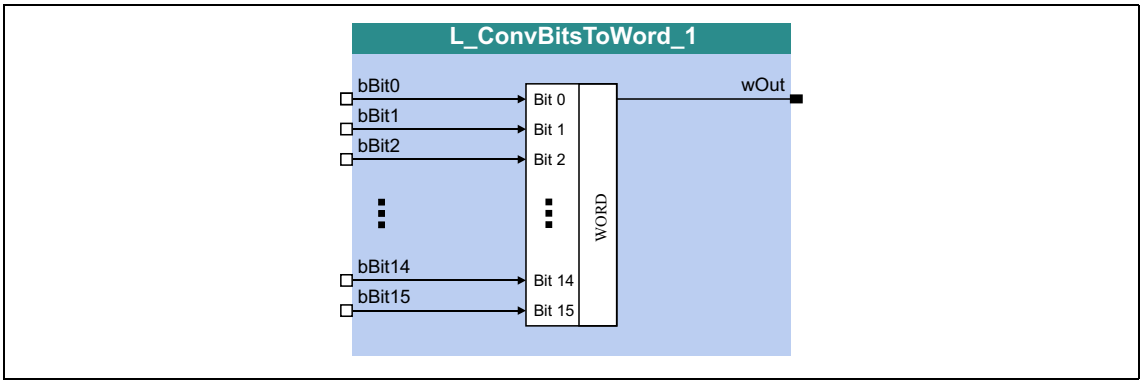
**Note!**

Conversion is remainder considered.

$$dnOut_p = nIn_a \cdot \frac{C01045/5}{C01045/6}$$

### 19.1.40 L_ConvBitsToWord_1

This FB converts 16 bit input values of the type "BOOL" into an output value of the type "WORD".




#### Inputs

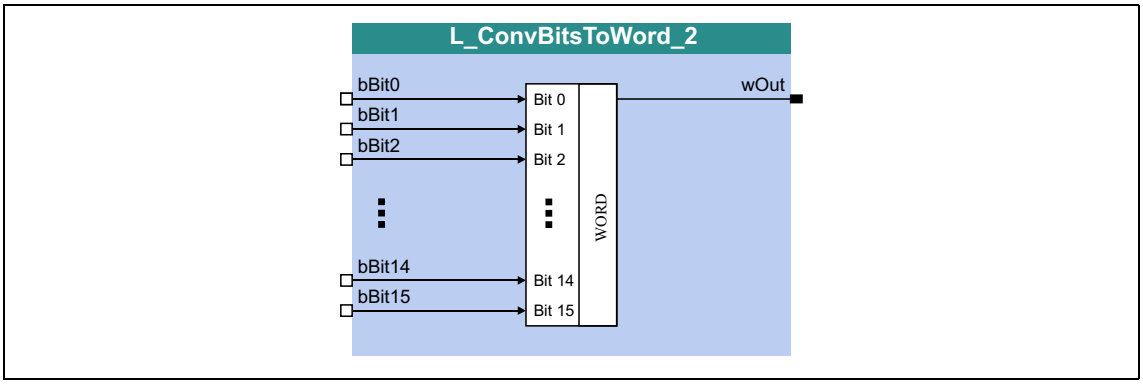
| Designator<br>Data type | Information/possible settings |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### 19.1.41 L_ConvBitsToWord_2

This FB converts 16 bit input values of the type "BOOL" into an output value of the type "WORD".




#### Inputs

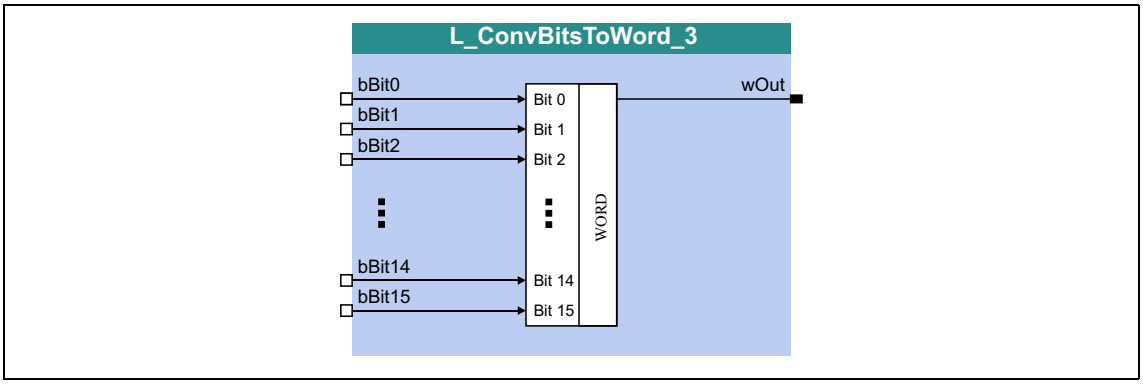
| Designator<br>Data type | Information/possible settings |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### 19.1.42 L_ConvBitsToWord_3

This FB converts 16 bit input values of the type "BOOL" into an output value of the type "WORD".




#### Inputs

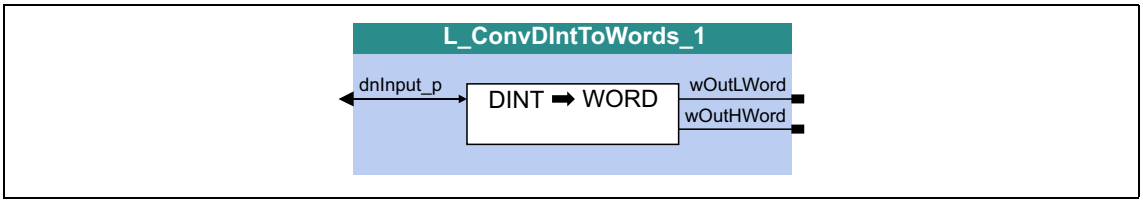
| Designator<br>Data type | Information/possible settings |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### 19.1.43 L_ConvDIntToWords_1

This FB converts an input value of the type "DINT" into two output values of the type "WORD".




#### Inputs

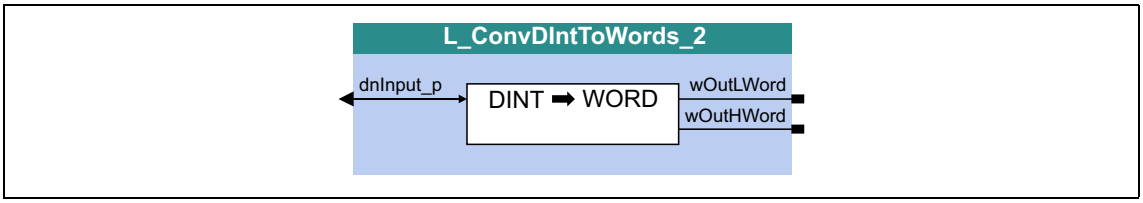
| Designator<br>Data type | Information/possible settings |
|---|---|
| dnInput_p<br>DINT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOutLWord<br>WORD | Output signal Low Word |
| wOutHWord<br>WORD | Output signal High Word |

### 19.1.44 L_ConvDIntToWords_2

This FB converts an input value of the type "DINT" into two output values of the type "WORD".




#### Inputs

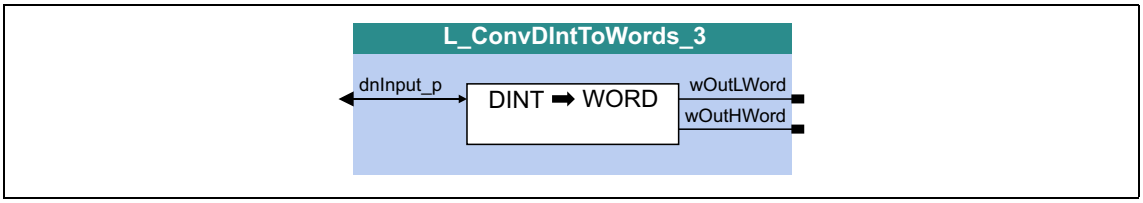
| Designator<br>Data type | Information/possible settings |
|---|---|
| dnInput_p<br>DINT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOutLWord<br>WORD | Output signal Low Word |
| wOutHWord<br>WORD | Output signal High Word |

### 19.1.45 L_ConvDIntToWords_3

This FB converts an input value of the type "DINT" into two output values of the type "WORD".




#### Inputs

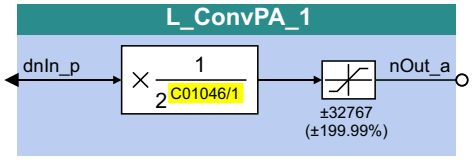
| Designator<br>Data type | Information/possible settings |
|---|---|
| dnInput_p<br>DINT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOutLWord<br>WORD | Output signal Low Word |
| wOutHWord<br>WORD | Output signal High Word |

### 19.1.46 L_ConvPA_1

This FB converts a position into an analog value.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01046/1 | 0 | | 31 | Division factor<br>• Lenze setting: 1 |

#### Function

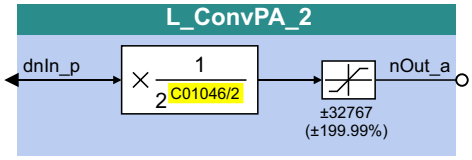
**Note!**

Conversion is remainder considered.

$$nOut_a = dnIn_p \cdot \frac{1}{2^{C01046/1}}$$

### 19.1.47 L_ConvPA_2

This FB converts a position into an analog value.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01046/2 | 0 | | 31 | Division factor<br>• Lenze setting: 1 |

#### Function

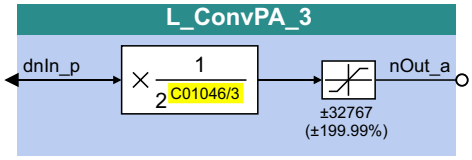
**Note!**

Conversion is remainder considered.

$$nOut_a = dnIn_p \cdot \frac{1}{2^{C01046/2}}$$

## 19.1.48 L_ConvPA_3

This FB converts a position into an analog value.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01046/3 | 0 | | 31 | Division factor<br>• Lenze setting: 1 |

### Function

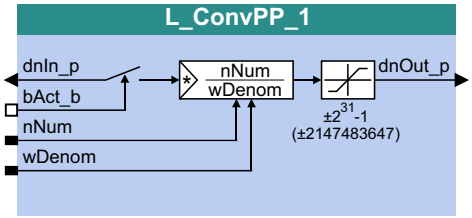
**Note!**

Conversion is remainder considered.

$$nOut_a = dnIn_p \cdot \frac{1}{2^{C01046/3}}$$

### 19.1.49 L_ConvPP_1

This FB converts a position with dynamic fraction.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn_p<br>DINT | Input signal | |
| bAct_b<br>BOOL | Conversion mode | |
| | FALSE | $dnOut_p = Remainder \cdot \frac{nNum}{wDenom}$ |
| | TRUE | $dnOut_p = dnIn_p \cdot \frac{nNum}{wDenom}$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

#### Outputs

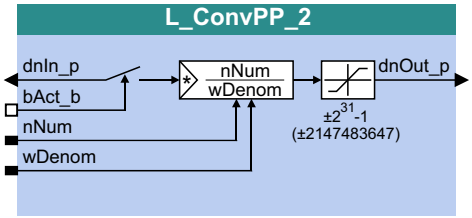
| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±$2^{31}$-1 (±2147483647) |

**Note!**

Conversion is remainder considered.

### 19.1.50 L_ConvPP_2

This FB converts a position with dynamic fraction.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn_p<br>DINT | Input signal | |
| bAct_b<br>BOOL | Conversion mode | |
| | FALSE | $dnOut_p = Remainder \cdot \frac{nNum}{wDenom}$ |
| | TRUE | $dnOut_p = dnIn_p \cdot \frac{nNum}{wDenom}$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

#### Outputs

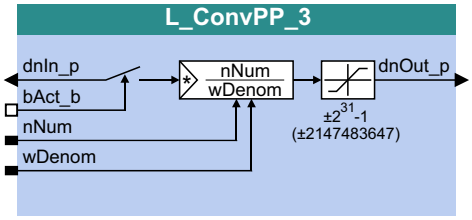
| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}-1$ (±2147483647) |

**Note!**

Conversion is remainder considered.

### 19.1.51 L_ConvPP_3

This FB converts a position with dynamic fraction.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn_p<br>DINT | Input signal | |
| bAct_b<br>BOOL | Conversion mode | |
| | FALSE | $dnOut_p = Remainder \cdot \frac{nNum}{wDenom}$ |
| | TRUE | $dnOut_p = dnIn_p \cdot \frac{nNum}{wDenom}$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±$2^{31}$-1 (±2147483647) |

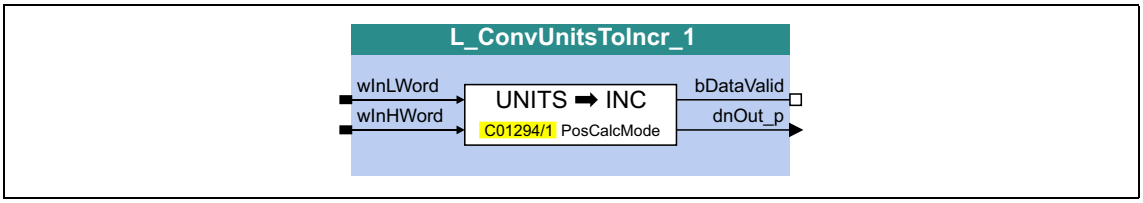
**Note!**

Conversion is remainder considered.

### 19.1.52 L_ConvUnitsToIncr_1

Taking into account the machine parameters, this FB converts a position value provided in the real unit of the machine into an internal 32-bit position value.

- The conversion mode has to be set in C01294/1. In the Lenze setting, no conversion takes place due to compatibility reasons.
- Conversion and provision of the result to *dnOut_p* do not take place in real time! Hence, the *bDataValid* output signalises when the conversion has been completed and the *dnOut_p* output value is consistent for transfer to following processes.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| wInLWord / wInHWord<br>WORD | Input signal Low Word / High Word |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bDataValid<br>BOOL | FALSE | Conversion is active, *dnOut_p* is not valid. |
| | TRUE | Conversion is completed, *dnOut_p* is valid (consistent). |
| dnOut_p<br>DINT | Result of conversion in [increments] | |

#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01294/1 | | | Mode: Position calculation |
| | 0 | dnOut_p=HW+LW | No conversion (Lenze setting)<br>**Note!**<br>If a cycle length is set in C01201/1, a calculation is made under the following conditions:<br>• Position specification is higher than or equal to the cycle length.<br>• Position specification is negative.<br>For a correct positioning process, wait until *bDataValid* = TRUE. |
| | 1 | 16 bits: LW=+/-32767 | *wInLWord* = ±32767 [Unit] |
| | 2 | 16 bits: HW=+/-; LW=0..65535 | *wInLWord* = 0 ... 65535 [Unit]<br>*wInHWord* = sign<br>(0 = positive; ≠0 = negative) |
| | 3 | 32 bits: HW_LW=+/-214748_3647 | *wInHWord* and *wInLWord* = ±214748.3647 [Unit] |

### Conversion formula for modes 1 ... 3

$$\text{dnOut_p [incr.]} = \text{Input value [Unit]} \cdot \frac{\text{C01202/1}}{\text{C01202/2}} \cdot \frac{\text{C01203/2}}{\text{C01203/1}} \cdot \frac{65536\,[\text{Incr./rev.}]}{\text{C01204}}$$

C01202/1: Gearbox factor (numerator)
C01202/2: Gearbox factor (denominator)
C01203/1: Encoder gearbox factor (numerator)
C01203/2: Encoder gearbox factor (denominator)
C01204: Feed constant

[19-18] Conversion formula for [Unit] → [increments]

### Example

- All gearbox factors = 1
- Feed constant = 360°/revolution

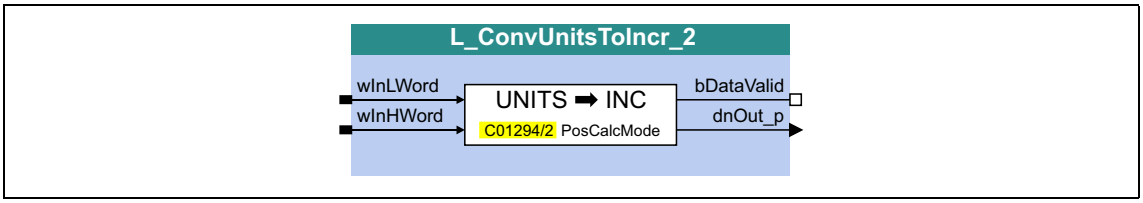
| Conversion mode (C01294) | | wInHWord | wInLWord | Input value [Unit] | dnOut_p [Increments |
|---|---|---|---|---|---|
| 0 | dnOut_p=HW+LW (no conversion) | 32-bit input value* | | -32000 | -32000 |
| | | 65535 | 33536 | | |
| 1 | 16 bits: LW=+/-32767 | No meaning | 16-bit input value* | -32000 | -5825422 |
| | | 65535 | 33536 | | |
| 2 | 16 bits: HW=+/-; LW=0..65535 | Sign | 16-bit input value | -33536 | -6105042 |
| | | 65535 ≡ negative | 33536 | | |
| 3 | 32 bits: HW_LW=+/-214748_3647 | 32-bit input value** | | -3.2000 | -582 |
| | | 65535 | 33536 | | |

* Two's complement ** Two's complement with four decimal positions

### 19.1.53 L_ConvUnitsToIncr_2

Taking into account the machine parameters, this FB converts a position value provided in the real unit of the machine into an internal 32-bit position value.

- The conversion mode has to be set in C01294/2. In the Lenze setting, no conversion takes place due to compatibility reasons.
- Conversion and provision of the result to *dnOut_p* do not take place in real time! Hence, the *bDataValid* output signalises when the conversion has been completed and the *dnOut_p* output value is consistent for transfer to following processes.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| wInLWord / wInHWord<br>WORD | Input signal Low Word / High Word |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bDataValid<br>BOOL | FALSE | Conversion is active, *dnOut_p* is not valid. |
| | TRUE | Conversion is completed, *dnOut_p* is valid (consistent). |
| dnOut_p<br>DINT | Result of conversion in [increments] | |

#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01294/2 | | | Mode: Position calculation |
| | 0 | dnOut_p=HW+LW | No conversion (Lenze setting)<br>**Note!**<br>If a cycle length is set in C01201/1, a calculation is made under the following conditions:<br>• Position specification is higher than or equal to the cycle length.<br>• Position specification is negative.<br>For a correct positioning process, wait until *bDataValid* = TRUE. |
| | 1 | 16 bits: LW=+/-32767 | *wInLWord* = ±32767 [Unit] |
| | 2 | 16 bits: HW=+/-; LW=0..65535 | *wInLWord* = 0 ... 65535 [Unit]<br>*wInHWord* = sign<br>(0 = positive; ≠0 = negative) |
| | 3 | 32 bits: HW_LW=+/-214748_3647 | *wInHWord* and *wInLWord* = ±214748.3647 [Unit] |

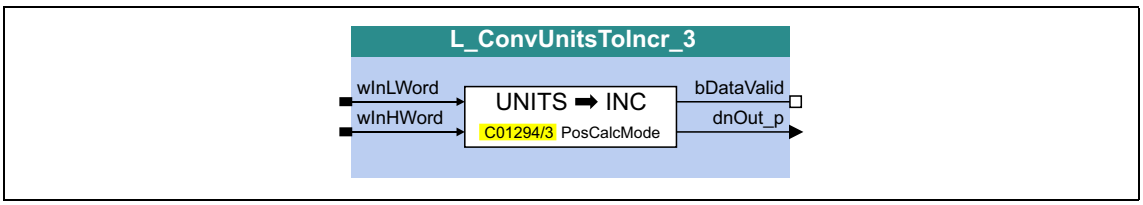


For a detailed functional description see L_ConvUnitsToIncr_1.

### 19.1.54 L_ConvUnitsToIncr_3

Taking into account the machine parameters, this FB converts a position value provided in the real unit of the machine into an internal 32-bit position value.

- The conversion mode has to be set in C01294/3. In the Lenze setting, no conversion takes place due to compatibility reasons.
- Conversion and provision of the result to *dnOut_p* do not take place in real time! Hence, the *bDataValid* output signalises when the conversion has been completed and the *dnOut_p* output value is consistent for transfer to following processes.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| wInLWord / wInHWord<br>WORD | Input signal Low Word / High Word |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bDataValid<br>BOOL | FALSE | Conversion is active, *dnOut_p* is not valid. |
| | TRUE | Conversion is completed, *dnOut_p* is valid (consistent). |
| dnOut_p<br>DINT | Result of conversion in [increments] | |

#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01294/3 | | | Mode: Position calculation |
| | 0 | dnOut_p=HW+LW | No conversion (Lenze setting)<br>**Note!**<br>If a cycle length is set in C01201/1, a calculation is made under the following conditions:<br>• Position specification is higher than or equal to the cycle length.<br>• Position specification is negative.<br>For a correct positioning process, wait until *bDataValid* = TRUE. |
| | 1 | 16 bits: LW=+/-32767 | *wInLWord* = ±32767 [Unit] |
| | 2 | 16 bits: HW=+/-; LW=0..65535 | *wInLWord* = 0 ... 65535 [Unit]<br>*wInHWord* = sign<br>(0 = positive; ≠0 = negative) |
| | 3 | 32 bits: HW_LW=+/-214748_3647 | *wInHWord* and *wInLWord* = ±214748.3647 [Unit] |

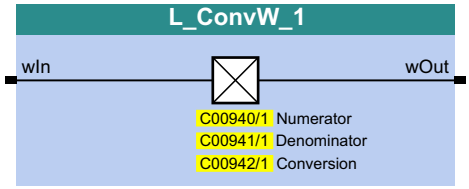


For a detailed functional description see L_ConvUnitsToIncr_1.

### 19.1.55 L_ConvW_1

This FB serves to convert analog signal forms. The following conversions per parameter can be selected:

- [%] → [incr/ms]
- [incr/ms] → [%]
- Conversion via parameterisable factors
- Input signal is passed through without conversion




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| wIn<br>WORD | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00940/1 | -32767 | | 32767 | Numerator<br>• Lenze setting: 1 |
| C00941/1 | -32767 | | 32767 | Denominator<br>• Lenze setting: 1 |
| C00942/1 | | | | Selection of the conversion |
| | 0 | *wOut* = *wIn* (no conversion) | | |
| | 1 | [%] → [incr/ms] | | |
| | 2 | [incr/ms] → [%] | | |
| | 3 | *wOut* = *wIn* * C00940 / C00941 | | |

**Conversion formulae**

**Note!**

Division is remainder considered.

$$wOut\ [incr/ms] = \frac{wIn\ [\%] \cdot C00011\ [rpm] \cdot 65536\ [incr/rev.]}{100\ \% \cdot 60\ [s/min] \cdot 1000\ [ms/s]}$$

[19-19] Conversion formula for selection 1: [%] → [incr/ms]

$$wOut\ [\%] = \frac{wIn\ [incr/ms] \cdot 100\ \% \cdot 60\ [s/min] \cdot 1000\ [ms/s]}{C00011\ [rpm] \cdot 65536\ [incr/rev.]}$$

[19-20] Conversion formula for selection 2: [incr/ms] → [%]

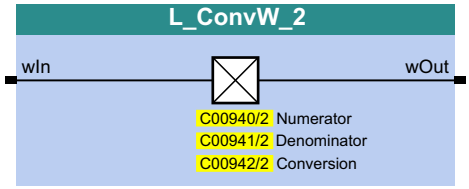
$$wOut = wIn \cdot \frac{C00940}{C00941}$$

[19-21] Conversion formula for selection 3: Parameterisable factors

## 19.1.56 L_ConvW_2

This FB serves to convert analog signal forms. The following conversions per parameter can be selected:

- [%] → [incr/ms]
- [incr/ms] → [%]
- Conversion via parameterisable factors
- Input signal is passed through without conversion




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| wIn<br>WORD | Input signal |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### Parameters

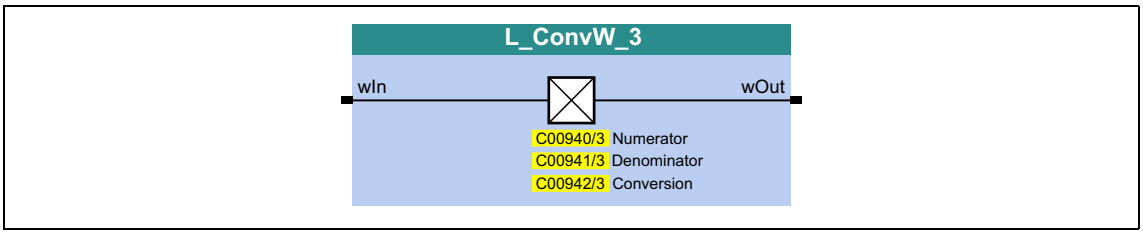
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00940/2 | -32767 | | 32767 | Numerator<br>• Lenze setting: 1 |
| C00941/2 | -32767 | | 32767 | Denominator<br>• Lenze setting: 1 |
| C00942/2 | | | | Selection of the conversion |
| | 0 | *wOut* = *wIn* (no conversion) | | |
| | 1 | [%] → [incr/ms] | | |
| | 2 | [incr/ms] → [%] | | |
| | 3 | *wOut* = *wIn* * C00940 / C00941 | | |

For conversion formulae see L_ConvW_1.

### 19.1.57 L_ConvW_3

This FB serves to convert analog signal forms. The following conversions per parameter can be selected:

- [%] → [incr/ms]
- [incr/ms] → [%]
- Conversion via parameterisable factors
- Input signal is passed through without conversion




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| wIn<br>WORD | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

#### Parameters

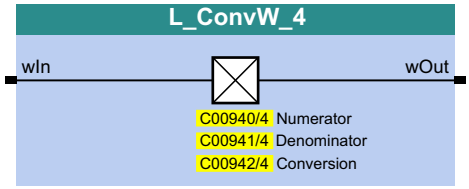
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00940/3 | -32767 | | 32767 | Numerator<br>• Lenze setting: 1 |
| C00941/3 | -32767 | | 32767 | Denominator<br>• Lenze setting: 1 |
| C00942/3 | | | | Selection of the conversion |
| | 0 | *wOut* = *wIn* (no conversion) | | |
| | 1 | [%] → [incr/ms] | | |
| | 2 | [incr/ms] → [%] | | |
| | 3 | *wOut* = *wIn* * C00940 / C00941 | | |

For conversion formulae see L_ConvW_1.

### 19.1.58 L_ConvW_4

This FB serves to convert analog signal forms. The following conversions per parameter can be selected:

- [%] → [incr/ms]
- [incr/ms] → [%]
- Conversion via parameterisable factors
- Input signal is passed through without conversion




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| wIn<br>WORD | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

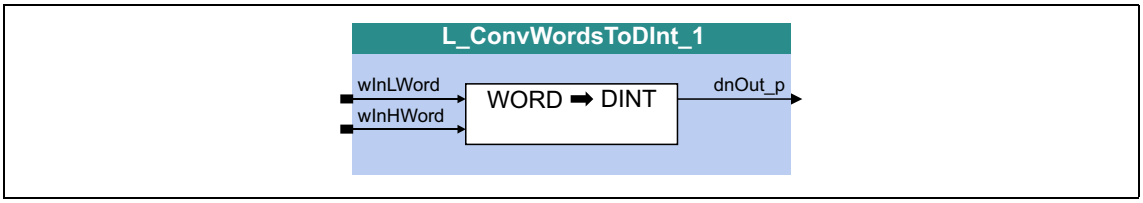
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00940/4 | -32767 | | 32767 | Numerator<br>• Lenze setting: 1 |
| C00941/4 | -32767 | | 32767 | Denominator<br>• Lenze setting: 1 |
| C00942/4 | | | | Selection of the conversion |
| | 0 | *wOut* = *wIn* (no conversion) | | |
| | 1 | [%] → [incr/ms] | | |
| | 2 | [incr/ms] → [%] | | |
| | 3 | *wOut* = *wIn* * C00940 / C00941 | | |

For conversion formulae see L_ConvW_1.

### 19.1.59 L_ConvWordsToDInt_1

This FB converts two input values of the type "WORD" into one output value of the type "DINT".




#### Inputs

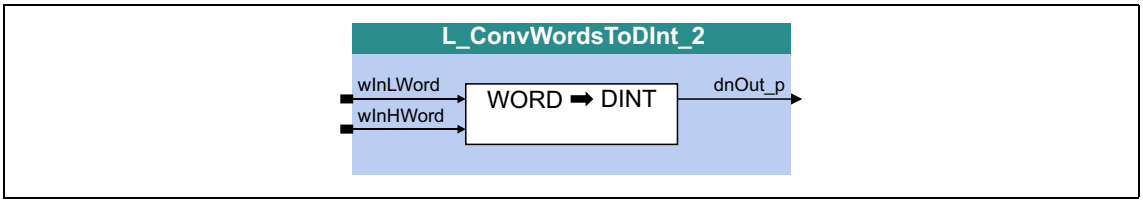
| Designator<br>Data type | Information/possible settings |
|---|---|
| wInLWord<br>WORD | Input signal Low Word |
| wInHWord<br>WORD | Input signal High Word |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

### 19.1.60 L_ConvWordsToDInt_2

This FB converts two input values of the type "WORD" into one output value of the type "DINT".




#### Inputs

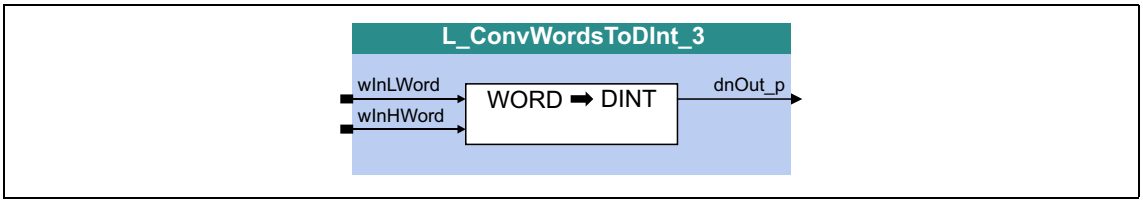
| Designator<br>Data type | Information/possible settings |
|---|---|
| wInLWord<br>WORD | Input signal Low Word |
| wInHWord<br>WORD | Input signal High Word |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

### 19.1.61 L_ConvWordsToDInt_3

This FB converts two input values of the type "WORD" into one output value of the type "DINT".




#### Inputs

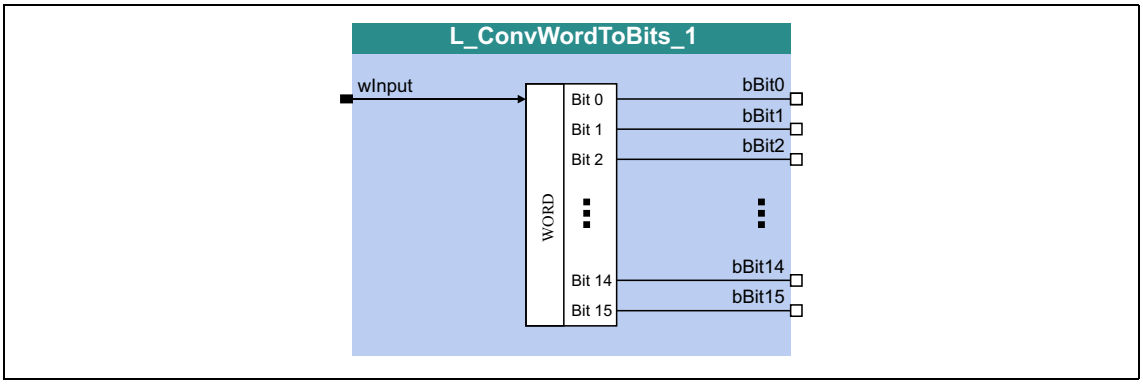
| Designator<br>Data type | Information/possible settings |
|---|---|
| wInLWord<br>WORD | Input signal Low Word |
| wInHWord<br>WORD | Input signal High Word |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

### 19.1.62 L_ConvWordToBits_1

This FB converts an input value of "WORD" type into 16 individual binary signals.




#### Inputs

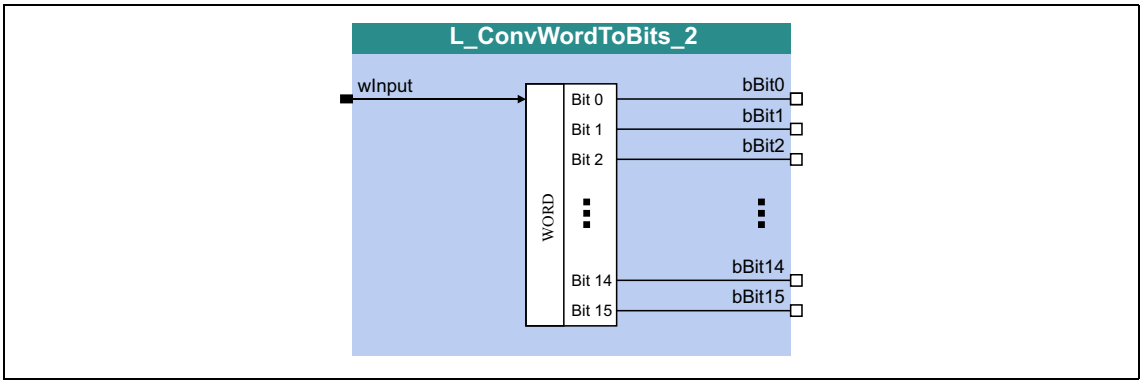
| Designator<br>Data type | Information/possible settings |
|---|---|
| wInput<br>WORD | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Output signal |

### 19.1.63 L_ConvWordToBits_2

This FB converts an input value of "WORD" type into 16 individual binary signals.




#### Inputs

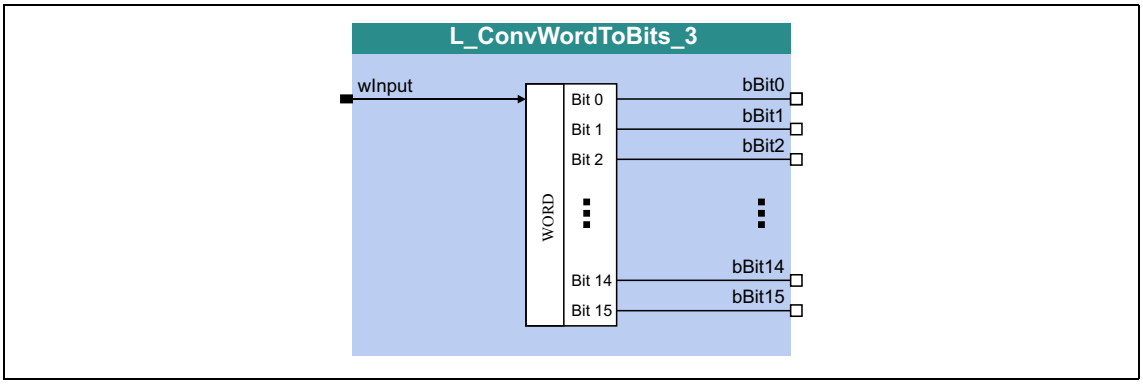
| Designator<br>Data type | Information/possible settings |
|---|---|
| wInput<br>WORD | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Output signal |

### 19.1.64 L_ConvWordToBits_3

This FB converts an input value of "WORD" type into 16 individual binary signals.




#### Inputs

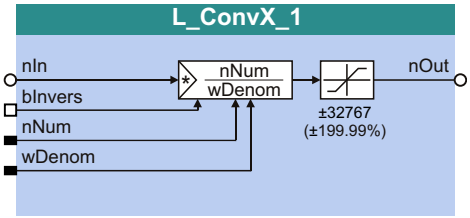
| Designator<br>Data type | Information/possible settings |
|---|---|
| wInput<br>WORD | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Output signal |

### 19.1.65 L_ConvX_1

This FB scales an analog value.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn<br>INT | Input signal | |
| bInvers<br>BOOL | Invert sign of the output signal | |
| | FALSE | $nOut = nIn \cdot \frac{nNum}{wDenom}$ |
| | TRUE | $nOut = nIn \cdot \frac{nNum}{wDenom} \cdot (-1)$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

#### Outputs

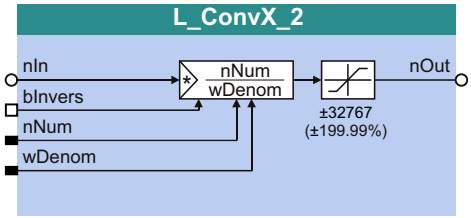
| Designator<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

**Note!**

Conversion is remainder considered.

### 19.1.66 L_ConvX_2

This FB scales an analog value.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn<br>INT | Input signal | |
| bInvers<br>BOOL | Invert sign of the output signal | |
| | FALSE | $nOut = nIn \cdot \frac{nNum}{wDenom}$ |
| | TRUE | $nOut = nIn \cdot \frac{nNum}{wDenom} \cdot (-1)$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

#### Outputs

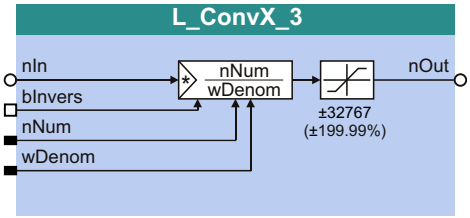
| Designator<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

**Note!**

Conversion is remainder considered.

### 19.1.67 L_ConvX_3

This FB scales an analog value.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn<br>INT | Input signal | |
| bInvers<br>BOOL | Invert sign of the output signal | |
| | FALSE | $nOut = nIn \cdot \frac{nNum}{wDenom}$ |
| | TRUE | $nOut = nIn \cdot \frac{nNum}{wDenom} \cdot (-1)$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

#### Outputs

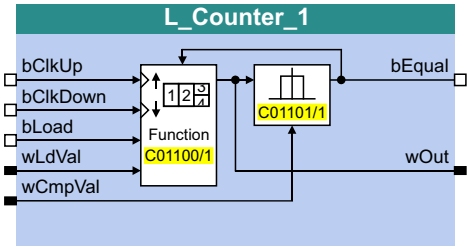
| Designator<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

**Note!**

Conversion is remainder considered.

## 19.1.68 L_Counter_1

This FB is a digital upcounter and downcounter with a parameterisable comparison operation.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bClkUp<br>BOOL | Clock input<br>• With each edge, the module counts up by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bClkDown<br>BOOL | Clock input<br>• With each edge, the module counts down by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bLoad<br>BOOL | Load input<br>• The input has the highest priority.<br>TRUE: Accept starting value *wLdVal*. |
| wLdVal<br>WORD | Starting value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |
| wCmpVal<br>WORD | Comparison value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bEqual<br>BOOL | Status signal "Comparison statement is true"<br>• The TRUE output is active in the Lenze setting if the current counter content is greater than or equal to the comparison value *wCmpVal*.<br>TRUE: The statement of the comparison mode selected in C01101/1 is true. |
| wOut<br>WORD | Counter content<br>• Internal limitation to ± 32767<br>• The most significant bit determines the sign! |

### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01100/1 | | | Function selection<br>• Lenze setting: Normal counting |
| | 0 | Normal counting | |
| | 1 | Auto reset | |
| | 2 | Manual reset | |

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01101/1 | | | Selection of comparison operation<br>• Lenze setting: Counter content ≥ comparison value |
| | 0 | Counter content ≥ comparison value | |
| | 1 | Counter content ≤ comparison value | |
| | 2 | Counter content = comparison value | |

**General function**

- Every FALSE/TRUE edge at the *bClkUp* input causes the block to count upwards by "1".
- Every FALSE/TRUE edge at the *bClkDown* input causes the block to count downwards by "1".

**Function "Normal counting"**

If the statement of the comparison mode selected in C01101/1 is true, the *bCompare* output is set to TRUE.

**Function "Auto reset"**

If the statement of the comparison mode selected in C01101/1 is true, the *bCompare* output is set to TRUE for 1 ms and the counter is reset to the *wLdVal* starting value.

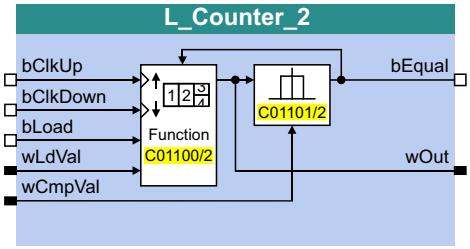
**Function "Manual reset"**

If the statement of the comparison mode selected in C01101/1 is true, the *bCompare* output is set to TRUE and the counter stops.

- Edges at *bClkUp* and *bClkDown* are ignored.
- The counter must be reset via the *bLoad* input.

### 19.1.69 L_Counter_2

This FB is a digital upcounter and downcounter with a parameterisable comparison operation.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bClkUp<br>BOOL | Clock input<br>• With each edge, the module counts up by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bClkDown<br>BOOL | Clock input<br>• With each edge, the module counts down by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bLoad<br>BOOL | Load input<br>• The input has the highest priority. |
| | TRUE: Accept starting value *wLdVal*. |
| wLdVal<br>WORD | Starting value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |
| wCmpVal<br>WORD | Comparison value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bEqual<br>BOOL | Status signal "Comparison statement is true"<br>• The TRUE output is active in the Lenze setting if the current counter content is greater than or equal to the comparison value *wCmpVal*. |
| | TRUE: The statement of the comparison mode selected in C01101/2 is true. |
| wOut<br>WORD | Counter content<br>• Internal limitation to ± 32767<br>• The most significant bit determines the sign! |

### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01100/2 | | | Function selection<br>• Lenze setting: Normal counting |
| | 0 | Normal counting | |
| | 1 | Auto reset | |
| | 2 | Manual reset | |
| C01101/2 | | | Selection of comparison operation<br>• Lenze setting: Counter content ≥ comparison value |
| | 0 | Counter content ≥ comparison value | |
| | 1 | Counter content ≤ comparison value | |
| | 2 | Counter content = comparison value | |

### General function

- Every FALSE/TRUE edge at the *bClkUp* input causes the block to count upwards by "1".
- Every FALSE/TRUE edge at the *bClkDown* input causes the block to count downwards by "1".

### Function "Normal counting"

If the statement of the comparison mode selected in C01101/2 is true, the *bCompare* output is set to TRUE.

### Function "Auto reset"

If the statement of the comparison mode selected in C01101/2 is true, the *bCompare* output is set to TRUE for 1 ms and the counter is reset to the *wLdVal* starting value.

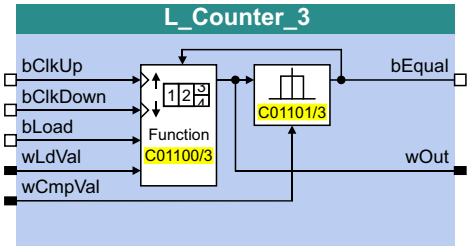
### Function "Manual reset"

If the statement of the comparison mode selected in C01101/2 is true, the *bCompare* output is set to TRUE and the counter stops.

- Edges at *bClkUp* and *bClkDown* are ignored.
- The counter must be reset via the *bLoad* input.

### 19.1.70 L_Counter_3

This FB is a digital upcounter and downcounter with a parameterisable comparison operation.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bClkUp<br>BOOL | Clock input<br>• With each edge, the module counts up by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bClkDown<br>BOOL | Clock input<br>• With each edge, the module counts down by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bLoad<br>BOOL | Load input<br>• The input has the highest priority. |
| | TRUE: Accept starting value *wLdVal*. |
| wLdVal<br>WORD | Starting value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |
| wCmpVal<br>WORD | Comparison value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bEqual<br>BOOL | Status signal "Comparison statement is true"<br>• The TRUE output is active in the Lenze setting if the current counter content is greater than or equal to the comparison value *wCmpVal*. |
| | TRUE: The statement of the comparison mode selected in C01101/3 is true. |
| wOut<br>WORD | Counter content<br>• Internal limitation to ± 32767<br>• The most significant bit determines the sign! |

### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01100/3 | | | Function selection<br>• Lenze setting: Normal counting |
| | 0 | Normal counting | |
| | 1 | Auto reset | |
| | 2 | Manual reset | |
| C01101/3 | | | Selection of comparison operation<br>• Lenze setting: Counter content ≥ comparison value |
| | 0 | Counter content ≥ comparison value | |
| | 1 | Counter content ≤ comparison value | |
| | 2 | Counter content = comparison value | |

### General function

- Every FALSE/TRUE edge at the *bClkUp* input causes the block to count upwards by "1".
- Every FALSE/TRUE edge at the *bClkDown* input causes the block to count downwards by "1".

### Function "Normal counting"

If the statement of the comparison mode selected in C01101/3 is true, the *bCompare* output is set to TRUE.

### Function "Auto reset"

If the statement of the comparison mode selected in C01101/3 is true, the *bCompare* output is set to TRUE for 1 ms and the counter is reset to the *wLdVal* starting value.

### Function "Manual reset"

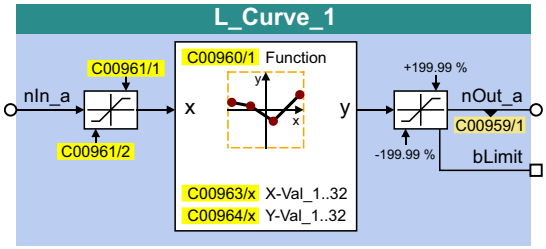
If the statement of the comparison mode selected in C01101/3 is true, the *bCompare* output is set to TRUE and the counter stops.

- Edges at *bClkUp* and *bClkDown* are ignored.
- The counter must be reset via the *bLoad* input.

### 19.1.71 L_Curve_1

Alternatively, this FB represents a characteristic curve or cam function y=f(x), whereby the input signal of the X-axis and the output signal of the Y-axis correspond.

- A limitation of the input value can be set in C00961/1 (upper limit) and C00961/2 (lower limit).
- The output value is internally limited to ±199.99 %. If a limitation of the output value is active, *bLimit* is set to TRUE.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value (x)<br>• ±16384 ≡ ±100 %<br>• A limitation of the input value can be set in C00961/1 (upper limit) and C00961/2 (lower limit). |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| nOut_a<br>INT | Output value (y)<br>• ±16384 ≡ ±100 %<br>• Internal limitation to ±32767 (±199.99 %) | |
| bLimit<br>BOOL | "Limitation active" status signal | |
| | TRUE | The output value is limited to ±199.99 %. |

#### Parameters

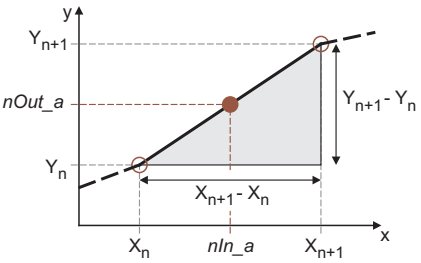
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00959/1<br>From version 02.00.00 | -199.99 | % | 199.99 | Output value<br>• Read only |
| C00960/1 | | | | Function selection |
| | 0 | Out = 0 | | FB deactivated (Lenze setting) |
| | 1 | Out = In | | |
| | 2 | Out = f(In) | | In preparation! |
| | 3 | Out = f(table) | | ▸ Function 3: nOut_a = f(table) |
| C00961/1 | -199.99 | % | 199.99 | Upper limit for input value<br>• Lenze setting: 199.99 % |
| C00961/2 | -199.99 | % | 199.99 | Lower limit for input value<br>• Lenze setting: -199.99 % |

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00963/1...32 | -32767 | | 32767 | X values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values X1 ... X32.<br>• ±16384 ≡ ±100 %<br>• Lenze setting: 0<br>▸ Function 3: nOut_a = f(table) |
| C00964/1...32 | -32767 | | 32767 | Y values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values Y1 ... Y32.<br>• ±16384 ≡ ±100 %<br>• Lenze setting: 0<br>▸ Function 3: nOut_a = f(table) |

### 19.1.71.1 Function 3: nOut_a = f(table)

If function 3 has been selected in C00960, the output value is calculated according to a characteristic.

- The characteristic may comprise up to 32 interpolation points which are defined by parameters.
- If the *nIn_a* input value is equal to one of the X interpolation points, the corresponding Y interpolation point will be output to *nOut_a*.
- If the *nIn_a* input value lies between two X interpolation points, the *nOut_a* output value will be interpolated linearly:




As per the equation,

$$\frac{X_{n+1} - X_n}{Y_{n+1} - Y_n} = \frac{nIn_a - X_n}{nOut_a - Y_n}$$

output value *nOut_a* for *nIn_a* is calculated as follows:

$$nOut_a = \frac{(nIn_a - X_n) \cdot (Y_{n+1} - Y_n)}{X_{n+1} - X_n} + Y_n$$

[19-22] Linear interpolation between two interpolation points

### Selection of the characteristic

The max. 32 interpolation points of the characteristic are selected via the subcodes of C00963 and C00964.

- The values are to be set as raw values (±16384 ≡ ±100 %).
- The same subcodes of C00963 and C00964 correspond to one pair of variates/interpolation point ($x_n$, $y_n$).

| | 1 | 2 | ... | 32 |
|---|---|---|---|---|
| **X** | C00963/**1** | C00963/**2** | C00963/... | C00963/**32** |
| **Y** | C00964/**1** | C00964/**2** | C00964/... | C00964/**32** |

- The first pair of variates (C00963/1 and C00964/1) is always valid.
- The X values of the characteristic must be entered in ascending order (X1 < X2 < ... < X32).
- An interruption of the ascending order of the X values represents the end of the characteristic (in our example: X5 < X4):

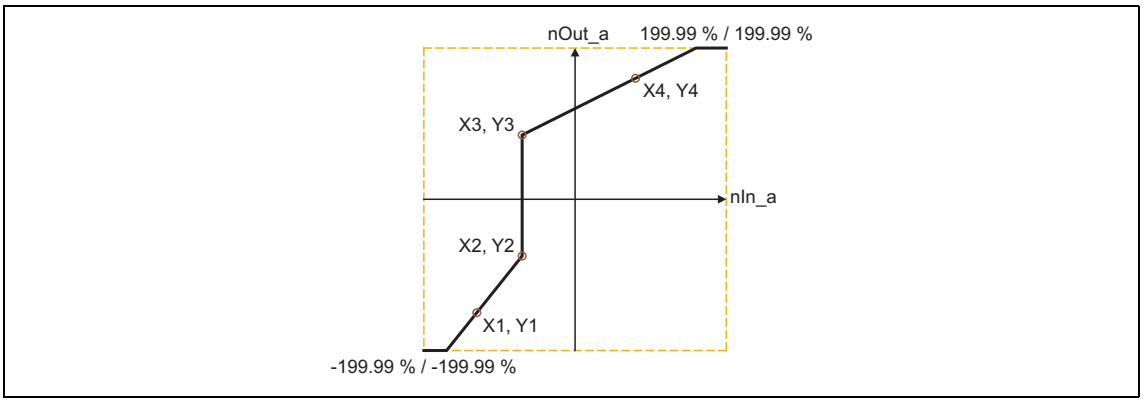
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... | 32 |
|---|---|---|---|---|---|---|---|---|---|
| **X** | -22938 (-140 %) | -14746 (-90 %) | 11469 (70 %) | 23757 (145 %) | 0 | 0 | 0 | 0 | 0 |
| **Y** | -22938 (-140 %) | -6554 (-40 %) | 18842 (115 %) | 26215 (160 %) | 0 | 0 | 0 | 0 | 0 |

- Starting at the last valid pairs of variates, extrapolation is performed on both sides until the end of the X value range (±199.99 %).

### Special cases

- Step changes/discontinuities can be displayed by using the same parameter setting for two consecutive X values. A step change is only valid if it has a valid pair of variates. Otherwise, extrapolation continues, starting at the last valid pair of variates.

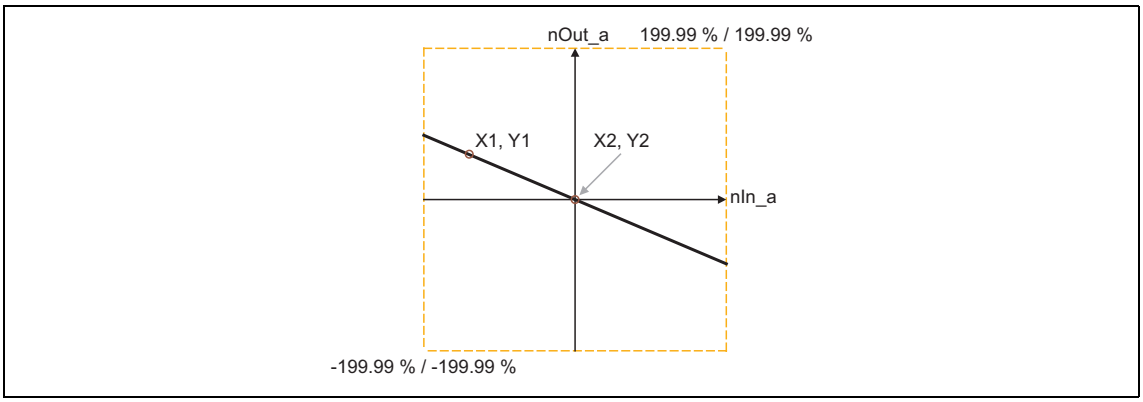
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... | 32 |
|---|---|---|---|---|---|---|---|---|---|
| **X** | -21300 (-130 %) | -11469 (-70 %) | -11469 (-70 %) | 13107 (80 %) | 0 | 0 | 0 | 0 | 0 |
| **Y** | -24576 (-150 %) | -12288 (-75 %) | 13927 (85 %) | 26215 (160 %) | 0 | 0 | 0 | 0 | 0 |




[19-23] Example: Characteristic with step change

- If only one interpolation point, X1, in the negative range is defined, X2 exists with the "0" Lenze setting being the valid interpolation point (X2 > X1), and a line is drawn through both interpolation points:

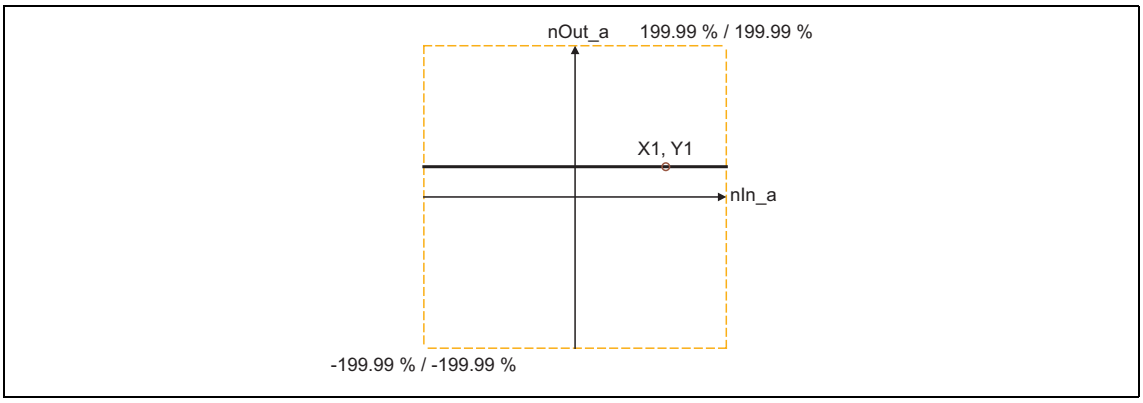
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... | 32 |
|---|---|---|---|---|---|---|---|---|---|
| **X** | -21300 (-130 %) | 0 (0 %) | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Y** | 6554 (40 %) | 0 (0 %) | 0 | 0 | 0 | 0 | 0 | 0 | 0 |




[19-24] Example: Characteristic with only one defined interpolation point and X1 < 0

- If only one interpolation point, X1, is defined in the positive range, a line is extrapolated:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... | 32 |
|---|---|---|---|---|---|---|---|---|---|
| **X** | 19661 (120 %) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Y** | 6554 (40 %) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

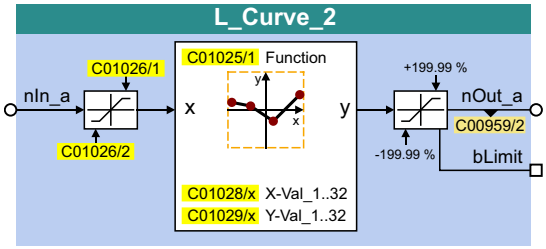



[19-25] Example: Characteristic with only one defined interpolation point and X1 > 0

## 19.1.72 L_Curve_2

Alternatively, this FB represents a characteristic curve or cam function y=f(x), whereby the input signal of the X-axis and the output signal of the Y-axis correspond.

- A limitation of the input value can be set in C01026/1 (upper limit) and C01026/2 (lower limit).
- The output value is internally limited to ±199.99 %. If a limitation of the output value is active, *bLimit* is set to TRUE.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value (x)<br>• ±16384 ≡ ±100 %<br>• A limitation of the input value can be set in C01026/1 (upper limit) and C01026/2 (lower limit). |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| nOut_a<br>INT | Output value (y)<br>• ±16384 ≡ ±100 %<br>• Internal limitation to ±32767 (±199.99 %) | |
| bLimit<br>BOOL | "Limitation active" status signal | |
| | TRUE | The output value is limited to ±199.99 %. |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00959/2 | -199.99 | % | 199.99 | Output value<br>• Read only |
| C001025/1 | | | | Function selection |
| | 0 | Out = 0 | | FB deactivated (Lenze setting) |
| | 1 | Out = In | | |
| | 2 | Out = f(In) | | In preparation! |
| | 3 | Out = f(table) | | See description of **L_Curve_1**:<br>▸ Function 3: nOut_a = f(table) |
| C01026/1 | -199.99 | % | 199.99 | Upper limit for input value<br>• Lenze setting: 199.99 % |
| C01026/2 | -199.99 | % | 199.99 | Lower limit for input value<br>• Lenze setting: -199.99 % |

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01028/1...32 | -32767 | | 32767 | X values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values X1 ... X32.<br>• ±16384 ≡ ±100 %<br>• Lenze setting: 0<br>See description of **L_Curve_1**:<br>▸ Function 3: nOut_a = f(table) |
| C01029/1...32 | -32767 | | 32767 | Y values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values Y1 ... Y32.<br>• ±16384 ≡ ±100 %<br>• Lenze setting: 0<br>See description of **L_Curve_1**:<br>▸ Function 3: nOut_a = f(table) |

### 19.1.73 L_Curve_3

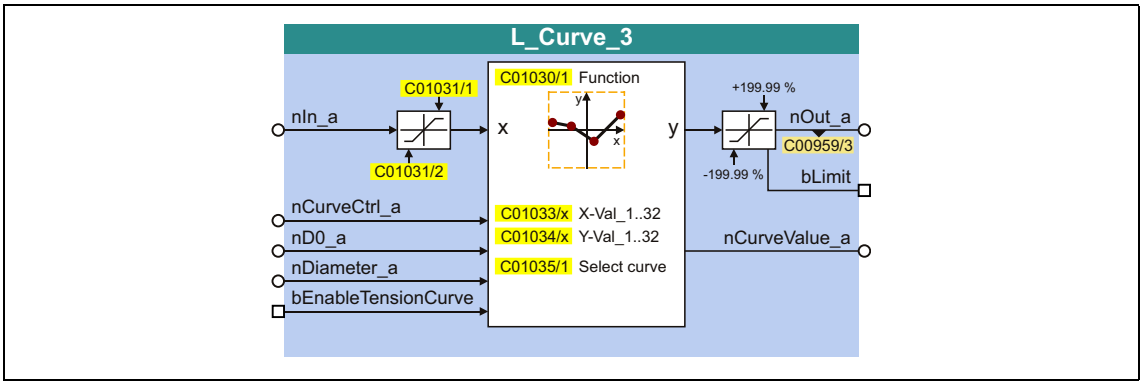
Alternatively, this FB represents a characteristic curve or cam function y=f(x), whereby the input signal of the X-axis and the output signal of the Y-axis correspond.

- A limitation of the input value can be set in C01031/1 (upper limit) and C01031/2 (lower limit).
- The output value is internally limited to ±199.99 %. If a limitation of the output value is active, *bLimit* is set to TRUE.

**Tip!**

From version 12.00.00 onwards the FB **L_Curve_3** can be used to create a tensiel force dependingt on the diameter for winding applications. For this purpose, the FB has been extended by further inputs/outputs and parameters.

▸ Use of the L_Curve_3 for tensile force characteristic




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_a<br>INT | Input value (x)<br>• ±16384 ≡ ±100 %<br>• A limitation of the input value can be set in C01031/1 (upper limit) and C01031/2 (lower limit). | |
| | When using the FB for tensile force characteristic (from V12.00.00):<br>Maximum tensile force setpoint in [%] | |
| The following inputs are only relevant when the FB is used for tensile force characteristic: | | |
| nCurveCtrl_a<br>INT<br>From version 12.00.00 | Slope of the characteristic (torque characteristic) in [%]<br>• ±16384 ≡ ±100 %<br>• Internal limitation to -100 ... +200 %<br>• With 100 %, there is a constant tensile force curve and thus a torque proportionally rising with D. | |
| nD0_a<br>INT<br>From version 12.00.00 | Initial point of the characteristic<br>• 16384 ≡ 100 % ≡ maximum diameter | |
| nDiameter_a<br>INT<br>From version 12.00.00 | Current diameter in [%]<br>• 16384 ≡ 100 % ≡ maximum diameter | |
| bEnableTensionCurve<br>BOOL<br>From version 12.00.00 | Activate tensile force setpoint | |
| | FALSE | Tensile force setpoint is not influenced. |
| | TRUE | Tensile force setpoint is multiplied by the characteristic value. |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| nOut_a<br>INT | Output value (y)<br>• ±16384 ≡ ±100 %<br>• Internal limitation to ±32767 (±199.99 %) | |
| | When using the FB for tensile force characteristic (from V12.00.00):<br>Current tensile force setpoint in [%]<br>• 16384 ≡ 100 % | |
| bLimit<br>BOOL | "Limitation active" status signal | |
| | TRUE | The output value is limited to ±199.99 %. |
| nCurveValue_a<br>INT<br>From version 12.00.00 | Current characteristic value of the tensile force characteristic in [%] | |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00959/3 | -199.99 | % | 199.99 | Output value<br>• Read only |
| C01030/1 | | | | Function selection |
| | 0 | Out = 0 | | FB deactivated (Lenze setting) |
| | 1 | Out = In | | |
| | 2 | Out = f(In) | | In preparation! |
| | 3 | Out = f(table) | | See description of **L_Curve_1**:<br>▸ Function 3: nOut_a = f(table) |
| | 4 | Out = f(characteristic) | | ▸ Use of the L_Curve_3 for tensile force characteristic |
| C01031/1 | -199.99 | % | 199.99 | Upper limit for input value<br>• Lenze setting: 199.99 % |
| C01031/2 | -199.99 | % | 199.99 | Lower limit for input value<br>• Lenze setting: -199.99 % |
| C01033/1...32 | -32767 | | 32767 | X values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values X1 ... X32.<br>• ±16384 ≡ ±100 %<br>• Lenze setting: 0<br>See description of **L_Curve_1**:<br>▸ Function 3: nOut_a = f(table) |
| C01034/1...32 | -32767 | | 32767 | Y values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values Y1 ... Y32.<br>• ±16384 ≡ ±100 %<br>• Lenze setting: 0<br>See description of **L_Curve_1**:<br>▸ Function 3: nOut_a = f(table) |

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01035/1<br>(from V12.00.00) | | | Selection of the tensile force profile<br>▸ Use of the L_Curve_3 for tensile force characteristic |
| | 0 | Linear tensile force profile | Lenze setting |
| | 1 | Linear torque profile | |
| | 2 | Tensile force profile according to characteristic | |

### 19.1.73.1 Use of the L_Curve_3 for tensile force characteristic

In case of centre winders, the drive torque is transmitted from the centre via the individual layers up to the winder surface. In case of smooth materials and high diameter areas, it is mostly required to reduce the tensile force depending on the diameter.

The FB **L_Curve_3** can be used to generate a tensile force profile depending on the diameter. For this purpose, the selection "4: Out = f(characteristic)" has to be set in C01030/1.

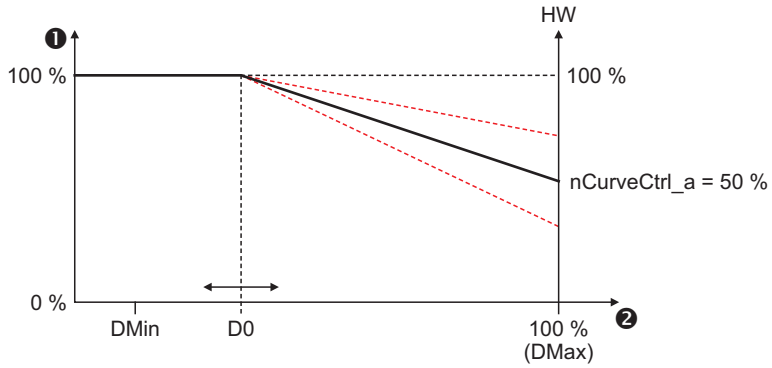
The characteristic is marked by an initial range with constant evaluation (100 %) and a second range where the tensile force is adapted to the diameter.

The part of the characteristic that depends on the diameter can be generated for a linear tensile force profile, a linear torque profile or based on a specified characteristic. The respective selection is made in C01035/1.

#### Linear tensile force profile

Characteristic for a linear tensile force profile (C01035/1 = 1)

**Characteristic**
❶ Tensile force
❷ Diameter



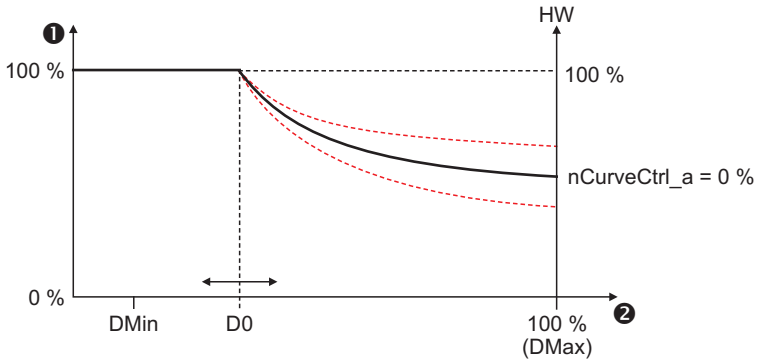

**Function for D > D0:**

$$100\,\% - \frac{D - D0}{100\,\% - D0} \cdot (100\,\% - HW)$$

| | |
|---|---|
| D | Current diameter (*nDiameter_a*) |
| D0 | Initial point for diameter dependency (*nD0_a*) |
| HW | Tensile force valuation (*nCurveCtrl_a*) |

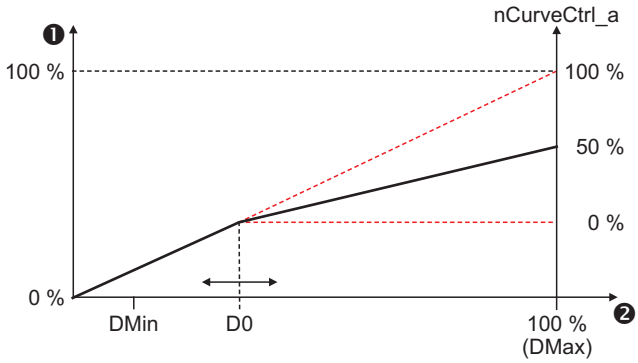
- *nD0_a* serves to define at which diameter the tensile force decrease is to start.

### Linear torque profile

| Characteristic for a linear torque profile (C01035/1 = 2) |
|---|
| **Characteristic**<br>❶ Tensile force<br>❷ Diameter<br>  <br><br>**Function for D > D0:**<br>$HW + \frac{(100\,\% - HW) \cdot D0}{D}$ |

| | |
|---|---|
| D | Current diameter (*nDiameter_a*) |
| D0 | Initial point for diameter dependency (*nD0_a*) |
| HW | Tensile force valuation (*nCurveCtrl_a*) |

- *nD0_a* serves to define at which diameter the tensile force decrease is to start.
- Since in this characteristic selection, the linear torque profile is in the focus, define in this case the slope of the torque curve via *dnCurveCtrl_n*:
  - When *nCurveCtrl_a* = 100 %, the tensile force remains unchanged.
  - When *nCurveCtrl_a* = 0 %, the tensile force is reduced up to D0/DMax.

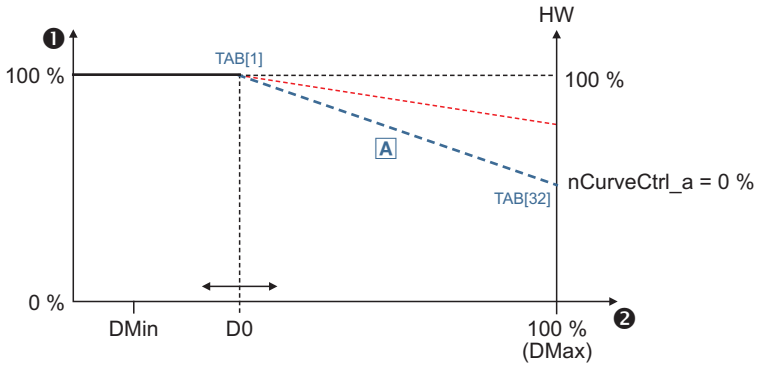
| Example: Steepness of the torque curve |
|---|
| **Characteristic**<br>❶ Torque<br>❷ Diameter<br>  <br> |

| | |
|---|---|
| D | Current diameter (*nDiameter_a*) |
| D0 | Initial point for diameter dependency (*nD0_a*) |

### Tensile force profile according to characteristic

The characteristic (A) can be parameterised in C01033/1...32.

- The characteristic comprises 32 values, regularly distributed over the diameter range of D0 ... DMax.
- The values are to be set as raw values (±16384 ≡ ±100 %).

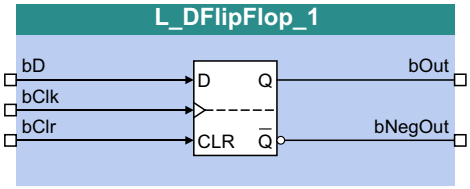
| Tensile force profile according to defined characteristic (C01035/3 = 6) |
|---|
| **Characteristic**<br>❶ Tensile force<br>❷ Diameter<br>  <br>**Function for D > D0:**<br>$100\,\% - \frac{(100\,\% - \mathrm{TAB[D\text{-}D0]}) \cdot (100\,\% - \mathrm{HW})}{100\,\%}$ |

| | |
|---|---|
| A | Table function (e.g. line) |
| TAB[D-D0] | Table value [1...32]for D-D0 |
| D0 | Initial point for diameter dependency (*nD0_a*) |
| HW | Tensile force valuation (*nCurveCtrl_a*) |

- *nD0_a* serves to define the start of the table range (D0).
- *nCurveCtrl_a* serves to change the slope of the characteristic.
  - When *nCurveCtrl_a* = 0 %, the slope of the characteristic remains unchanged.

### 19.1.74 L_DFlipFlop_1

The FB saves binary signals (DFlipFlop) in a clock-controlled way.




#### Inputs

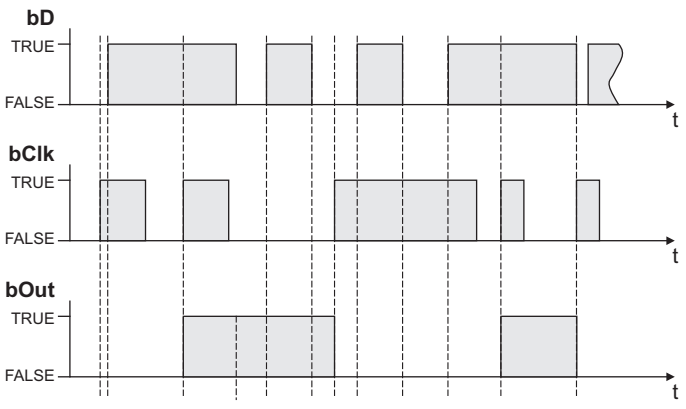
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bD<br>BOOL | Data input | |
| bClk<br>BOOL | Clock input<br>• Only FALSE/TRUE edges are evaluated | |
| bClr<br>BOOL | Reset input | |
| | TRUE | • The *bOut* output is set to FALSE.<br>• The *bNegOut* output is set to TRUE. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |
| bNegOut<br>BOOL | Output signal, inverted |

#### Function

If the *bClr* input = FALSE, a signal edge at the *bClk* input switches the static input signal *bD* to the *bOut* output, where it is retained:

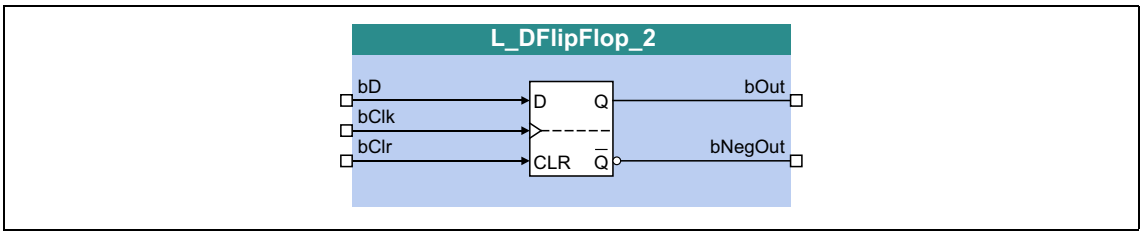


If the *bClr* input = TRUE:

- Due to the priority *bClr* > *bClk*, *bD* the *bOut* output signal can be set any time to the *FALSE* status by the *bClr* input signal = TRUE.
- The output signal is kept in this status independent of the other input signals.

## 19.1.75 L_DFlipFlop_2

The FB saves binary signals (DFlipFlop) in a clock-controlled way.




### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bD<br>BOOL | Data input | |
| bClk<br>BOOL | Clock input<br>• Only FALSE/TRUE edges are evaluated | |
| bClr<br>BOOL | Reset input | |
| | TRUE | • The *bOut* output is set to FALSE.<br>• The *bNegOut* output is set to TRUE. |

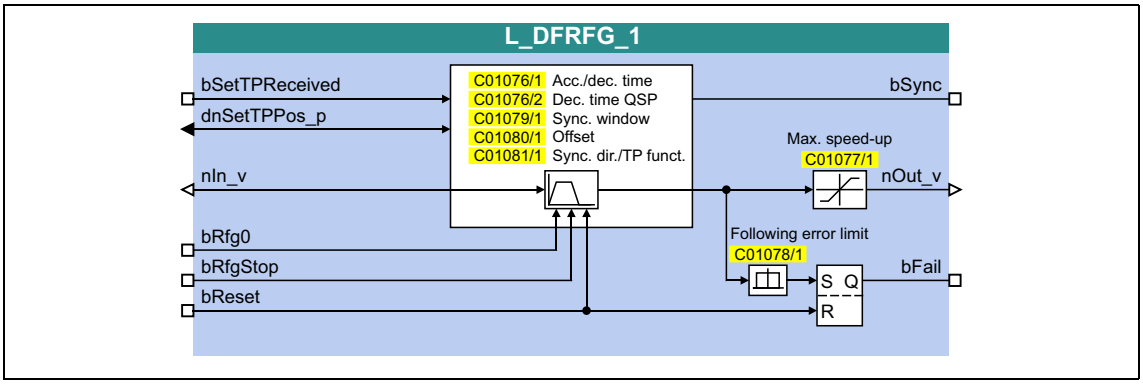
### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |
| bNegOut<br>BOOL | Output signal, inverted |

For a detailed functional description see L_DFlipFlop_1.

### 19.1.76 L_DFRFG_1

This FB synchronises a drive (slave) to the master value of a master drive and then executes a angular synchronism with regard to the digital frequency.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bSetTPReceived<br>BOOL | Input for status signal "Touch probe detected" | |
| | FALSE↗TRUE | Touch probe or zero pulse received. |
| dnSetTPPos_p<br>DINT | Input for accepting the position measured via touch probe | |
| nIn_v<br>INT | Speed setpoint of the master<br>• Scaling: 16384 ≡ 15000 rpm | |
| bRfg0<br>BOOL | Ramp down ramp function generator to "0" (quick stop function)<br>• This input can, for instance, be connected to the status signal "quick stop active". | |
| | TRUE | Ramp down the ramp function generator to "0"<br>• The drive is brought to a standstill with the deceleration time set in C01076/2. |
| bRfgStop<br>BOOL | Stop ramp function generator | |
| | TRUE | Stop ramp function generator<br>• The last status is output to *nOut_v*.<br>• The speed/angle setpoint *nIn_v* is saved.<br>• After *bStop* is reset to FALSE, the setpoint angle is approached via the ramp function generator. |
| bReset<br>BOOL | Reset added angle setpoint | |
| | TRUE | The internally added angle setpoint is reset and the ramp function generator is activated. If the status signal *bFail* is set, it will also be reset. |
| | TRUE↘FALSE | Detect speed/angle setpoint. |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bSync<br>BOOL | Status signal "Drive is running synchronously" | |
| | TRUE | The drive is running synchronously, the speed input id directly applied to the speed output. |
| nOut_v<br>INT | Speed/angle setpoint<br>• Scaling: 16384 ≡ 15000 rpm | |
| bFail<br>BOOL | Status signal "Angular difference exceeded" | |
| | TRUE | The angular difference set in C01078/1 has been exceeded. |

### Parameters

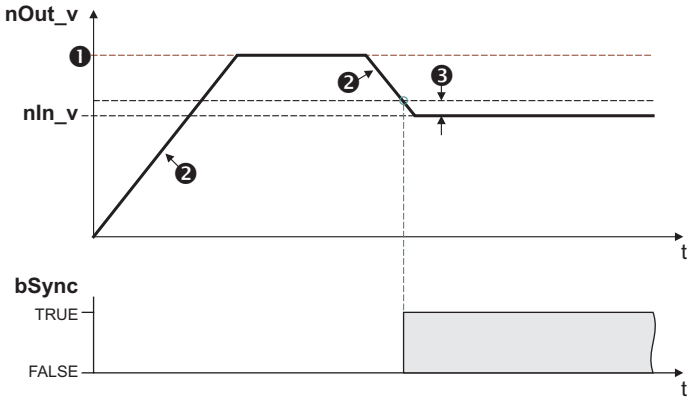
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01076/1 | 0.000 | s | 999.900 | Acceleration and deceleration time<br>• Relating to the reference speed (constant acceleration and deceleration) set in C00011.<br>• It is a maximum value which limits the acceleration of the ramp function generator.<br>• Lenze setting: 1.000 s |
| C01076/2 | 0.000 | s | 999.900 | Deceleration time for quick stop of the slave drive<br>• Relating to the reference speed (constant acceleration and deceleration) set in C00011.<br>• It is a maximum value which limits the deceleration of the ramp function generator.<br>• Lenze setting: 0.000 s |
| C01077/1 | 1 | rpm | 15000 | Max. speed-up<br>• Lenze setting: 3000 rpm |
| C01078/1 | 10 | Incr. | 2000000000 | Following error limit<br>• Scaling: A revolution is displayed with 65536 increments or steps.<br>• Lenze setting: 2000000000 incr. |
| C01079/1 | 0 | Incr. | 65535 | Synchronisation window (position)<br>• If the difference between the master and the slave is lower than the synchronisation window, the *bSync* status signal is set to TRUE.<br>• Lenze setting: 100 incr. |
| C01080/1 | -2147483647 | Incr. | 2147483647 | Offset<br>• Angular offset for the internal actual position of the FB<br>• Scaling: A revolution is displayed with 65536 increments or steps.<br>• Lenze setting: 0 incr. |

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01081/1 | | | Sync. direction/TP function<br>• Selection of the direction of rotation for synchronising<br>• With high offset values and low input speed, the drive may change its direction of rotation. In order to prevent this, a direction of rotation can be permanently defined via this parameter. |
| | 1 | cw/ccw - without TP (Lenze setting) | |
| | 2 | cw - without TP | |
| | 3 | ccw - without TP | |
| | 4 | cw/ccw - with TP | |
| | 5 | cw - with TP | |
| | 6 | ccw - with TP | |

### 19.1.76.1 Ramp function generator (profile generator)

The ramp function generator accelerates the slave drive to the input master speed.

- Synchronising starts when the FB receives a touch probe edge from the master axis (e.g. a conveying belt) at the *bSetTPReceived* input.
- When the slave reaches the setpoint speed of the master and the made up path difference is lower than the synchronisation window set in C01079/1, the *bSync* output is set to TRUE. At the same time, the FB switches the profile generator to inactive and outputs the *nIn_v* input speed to *nOut_v*.

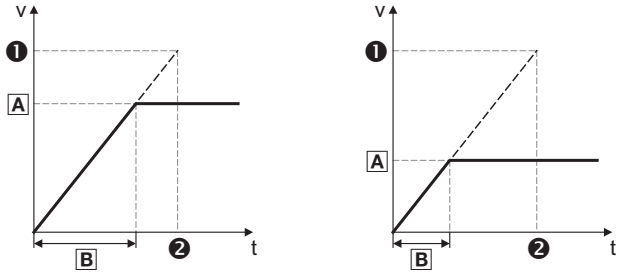



❶ C01077/1: Maximum speed
❷ C01076/1: Acceleration/deceleration time
❸ C01079/1: Synchronisation window

[19-26] Example: Synchronisation

- C01077/1 serves to set a maximum speed which limits the speed of the slave.

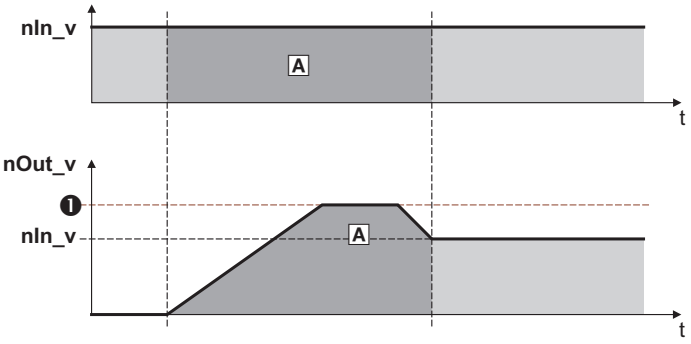
- The acceleration or deceleration in the synchronous point results from the acceleration/deceleration time set in C01076/1.
  - Reference for the acceleration/deceleration time is the reference speed (C00011):




| | |
|---|---|
| ❶ C00011: Reference speed<br>❷ C01076/1: Acceleration/deceleration time | A Effective target speed<br>B Required time |

[19-27] Connection between acceleration time and acceleration

- Based on the input speed of the master axis, a setpoint angle is calculated from the starting time onwards which leads the actual angle of the slave.
- Dependent on the master speed and the settings for acceleration and offset, the FB may travel oversynchronously for reducing the angular difference, i.e. *nOut_v* is higher than *nIn_v*:




❶ C01077/1: Maximum speed
A Identical surface below the speed profile that represents the covered path.

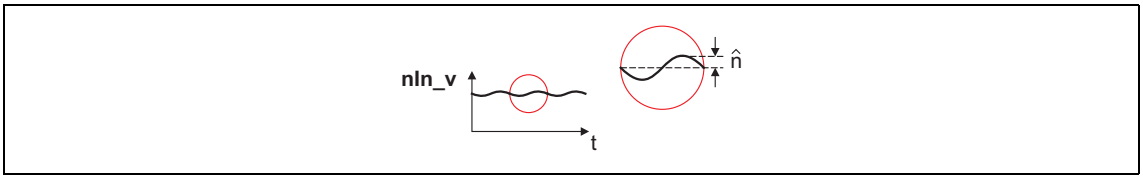
[19-28] Speed/time diagram

**Note!**

- Set the maximum speed in C01077/1 higher than the master speed to be expected. The speed is selected on the motor side and is independent of the reference speed (C00011). The higher the difference between maximum speed and master speed, the less time to the synchronous time is required.
- In case of a heavily oscillating input speed it may occur that directly after setting the *bSync* status signal to TRUE the FB still executes slight angle corrections.
- Generally avoid acceleration or deceleration processes of the master axis while the slave axes are synchronising.

**Tip!**

Set the synchronisation window in C01079/1 higher than the amplitude of beat available on the input signal:

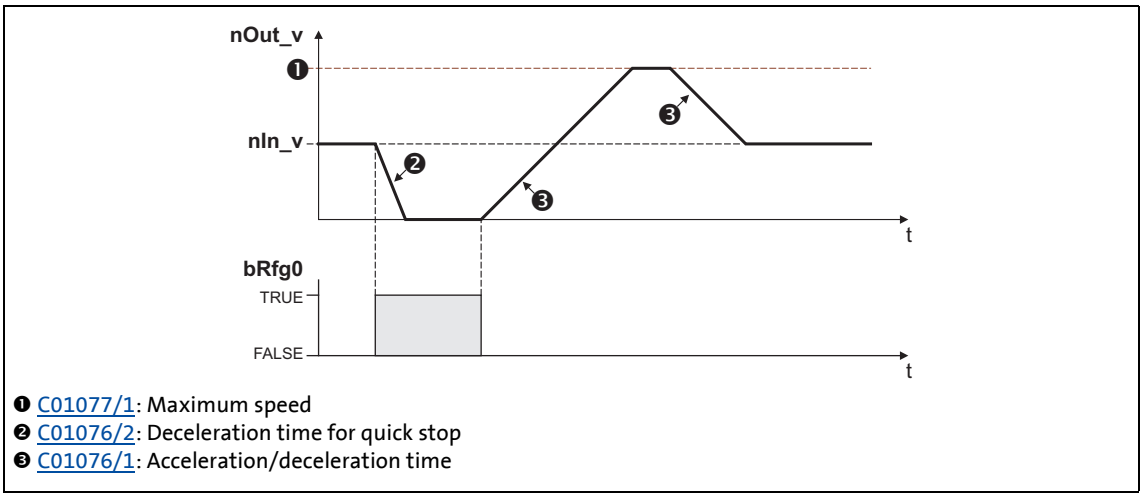



[19-29] Example: Beat on the input signal

### 19.1.76.2 Quick stop

The quick stop function removes the drive from the interconnection and leads it to standstill.

- Quick stop is activated by setting *bRfg0* to TRUE.
- C01076/2 is used to set the delay time in [s] which refers as the acceleration/deceleration time to the reference speed (C00011).
- The angle setpoint created internally on the basis of the input signal *nIn_v* is saved.
- After *bRfg0* is reset to FALSE, the angle setpoint is approached via the ramp function generator again.



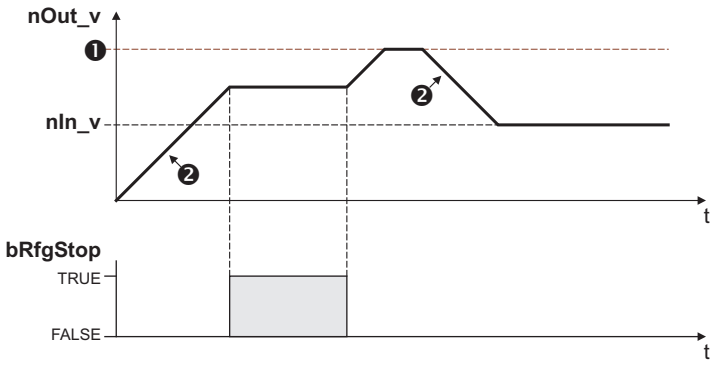

❶ C01077/1: Maximum speed
❷ C01076/2: Deceleration time for quick stop
❸ C01076/1: Acceleration/deceleration time

[19-30] Example: Quick stop function (QSP)

### 19.1.76.3 Ramp function generator stop

The stop function holds the status of the ramp function generator during operation.

- The stop function is activated by setting *bRfgStop* to TRUE.
- The last status is output to *nOut_v* (16384 ≡ 15000 rpm).
- The angle setpoint created internally on the basis of the input signal *nIn_v* is saved, thus the monitoring function of the angular difference between the created setpoint angle and actual angle cannot be activated.
- After *bRfgStop* is reset to FALSE, the angle setpoint is approached via the ramp function generator again.




❶ C01077/1: Maximum speed
❷ C01076/1: Acceleration/deceleration time

[19-31] Example: Stop function

### 19.1.76.4 Reset angle setpoint

By setting *bReset* to TRUE, the internally created angle setpoint is set to "0" and the ramp function generator is activated.

### 19.1.76.5 Detecting the angular difference

In C01078/1 , a limit value for monitoring the angular difference between the created setpoint and actual angles can be set.

- Scaling: A revolution is displayed with 65536 increments or steps.
- The ramp function generator can accept a angular difference of up to ±2140000000 increments (≈32000 revolutions).
- If monitoring responds, the status signal *bFail* is set to TRUE.
- If the internally added angle setpoint is reset by setting *bReset* to TRUE, the *bFail* status signal is reset to FALSE.

### 19.1.76.6 Offset setting

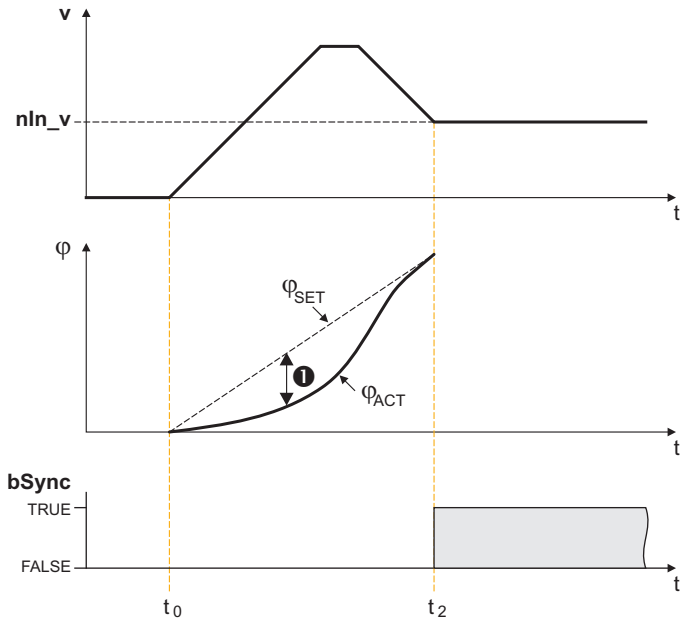
If the time to synchronicity is too long or the slave drive must not be operated oversynchronously, either the acceleration can be adapted or the touch probe initiator can be shifted. Since this is only possible to a limited extent due to construction conditions, in this case the better solution is selecting a virtual angular offset.

- In order to reach angle synchronicity, the difference between created setpoint and actual angle must be zero:

$$\varphi_{DIFF} = \varphi_{SET} - \varphi_{ACT} = 0$$

**Behaviour without offset**

If the actual angle starts with an offset = 0, a higher difference between setpoint angle and actual angle develops:

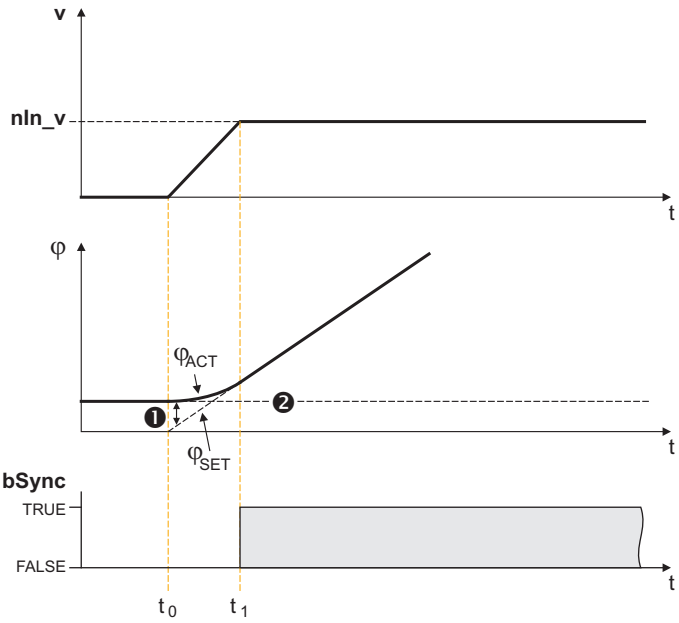



❶ Difference between setpoint angle and actual angle ($\varphi_{DIFF}$)
$t_0$ = touch probe event
$t_2$ = synchronicity reached

[19-32] Example: Without offset selection (offset = 0)

### Behaviour with offset

By selecting an offset, the actual position is preloaded with a non-zero value. This reduces the distance and the time to synchronicity.




❶ Difference between setpoint angle and actual angle ($\varphi_{DIFF}$)
❷ Offset [inc]
$t_0$ = touch probe event
$t_1$ = synchronicity achieved

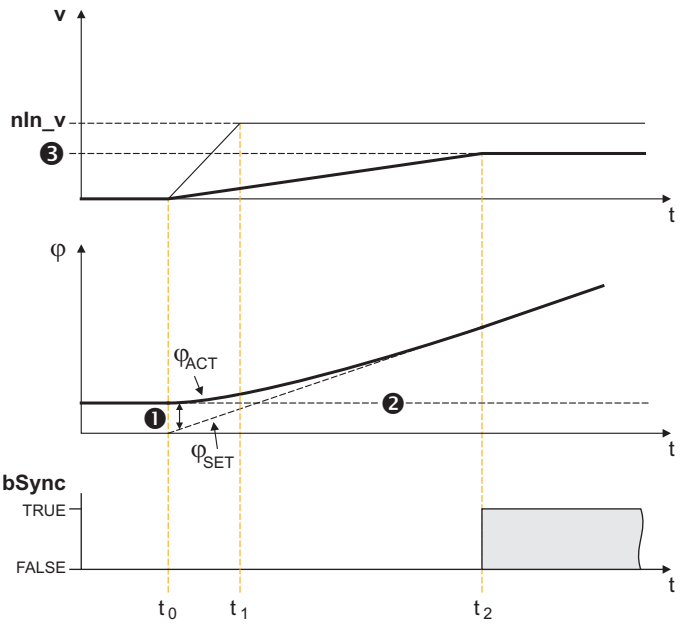
[19-33] Example 1: With offset selection

- When a positive offset is selected, the sign of the distance between setpoint and actual value changes. This distance is reduced due to the acceleration of the slave from the input of the touch probe pulse. Thus, the time to synchronicity is shorter.
- The offset refers to the master value selection and is scaled with an encoder revolution (≡ 65536 increments).
- The offset can be determined empirically, but it is sensible to calculate the required acceleration distance and select this value as offset:

$$\varphi_{ACC} = \frac{1}{2} \cdot \frac{(v_{SET})^2}{C00011\ [rpm]} \cdot C01076/1\ [ms] \cdot \frac{16384\ [incr./ms]}{15000\ [rpm]}$$

$\varphi_{ACC}$ = acceleration distance in [increments]
$v_{SET}$ = setpoint speed in [rpm]
C00011 = reference speed of the motor in [rpm]
C01076/1 = acceleration/deceleration time of the ramp generator in [ms]

The acceleration calculated from the acceleration/deceleration time (C01076/1) is a maximum acceleration that will not be reached if the input speed is reduced and the FB has to reach its target with a too high offset.




❶ Difference between setpoint angle and actual angle ($\varphi_{DIFF}$)
❷ Offset [inc]
❸ Reduced input speed
$t_0$ = touch probe event
$t_1$ = reaching synchronicity at original input speed
$t_2$ = synchronicity reached

[19-34] Example 2: With offset selection and reduced input speed

## Note!

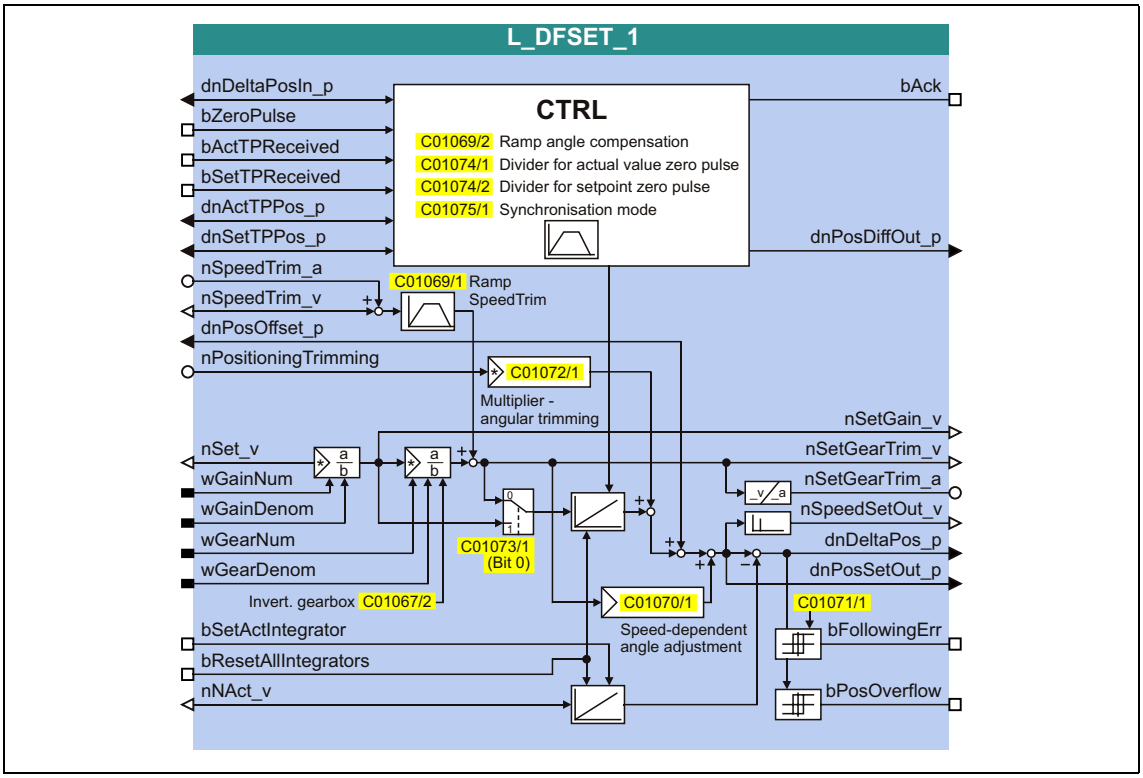
With a very low input speed, the acceleration can also get negative which causes the slave drive to change its direction of rotation.

In order to prevent a change of the direction of rotation, C01069/1 serves to select a permanent direction of rotation.

### 19.1.77 L_DFSET_1

This FB prepares the master value for a slave drive. This FB enables the inverter to follow the master drive true to speed and angle.

- You can select values for the stretch factor and gearbox factor of the slave.
- A speed trimming or angular trimming is possible.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnDeltaPosIn_p<br>DINT | Input for considering a following error for the mark correction<br>• The value of this input is used for the following error compensation if the *nNAct_v* input is not assigned. | |
| bZeroPulse<br>BOOL | Enable of the zero pulse-/touch probe synchronisation for the modes 2, 10 ... 13<br>• The mode is selected in C01075/1. | |
| | FALSE↗TRUE<br>TRUE | Synchronisation enabled. |
| bActTPReceived<br>BOOL | Input for detecting an actual value touch probe or zero pulse | |
| | FALSE↗TRUE | Actual value touch probe or zero pulse received. |
| bSetTPReceived<br>BOOL | Input for status signal "Touch probe detected" | |
| | FALSE↗TRUE | Touch probe or zero pulse received. |
| dnActTPPos_p<br>DINT | Input for accepting the position measured via touch probe at *bActTPReceived* | |
| dnSetTPPos_p<br>DINT | Input for accepting the position measured via touch probe at *bSetTPReceived* | |
| nSpeedTrim_a<br>INT | Speed trimming in [%]<br>• Scaling: 16384 ≡ 100 % | |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nSpeedTrim_v<br>INT | Speed trimming in [increments/ms]<br>• Scaling: 16384 ≡ 15000 rpm<br>• Speed trimming via this input is more precise. | |
| dnPosOffset_p<br>DINT | Angular offset in [increments]<br>• Scaling: A motor revolution is mapped with 65536 increments or steps. | |
| nPositionTrimming<br>INT | Angular trimming in [increments]<br>• Scaling: A motor revolution is mapped with 65536 increments or steps.<br>• When analog values are selected: 100 % ≡ 1/4 revolution ≡ 16384 increments<br>• The setting range can be extended via C01072/1. | |
| nSet_v<br>INT | Speed setpoint<br>• Scaling: 16384 ≡ 15000 rpm | |
| wGainNum<br>WORD | Stretch factor (numerator) | |
| wGainDenom<br>WORD | Stretch factor (denominator) | |
| wGearNum<br>WORD | Gearbox factor (numerator) | |
| wGearDenom<br>WORD | Gearbox factor (denominator) | |
| bSetActIntegrator<br>BOOL | Equalise angle integrators (current position = set position)<br>• This input has a higher priority than the *bResetAllIntegrators* input. | |
| | FALSE↗TRUE | Set following error *dnDeltaPos_p* = 0. |
| | TRUE | Equalise angle integrators. |
| bResetAllIntegrators<br>BOOL | Reset angle integrators | |
| | TRUE | Positional deviation, *dnPosSetOut_p* and *dnDeltaPos_p* are set to "0". |
| nNAct_v<br>INT | Actual value in [increments/ms] for calculating the actual position<br>• Scaling: 16384 ≡ 15000 rpm<br>• If this input is connected or assigned to an output, the following error created from integration of this input and difference with the set position is used for the following error compensation in case of mark correction. | |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bAck<br>BOOL | Status signal of the internal angle correction (or control signal for external angle correction) | |
| | TRUE | Status signal "Synchronising is executed"<br>• When the external angle correction is deactivated (C01073/1 - bit 2 = "0") and a ramp for the angle compensation is set in C01069/2. |
| | FALSE↗TRUE | Control signal "Start external angle correction"<br>• When the external angle correction is activated (C01073/1 - bit 2 = "1") and the angular offset is measured. |
| dnPosDiffOut_p<br>DINT | Angular offset in [increments]<br>• Difference between two positions measured by means of touch probe. | |
| nSetGain_v<br>INT | Speed setpoint in [increments/ms] evaluated with stretch factor<br>• Scaling: 16384 ≡ 15000 rpm | |
| nSetGearTrim_v<br>INT | Speed setpoint in [increments/ms] evaluated with stretch factor and gearbox factor<br>• Scaling: 16384 ≡ 15000 rpm | |
| nSetGearTrim_a<br>INT | Speed setpoint in [%] evaluated with stretch factor and gearbox factor<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| nSpeedSetOut_v<br>INT | Speed setpoint in [increments/ms]<br>• Scaling: 16384 ≡ 15000 rpm<br>• Derived from angle setpoint *dnPosSetOut_p*.<br>• Can change abruptly if e.g. the angular offset changes abruptly. | |
| dnDeltaPos_p<br>DINT | Following error between setpoint position and actual position in [increments]<br>• This output can be used for display purposes. | |
| dnPosSetOut_p<br>DINT | Angle setpoint in [increments]<br>• Scaling: A revolution is displayed with 65536 increments or steps. | |
| bFollowingErr<br>BOOL | Status signal "Following error" | |
| | TRUE | Following error occurred.<br>• The limit value set in C01071/1 has been exceeded. |
| bPosOverflow<br>BOOL | Status signal "angle controller overflow"<br>• An angle controller overflow occurs from a limit value of 2130706432 increments. | |
| | TRUE | Angle controller overflow occurred. |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01067/2 | | | | Invert. gearbox nSet_v |
| | 0 | not inverted | | |
| | 1 | inverted | | |
| | 2 | Automatically from MCK | | |
| C01069/1 | 0 | Incr./ms | 32767 | Ramp SpeedTrim<br>• Lenze setting: 10 incr./ms<br>• From version 12.00.00 onwards, this setting has a resolution increased by the factor 100: A value of 32767 is internally evaluated with 327.67. Hence, the unit is [incr./100 ms]. |
| C01069/2 | 0 | Incr./ms | 32767 | Ramp angle compensation<br>• The increments for the angle compensation in case of mark correction are specified in the Lenze setting with downstream 3rd-order polynomial.<br>• Lenze setting: 100 incr./ms |
| C01070/1 | -134217728 | Incr. | 134217728 | Speed-dependent angle adjustment<br>• Scaling: A revolution is displayed with 65536 increments or steps.<br>• Lenze setting: 0 incr. |
| C01071/1 | 10 | Incr. | 2147483647 | Following error limit<br>• Scaling: A motor revolution is mapped with 65536 increments or steps.<br>• Lenze setting: 32768 incr. |
| C01072/1 | -20000 | | 20000 | Multiplier - angular trimming<br>• Lenze setting: 1 |

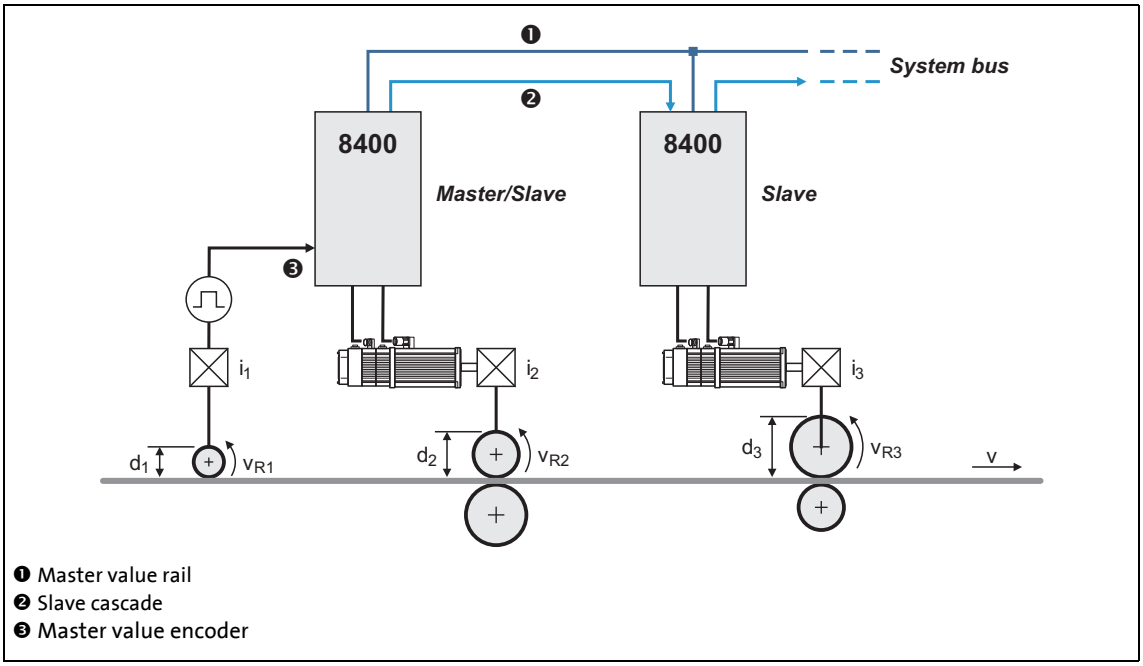
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01073/1 | Setting is bit coded: | | | Setting<br>• Lenze setting: 0x02 |
| | Bit 0 | Setpoint angle without gearbox factor | | 0 = Evaluation of the setpoint angle with gearbox factor.<br>1 = Evaluation of the setpoint angle without gearbox factor. |
| | Bit 1 | Angle correction with polynomial | | 0 = Angle correction without downstream polynomial.<br>1 = Angle correction with downstream polynomial. |
| | Bit 2 | External angle correction | | 0 = Internal angle correction<br>1 = External angle correction<br>• From version 13.00.00 |
| | Bit 3 | Reserved | | |
| | ... | | | |
| | Bit 7 | | | |
| C01074/1 | 0 | | 16384 | Divisor for actual value zero pulse<br>• Number of signals to be received at least until synchronisation gets active.<br>• Initialisation: 0 (≡ 1) |
| | **Note:** The settings 0 and 1 are functionally identical. If 0 or 1 is set, at least one pulse is required in order that the synchronisation gets active. | | | |
| C01074/2 | 0 | | 16384 | Divisor for setpoint zero pulse<br>• Number of signals to be received at least until synchronisation gets active.<br>• Initialisation: 0 (≡ 1) |
| | **Note:** The settings 0 and 1 are functionally identical. If 0 or 1 is set, at least one pulse is required in order that the synchronisation gets active. | | | |

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01075/1 | | | Synchronisation mode - mark correction |
| | 0 | Synchronisation inactive | Lenze setting |
| | 1 | Permanent synchronisation without enable via the *bZeroPulse* input<br>• Correction is executed over the shortest possible path. | The synchronisation only takes place after 2 setpoint pulses have been received at the *bSetTPReceived* input. (required pulse order: set-actual-set) |
| | 2 | Permanent synchronisation only with enable via the *bZeroPulse* input<br>• When *bZeroPulse* is set to TRUE, a permanent zero pulse/touch probe synchronisation is executed.<br>• Correction is executed over the shortest possible path. | |
| | 10 | One-time synchronisation<br>• An angular difference is compensated over the shortest possible path. | |
| | 11 | One-time synchronisation<br>• An angle difference is compensated in CW direction. | |
| | 12 | One-time synchronisation<br>• An angular difference is compensated in CCW direction. | |
| | 13 | One-time synchronisation<br>• An angular difference is compensated over the shortest possible path. | Synchronisation takes place after the first setpoint pulse has been received at the *bSetTPReceived* input and/or actual value pulse at the *bActTPReceived* input.<br>(Required clock sequence: Set/actual or actual/set) |

### 19.1.77.1 Master value rail/slave cascade

If there is only one master speed that is transmitted to all slave drives involved, e.g. via system bus (CAN), it is called master value rail.

If a slave drive takes over the master value generation for the following slave, it is called slave cascade.




❶ Master value rail
❷ Slave cascade
❸ Master value encoder

[19-35] Example: Master value rail/slave cascade

### 19.1.77.2 Setpoint conditioning with stretch factor and gearbox factor

#### Stretching factor

The stretch factor is required for the "speed synchronism via master value cascade" mode. It defines the ratio the slave drive is to be running with regard to its master value.

- The stretch factor evaluates the setpoint at the *nSet_v* input.
- The stretch factor must be selected via the *wGainNum* and *wGainDenom* inputs in the form of numerators and denominators.
- The result is provided at the *nSetGain_v* output.
  - Scaling: 16384 ≡ 15000 rpm

$$\text{nSetGain_v} = \text{nSet_v} \cdot \frac{\text{wGainNum}}{\text{wGainDenom}}$$

- If the stretch factor is 1 and the gearbox factors are selected correctly, the circumferential speeds of the rolls for master and slave 1 are identical in the example shown in the illustration [19-35].

#### Gearbox factor

The gearbox factor defines the gearbox ratio of the drive. Enter the ratio of the drive.

- The gearbox factor evaluates the setpoint at the *nSet_v* input multiplied by the stretch factor.
- The gearbox factor has to be selected via the inputs *wGearNum* and *wGearDenom* in the form of numerators and denominators.
- The bit 0 in C01073/1 serves to simply deactivate the evaluation with the gearbox factor (does not apply to the speed outputs *nSetGain_v*, *nSetGearTrim_v* and *nSetGearTrim_a*).
- The result is provided at the outputs *nSetGearTrim_v* and *nSetGearTrim_a*.
  - Scaling *nSetGearTrim_v*: 16384 ≡ 15000 rpm
  - Scaling *nSetGearTrim_a*: 16384 ≡ 100 % reference speed (C00011)

$$\text{nSetGearTrim_v} = \text{nSet_v} \cdot \text{Stretching factor} \cdot \frac{\text{wGearNum}}{\text{wGearDenom}}$$

$$\text{nSetGearTrim_v} = \text{nSet_v} \cdot \frac{\text{wGainNum}}{\text{wGainDenom}} \cdot \frac{\text{wGearNum}}{\text{wGearDenom}}$$

$$\text{nSetGearTrim_a} = \text{nSet_v} \cdot \frac{\text{C00011}}{15000} \cdot \frac{\text{wGainNum}}{\text{wGainDenom}} \cdot \frac{\text{wGearNum}}{\text{wGearDenom}}$$

**Note!**

The intermediate results and the result of the evaluation are limited to ±32767 increments (16 bits).

### 19.1.77.3 Processing correction values

#### Speed trimming

Speed trimming enables the connection of correction values, e.g. from a higher-level control loop. This permits an acceleration or deceleration of the drive.

The correction value can either be transmitted as an analog value via the *nSpeedTrim_a* input or for exact speed trimming as a speed value via the *nSpeedTrim_v* input.

- Scaling *nSpeedTrim_a*: 16384 ≡ 100 % reference speed (C00011)
- Scaling *nSpeedTrim_v*: 16384 ≡ 15000 rpm

Speed variations of the trimming values are specified via the SpeedTrim ramp (C01069/1).

### Angular trimming

Angular trimming enables the rotor position to be put forward or back with regard to the setpoint (the drive is leading or lagging).

The correction value must has to be defined in [increments] via the *nPositionTrimming* input. It is internally added to the angle setpoint.

- Scaling: A motor revolution is mapped with 65536 increments or steps.
- The maximum angular trimming amounts to ±½ motor revolution (±32767 increments).
- When analog values are selected: 100 % ≡ ¼ motor revolution ≡ 16384 increments.
- The setting range can be extended with the multiplier to be set in C01072/1.

### Angular offset

The *dnPosOffset_p* input serves to define a permanent angular offset for the setpoint of the drive.

- An adjustment takes place via the following error (*dnDeltaPos_p*) output and via the speed change at the *nSpeedSetOut_v* output.
- Example: A angular offset of 90° is to be set on the load side with a gearbox ratio of 3.8147:

$$\text{dnPosOffset_p} = \frac{90°}{360°} \cdot 65536 \left[\frac{\text{Increments}}{\text{Motor revolution}}\right] \cdot 3.8147 = 62500\ \text{[increments]}$$

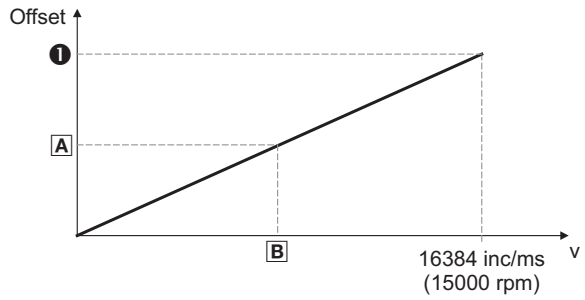
**Stop!**

Change the angular offset only step by step as the FB does not contain a ram generator for preventing torque impulses.

### Speed-proportional angle adjustment

The speed-proportional angle adjustment enables a leading or lagging of the angle with increasing speed and serves to compensate dead times and transmission times.

The correction value selected in C01070/1. in [increments] refers to a speed of 15000 rpm (linear relationship):




❶ C01070/1: Speed-proportional angle adjustment
A Effective offset
B Current speed

[19-36] Connection between offset/speed

### 19.1.77.4 Synchronising slave drive to master drive

If an angular synchronism of the drives is required (e.g. in printing units), the slave drives must be positioned to the master position since the FB **L_DFSET** as a master value can only receive and process one relative signal. The FB **L_DFSET** receives the master value via a real-time capable MotionBus as the system bus (CAN).

**Tip!**

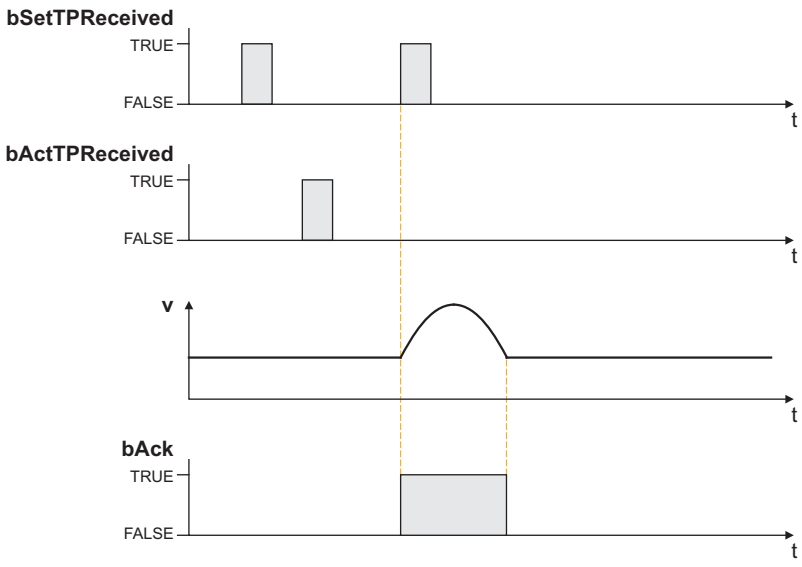
The 8400 TopLine serves to realise very easily a master value cascade via the cross communication with the Axis bus of the drives.

For synchronisation purposes, the slave drive is informed cyclically or once about its drive position by a pulse generated by the master drive. This pulse is either the zero pulse of the master feedback or the edge of a touch probe sensor. Moreover, a pulse must be generated by the slave drive. Only if the FB **L_DFSET** receives both signals, it can execute a synchronisation.

#### Synchronisation mode - mark correction

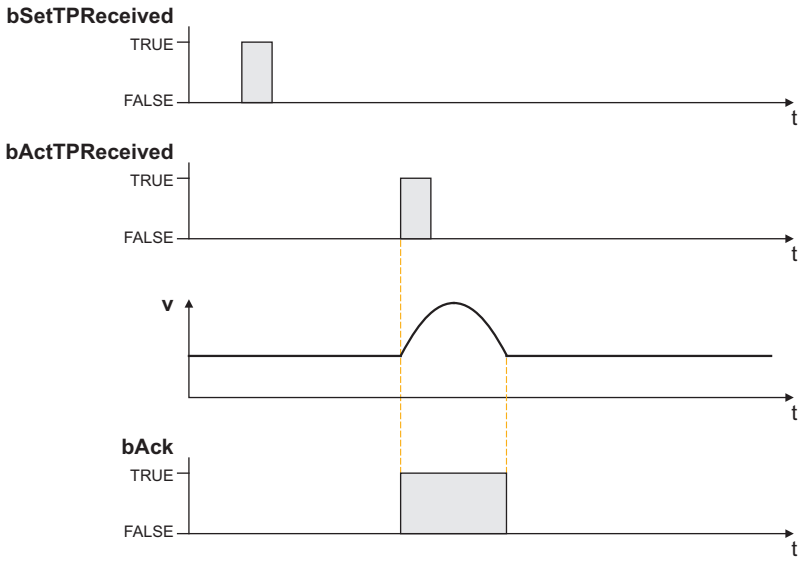
For synchronisation, the following modes are available in C01075/1 :

| Mode | | Info |
|---|---|---|
| 0 | Synchronisation inactive | Lenze setting |
| 1 | Permanent synchronisation without enable via the *bZeroPulse* input<br>• Correction is executed over the shortest possible path. | The synchronisation only takes place after 2 setpoint pulses have been received at the *bSetTPReceived* input.<br>(required pulse order: set-actual-set)<br>• See illustration [19-37]. (📖 1551) |
| 2 | Permanent synchronisation only with enable via the *bZeroPulse* input<br>• When *bZeroPulse* is set to TRUE, a permanent zero pulse/touch probe synchronisation is executed.<br>• Correction is executed over the shortest possible path. | |
| 10 | One-time synchronisation only with enable via the *bZeroPulse* input<br>• An angular difference is compensated over the shortest possible path. | |
| 11 | One-time synchronisation only with enable via the *bZeroPulse* input<br>• An angle difference is compensated in CW direction. | |
| 12 | One-time synchronisation only with enable via the *bZeroPulse* input<br>• An angular difference is compensated in CCW direction. | |
| 13 | One-time synchronisation only with enable via the *bZeroPulse* input<br>• An angular difference is compensated over the shortest possible path. | Synchronisation takes place after the first setpoint pulse has been received at the *bSetTPReceived* input and/or actual value pulse at the *bActTPReceived* input.<br>(Required clock sequence: Set/actual or actual/set)<br>• See illustration [19-38]. (📖 1551) |




Synchronisation mode (C01075/1) = 1, 2, 10, 11, 12
Divisor for actual value zero pulse (C01074/1) = 0 or 1
Divisor for setpoint zero pulse (C01074/2) = 0 or 1

[19-37] Synchronisation process in the modes 1, 2, 10, 11, 12




Synchronisation mode (C01075/1) = 13
Divisor for actual value zero pulse (C01074/1) = 0 or 1
Divisor for setpoint zero pulse (C01074/2) = 0 or 1

[19-38] Synchronisation process in mode 13

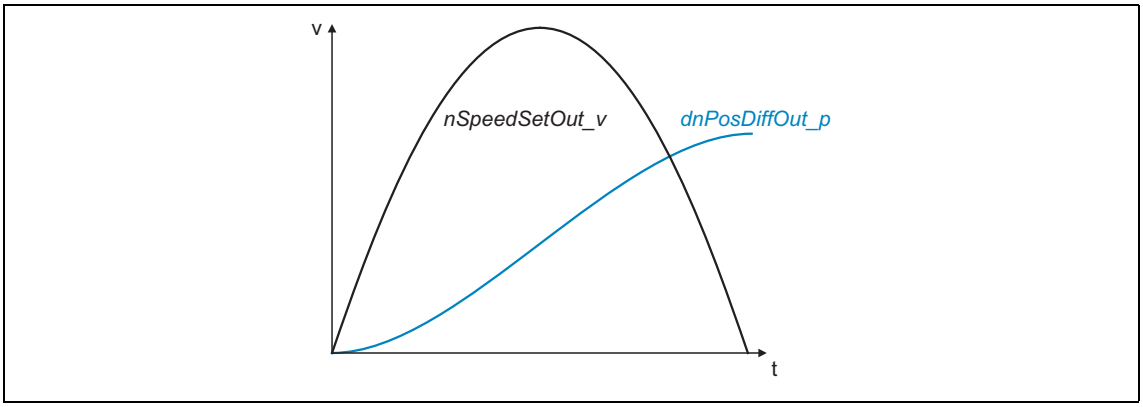
### Compensation process

At the input of the second set or actual touch probe pulse, the difference between master and slave position is detected which is then provided via the *nSpeedSetOut_v* speed output and equivalently as position at .

In C01073/1, the following bit coded settings can be made for the compensation procedure:

| Setting | | Info |
|---|---|---|
| Bit 1 | Angle correction with polynomial | If this bit is set (Lenze setting),the compensating movement is rounded by polynomial.<br>• The angle error to be compensated is limited to ±1073741823 increments due to the system. This applies in general when the **L_DFSET_1** carries out the compensation.<br>• The maximum compensating speed can be set in C1069/2. When "0" is set in C1069/2 if activated in C01075/1, the angle error is measured and output to *dnPosDiffOut_p*. |
| Bit 2 | External angle correction<br>From version 13.00.00 | If this bit is set, the FB **L_DFSET_1** does not compensate any angle errors.<br>• Every time the angle errors have been measured, the angular difference is output to *dnPosDiffOut_p*.<br>• In addition, a FALSE/TRUE edge is output to *bAck*. This signal serves to trigger an external profile generator which compensates the angular offset via a speed profile additionally. |

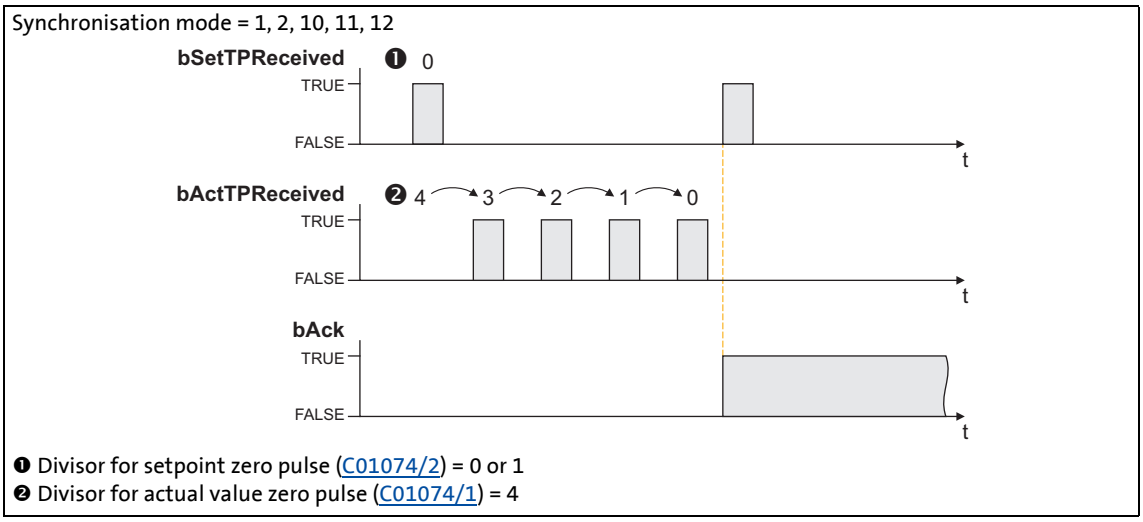
The following illustration shows a compensating process after the marks have been detected. The speed is added here as a parabola to the line speed. (The scaling in this diagram does not correspond to reality)




[19-39] Example: Compensation process

### 19.1.77.5 Masking out touch probe signals

When passing-through material is used, e.g. printed foil, touch probe initiators may respond several times per cycle. For suppressing such "interference pulses", count values can be selected in C01074/1 and C01074/2 that are decremented when the touch probe pulse has been received. Only when the counter content is "0", the synchronisation will be enabled. Please note that the settings 0 and 1 are functionally identical. If 0 or 1 is set, at least one pulse is required to activate the synchronisation.




[19-40] Synchronisation process in the modes 1, 2, 10, 11, 12 with a masking out of the touch probe pulses

### 19.1.77.6 Process monitoring functions

#### Following error

The *bFollowingErr* status output is set to TRUE if the drive cannot follow its setpoint angle.

- Possible causes:
  - The centrifugal mass is too high for the set acceleration or deceleration time.
  - The torque limit has been reached (load torque > drive torque).
- Remedy: Unload drive or increase torque limit at the servo controller (if the power limits of the inverter have not yet been reached).

The following error is derived from the angular difference of the setpoint angle integrator minus the actual angle integrator. The comparison value (following error limit) can be set in C01071/1.

#### Angle controller overflow (bPosOverflow = TRUE)

The *bPosOverflow* status output is set to TRUE if the angular difference that can be displayed device-internally has been exceeded. Home positions get lost here.

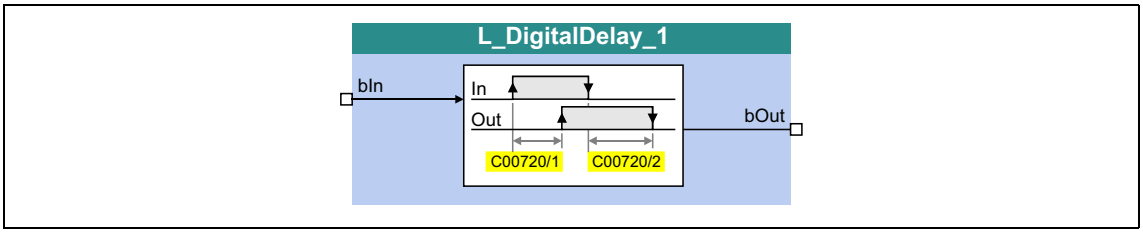
**Tip!**

If an error response is to be triggered in case of a following error and/or angle controller overflow, connect the corresponding status output with a free *bSetError* input of the SB LS_SetError_2 and parameterise the requested error response for this input in C00581.

## 19.1.78 L_DigitalDelay_1

This FB applies a time delay to binary signals.

- ON and OFF-deceleration can be parameterised separately.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input signal |

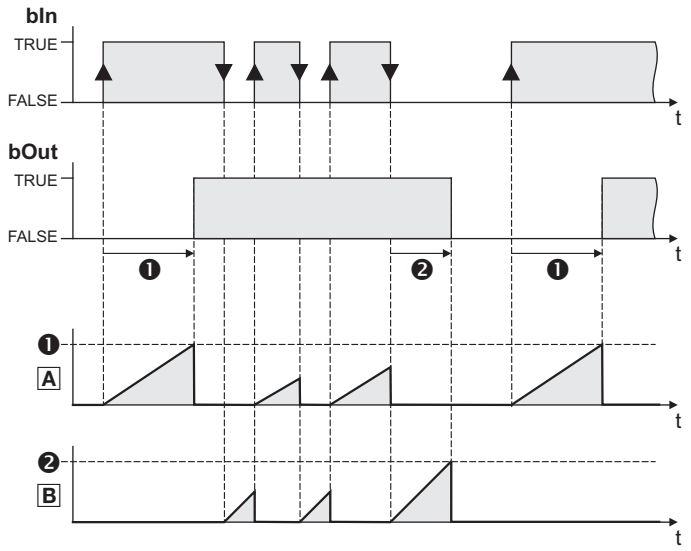
### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal (time-delayed input signal) |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00720/1 | 0.000 | s | 3600.000 | ON-deceleration<br>• Lenze setting: 0.000 s |
| C00720/2 | 0.000 | s | 3600.000 | OFF-deceleration<br>• Lenze setting: 0.000 s |

### Function



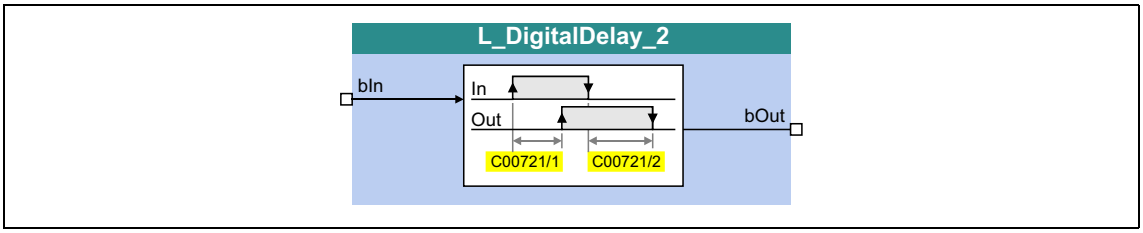

❶ ON delay
❷ OFF delay
A Internal timing element for ON delay (retriggerable)
B Internal timing element for OFF delay (retriggerable)

1. A FALSE-TRUE edge at b*In* starts the internal timing element for the ON delay.
2. After the defined ON delay, the input signal *bIn* is output at *bOut*.
3. A TRUE-FALSE edge at b*In* starts the internal timing element for the OFF delay.
4. After the defined OFF delay, the input signal *bIn* is output at *bOut*.

### 19.1.79 L_DigitalDelay_2

This FB applies a time delay to binary signals.

- ON and OFF-deceleration can be parameterised separately.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal (time-delayed input signal) |

#### Parameters

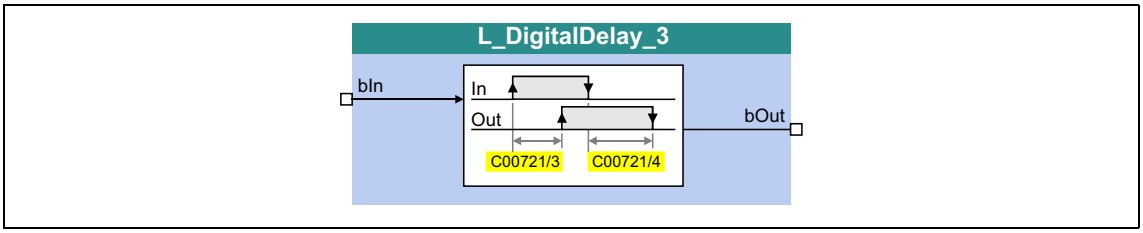
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00721/1 | 0.000 | s | 3600.000 | ON-deceleration<br>• Lenze setting: 0.000 s |
| C00721/2 | 0.000 | s | 3600.000 | OFF-deceleration<br>• Lenze setting: 0.000 s |

For a detailed description see L_DigitalDelay_1.

## 19.1.80 L_DigitalDelay_3

This FB applies a time delay to binary signals.

- ON and OFF-deceleration can be parameterised separately.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input signal |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal (time-delayed input signal) |

### Parameters

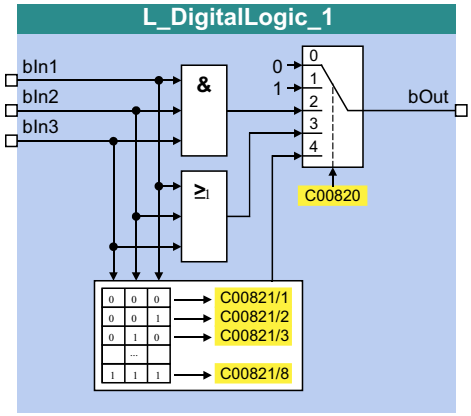
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00721/3 | 0.000 | s | 3600.000 | ON-deceleration<br>• Lenze setting: 0.000 s |
| C00721/4 | 0.000 | s | 3600.000 | OFF-deceleration<br>• Lenze setting: 0.000 s |

For a detailed description see L_DigitalDelay_1.

## 19.1.81 L_DigitalLogic_1

This FB provides a binary output signal created by a logic operation of the input signals. Optionally, one of the constant binary values independent from the input signals can be output.

- Output of a constant binary value
- AND operation of the inputs
- OR operation of the inputs
- Output depending on the combination of the input signals




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>BOOL | Input signal 1 |
| bIn2<br>BOOL | Input signal 2 |
| bIn3<br>BOOL | Input signal 3 |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00820 | | | Function selection |
| | 0 | "0" | Constant value "FALSE" |
| | 1 | "1" | Constant value "TRUE" |
| | 2 | bOut = bIn1 ∧ bIn2 ∧ bIn3 | AND operation |
| | 3 | bOut = bIn1 ∨ bIn2 ∨ bIn3 | OR operation |
| | 4 | bOut = f (truth table) | The output value depends on the truth table parameterised in C00821/1...8 |

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00821/1...8 | | | Truth table for function "4: bOut = f (truth table)"<br>• Each of the eight possible input combinations can be assigned to the output value FALSE or TRUE.<br>• For an application example see the following section. |
| | 0 | FALSE | |
| | 1 | TRUE | |

**Function "4: bOut = f (Truth table)"**

When the function "4: bOut = f (truth table)" is selected in C00820, the output value *bOut* depends on the truth table parameterised in C00821/1...8.

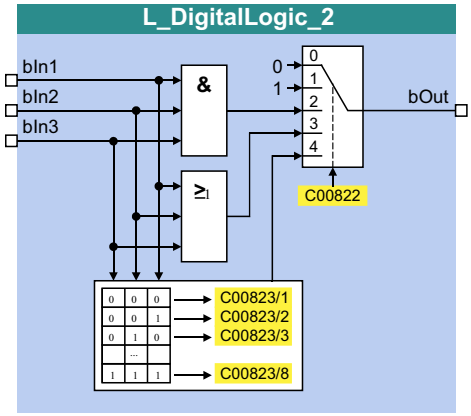
The following table shows which setting may be required in C00821/1...8 to realise the logic operations NAND, NOR, XOR and XNOR:

| Input signals | | | Out | Parameter setting for logic operation: | | | |
|---|---|---|---|---|---|---|---|
| *bIn3* | *bIn2* | *bIn1* | *bOut* | NAND | NOR | XOR | XNOR |
| 0 | 0 | 0 | C00821/1 = | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | C00821/2 = | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | C00821/3 = | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | C00821/4 = | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | C00821/5 = | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | C00821/6 = | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | C00821/7 = | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | C00821/8 = | 0 | 0 | 0 | 1 |

## 19.1.82 L_DigitalLogic_2

This FB provides a binary output signal created by a logic operation of the input signals. Optionally, one of the constant binary values independent from the input signals can be output.

- Output of a constant binary value
- AND operation of the inputs
- OR operation of the inputs
- Output depending on the combination of the input signals




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>BOOL | Input signal 1 |
| bIn2<br>BOOL | Input signal 2 |
| bIn3<br>BOOL | Input signal 3 |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00822 | | | Function selection |
| | 0 | "0" | Constant value "FALSE" |
| | 1 | "1" | Constant value "TRUE" |
| | 2 | bOut = bIn1 ∧ bIn2 ∧ bIn3 | AND operation |
| | 3 | bOut = bIn1 ∨ bIn2 ∨ bIn3 | OR operation |
| | 4 | bOut = f (truth table) | The output value depends on the truth table parameterised in C00823/1...8 |

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00823/1...8 | | | Truth table for function "4: bOut = f (truth table)"<br>• Each of the eight possible input combinations can be assigned to the output value FALSE or TRUE.<br>• For an application example see the following section. |
| | 0 | FALSE | |
| | 1 | TRUE | |

**Function "4: bOut = f (Truth table)"**

When the function "4: bOut = f (truth table)" is selected in C00822, the output value *bOut* depends on the truth table parameterised in C00823/1...8.

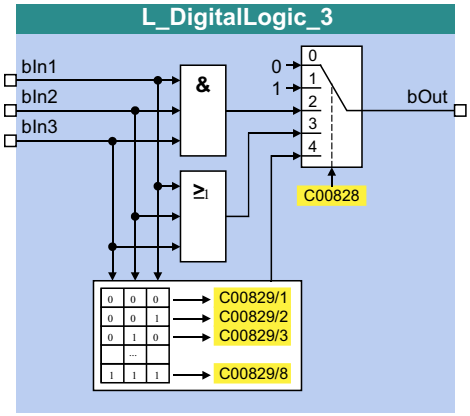
The following table shows which setting may be required in C00823/1...8 to realise the logic operations NAND, NOR, XOR and XNOR:

| Input signals | | | Out | Parameter setting for logic operation: | | | |
|---|---|---|---|---|---|---|---|
| *bIn3* | *bIn2* | *bIn1* | *bOut* | NAND | NOR | XOR | XNOR |
| 0 | 0 | 0 | C00823/1 = | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | C00823/2 = | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | C00823/3 = | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | C00823/4 = | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | C00823/5 = | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | C00823/6 = | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | C00823/7 = | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | C00823/8 = | 0 | 0 | 0 | 1 |

### 19.1.83 L_DigitalLogic_3

This FB provides a binary output signal created by a logic operation of the input signals. Optionally, one of the constant binary values independent from the input signals can be output.

- Output of a constant binary value
- AND operation of the inputs
- OR operation of the inputs
- Output depending on the combination of the input signals




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>BOOL | Input signal 1 |
| bIn2<br>BOOL | Input signal 2 |
| bIn3<br>BOOL | Input signal 3 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00828 | | | Function selection |
| | 0 | "0" | Constant value "FALSE" |
| | 1 | "1" | Constant value "TRUE" |
| | 2 | bOut = bIn1 ∧ bIn2 ∧ bIn3 | AND operation |
| | 3 | bOut = bIn1 ∨ bIn2 ∨ bIn3 | OR operation |
| | 4 | bOut = f (truth table) | The output value depends on the truth table parameterised in C00829/1...8 |

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00829/1...8 | | | Truth table for function "4: bOut = f (truth table)"<br>• Each of the eight possible input combinations can be assigned to the output value FALSE or TRUE.<br>• For an application example see the following section. |
| | 0 | FALSE | |
| | 1 | TRUE | |

**Function "4: bOut = f (Truth table)"**

When the function "4: bOut = f (truth table)" is selected in C00828, the output value *bOut* depends on the truth table parameterised in C00829/1...8.

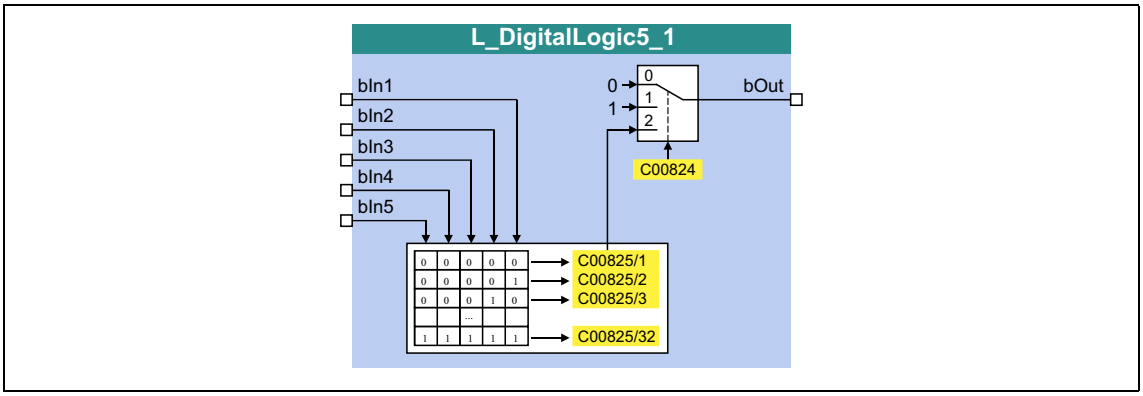
The following table shows which setting may be required in C00829/1...8 to realise the logic operations NAND, NOR, XOR and XNOR:

| Input signals | | | Out | Parameter setting for logic operation: | | | |
|---|---|---|---|---|---|---|---|
| *bIn3* | *bIn2* | *bIn1* | *bOut* | NAND | NOR | XOR | XNOR |
| 0 | 0 | 0 | C00829/1 = | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | C00829/2 = | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | C00829/3 = | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | C00829/4 = | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | C00829/5 = | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | C00829/6 = | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | C00829/7 = | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | C00829/8 = | 0 | 0 | 0 | 1 |

### 19.1.84 L_DigitalLogic5_1

This FB provides a binary output signal created by a logic operation of the input signals. Optionally, one of the constant binary values independent from the input signals can be output.

- Output of a constant binary value
- Output depending on the combination of the input signals




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal 1 ... 5 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00824 | | | Function selection |
| | 0 | "0" | Constant value "FALSE" |
| | 1 | "1" | Constant value "TRUE" |
| | 2 | bOut = f (truth table) | The output value depends on the parameterised truth table |
| C00825 | see truth table | | Truth table<br>Each of the 32 possible input combinations can be assigned to the output value FALSE or TRUE. |

#### Truth table for C00824 = 4

| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | FALSE | C00825/1 (FALSE or TRUE) |
| FALSE | FALSE | FALSE | FALSE | TRUE | C00825/2 (FALSE or TRUE) |

| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|---|---|
| FALSE | FALSE | FALSE | TRUE | FALSE | C00825/3 (FALSE or TRUE) |
| ... | | | | | C00825/... (FALSE or TRUE) |
| TRUE | TRUE | TRUE | FALSE | TRUE | C00825/30 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | TRUE | FALSE | C00825/31 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | TRUE | TRUE | C00825/32 (FALSE or TRUE) |

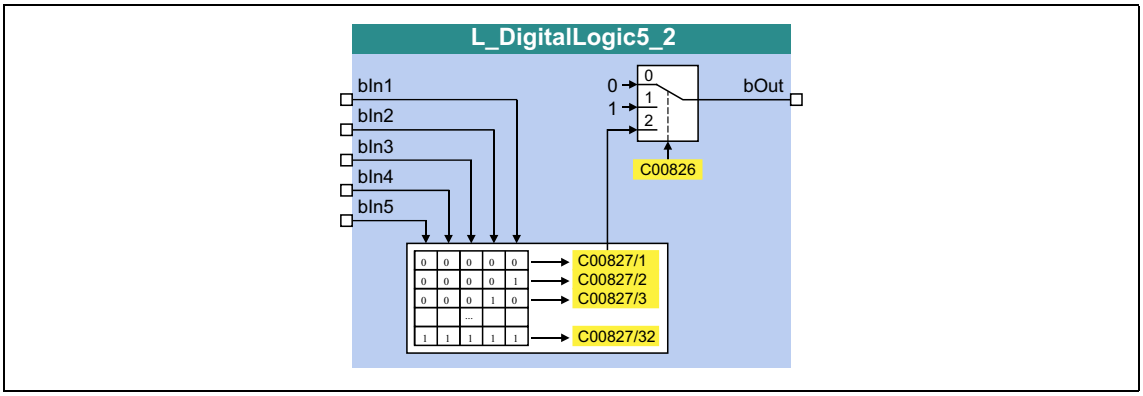
Example: If in case of the signal combination *bIn1* = TRUE, *bIn2* = FALSE, *bIn3* = TRUE, *bIn4* = TRUE and *bIn5* = TRUE, the output signal *bOut* is to be = TRUE, C00825/30 must be set to "TRUE":

| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|---|---|
| TRUE | TRUE | TRUE | FALSE | TRUE | C00825/30 (TRUE) |

## 19.1.85 L_DigitalLogic5_2

This FB provides a binary output signal created by a logic operation of the input signals. Optionally, one of the constant binary values independent from the input signals can be output.

- Output of a constant binary value
- Output depending on the combination of the input signals




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal 1 ... 5 |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00826 | | | Function selection |
| | 0 | "0" | Constant value "FALSE" |
| | 1 | "1" | Constant value "TRUE" |
| | 2 | bOut = f (truth table) | The output value depends on the parameterised truth table |
| C00827 | see truth table | | Truth table<br>Each of the 32 possible input combinations can be assigned to the output value FALSE or TRUE. |

### Truth table for C00826 = 4

| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | FALSE | C00827/1 (FALSE or TRUE) |
| FALSE | FALSE | FALSE | FALSE | TRUE | C00827/2 (FALSE or TRUE) |

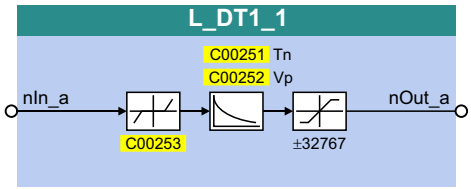
| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|---|---|
| FALSE | FALSE | FALSE | TRUE | FALSE | C00827/3 (FALSE or TRUE) |
| ... | | | | | C00827/... (FALSE or TRUE) |
| TRUE | TRUE | TRUE | FALSE | TRUE | C00827/30 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | TRUE | FALSE | C00827/31 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | TRUE | TRUE | C00827/32 (FALSE or TRUE) |

Example: If in case of the signal combination *bIn1* = TRUE, *bIn2* = FALSE, *bIn3* = TRUE, *bIn4* = TRUE and *bIn5* = TRUE, the output signal *bOut* is to be = TRUE, C00827/30 must be set to "TRUE":

| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|---|---|
| TRUE | TRUE | TRUE | FALSE | TRUE | C00827/30 (TRUE) |

### 19.1.86 L_DT1_1

This FB differentiates signals. The function block can be used for acceleration feedforward (dv/dt), for example.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

#### Outputs

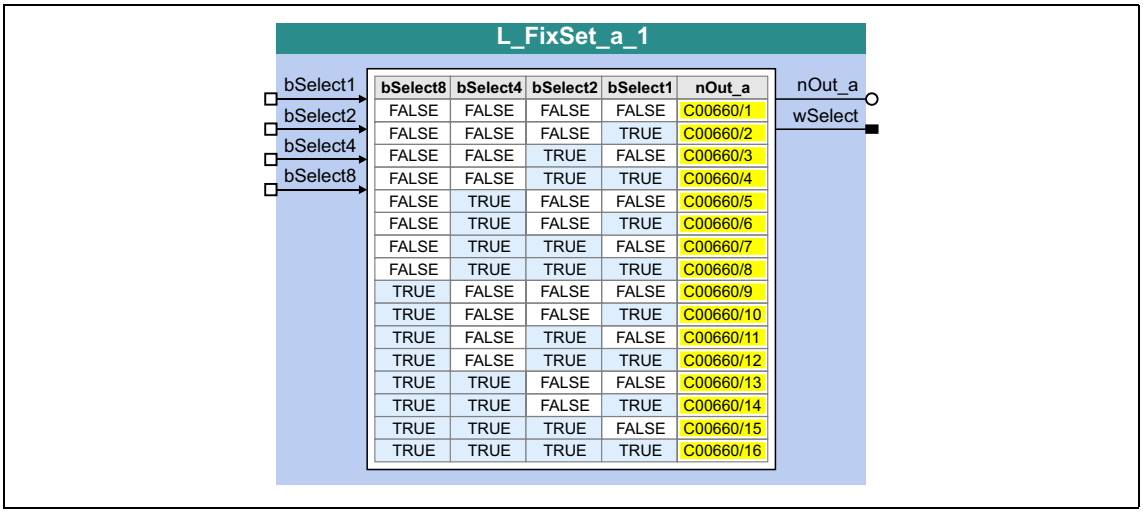
| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00251 | 10 | ms | 5000 | Time constant Tn<br>• Lenze setting: 1000 ms |
| C00252 | -320.00 | | 320.00 | Gain factor Vp<br>• Lenze setting: 1.00 |
| C00253 | **Note:** The most significant bit determines the sign of the value, the remaining bits determine the numerical value. | | | Selection of sensitivity<br>• Depending on the selection, the number of indicated higher-order bits is evaluated.<br>• Lenze setting: 15 bits |
| | 1 | 15 bits | | Bit 0 ... bit 14 are evaluated |
| | 2 | 14 bits | | Bit 0 ... bit 13 are evaluated |
| | 3 | 13 bits | | Bit 0 ... bit 12 are evaluated |
| | 4 | 12 bits | | Bit 0 ... bit 11 are evaluated |
| | 5 | 11 Bit | | Bit 0 ... bit 10 are evaluated |
| | 6 | 10 Bit | | Bit 0 ... bit 9 are evaluated |
| | 7 | 9 Bit | | Bit 0 ... bit 8 are evaluated |

### 19.1.87 L_FixSet_a_1

This FB outputs one of 16 parameterisable analog signals. Binary coded selection of the "fixed value" to be output via the four selection inputs.




| bSelect8 | bSelect4 | bSelect2 | bSelect1 | nOut_a |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | C00660/1 |
| FALSE | FALSE | FALSE | TRUE | C00660/2 |
| FALSE | FALSE | TRUE | FALSE | C00660/3 |
| FALSE | FALSE | TRUE | TRUE | C00660/4 |
| FALSE | TRUE | FALSE | FALSE | C00660/5 |
| FALSE | TRUE | FALSE | TRUE | C00660/6 |
| FALSE | TRUE | TRUE | FALSE | C00660/7 |
| FALSE | TRUE | TRUE | TRUE | C00660/8 |
| TRUE | FALSE | FALSE | FALSE | C00660/9 |
| TRUE | FALSE | FALSE | TRUE | C00660/10 |
| TRUE | FALSE | TRUE | FALSE | C00660/11 |
| TRUE | FALSE | TRUE | TRUE | C00660/12 |
| TRUE | TRUE | FALSE | FALSE | C00660/13 |
| TRUE | TRUE | FALSE | TRUE | C00660/14 |
| TRUE | TRUE | TRUE | FALSE | C00660/15 |
| TRUE | TRUE | TRUE | TRUE | C00660/16 |

#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bSelect1 ... bSelect8<br>BOOL | Binary coded selection of the fixed value to be output<br>• See truth table displayed in the FB above. |

#### Outputs

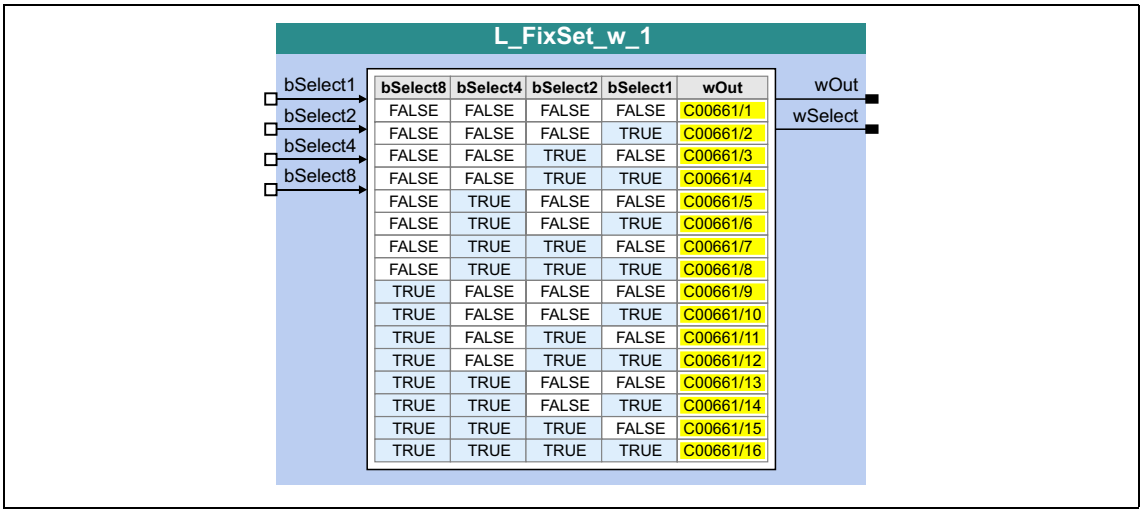
| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output of the selected fixed value |
| wSelect<br>WORD | Current selection (0 ... 15) |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00660/1...16 | -199.99 | % | 199.99 | Fixed value 0 ... 15<br>• Lenze setting: 0.00 % |

## 19.1.88 L_FixSet_w_1

This FB outputs one of 16 parameterisable data words. Binary coded selection of the "fixed value" to be output via the four selection inputs.




| bSelect8 | bSelect4 | bSelect2 | bSelect1 | wOut |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | C00661/1 |
| FALSE | FALSE | FALSE | TRUE | C00661/2 |
| FALSE | FALSE | TRUE | FALSE | C00661/3 |
| FALSE | FALSE | TRUE | TRUE | C00661/4 |
| FALSE | TRUE | FALSE | FALSE | C00661/5 |
| FALSE | TRUE | FALSE | TRUE | C00661/6 |
| FALSE | TRUE | TRUE | FALSE | C00661/7 |
| FALSE | TRUE | TRUE | TRUE | C00661/8 |
| TRUE | FALSE | FALSE | FALSE | C00661/9 |
| TRUE | FALSE | FALSE | TRUE | C00661/10 |
| TRUE | FALSE | TRUE | FALSE | C00661/11 |
| TRUE | FALSE | TRUE | TRUE | C00661/12 |
| TRUE | TRUE | FALSE | FALSE | C00661/13 |
| TRUE | TRUE | FALSE | TRUE | C00661/14 |
| TRUE | TRUE | TRUE | FALSE | C00661/15 |
| TRUE | TRUE | TRUE | TRUE | C00661/16 |

### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bSelect1 ... bSelect8<br>BOOL | Binary coded selection of the fixed value to be output<br>• See truth table displayed in the FB above. |

### Outputs

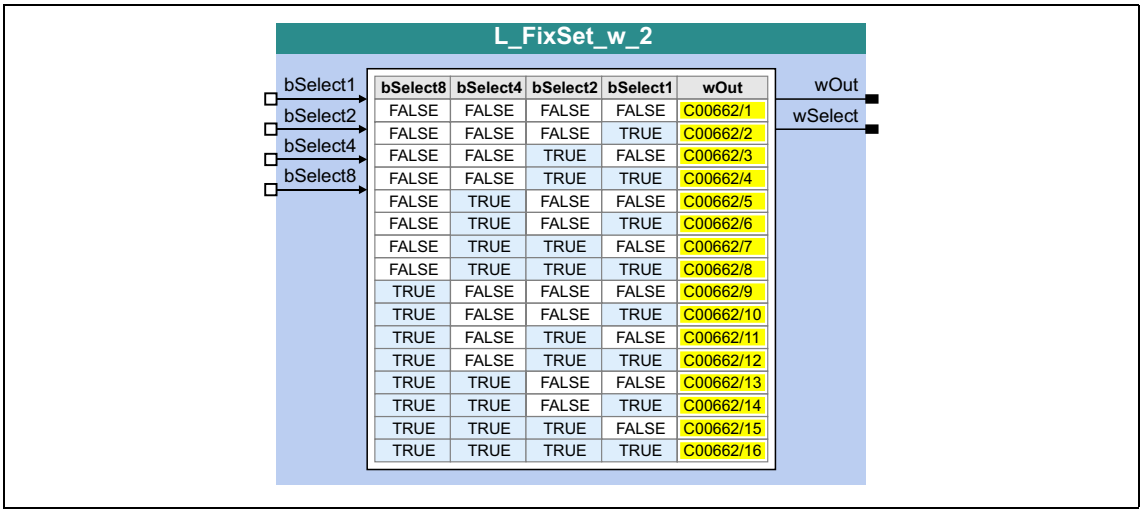
| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output of the selected fixed value |
| wSelect<br>WORD | Current selection (0 ... 15) |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00661/1...16 | 0 | | 65535 | Fixed values 0 ... 15<br>• Lenze setting: 0 |

### 19.1.89 L_FixSet_w_2

This FB outputs one of 16 parameterisable data words. Binary coded selection of the "fixed value" to be output via the four selection inputs.




| bSelect8 | bSelect4 | bSelect2 | bSelect1 | wOut |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | C00662/1 |
| FALSE | FALSE | FALSE | TRUE | C00662/2 |
| FALSE | FALSE | TRUE | FALSE | C00662/3 |
| FALSE | FALSE | TRUE | TRUE | C00662/4 |
| FALSE | TRUE | FALSE | FALSE | C00662/5 |
| FALSE | TRUE | FALSE | TRUE | C00662/6 |
| FALSE | TRUE | TRUE | FALSE | C00662/7 |
| FALSE | TRUE | TRUE | TRUE | C00662/8 |
| TRUE | FALSE | FALSE | FALSE | C00662/9 |
| TRUE | FALSE | FALSE | TRUE | C00662/10 |
| TRUE | FALSE | TRUE | FALSE | C00662/11 |
| TRUE | FALSE | TRUE | TRUE | C00662/12 |
| TRUE | TRUE | FALSE | FALSE | C00662/13 |
| TRUE | TRUE | FALSE | TRUE | C00662/14 |
| TRUE | TRUE | TRUE | FALSE | C00662/15 |
| TRUE | TRUE | TRUE | TRUE | C00662/16 |

#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bSelect1 ... bSelect8<br>BOOL | Binary coded selection of the fixed value to be output<br>• See truth table displayed in the FB above. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output of the selected fixed value |
| wSelect<br>WORD | Current selection (0 ... 15) |

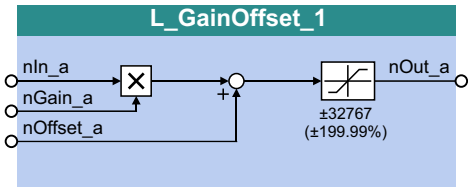
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00662/1...16 | 0 | | 65535 | Fixed values 0 ... 15<br>• Lenze setting: 0 |

## 19.1.90 L_GainOffset_1

This FB can amplify an analog input signal and add an offset to it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

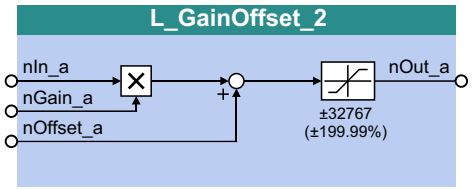
### Function

$$nOut_a = (nIn_a \cdot Gain\ factor) + Offset$$

## 19.1.91 L_GainOffset_2

This FB can amplify an analog input signal and add an offset to it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

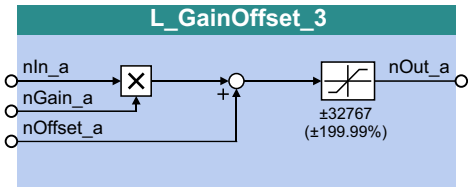
### Function

$$nOut_a = (nIn_a \cdot Gain\ factor) + Offset$$

### 19.1.92 L_GainOffset_3

This FB can amplify an analog input signal and add an offset to it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

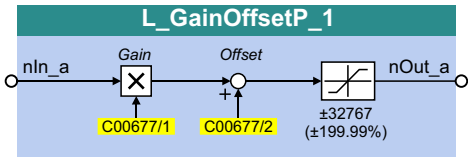
#### Function

$$nOut_a = (nIn_a \cdot Gain\ factor) + Offset$$

## 19.1.93 L_GainOffsetP_1

This FB can amplify an analog input signal and add an offset to it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00677/1 | -199.99 | % | 199.99 | Gain factor<br>• Lenze setting: 100.00 %<br>• 199.99 % ≈2 |
| C00677/2 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

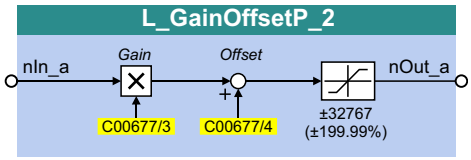
### Function

$$nOut_a = (nIn_a \cdot Gain\ factor) + Offset$$

## 19.1.94 L_GainOffsetP_2

This FB can amplify an analog input signal and add an offset to it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00677/3 | -199.99 | % | 199.99 | Gain factor<br>• Lenze setting: 100.00 %<br>• 199.99 % ≈2 |
| C00677/4 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

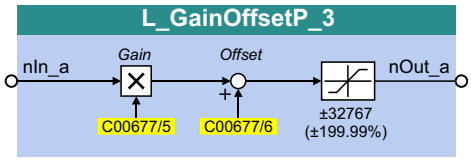
### Function

$$nOut_a = (nIn_a \cdot Gain\ factor) + Offset$$

## 19.1.95 L_GainOffsetP_3

This FB can amplify an analog input signal and add an offset to it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00677/5 | -199.99 | % | 199.99 | Gain factor<br>• Lenze setting: 100.00 %<br>• 199.99 % ≈2 |
| C00677/6 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

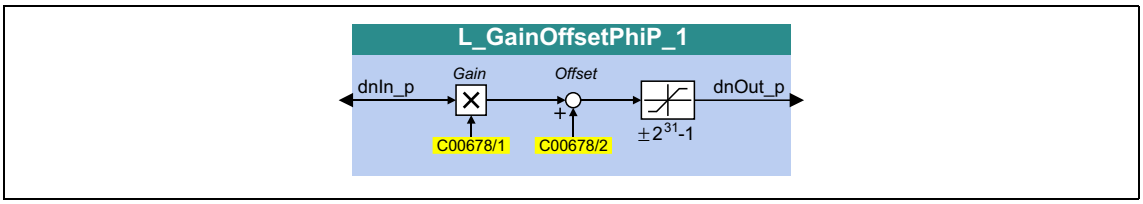
### Function

$$nOut_a = (nIn_a \cdot Gain\ factor) + Offset$$

## 19.1.96 L_GainOffsetPhiP_1

This FB can amplify an angle signal and add an offset to it afterwards.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via parameters.
- The value provided at the *dnOut_p* output is internally limited to $\pm 2^{31}-1$.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}-1$ (±2147483647) |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00678/1 | -2147483647 | Incr. | 2147483647 | Offset<br>• Lenze setting: 0 incr. |
| C00678/2 | -2147483647 | | 2147483647 | Gain factor<br>• Lenze setting: 65536 |

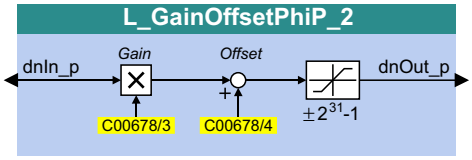
### Function

$$dnOut_p = (dnIn_p \cdot \text{Gain factor}) + \text{Offset}$$

## 19.1.97 L_GainOffsetPhiP_2

This FB can amplify an angle signal and add an offset to it afterwards.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via parameters.
- The value provided at the *dnOut_p* output is internally limited to $\pm 2^{31}$-1.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

### Parameters

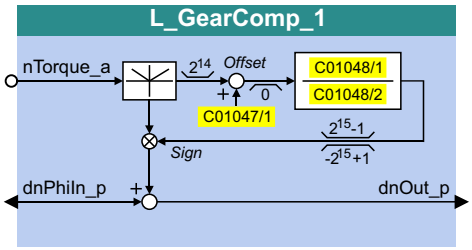
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00678/3 | -2147483647 | Incr. | 2147483647 | Offset<br>• Lenze setting: 0 incr. |
| C00678/4 | -2147483647 | | 2147483647 | Gain factor<br>• Lenze setting: 65536 |

### Function

$$dnOut_p = (dnIn_p \cdot \text{Gain factor}) + \text{Offset}$$

## 19.1.98 L_GearComp_1

This FB is used for dynamic compensation of elasticities in the drive train that arise, e..g. by elastic coupling of speed reduction gearboxes or long transmission shafts.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nTorque_a<br>INT | Torque demand<br>• 16384 ≡ 100 % |
| dnPhiIn_p<br>DINT | Angle setpoint<br>• 65536 increments ≡ 1 motor revolution |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Corrected angle setpoint<br>• 65536 increments ≡ 1 motor revolution |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01047/1 | -16383 | | 16383 | Offset<br>• Static torque offset<br>• Lenze setting: 0 |
| C01048/1 | -32767 | | 32767 | Counter<br>• Elastic constant<br>• Lenze setting: 1 |
| C01048/2 | 1 | | 32767 | Denominator<br>• Elastic constant<br>• Lenze setting: 1 |

**Function**

Elasticity is a measure of how far the load have moved from the ideal setpoint position at motor standstill due to mechanical force effect.

- Example: "Hoist":
  Due to elasticity of the mechanical transmission elements, the real position of the "hook" varies in loaded and unloaded status.
- In order to compensate errors caused by elasticity, the actual torque at the *nTorque_a* input is applied. This torque is a measure for the current load.
- The multiplication by an elasticity factor results in an angle compensation value which is added to the setpoint angle correctly signed depending on the direction of the torque. This serves to correct the false position of the load.
- The elasticity factor is selected in the form of numerator and denominator via C01048/1 and C01048/2.
- C01047/1 serves to select a static correction value (offset).

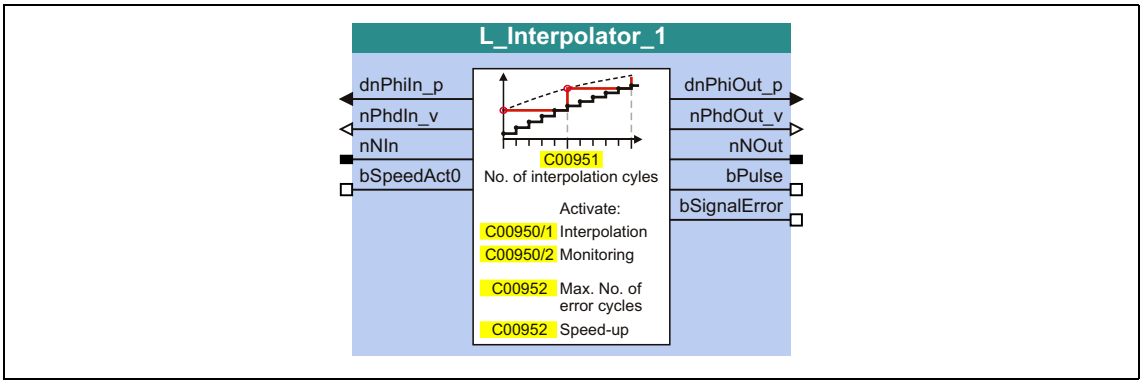


**Note!**

Please note that the compensation may be reversed in case of a signed numerator/denominator selection of the elasticity factor!

## 19.1.99 L_Interpolator_1

This FB interpolates a position setpoint and/or an analog value e.g. to compensate for larger bus transmission cycles or to continue signal characteristics if data telegrams are missing.




### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnPhIn_p<br>DINT | Position setpoint<br>• Is interpolated and completed when signal interpolation is activated. | |
| nPhdIn_v<br>INT | Angular velocity<br>• Is only passed through to the *nPhdOut_v* output. | |
| nNIn<br>INT | Analog value<br>• Is interpolated when signal interpolation is activated. | |
| bSpeedAct0<br>BOOL | Input for detecting the "Current speed is zero" status<br>• This status signal needs to be transmitted by the setpoint source to ensure trouble-free operation. | |
| | TRUE | The current speed is zero. The interpolator in the follow-up controller is kept to the *dnPhiIn_p* input.<br>• This function is for instance required for referencing the axis. Since in this case, the axis can move independently of the master position, the follow-up controller is not allowed to intervene. |

### Outputs

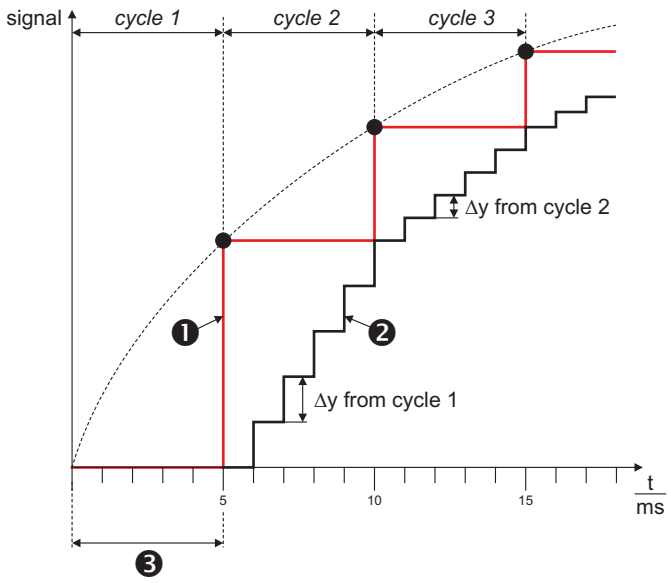
| Designator<br>Data type | Value/meaning | |
|---|---|---|
| dnPhOut_p<br>DINT | Output of the *dnPhIn_p* position setpoint which, if applicable, has been interpolated and completed | |
| nPhdOut_v<br>INT | Output of the *nPhdIn_v* angular velocity | |
| nNOut<br>INT | Output of the *nNIn* analog value which, if applicable, has been interpolated | |
| bPulse<br>BOOL | "Input values have been accepted" status signal | |
| | TRUE | The input values have been accepted during this cycle. |
| bSignalError<br>BOOL | "Signal error" status signal<br>• Only if signal monitoring is active (C00950/2 = "1: On"). | |
| | TRUE | The number of missing data telegrams has exceeded the limit value parameterised in C00952. |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00950/1 | | | | Signal interpolation of the *dnPhIn_p* and *nNIn* input signals<br>• Lenze setting: Off<br>▸ Signal interpolation (📖 1583) |
| | 0 | Off | | |
| | 1 | On | | |
| C00950/2 | | | | Signal monitoring of the *dnPhIn_p* input signal<br>• Lenze setting: Off<br>▸ Signal monitoring (📖 1584) |
| | 0 | Off | | |
| | 1 | On | | |
| C00951 | 1 | | 65535 | Number of interpolation steps<br>• Corresponds to the receive cycle of the data telegrams in [ms].<br>• Lenze setting: 1 |
| C00952 | 0 | | 65535 | Limit value for missing data telegrams<br>• Lenze setting: 5<br>▸ Signal monitoring (📖 1584) |
| C00953 | 0 | | 100 | Speed-up<br>• Limitation of the amount of correction increments per cycle<br>• Scaling: 1 increment/ms ≡ 0.9155 rpm |

## 19.1.99.1 Signal interpolation

If signal interpolation is active (C00950/1 = 1), the output signal will not reach the level of the corresponding input signal until all interpolation steps parameterised in C00951 have been performed:




❶ Setpoint signal
❷ Interpolated setpoint signal
❸ Receive cycle in [ms] = Number of interpolation steps (in our example: 5)

[19-41] Signal characteristic

**Note!**

Do not change the number of interpolation steps during operation. Otherwise the interpolation becomes inaccurate.

### 19.1.99.2 Signal monitoring

If signal monitoring is active (C00950/2 = 1), the signal characteristic of the *dnPhIn_p* input signal is continued even if the data telegram is missing (setpoint selection via CAN).

Monitoring is performed on the basis of the *dnPhIn_p* position setpoint and the *bSpeedAct0* status signal:

- If the *dnPhIn_p* position setpoint remains the same in the next device cycle, it is either because the speed is zero or because no data telegram has been received.
- The evaluation of the *bSpeedAct0* status signal gives information about which reason applies. This status signal needs to be transmitted by the setpoint source to ensure trouble-free operation:
  - *bSpeedAct0* = FALSE means that the speed is not zero, so an error is assumed: The signal characteristic of the *dnPhIn_p* input signal is completed (the current slope is retained).
  - *bSpeedAct0* = TRUE means that the speed is zero, so the unchanged position setpoint is not treated as an error.
- If the number of missing data telegrams exceeds the limit value parameterised in C00952, the *bSignalError* output is set to TRUE.
  - The *bSignalError* output is automatically reset to FALSE if correct signals are detected at *dnPhIn_p* and *bSpeedAct0* again.

**Note!**

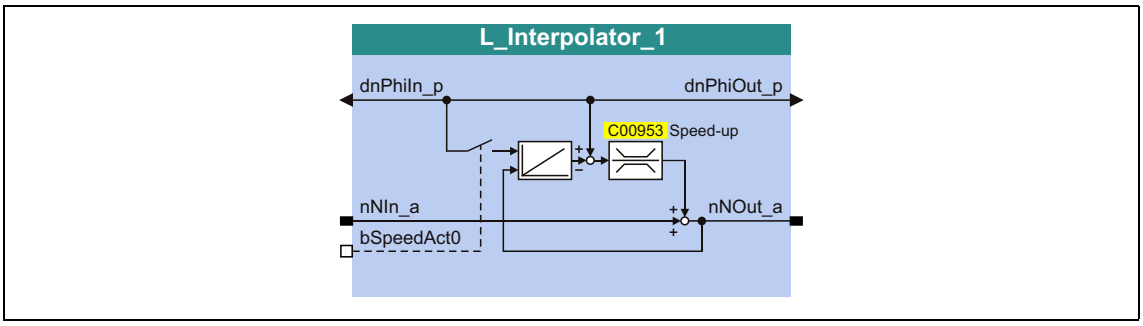
The analog value *nNIn* is not monitored!

### 19.1.99.3 Angle correction in case of transmission errors

If an angular offset between master and slave is caused due to transmission errors (missing data telegrams), it will be corrected by a catch-up function in the FB. For this purpose, the following connections and parameter settings are required for the slave at the FB **L_Interpolator_1**:

1. The master angle of the master is connected to the *dnPhiIn_p* input.
2. The speed signal of the master is connected to the *nNIn* input.
3. The *nNOut* output is connected to the *nSet_v* input of the FB L_DFSET_1.

The following illustration shows the principle of the catch-up function in the FB **L_Interpolator_1**:




[19-42] Principle of the catch-up function

The speed signal at *nNIn* is provided almost 1:1 at the *nNOut* output. In case of a telegram error, a correction value can be added to the signal. This correction value results from the subtraction of the integrated speed signal from the position value applied at *dnPhiIn_p*.

If, for example, a data telegram should fail, the input values remain constant for one program cycle. During the next cycle, the correct position and the correct speed are restored.

"Holding" the position at *dnPhiIn_p* results in a difference between the position values at *dnPhiIn_p* and at the output of the integrator. This difference is added to the *nNOut_a* output signal.

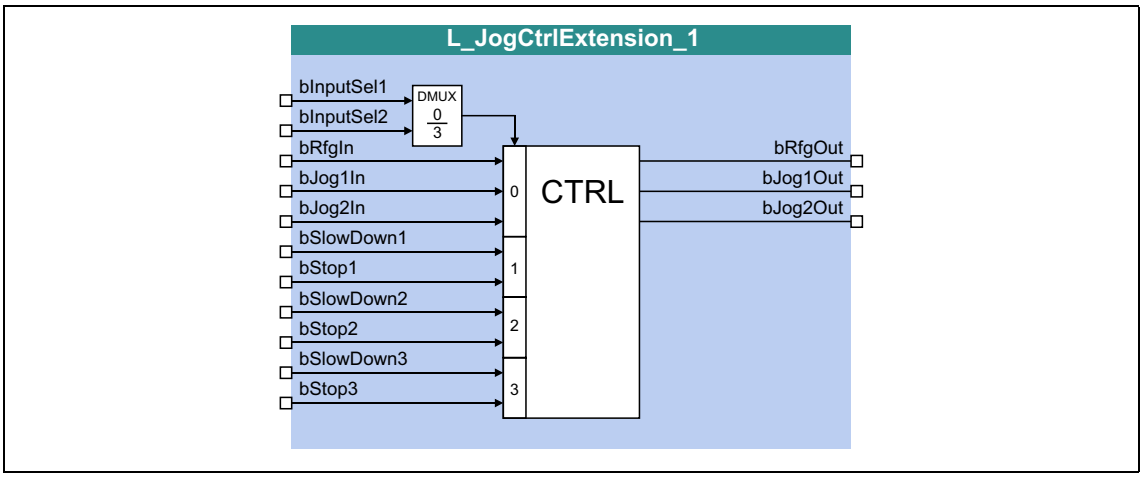
In order that this angle correction does not lead to a strong jerk in the master value, the amount of correction increments is limited per cycle (catch-up cycle) using C00953. A typical correction value is for instance 10 increments/ms.

When the controller is inhibited, the integrator is to be loaded with the position value pending at *dnPhiIn_p* by setting the *bSpeedAct0* input to TRUE. When the controller is enabled, the *nNOut* speed signal is integrated.

### 19.1.100 L_JogCtrlExtension_1

This FB can be connected upstream to the L_NSet ramp function generator/setpoint generator to implement a switch-off positioning at limit switch.

- Detailed information on this operating mode can be found in the description
  ▸ TA "Switch-off positioning" (📖 527).




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bInputSel1<br>bInputSel2<br>BOOL | Activation of the *bSlowDown1/bStop1*, *bSlowDown2/bStop2* and *bSlowDown3/bStop3* signal pairs according to the Truth table |
| bRfgIn<br>BOOL | Ramping down of the setpoint generator in the downstream L_NSet FB according to the Truth table |
| bJog1In<br>bJog2In<br>BOOL | Selection inputs for setting fixed speeds in the setpoint generator<br>• If the pre-switch off is inactive (*bInputSel1* and *bInputSel2* are both set to FALSE), the two control signals are output one-to-one at the *bJog1Out* and *bJog2Out* outputs.<br>• To achieve the desired behaviour (starting at high speed, pre-switch off at low speed), both inputs must be set to TRUE.<br>• Fixed setpoint 2 must be less than fixed setpoint 3!<br>Otherwise, the drive will start at a low speed and accelerate after the pre-switch off.<br>• If, in addition to the *bJog1In* and *bJog2In* inputs, other jog signals are set at the L_NSet FB, new fixed setpoints are reached, and the drive traverses at speeds that differ from the selection via *bJog1In* and *bJog2In*. |
| bSlowDown1<br>bSlowDown2<br>bSlowDown3<br>BOOL | Activation of fixed setpoint 2 in the downstream L_NSet FB<br>• These inputs only fulfil a function if they have been activated via *bInputSel1* and *bInputSel2* previously (see Truth table). |
| bStop1<br>bStop2<br>bStop3<br>BOOL | Ramping down of the ramp function generator in the downstream L_NSet FB<br>• These inputs only fulfil a function if they have been activated via *bInputSel1* and *bInputSel2* previously (see Truth table). |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bRfgOut<br>BOOL | Control signal for ramping down the setpoint generator<br>• Connect this output to the *bRfg0* input of the L_NSet FB. |
| bJog1Out<br>BOOL | Control signal for setting fixed speeds in the setpoint generator<br>• Connect this output to the *bJog1* input of the L_NSet FB. |
| bJog2Out<br>BOOL | Control signal for setting fixed speeds in the setpoint generator<br>• Connect this output to the *bJog2* input of the L_NSet FB. |

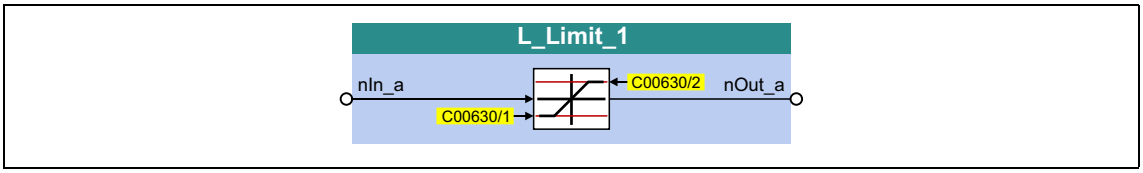
### Truth table

| Input | | Function | Response in the L_NSet FB |
|---|---|---|---|
| bInputSel1 | bInputSel2 | | |
| FALSE | FALSE | Pre-switch off inactive | No response<br>• The *bRfgIn* input signal is directly provided at the *bRfgOut* output.<br>• The *bJogIn1* and *bJogIn2* input signals are directly output at the *bJog1Out* and *bJog2Out* outputs. |
| TRUE | FALSE | The *bSlowDown1* and *bStop1* inputs are evaluated. | **Pre-switch off can be activated**<br>• If the SlowDown function is activated via the selected *bSlowDown* input, fixed setpoint 2 in the setpoint generator is activated via the *bJog1Out* and *bJog2Out* outputs.<br>• If the Stop function is activated via the selected *bStop* input, the *bRfgOut* output is set to TRUE and hence the setpoint generator is deactivated. |
| FALSE | TRUE | The *bSlowDown2* and *bStop2* inputs are evaluated. | |
| TRUE | TRUE | The *bSlowDown3* and *bStop3* inputs are evaluated. | |

[19-1] Truth table for activating the pre-switch off

### 19.1.101 L_Limit_1

This FB limits an analog input signal to a value range whose upper and lower limit can be set via parameters.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Scaling: 16384 ≡ 100 % |

#### Parameters

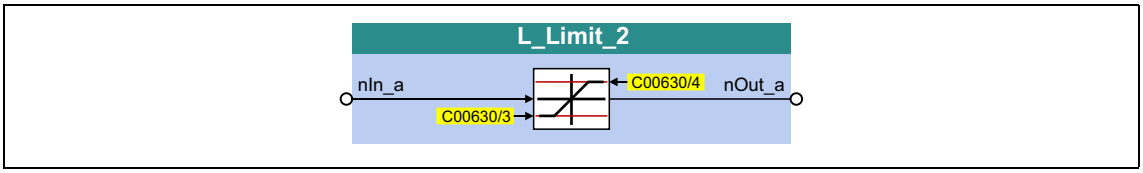
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00630/1 | -199.99 | % | 199.99 | Lower limit<br>• Lenze setting: -100.00 % |
| C00630/2 | -199.99 | % | 199.99 | Upper limit<br>• Lenze setting: 100.00 % |

**Tip!**

Always set the lower limit lower than the upper limit, otherwise value "0" is provided at the *nOut_a* output.

### 19.1.102 L_Limit_2

This FB limits an analog input signal to a value range whose upper and lower limit can be set via parameters.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Scaling: 16384 ≡ 100 % |

#### Parameters

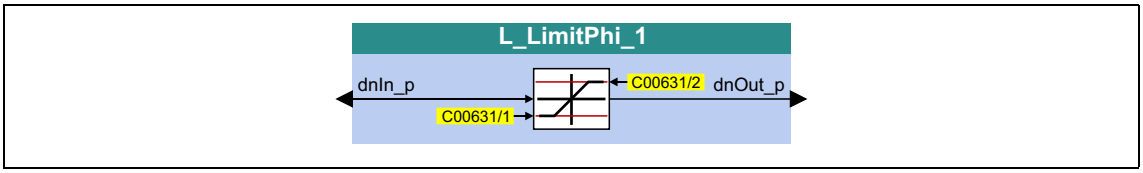
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00630/3 | -199.99 | % | 199.99 | Lower limit<br>• Lenze setting: -100.00 % |
| C00630/4 | -199.99 | % | 199.99 | Upper limit<br>• Lenze setting: 100.00 % |

**Tip!**

Always set the lower limit lower than the upper limit, otherwise value "0" is provided at the *nOut_a* output.

### 19.1.103 L_LimitPhi_1

This FB restricts an angle signal to one value range, whose upper and lower limit can be set via parameters.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>INT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>INT | Output signal |

#### Parameters

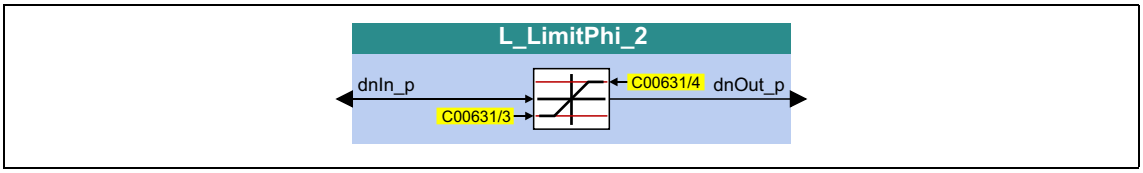
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00631/1 | -2147483647 | Incr. | 2147483647 | Lower limit<br>• Lenze setting: -2147483647 incr. |
| C00631/2 | -2147483647 | Incr. | 2147483647 | Upper limit<br>• Lenze setting: 2147483647 incr. |

**Tip!**

Always set the lower limit lower than the upper limit, otherwise value "0" is provided at the *dnOut_p* output.

### 19.1.104 L_LimitPhi_2

This FB restricts an angle signal to one value range, whose upper and lower limit can be set via parameters.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>INT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>INT | Output signal |

#### Parameters

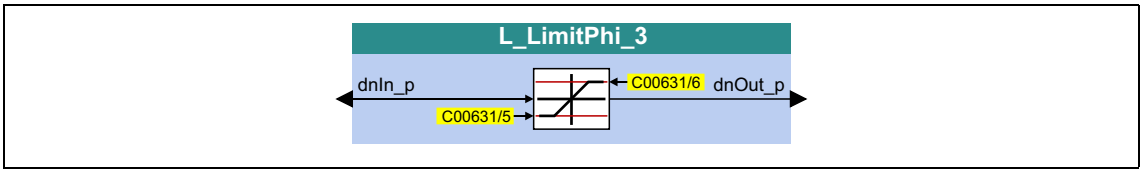
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00631/3 | -2147483647 | Incr. | 2147483647 | Lower limit<br>• Lenze setting: -2147483647 incr. |
| C00631/4 | -2147483647 | Incr. | 2147483647 | Upper limit<br>• Lenze setting: 2147483647 incr. |

**Tip!**

Always set the lower limit lower than the upper limit, otherwise value "0" is provided at the *dnOut_p* output.

## 19.1.105 L_LimitPhi_3

This FB restricts an angle signal to one value range, whose upper and lower limit can be set via parameters.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>INT | Input signal |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>INT | Output signal |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00631/5 | -2147483647 | Incr. | 2147483647 | Lower limit<br>• Lenze setting: -2147483647 incr. |
| C00631/6 | -2147483647 | Incr. | 2147483647 | Upper limit<br>• Lenze setting: 2147483647 incr. |

**Tip!**

Always set the lower limit lower than the upper limit, otherwise value "0" is provided at the *dnOut_p* output.

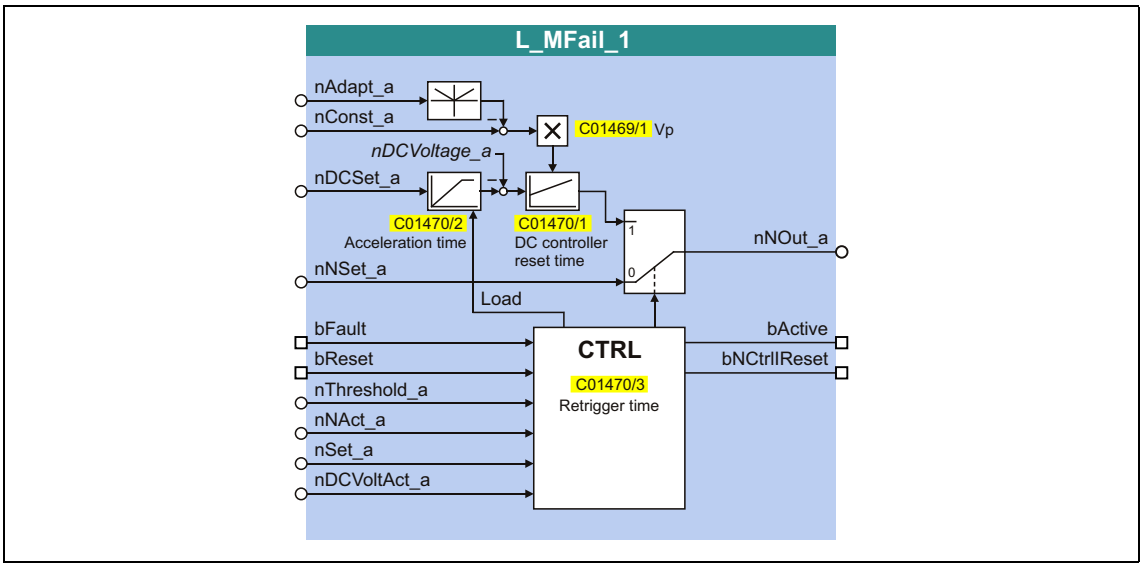
### 19.1.106 L_MFail_1

In case the supply voltage fails, this FB serves to stop (brake) the drive or drive system in a controlled way via L1, L2, L3 or +UG, -UG to prevent coasting.

**Stop!**

For drive systems that are coupled via digital frequency (a master drive and one or several slave drives) the following applies:

- The mains failure detection and control may only be activated for the master drive.
- All inverters must be operated via the terminals +UG, -UG in the DC bus connection.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nAdapt_a<br>INT | Dynamic adaptation of the proportional gain of the DC-bus voltage controller<br>• The resulting proportional gain results from:<br>$Vp = C01469/1 \cdot \frac{nConst_a\,[\%] - \lvert nAdapt_a\,[\%]\rvert}{100\,\%}$<br>• Scaling: 16384 ≡ 100 % |
| nConst_a<br>INT | |
| nDCSet_a<br>INT | Voltage setpoint on which the DC-bus voltage is to be kept.<br>• Scaling: 16384 ≡ 1000 V |
| nNset_a<br>INT | Speed setpoint in [%]<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| bFault<br>BOOL | Activating the mains failure control<br>TRUE: Activate mains failure control. |
| bReset<br>BOOL | Reset mains failure control<br>• A reset is always required when the restart protection is active or the restart protection is used and the supply (mains or DC supply) has been switched on.<br>TRUE: Reset mains failure control. |
| nThreshold_a<br>INT | Restart threshold in [%]<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |

| Designator<br>Data type | Information/possible settings |
|---|---|
| nNAct_a<br>INT | Comparison value for the restart threshold in [%]<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| nSet_a<br>INT | Speed starting point for the deceleration in [%]<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| nDCVoltAct_a<br>INT | Actual DC-bus voltage<br>• Scaling: 16384 ≡ 1000 V |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nNOut_a<br>INT | Speed setpoint in [%]<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| bActive<br>BOOL | Status signal "Mains failure control active"<br>TRUE: The mains failure control is active. |
| bNCtrlIReset<br>BOOL | Control signal for resetting the I component of the speed controller in the motor control<br>• Connect this output with the *bSpeedCtrlIOn* input of the SB LS_MotionControlKernel (if this is connected upstream to the motor control).<br>TRUE: Reset I component of the speed controller of the motor control. |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01469/1 | 0.001 | | 31.000 | Vp<br>• Lenze setting: 1.000 |
| C01470/1 | 0 | ms | 60000 | Vdc-bus controller reset time<br>• Lenze setting: 20 ms |
| | **Note:** The set value is internally limited to 20 ... 2000 ms!<br>From version 14.00.00 onwards, the limited value is written back to the code if the setting is outside the limits. | | | |
| C01470/2 | 0 | ms | 60000 | Acceleration time<br>• Lenze setting: 20 ms |
| | **Note:** The set value is internally set to 1 ... 16000 ms!<br>From version 14.00.00 onwards, the limited value is written back to the code if the setting is outside the limits. | | | |
| C01470/3 | 0 | ms | 60000 | Retrigger time<br>• Lenze setting: 20 ms |

### 19.1.106.1 Procedure of the mains failure control

A failure of the voltage supply of the power section can be detected by

- an evaluation of the DC-bus voltage
  and/or
- an external monitoring system (e.g. voltage measuring relay).

The type of mains failure control to be used depends on the used drive system.

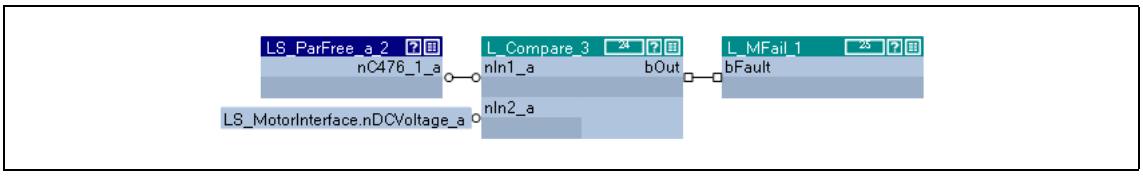
**Note!**

The following interconnection examples are not functional yet. For an error-free function, connect the FB **L_MFail_1** with further signals!

#### Evaluation of the DC-bus voltage

This proceeding is used for single drives or multi-axis drives which do not use an external monitoring system.

- For evaluating the DC-bus voltage, you can use a comparator (e.g. FB L_Compare_3) as shown in the following interconnection example 1.
  - In order that the mains failure control will be activated if the DC-bus voltage falls below a certain operating threshold, set the comparison function "2: In1 > In2" has to be set for the FB L_Compare_3 in C00690.
  - For specifying the operating threshold, the SB LS_ParFree_a_2 is used. This system block can output 16 parameterisable analog signals. In the shown example, the operating threshold has to be set in the "free parameter" C00476/1.

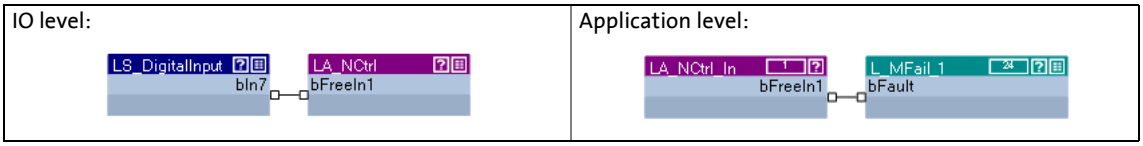



[19-43] Interconnection example 1: Evaluation of the DC-bus voltage with a comparator (cutout)

#### Use of an external monitoring system

In this proceeding, the digital status signal of an external monitoring system is connected to the FB **L_MFail_1** via a digital input of the inverter.

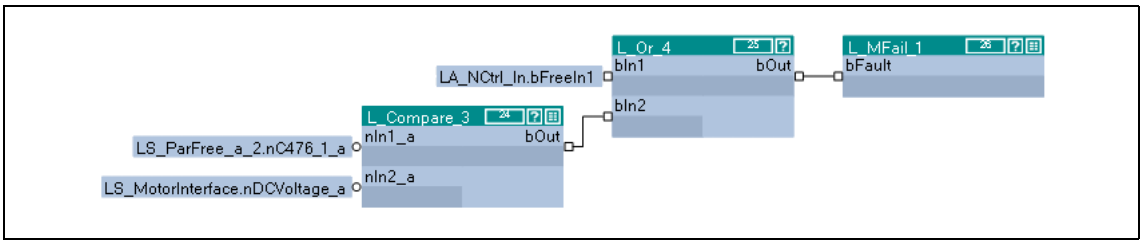
- In the following example, the digital input DI7 is used.
  - The active level (HIGH or LOW active) for DI7 has to be parameterised in C00114 in such a way that *bFault* becomes TRUE when the monitoring system is tripped.
  - A free input of the application block can be used to transfer the digital input signal from I/O level to application level.




[19-44] Interconnection example 2: Use of the digital status signal of an external monitoring system

**Combination of the above shown proceedings**

The combination of the proceedings can be simply realised via an OR operation:



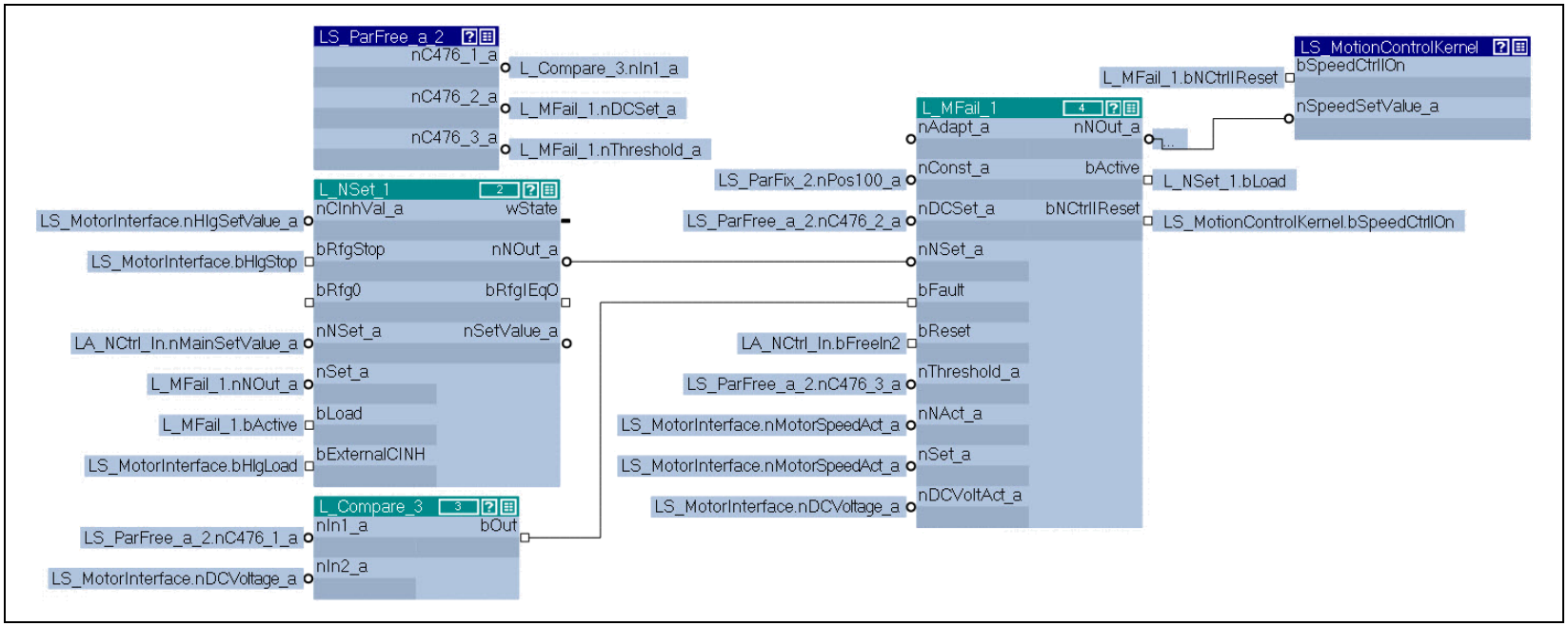

[19-45] Interconnection example 3: Mains failure control of different sources

### 19.1.106.2 Signal flow for mains failure control

The following illustration shows the basic signal flow (cutout) for a mains failure control.

**Note!**

The following interconnection example is not functional yet. For an error-free function, connect e.g. the FB L_NSet_1 with further signals!




[19-46] Signal flow (detail)

| L_MFail inputs | Function |
|---|---|
| nAdapt_a | Dynamic adaptation of the proportional gain of the DC-bus voltage controller.<br>• Input not interconnected ➞ Adaptation = 0 %. |
| nConst_a | Proportional gain of the DC-bus voltage controller<br>• Input with *nPos100_a* constant interconnected ➞ proportional gain = 100 %. |
| nDCSet_a | Voltage setpoint on which the DC-bus voltage is to be kept.<br>• The voltage setpoint has to be set in the "free parameter" C00476/2. |
| nNset_a | Speed setpoint path (output: nNOut_a) |
| bFault | Activating the mains failure control<br>▸ Procedure of the mains failure control (📖 1595) |
| bReset | Reset mains failure control<br>• The free *bFreeIn2* input of the application block used in the example can be e.g. connected to a digital input on I/O level. |
| nThreshold_a | Restart threshold<br>• The restart threshold (at first approx 2 %) has to be set in the "free parameter" C00476/3. |
| nNAct_a | Comparison value for the restart threshold<br>• Comparison value is the actual speed value |
| nSet_a | Speed starting point for the deceleration<br>• Speed starting point is the actual speed value |
| nDCVoltAct_a | Actual DC-bus voltage |

### 19.1.106.3 Activating the mains failure control

The mains failure control is activated by setting *bFault* to TRUE. The inverter now generates the required operational energy from the rotational energy of the drive. The drive is braked via the power loss of the inverter and the motor. Thus, the speed deceleration ramp is shorter than for an uncontrolled system (coasting drive).

**After activating the mains failure control:**

1. The acceleration time set in C01470/2 is used to control the DC-bus voltage to the value at *nDCSet_a*.
2. At *nNOut_a*, an internally generated speed setpoint is output which serves to brake the drive to a speed close to "0" (via the speed setpoint).
   - Starting value for the controlled deceleration is the value at *nSet_a*. This input is appropriately connected to the actual speed value *nMotorSpeedAct_a* of the SB LS_MotorInterface.
   - The speed deceleration ramp (and hence the braking torque) results from the moment of inertia of the load machine(s), the power loss of the drive (system) and the set parameter setting.




**Stop!**

If a connected braking unit responds, the drive is braked with a max. possible torque ($I_{max}$). In this case, adapt the parameter setting if required (see the following chapter).

If the power section is not supplied, the drive cannot create a standstill torque (important for active loads as e.g. hoists).

**Stopping/cancelling the mains failure control**

When *bFault* is reset to FALSE, an internal timing element is triggered. After the time set in C01470/3 has elapsed, the mains failure control is stopped/cancelled.

- When the restart protection is active, the drive is continued to be braked to standstill. ▸ Restart protection (📖 1603)
  - When the restart protection is active, the drive can only be reset by setting the *bReset* input to TRUE.
- When the restart protection not active, the drive is accelerated to the speed setpoint. ▸ Quick mains recovery (KU) (📖 1603)

### 19.1.106.4 Parameterising the mains failure control

The parameters to be set before initial commissioning strongly depend on the used motor, the moment of inertia of the drive and the drive configuration (single drive, drive system, master/slave operation etc.) and must hence be adapted to the corresponding application case.

**Note!**

The following data refers to the chapter "Procedure of the mains failure control". (📖 1595)

1. Measure the DC-bus voltage with an oscilloscope (channel 1).
   Option 1: With a corresponding voltage divider at the terminals +UG, -UG.
   Option 2: At the analog output terminals X3/O1U and X3/GA
   - For this purpose, adapt the signal flow in order that the DC-bus voltage is provided via the analog output 1: Go to the application level and connect the *nDCVoltage_a* signal of the SB LS_MotorInterface to a free output of the application block (e.g. *wFreeOut1*). Then connect this output in the IO level to the *nOut1V_a* input of the SB LS_AnalogOutput.
2. Measure the speed with an oscilloscope (channel 2) at the analog output terminals X3/O2U and X3/GA.
   - For this purpose, adapt the signal flow in order that the speed is provided via the analog output 2: Go to the application level and connect the *nMotorSpeedAct_a* signal of the SB LS_MotorInterface to a free output of the application block (e.g. *wFreeOut2*). Then connect this output in the IO level to the *nOut2V_a* input of the SB LS_AnalogOutput.
3. Set the operating threshold for the mains failure control in C00476/1.
   - Scaling: 100 % ≡ 1000 V
   - Recommended setting: approx. 50 V above the reset threshold for undervoltage in the DC bus ($LU_{off}$).
   - The operating thresholds for undervoltage (LU) and overvoltage (OU) depend on the setting in C00173:

| Mains voltage | | Undervoltage (LU) | | Overvoltage (OU) | |
|---|---|---|---|---|---|
| 1-phase | C00173 | Operating threshold $LU_{on}$ | Reset threshold $LU_{off}$ | Operating threshold $OU_{on}$ | Reset threshold $OU_{off}$ |
| $230V_{AC}$ | 0, 1, 2, 3 | 180 $V_{DC}$ | 240 $V_{DC}$ | 400 $V_{DC}$ | 390 $V_{DC}$ |

| Mains voltage | | Undervoltage (LU) | | Overvoltage (OU) | |
|---|---|---|---|---|---|
| 3-phase | C00173 | Operating threshold $LU_{on}$ | Reset threshold $LU_{off}$ | Operating threshold $OU_{on}$ | Reset threshold $OU_{off}$ |
| $400V_{AC}$ | 0 | 285 $V_{DC}$ | 430 $V_{DC}$ | 800 $V_{DC}$ | 790 $V_{DC}$ |
| $440V_{AC}$ | 1 | 400 $V_{DC}$ | 430 $V_{DC}$ | | |
| $480V_{AC}$ | 2 | 490 $V_{DC}$ | 535 $V_{DC}$ | | |
| $500V_{AC}$ | 3 | 540 $V_{DC}$ | 585 $V_{DC}$ | | |

4. Set the voltage setpoint in C00476/2 the DC-bus voltage is to be controlled to.
   - Scaling: 100 % ≡ 1000 V
   - Recommended setting: approx. 700 V → C00476/2 = 70 %

**Stop!**

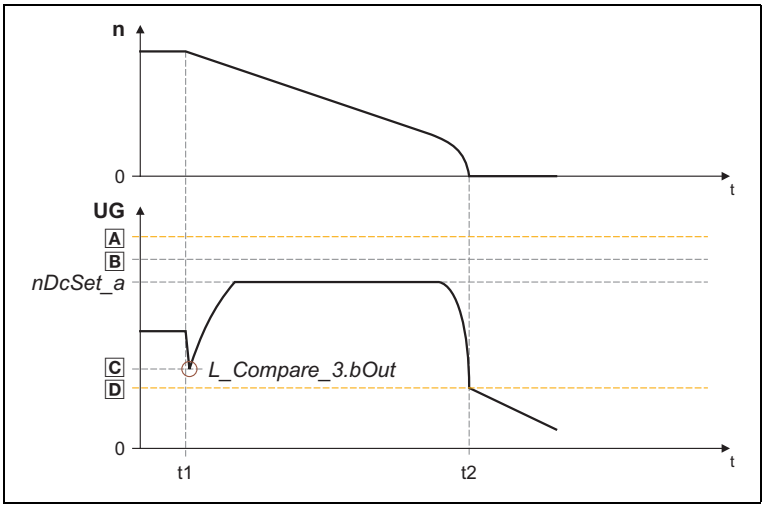
The voltage setpoint must be below the operating threshold of a braking unit if connected. If a connected braking unit responds, the drive with max. possible torque ($I_{max}$) will be braked. The desired operational performance gets lost.

### 19.1.106.5 Commissioning the mains failure control

Commissioning should be executed with motors without load.

1. Start the drive.
2. Set the acceleration time:
   - Set the speed setpoint to 100 %, operate the inverter with max. speed.
   - Set controller inhibit and measure deceleration time up to standstill.
   - Set approx. 1/10 of the measured deceleration time in C01470/2.
3. Set retrigger time:
   - In case of mains failure control via detection of the DC-bus voltage level:
     Set the deceleration time measured under point 2 in C01470/3.
   - In case of mains failure control via an external system:
     Set the time in which the drive is to be continued to be brought to a standstill under control at short-term mains recovery in C01470/3.
4. Switch off the supply voltage (mains or DC bus).

The following profile is to be shown on the oscilloscope:




n = speed of the drive

A Operating threshold OU
B Switch-on threshold - braking unit
C Operating threshold - mains failure
D Operating threshold LU

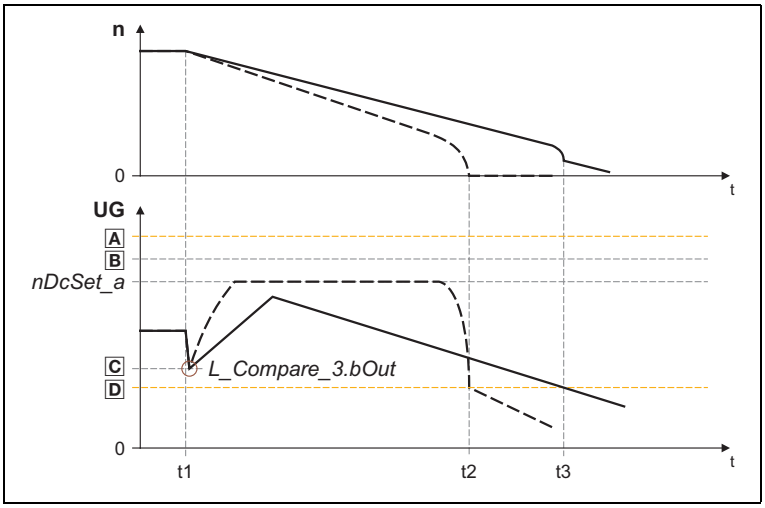
[19-47] Schematic diagram with activated mains failure control (ideal profile)

| Point in time | Info |
|---|---|
| t = t1 | Mains failure |
| t = t2 | Speed "0" reached. |

### 19.1.106.6 Fine adjustment of the mains failure control

For the fine adjustment, repeat the following points several times:

1. A final speed as low as possible has to be reached before the inverter reaches the operating threshold for undervoltage (LU):
   - Increase proportional gain Vp in C01469/1 .
   - Reduce Vdc-bus controller reset time in C01470/1.
2. The responding of the braking unit or the reaching of the operating threshold for overvoltage (OU) has to be prevented:
   - Increase Vdc controller reset time in C01470/1 until the ideal profile shown in [19-47] will be almost reached.
   - If required, reduce the *nDCSet_a* voltage setpoint to which the DC-bus voltage is to be controlled (in the interconnection example via the free parameter C00476/2).
3. Increasing the deceleration time or reducing the braking torque is only possible to a limited extent:
   - Increasing the acceleration time in C01470/2 reduces the initial braking torque and simultaneously increases the deceleration time.
   - Increasing the Vdc controller reset time in C01470/1 reduces the braking torque and simultaneously increases the deceleration time. If the reset times are too high, the inverter reaches the operating threshold for undervoltage (LU) before standstill is reached. Hence, the drive is not controlled anymore.




[19-48] Schematic diagram with different braking torques

| Point in time | Info |
|---|---|
| t = t1 | Mains failure |
| t = t2 | Speed "0" with higher braking torque reached (short reset time). |
| t = t3 | With a lower braking torque (higher reset time), the drive reaches the operating threshold for undervoltage (LU) without reaching speed "0". |
| t > t3 | Drive is not controlled anymore (is braked by friction). |

### 19.1.106.7 Reset mains failure control

The mains failure control is rest by setting *bReset* to TRUE.

- A reset via *bReset* is always required when the restart protection is active or the restart protection is used and the supply (mains or DC supply) has been switched on.
- The *bReset* input can, for instance, be connected to a digital input.

### 19.1.106.8 Restart protection

The integrated restart protection is to prevent a restart in the lower speed range when the supply voltage was only interrupted briefly (mains recovery before drive stands still).

- The *nThreshold_a* input serves to specify the restart threshold in [%] with regard to the reference speed (C00011) below which no start has to take place after mains recovery.
  - If in case of mains recovery the speed is below the restart threshold, the drive is continued to be braked in a controlled way. This function is only stopped by setting *bReset* to TRUE.
  - If the speed is above the restart threshold after mains recovery, the drive changes to its setpoint *nNSet_a*.
  - The function is switched inactive when *nThreshold_a* = 0 %.
- A reset takes place by setting *bReset* to TRUE and is required after every mains switching.
  - In this case, the *bActive* output is set to TRUE when the *bFault* input is set to FALSE.

### 19.1.106.9 Quick mains recovery (KU)

The quick mains recovery causes a restart of the inverter if the restart protection is not active anymore.

- In case, the drive reaches its setpoint.
- If this behaviour is not desired, you can delay the restart by setting a retrigger time in C01470/3 or prevent it by using the restart protection.
- Dependent on the system, the mains recovery is reported through the mains failure control via the level of the DC-bus voltage. ▸ Procedure of the mains failure control (📖 1595)

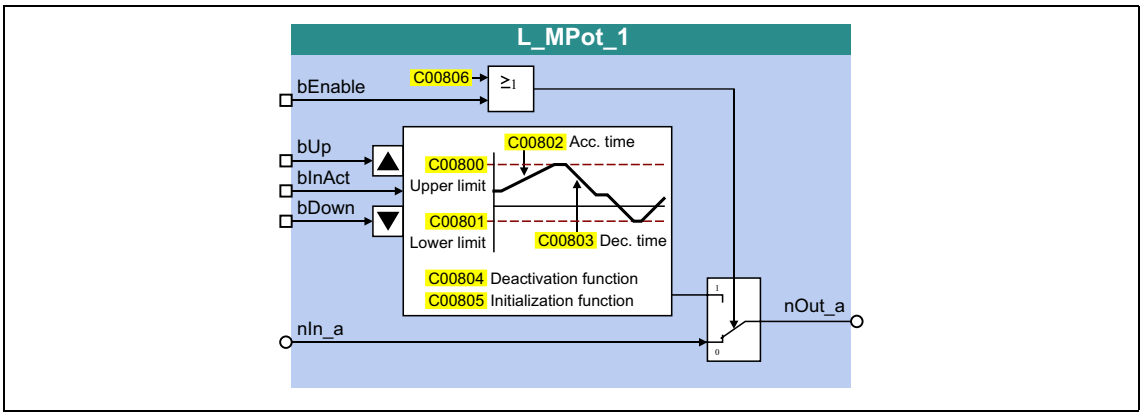
A quick mains recovery is caused by a "quick interruption" of the energy supply company (e.g. thunderstorm) and by faulty components in the supply cables (e.g. collector rings).

- Then, set the retrigger time in C01470/3 higher than the measured deceleration time reached in braking operation.

### 19.1.107 L_MPot_1

This FB replaces a hardware motor potentiometer and can be used as an alternative setpoint source controlled via two inputs.

- The signal is output via a ramp function generator with linear ramps.
- The acceleration and deceleration times are set via parameters.
- Constant ramping even with speed limit values changed online.
- The motor potentiometer function can be switched on/off online.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bEnable<br>BOOL | Switch over motor potentiometer function<br>*bEnable* input and C00806 code are ORed. | |
| | TRUE | Motor potentiometer function is active, setpoint can be changed via *bUp* and *bDown*.<br>• With switching to TRUE, the value applied to *nIn_a* is automatically transferred to the motor potentiometer. |
| | FALSE | The value applied to *nIn_a* is output at *nOut_a*. |
| nIn_a<br>INT | When bEnable = FALSE, the analog nIn_ is input signal switched to the nOut_a output. | |
| bUp<br>BOOL | Approaching of the upper speed limit value set in C00800. | |
| | TRUE | The *nOut_a* output signal runs to its upper limit value (*nHighLimit*).<br>• If the *bDown* input is simultaneously set to TRUE, the *nOut_a* output signal is not changed. |
| bDown<br>BOOL | Approaching of the lower speed limit value set in C00801. | |
| | TRUE | The *nOut_a* output signal runs to its lower limit value (*nLowLimit*).<br>• If the *bUp* input is simultaneously set to TRUE, the *nOut_a* output signal is not changed. |
| bInAct<br>BOOL | Deactivate motor potentiometer function<br>• This input has the highest priority.<br>• When the motor potentiometer is deactivated, the *nOut_a* output signal follows the function set with code C00804. | |
| | TRUE | Motor potentiometer function is deactivated. |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

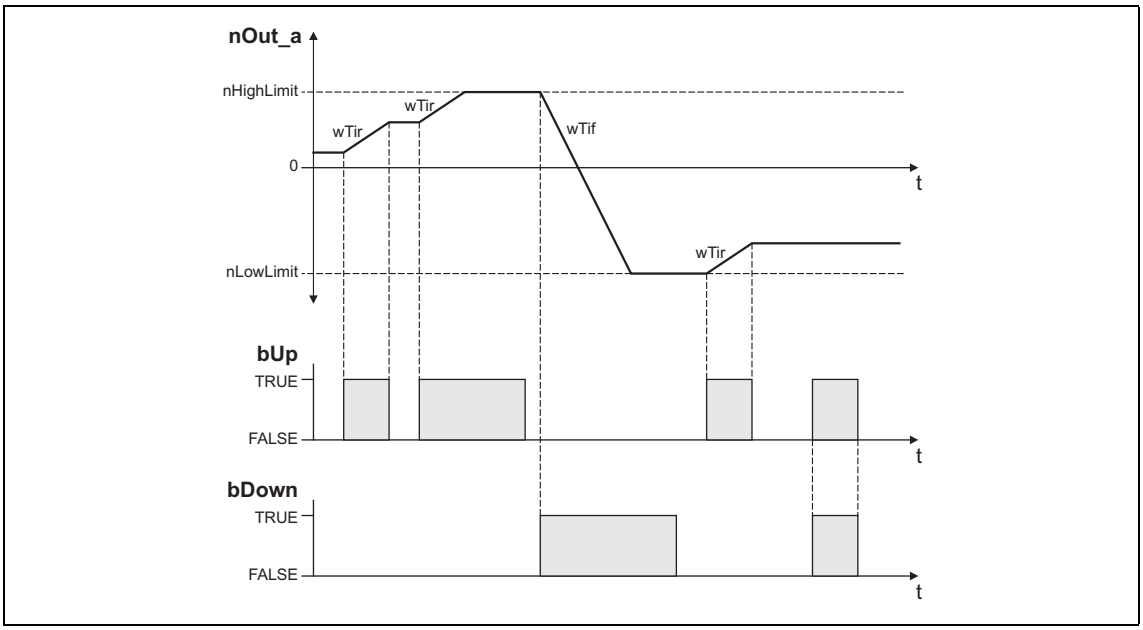
### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00800 | -199.99 | % | 199.99 | Upper limit<br>• Lenze setting: 100.00 % |
| C00801 | -199.99 | % | 199.99 | Lower limit<br>• Lenze setting: -100.00 % |
| C00802 | 0.1 | s | 6000.0 | Acceleration time<br>• Lenze setting: 10.0 s |
| C00803 | 0.1 | s | 6000.0 | Deceleration time<br>• Lenze setting: 10.0 s |
| C00804 | | | | Inactive function<br>• Selection of response when deactivating the motor potentiometer via the input *bInAct*.<br>• Lenze setting: 0 |
| | 0 | No further action; *nOut_a* retains its value. | | |
| | 1 | The motor potentiometer returns to 0 % within the deceleration time $T_{if}$ | | |
| | 2 | The motor potentiometer runs to the lower limit value (C00801) within the deceleration time $T_{if}$ | | |
| | 3 | The motor potentiometer output immediately changes to 0 % | | **Important for the emergency stop function** |
| | 4 | The motor potentiometer output immediately changes to the lower limit value (C00801) | | |
| | 5 | The motor potentiometer runs to the upper limit value (C00800) within the acceleration time $T_{ir}$ | | |
| C00805 | | | | Init function<br>• Selection of response when switching on the device.<br>• Lenze setting: 0 |
| | 0 | The output value being output during mains power-off is saved non-volatilely in the internal memory of the inverter. It will be reloaded during mains power-on. | | |
| | 1 | The lower limit value (C00801) is loaded during mains power-on. | | |
| | 2 | An output value = 0 % is loaded during mains power-on. | | |
| C00806 | | | | Use of the motor potentiometer<br>• When switching to 1: YES, the value applied to *nIn_a* is automatically transferred to the motor potentiometer.<br>• Lenze setting: 0 |
| | 0 | No | | |
| | 1 | Yes | | |

### 19.1.107.1 Activate & control motor potentiometer

When *bInAct* is set to FALSE, the motor potentiometer is activated.

- The currently active function depends on the current output signal *nOut_a*, the limit values set and the control signals at *bUp* and *bDown*.
- When the *nOut_a* output signal is outside the limits set, the output signal runs to the next limit with the Ti times set. This process is independent of the control signals at *bUp* and *bDown*.
- When the *nOut_a* output signal is inside the limits set, the output signal changes according to the control signals at *bUp* and *bDown*.

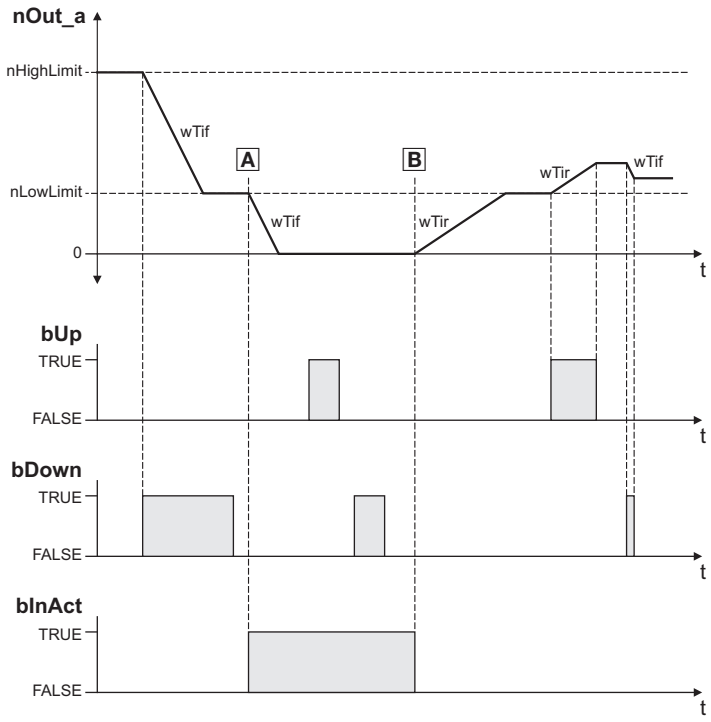



[19-49] Example: Control of the motor potentiometer

| bUp | bDown | bInact | Function |
|---|---|---|---|
| FALSE | FALSE | FALSE | The *nOut_a* output signal remains unchanged. |
| TRUE | FALSE | | The *nOut_a* output signal runs to its upper limit value (*nHighLimit*). |
| FALSE | TRUE | | The *nOut_a* output signal runs to its lower limit value (*nLowLimit*). |
| TRUE | TRUE | | The *nOut_a* output signal remains unchanged. |
| - | - | TRUE | The motor potentiometer function is deactivated. The *nOut_a* output signal responds according to the function selected via *Function*. |

### 19.1.107.2 Deactivate motor potentiometer

When the motor potentiometer is deactivated by setting *bInAct* to TRUE, the *nOut_a* output signal responds according to the function selected via *Function*.



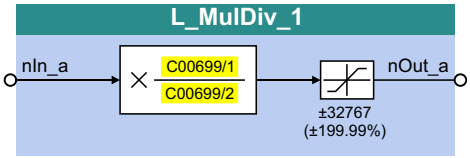

A When the motor potentiometer is deactivated, *nOut_a* is reduced to 0 % with the deceleration time *wTif*.
B If it is activated again, *nOut_a* is increased again to *nLowLimit* with the acceleration time *wTir*.

[19-50] Example: Deactivation of the motor potentiometer when the *Function* = 1 has been selected

### 19.1.108 L_MulDiv_1

This FB multiplies the analog input signal with a parameterisable factor.

- The value of the factor is determined by a quotient consisting of numerator and denominator .
- The value output at *nOut_a* is limited to ±199.99 %.
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1<br>INT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Product value (result of the multiplication)<br>• Internal limitation to ± 32767 |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00699/1 | -32767 | | 32767 | Counter |
| C00699/2 | -32767 | | 32767 | Denominator |

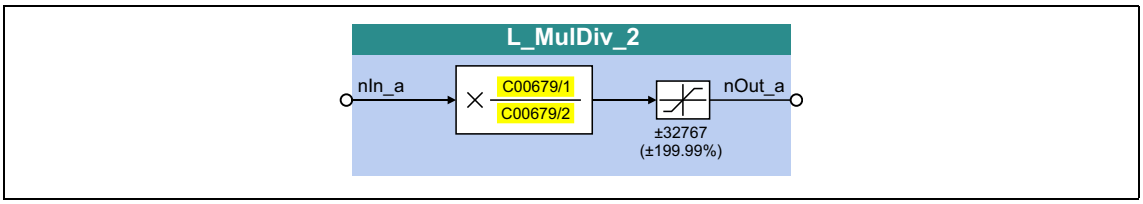
#### Function

$$nOut_a = nIn_a \times \frac{C00699/1}{C00699/2}$$

### 19.1.109 L_MulDiv_2

This FB multiplies the analog input signal with a parameterisable factor.

- The value of the factor is determined by a quotient consisting of numerator and denominator .
- The value output at *nOut_a* is limited to ±199.99 %.
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn1<br>INT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Product value (result of the multiplication)<br>• Internal limitation to ± 32767 |

#### Parameters

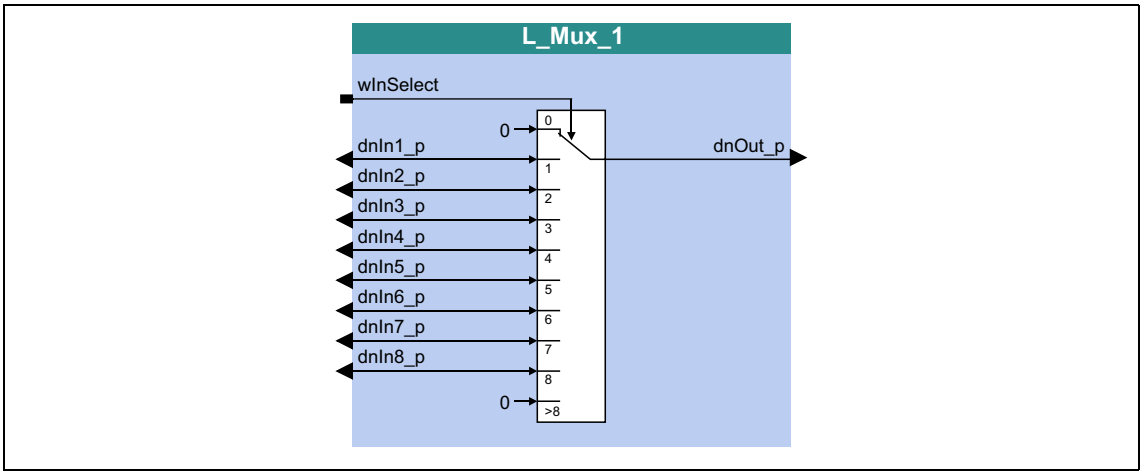
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00679/1 | -32767 | | 32767 | Counter |
| C00679/2 | -32767 | | 32767 | Denominator |

#### Function

$$nOut_a = nIn_a \times \frac{C00679/1}{C00679/2}$$

### 19.1.110 L_Mux_1

This FB provides one of the eight input signals *dnIn1_p* ... *dnIn8_p* at the output *dnOut_p*. The selection is made by means of the signal at the input *wInSelect*.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wInSelect<br>WORD | Input signal | |
| | 1 ... 8 | • The values "1" ... "8" select the input signal to be applied to the output.<br>• Values from"1" ... "8" set the *dnOut_p* output to "0". |
| dnIn1_p<br>DINT<br>...<br>dnIn8_p<br>DINT | Input signal | |

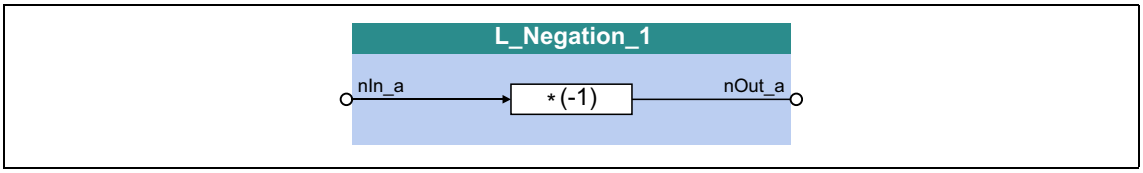
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

### 19.1.111 L_Negation_1

This FB converts the sign of the input signal, i.e. the input signal is multiplied by the value -1 and is then output.

- With the value - 32768 at the *nIn_a* input, the value + 32767 is provided at the *nOut_a* output.




#### Inputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nIn_a<br>INT | Input signal |

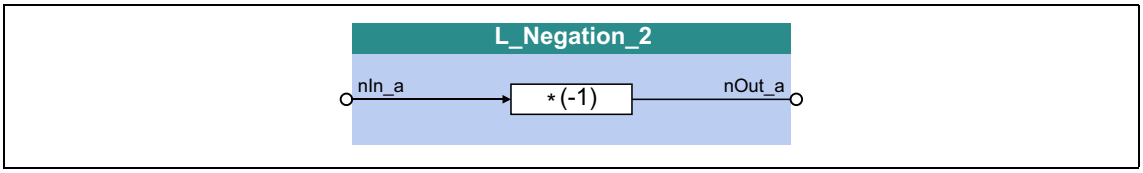
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 19.1.112 L_Negation_2

This FB converts the sign of the input signal, i.e. the input signal is multiplied by the value -1 and is then output.

- With the value - 32768 at the *nIn_a* input, the value + 32767 is provided at the *nOut_a* output.




#### Inputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nIn_a<br>INT | Input signal |

#### Outputs

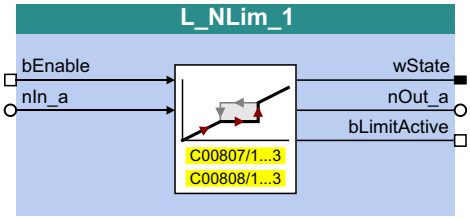
| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 19.1.113 L_NLim_1

This FB can hide up to three parameterisable blocking zones within one continuous signal characteristic.

**Note!**

A value of "0" cannot be masked out if the *nIn_a* input signal changes signs.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bEnable<br>BOOL | Activate zone masking | |
| | FALSE | The FB has been deactivated.<br>• The input signal is output one-to-one at the *nOut_a* output. |
| | TRUE | The FB has been activated.<br>• Zone masking of the input signal is carried out according to the parameterised blocking zones. |
| nIn_a<br>INT | Input signal | |

#### Outputs

| Identifier/data type | Value/meaning | |
|---|---|---|
| wState<br>WORD | Bit-coded status word<br>• Bits that are not listed are reserved for future extensions. | |
| | Bit 0 | No blocking zone active |
| | Bit 1 | Blocking zone 1 active |
| | Bit 2 | Blocking zone 2 active |
| | Bit 3 | Blocking zone 3 active |
| nOut_a<br>INT | Output signal<br>• If the FB has been activated, the output signal is outside the blocking zones. | |
| bLimitActive<br>BOOL | "Limitation active" status signal | |
| | TRUE | The input signal is inside a blocking zones and is limited to the respective boundary value of the blocking zone. |

### Parameters

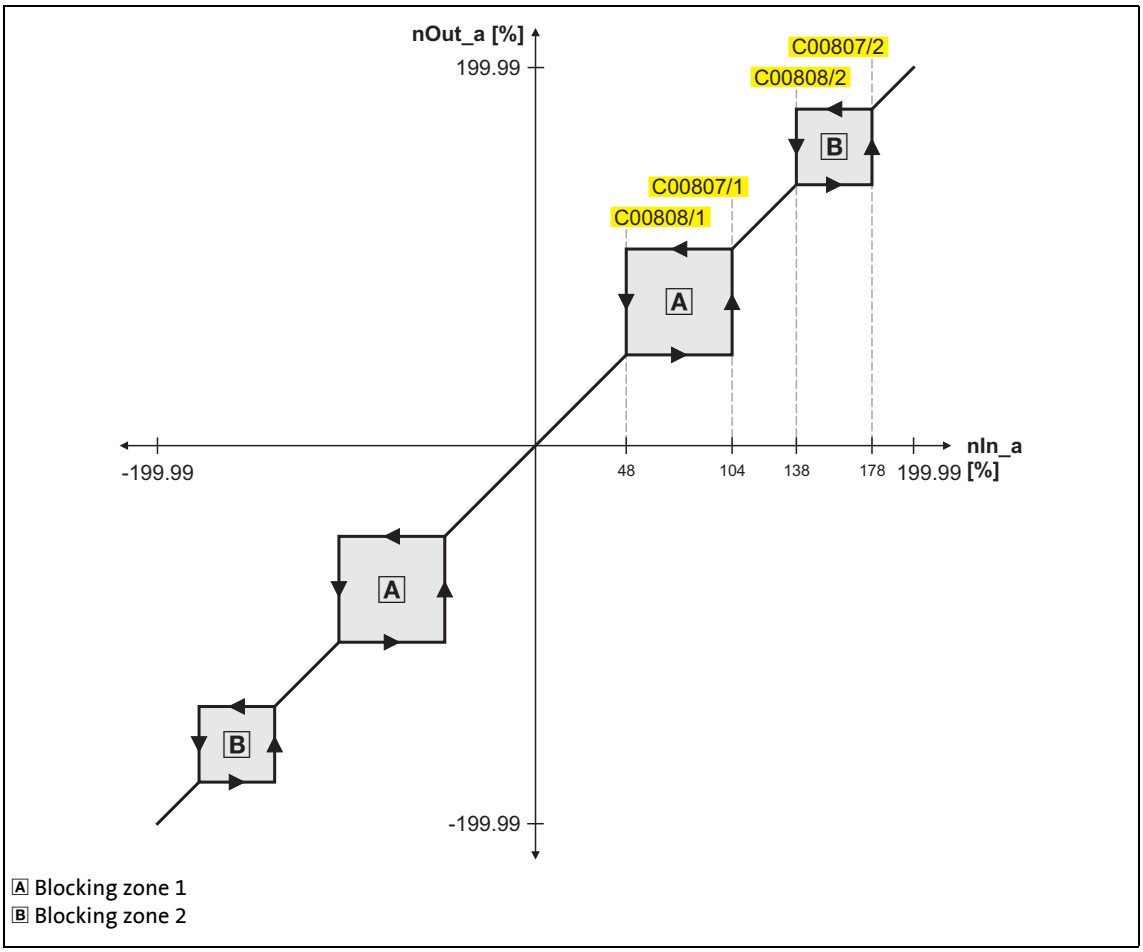
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00807/1...3 | 0.00 | % | 199.99 | Max. skip frequency 1 ... 3<br>• Lenze setting: 0.00 % |
| C00808/1...3 | 0.00 | % | 199.99 | Min. skip frequency 1 ... 3<br>• Lenze setting: 0.00 % |

### Definition of the blocking zones

Up to three zones can be parameterised which are to be skipped by the *nOut_a* output signal.

The example below shows the parameter setting of two blocking zones for the **L_NLim_1** FB:

| Parameters | Blocking zone 1 | | Blocking zone 2 | | Blocking zone 3 | |
|---|---|---|---|---|---|---|
| Minimum limit value | C00808/1: | 48 % | C00808/2: | 138 % | C00808/3: | 0 % |
| Maximum limit value | C00807/1: | 104 % | C00807/2: | 178 % | C00807/3: | 0 % |




[19-51] Zone masking by means of parameterisable blocking zones (in our example: L_NLim_1)

- The parameterised blocking zones have the same effect on negative input signals.
- A blocking zone is deactivated by entering identical limit values (in our example: blocking zone 3).

**Overlapping of blocking zones**

If blocking zones overlap, the lowest and highest value of the overlapping zones form a new zone.

In this case, the status display (*wState* output) will also just display one zone (the lower one of the two original zones).

**Abutting blocking zones**

If two blocking zones abut (e.g. 20 ... 30 % and 30 ... 40 %), the limit value between the two zones (in this example 30 %) is also passed through.

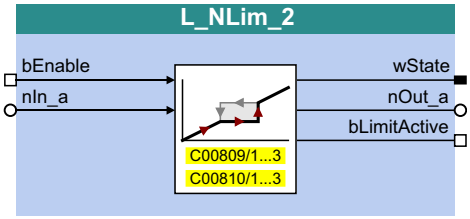
The same applies to a limit range of 0 ... xx %. At the zero crossing of the *nIn_a* input signal, a value of "0" is also output at the *nOut_a* output. Value "0" cannot be excluded. If the *nIn_a* input signal drops back to "0", the *nOut_a* output will retain the upper limit value.

### 19.1.114 L_NLim_2

This FB can hide up to three parameterisable blocking zones within one continuous signal characteristic.

**Note!**

A value of "0" cannot be masked out if the *nIn_a* input signal changes signs.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bEnable<br>BOOL | Activate zone masking | |
| | FALSE | The FB has been deactivated.<br>• The input signal is output one-to-one at the *nOut_a* output. |
| | TRUE | The FB has been activated.<br>• Zone masking of the input signal is carried out according to the parameterised blocking zones. |
| nIn_a<br>INT | Input signal | |

#### Outputs

| Identifier/data type | Value/meaning | |
|---|---|---|
| wState<br>WORD | Bit-coded status word<br>• Bits that are not listed are reserved for future extensions. | |
| | Bit 0 | No blocking zone active |
| | Bit 1 | Blocking zone 1 active |
| | Bit 2 | Blocking zone 2 active |
| | Bit 3 | Blocking zone 3 active |
| nOut_a<br>INT | Output signal<br>• If the FB has been activated, the output signal is outside the blocking zones. | |
| bLimitActive<br>BOOL | "Limitation active" status signal | |
| | TRUE | The input signal is inside a blocking zones and is limited to the respective boundary value of the blocking zone. |

### Parameters

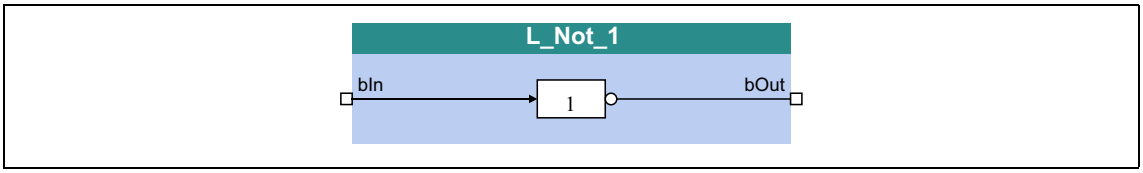
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00809/1...3 | 0.00 | % | 199.99 | Max. skip frequency 1 ... 3<br>• Lenze setting: 0.00 % |
| C00810/1...3 | 0.00 | % | 199.99 | Min. skip frequency 1 ... 3<br>• Lenze setting: 0.00 % |



For a detailed functional description see L_NLim_1.

### 19.1.115 L_Not_1

This FB negates a BOOL data type signal.




#### Inputs

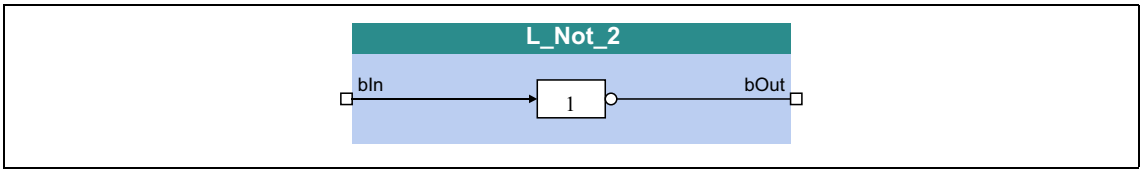
| Designator<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

### 19.1.116 L_Not_2

This FB negates a BOOL data type signal.




#### Inputs

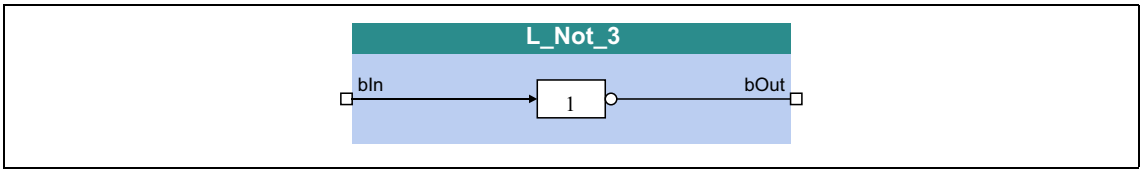
| Designator<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

### 19.1.117 L_Not_3

This FB negates a BOOL data type signal.




#### Inputs

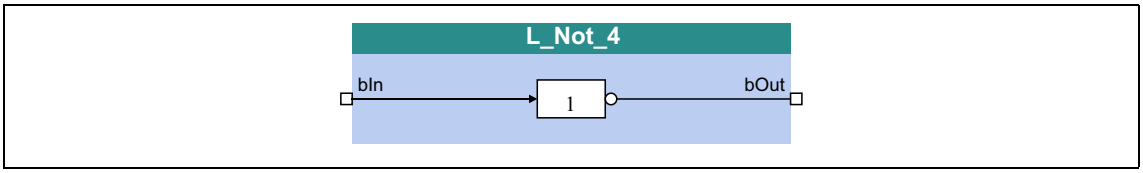
| Designator<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

### 19.1.118 L_Not_4

This FB negates a BOOL data type signal.




#### Inputs

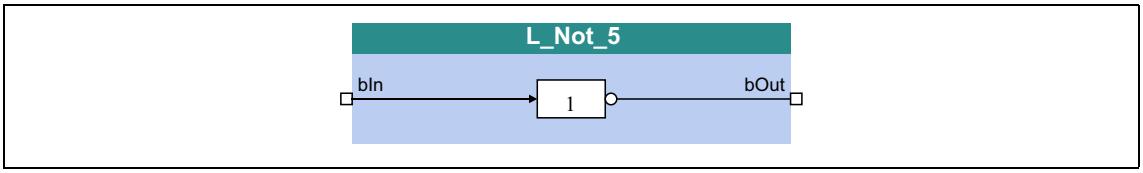
| Designator<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

### 19.1.119 L_Not_5

This FB negates a BOOL data type signal.




#### Inputs

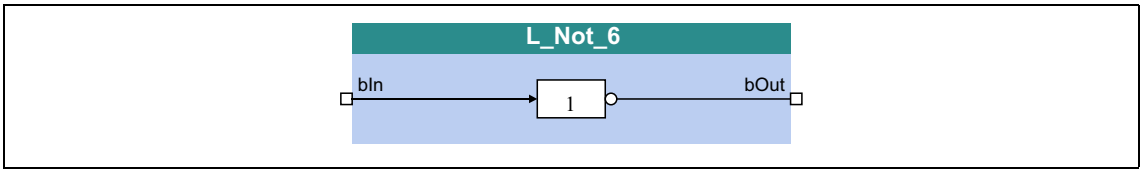
| Designator<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

### 19.1.120 L_Not_6

This FB negates a BOOL data type signal.




#### Inputs

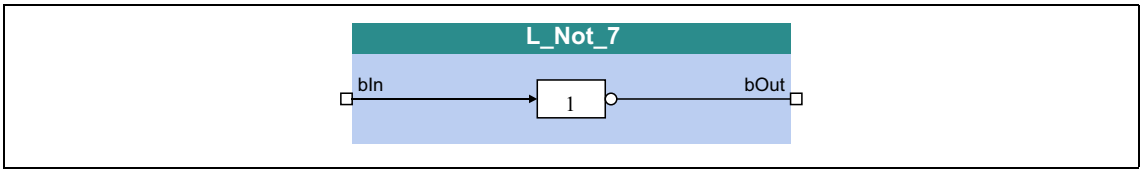
| Designator<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

### 19.1.121 L_Not_7

This FB negates a BOOL data type signal.




#### Inputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

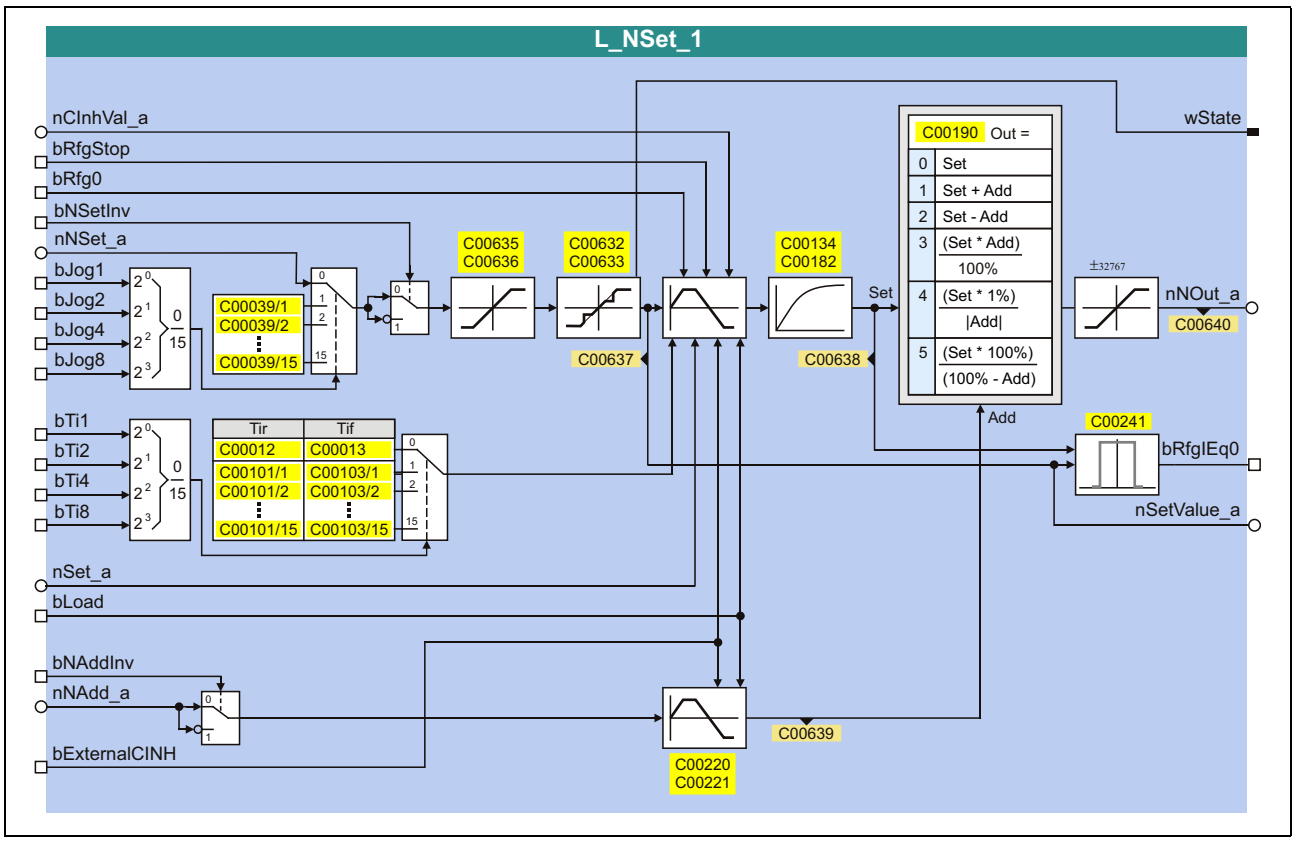
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

### 19.1.122 L_NSet_1

This FB is used for general signal processing of process values and is provided with the following functions:

- Ramp function generator
  - With linear ramps for main and additional setpoint path
  - With S-shaped ramp (PT1 rounding)
  - Setting and holding
- Internal limitation of the input signal
- 3 adjustable blocking zones
- Arithmetic function
- 15 fixed setpoints (JOG setpoints)
- 15 acceleration and deceleration times



### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nCInhVal_a<br>INT | Main setpoint signal which is to be accepted by the main setpoint integrator when the controller is inhibited. | |
| bRfgStop<br>BOOL | Holding (freezing) of the current value of the main setpoint integrator | |
| | TRUE | The current value of the main setpoint integrator is held. |
| bRfg0<br>BOOL | Leading the main setpoint integrator to 0 within the current Ti times | |
| | TRUE | The current value of the main setpoint integrator is led to "0" within the Ti time set. |
| bNSetInv<br>BOOL | Signal inversion for the main setpoint | |
| | TRUE | Main setpoint signal is inverted. |
| nNset_a<br>INT | Main setpoint signal<br>• Other signals are also permitted | |
| bJog1 ... bJog8<br>BOOL | Inputs for overriding fixed setpoints (JOG setpoints) for the main setpoint<br>• Selection inputs are binary coded. | |
| bTI1 ... bTI8<br>BOOL | Selection inputs for alternative acceleration/deceleration times for the main setpoint<br>• Selection inputs are binary coded. | |
| nSet_a<br>INT | Starting value which is loaded into the main setpoint integrator by setting *bLoad* to TRUE. | |
| bLoad<br>BOOL | Control of both ramp function generators in special situations, e.g. QSP | |
| | TRUE | The *nSet_a* input signal is loaded into the main setpoint integrator and the additional setpoint integrator is set to "0". |
| bAddInv<br>BOOL | Signal inversion for the additional setpoint | |
| | TRUE | Additional setpoint signal is inverted. |
| nNAdd_a<br>INT | Additional setpoint signal<br>• Other signals are also permitted | |
| bExternalCINH<br>BOOL | Additional load input for the main setpoint integrator and the additional setpoint integrator | |
| | TRUE | The main setpoint integrator is set to the value applied at *nCInhVal_a*.<br>The additional setpoint integrator is set to "0".<br>▸ Application example for the additional load function (📖 1632) |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nNOut_a<br>INT | Speed setpoint output signal<br>• Scaling: 16384 ≡ 100 % |
| bRfgIEqO<br>BOOL | Status signal "setpoint before ramp = setpoint after ramp (ramp inactive)" |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| wState<br>WORD | Bit-coded status word<br>• Bits that are not listed are reserved for future extensions. | |
| | Bit 0 | No blocking zone active |
| | Bit 1 | Blocking zone 1 active |
| | Bit 2 | Blocking zone 2 active |
| | Bit 3 | Blocking zone 3 active |
| | Bit 4 | Jog in blocking zone |
| | Bit 5 | MaxLimit active |
| | Bit 6 | MinLimit active |
| nSetValue_a<br>INT | Speed-setpoint input signal of the ramp function generator<br>• Scaling: 16384 ≡ 100 % | |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00012 | 0.000 | s | 999.900 | Acceleration time $T_{ir}$ for the main setpoint<br>• Lenze setting: 0.000 s |
| C00013 | 0.000 | s | 999.900 | Deceleration time $T_{if}$ for the main setpoint<br>• Lenze setting: 0.000 s |
| C00039/1..15 | -199.99 | % | 199.99 | Fixed setpoints (JOG setpoints)<br>• Lenze setting: 0.00 % |
| C00101/1..15 | 0.000 | s | 999.900 | Alternative acceleration times ($T_{ir}$) for the main setpoint<br>• Lenze setting: 0.000 s |
| C00103/1..15 | 0.000 | s | 999.900 | Alternative deceleration times ($T_{if}$) for the main setpoint<br>• Lenze setting: 0.000 s |
| C00134 | | | | Activates ramp rounding with PT1 behaviour for the main setpoint<br>• The corresponding S-ramp time must be set in C00182.<br>• Lenze setting: 0 (deactivated) |
| | 0 | Off | | |
| | 1 | PT1 behaviour | | |
| C00182 | 0.01 | s | 50.00 | S-ramp time PT1<br>• Lenze setting: 20.00 s |
| C00190 | | | | Selection of the arithmetic function for combining main and additional setpoint |
| | 0 | NOut = NSet | | Lenze setting<br>• The additional setpoint is not processed. |
| | 1 | NOut = NSet + NAdd | | |
| | 2 | NOut = NSet - NAdd | | |
| | 3 | NOut = (NSet * NAdd) / 100% | | |
| | 4 | NOut = (NSet * 1%) / \|NAdd\| | | |
| | 5 | NOut = (NSet * 100%) / (100% - NAdd) | | |
| C00220 | 0.000 | s | 999.900 | Acceleration time $T_{ir}$ for the additional setpoint<br>• Lenze setting: 0.000 s |

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00221 | 0.000 | s | 999.900 | Deceleration time $T_{if}$ for the additional setpoint<br>• Lenze setting: 0.000 s |
| C00241 | 0.00 | % | 100.00 | Hysteresis window for zero detection of speed output setpoint (output *bRfgIEqO*)<br>• Lenze setting: 0.50 % |
| C00632/1...3 | 0.00 | % | 199.99 | Maximum limit values for the speed blocking zones<br>• Selection of the maximum limit values for the blocking zones in which the speed must not be constant.<br>• Lenze setting: 0.00 % |
| C00633/1...3 | 0.00 | % | 199.99 | Minimum limit values for the speed blocking zones<br>• Selection of the minimum limit values for the blocking zones in which the speed must not be constant.<br>• Lenze setting: 0.00 % |
| C00634 | | | | Status (bit-coded)<br>• Bits that are not listed are reserved for future extensions. |
| | Bit 0 | No blocking zone active | | |
| | Bit 1 | Blocking zone 1 active | | |
| | Bit 2 | Blocking zone 2 active | | |
| | Bit 3 | Blocking zone 3 active | | |
| | Bit 4 | Jog in blocking zone | | |
| | Bit 5 | MaxLimit active | | |
| | Bit 6 | MinLimit active | | |
| C00635 | -199.99 | % | 199.99 | nMaxLimit<br>• Maximum speed setpoint for speed setpoint limitation<br>• Lenze setting: 199.99 % |
| C00636 | -199.99 | % | 199.99 | nMinLimit<br>• Minimum speed setpoint for speed setpoint limitation<br>• Lenze setting: -199.99 % |
| C00637 | -199.99 | % | 199.99 | Blocking zone output<br>• Display of the speed setpoint after being processed by blocking zone function. |
| C00638 | -199.99 | % | 199.99 | Ramp smoothing output<br>• Display of the speed setpoint after being processed by PT1 filter function. |
| C00639 | -199.99 | % | 199.99 | Additional value output<br>• Display of the additional speed setpoint after being processed by ramp generator. |
| C00640 | -199.99 | % | 199.99 | Output nNOut_a<br>• Display of the generated main speed setpoint at the *nNOut_a* output. |

### 19.1.122.1 Main setpoint path

- The signals in the main setpoint path are limited to a value range of ±32767.
- The signal at *nNSet_a* is first led via the JOG selection function.
- A selected JOG value switches the *nNSet_a* input inactive. Then, the subsequent signal conditioning operates with the JOG value.

### 19.1.122.2 JOG setpoints

In addition to the direct main setpoint selection via the *nNSet_a* input, so-called JOG setpoints can be preset in C00039/1...15.

- The JOG setpoints are binary-coded and can be called using the *bJog1* ... *bJog8* selection inputs so that 15 options are available:

| Selection inputs | | | | Main setpoint Main setpoint |
|---|---|---|---|---|
| bJog8 | bJog4 | bJog2 | bJog1 | |
| FALSE | FALSE | FALSE | FALSE | *nNset_a* |
| FALSE | FALSE | FALSE | TRUE | C00039/1 |
| FALSE | FALSE | TRUE | FALSE | C00039/2 |
| FALSE | FALSE | TRUE | TRUE | C00039/3 |
| FALSE | TRUE | FALSE | FALSE | C00039/4 |
| FALSE | TRUE | FALSE | TRUE | C00039/5 |
| FALSE | TRUE | TRUE | FALSE | C00039/6 |
| FALSE | TRUE | TRUE | TRUE | C00039/7 |
| TRUE | FALSE | FALSE | FALSE | C00039/8 |
| TRUE | FALSE | FALSE | TRUE | C00039/9 |
| TRUE | FALSE | TRUE | FALSE | C00039/10 |
| TRUE | FALSE | TRUE | TRUE | C00039/11 |
| TRUE | TRUE | FALSE | FALSE | C00039/12 |
| TRUE | TRUE | FALSE | TRUE | C00039/13 |
| TRUE | TRUE | TRUE | FALSE | C00039/14 |
| TRUE | TRUE | TRUE | TRUE | C00039/15 |

- The number of selection inputs to be assigned depends on the number of JOG setpoints required:

| Number of JOG setpoints required | Number of selection inputs to be assigned (bJog1 ... bJog8) |
|---|---|
| 1 | At least 1 |
| 2 ... 3 | at least 2 |
| 4 ... 7 | at least 3 |
| 8 ... 15 | 4 |

### 19.1.122.3 Setpoint inversion

The output signal of the JOG function is led via an inverter.

The sign of the setpoint changes if *bNSetInv* is set to TRUE.

### 19.1.122.4 Value range of the input signal

The value range of the input signal can be limited by using the following parameters:

- C00635: MaxLimit (default setting: +199.99 %)
- C00636: MinLimit (default setting: -199.99 %)

### 19.1.122.5 Skip frequency function

If the speed setpoints in speed-variable drives are linearly increasing, for instance, the frequency/speed range is divided into a number of equal time segments. Therefore, there may be speeds during acceleration time which must be bridged very fast (e.g. natural resonant frequencies).

The skip frequency function offers the opportunity to select a range in which the initial speed is maintained. If the speed setpoint leaves that range, the drive will be accelerated to reach the desired speed.

**Note!**

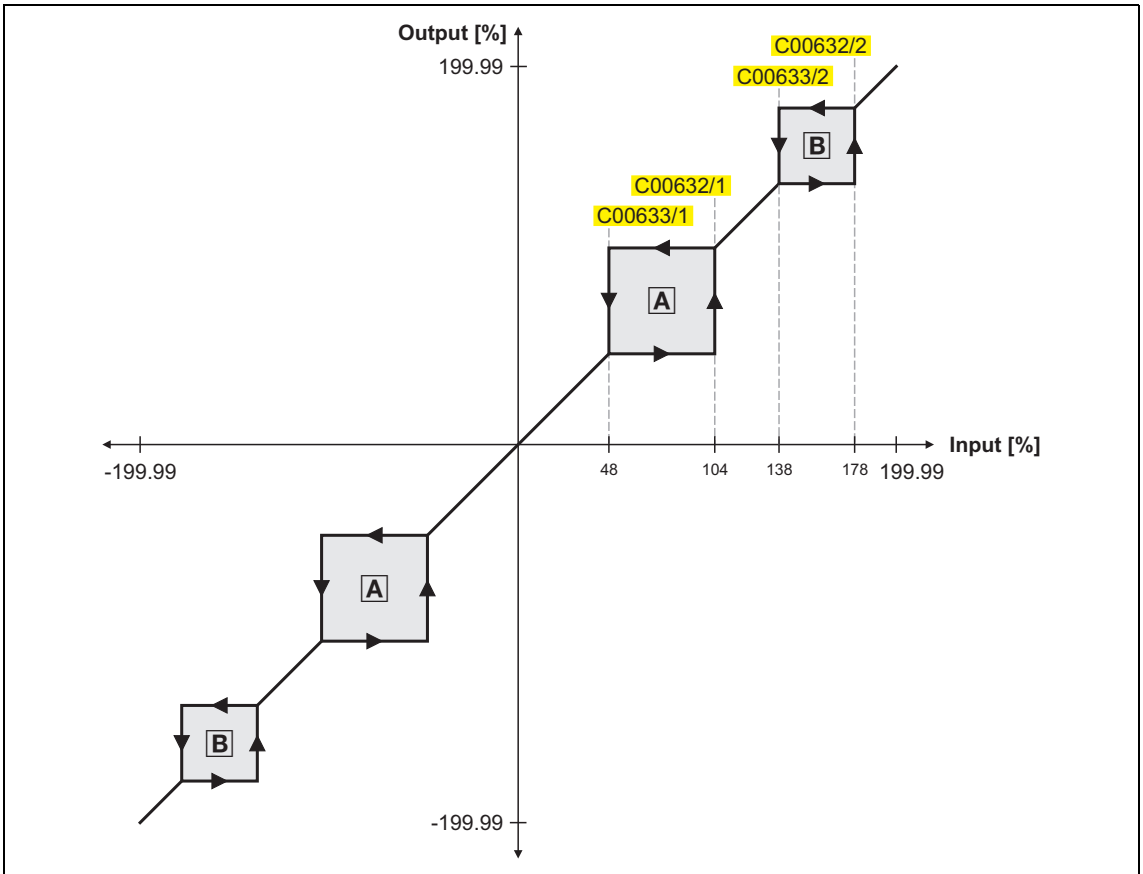
- Skip frequencies only affect main setpoints.
- It is not possible to exclude "0" speed if there is a sign reversal of the speed setpoint.

### Definition of the blocking zones

The subcodes of codes C00632 and C00633 can be used to define three zones which are to be skipped by the output setpoint and which are to be passed as fast as possible by the ramp function generator.

The example below shows the parameter setting of two blocking zones:

| Parameters | Blocking zone 1 | | Blocking zone 2 | | Blocking zone 3 | |
|---|---|---|---|---|---|---|
| Minimum limit value | C00633/1: | 48 % | C00633/2: | 138 % | C00633/3: | 0 % |
| Maximum limit value | C00632/1: | 104 % | C00632/2: | 178 % | C00632/3: | 0 % |




[19-52] Zone masking by means of parameterisable blocking zones

- The parameterised blocking zones have the same effect on negative input signals.
- A blocking zone is deactivated by entering identical limit values (in our example: blocking zone 3).

**Overlapping of blocking zones**

If blocking zones overlap, the lowest and highest value of the overlapping zones form a new zone.

In this case, the status display (output *wState* or display parameter C00634) only indicates one zone (the lower of the two original zones).

**Abutting blocking zones**

If two blocking zones abut (e.g. 20 ... 30 % and 30 ... 40 %), the limit value between the two zones (in this example 30 %) is also passed through.

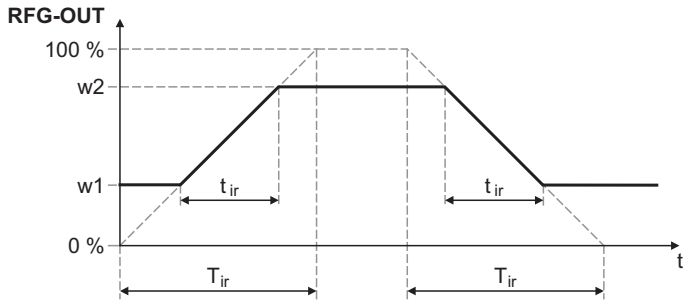
The same applies to a limit range of 0 ... xx %. During zero crossing of the speed setpoint, "0" speed is output as setpoint. It is possible to exclude "0" speed. However, in this case, the output speed will remain on the upper limit value when the input setpoint becomes "0".

**Tip!**

As described above, the acceleration phase starts after the blocking zones have been passed through. The ramp function generator integrated in the **L_Nset** function block limits the progression of the speed. For this reason, the time values set for the integrated ramp function generator should be as low as possible whereas the setpoint for the **L_NSet** function block should be generated by a ramp function generator with higher time values (e.g. L_MPot function block).

### 19.1.122.6 Ramp function generator for the main setpoint

The setpoint is now led via a ramp function generator with linear characteristic. The ramp function generator converts setpoint step-changes at the input into a ramp.




w1, w2 = change of the main setpoint as a function of $t_{ir}$ and $t_{if}$
RFG-OUT = output of the ramp function generator

[19-53] Acceleration and deceleration times

- $t_{ir}$ and $t_{if}$ are the desired times for changing between w1 and w2.
- S-ramps are possible by selecting S-ramp times.
- The $t_{ir}/t_{if}$ values are converted into the required Ti times according to the following formula:

| | |
|---|---|
| $T_{ir} = t_{ir} \cdot \frac{100\,\%}{w2 - w1}$ | $T_{if} = t_{if} \cdot \frac{100\,\%}{w2 - w1}$ |

### Setting and selection of Ti times

Via parameters, you can select 16 different Tir and Tif times each for the ramp function generator.

- The selection is made via the binary coded selection inputs *bTI1* ... *bTI8*:

| Selection inputs | | | | Used Acceleration time | Used Deceleration time |
|---|---|---|---|---|---|
| bTI8 | bTI4 | bTI2 | bTI1 | | |
| FALSE | FALSE | FALSE | FALSE | C00012 | C00013 |
| FALSE | FALSE | FALSE | TRUE | C00101/1 | C00103/1 |
| FALSE | FALSE | TRUE | FALSE | C00101/2 | C00103/2 |
| FALSE | FALSE | TRUE | TRUE | C00101/3 | C00103/3 |
| FALSE | TRUE | FALSE | FALSE | C00101/4 | C00103/4 |
| FALSE | TRUE | FALSE | TRUE | C00101/5 | C00103/5 |
| FALSE | TRUE | TRUE | FALSE | C00101/6 | C00103/6 |
| FALSE | TRUE | TRUE | TRUE | C00101/7 | C00103/7 |
| TRUE | FALSE | FALSE | FALSE | C00101/8 | C00103/8 |
| TRUE | FALSE | FALSE | TRUE | C00101/9 | C00103/9 |
| TRUE | FALSE | TRUE | FALSE | C00101/10 | C00103/10 |
| TRUE | FALSE | TRUE | TRUE | C00101/11 | C00103/11 |
| TRUE | TRUE | FALSE | FALSE | C00101/12 | C00103/12 |
| TRUE | TRUE | FALSE | TRUE | C00101/13 | C00103/13 |
| TRUE | TRUE | TRUE | FALSE | C00101/14 | C00103/14 |
| TRUE | TRUE | TRUE | TRUE | C00101/15 | C00103/15 |

### Function

- When the controller is inhibited (CINH), the ramp function generator accepts the value applied at *nCInhVal_a* and transfers it to the downstream function. This function has priority over all other functions.
- *bRfgStop* = TRUE
  - The ramp function generator is stopped. Changes at the input of the ramp function generator have no effect on the output signal.
- *bRfg0* = TRUE
  - The ramp function generator runs to 0 along its deceleration ramp.
- Furthermore it is possible to load the ramp function generator online with a defined value. For this purpose, *bLoad* must be set to TRUE. As long as this input is set, the value at *nSet_a* is transferred to the ramp function generator and provided at the output.

**Priorities:**

| CINH | bLoad | bRfg0 | bRfgStop | Function |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | The ramp function generator follows the input value via the set ramps. |
| FALSE | FALSE | FALSE | **TRUE** | **Stop the ramp function generator:**<br>The value at the output of the ramp function generator is held. |
| FALSE | FALSE | **TRUE** | FALSE | **Ramp down the ramp function generator:**<br>The ramp function generator runs to 0 within the set deceleration time. |
| FALSE | FALSE | **TRUE** | TRUE | |
| FALSE | **TRUE** | FALSE | FALSE | **Load ramp function generator online:**<br>The ramp function generator accepts the value at *nSet_a* and provides it at its output. |
| FALSE | **TRUE** | FALSE | TRUE | |
| FALSE | **TRUE** | TRUE | FALSE | |
| FALSE | **TRUE** | TRUE | TRUE | |
| **TRUE** | FALSE | FALSE | FALSE | **Controller inhibit:**<br>The ramp function generator accepts the value at *nCInhVal_a* and provides it at its output. |
| **TRUE** | FALSE | FALSE | TRUE | |
| **TRUE** | FALSE | TRUE | FALSE | |
| **TRUE** | FALSE | TRUE | TRUE | |
| **TRUE** | TRUE | FALSE | FALSE | |
| **TRUE** | TRUE | FALSE | TRUE | |
| **TRUE** | TRUE | TRUE | FALSE | |
| **TRUE** | TRUE | TRUE | TRUE | |

### 19.1.122.7 S-shaped ramp

A PT1 element is connected downstream of the linear ramp function generator. This arrangement implements an S-shaped ramp for a nearly jerk-free acceleration and deceleration.

- The PT1 element can be switched on/off via the *bSShapeActive* input.
- The corresponding S-ramp time can be set under C00182.

### 19.1.122.8 Additional setpoint

Use the *nNAdd_a* input to define an additional value (e.g. a correcting signal) and combine it arithmetically with the main setpoint *nNSet_a*.

- First, the additional setpoint is led via a ramp function generator with linear characteristic. Its Ti times can be set in C00220 (acceleration time) and C00221 (deceleration time).
- When the input *bNAddInv* is set to TRUE, the additional setpoint can be inverted before having an effect on the ramp function generator.
- When the input *bLoad* is set to TRUE, the ramp function generator is set to zero for the additional setpoint and held there without considering the Ti times. The same applies when the controller is inhibited.
- The following arithmetic combination of main setpoint and additional setpoint can be selected in C00190:

| Value in C00190 | Function | Info |
|---|---|---|
| 0 | *nNOut_a* = *nNSet_a* | The additional setpoint *nNAdd_a* is not processed. |
| 1 | *nNOut_a* = *nNSet_a* + *nNAdd_a* | |
| 2 | *nNOut_a* = *nNSet_a* - *nNAdd_a* | |
| 3 | *nNOut_a* = (*nNSet_a* * *nNAdd_a*) / 100 % | Internal scaling:<br>• 100 % ≡ 16384<br>• 1 % ≡ 164 |
| 4 | *nNOut_a* = (*nNSet_a* * 1 %) / \|*nNAdd_a*\| | |
| 5 | *nNOut_a* = (*nNSet_a* * 100 %) / (100 % - *nNAdd_a*) | |

### 19.1.122.9 Application example for the additional load function

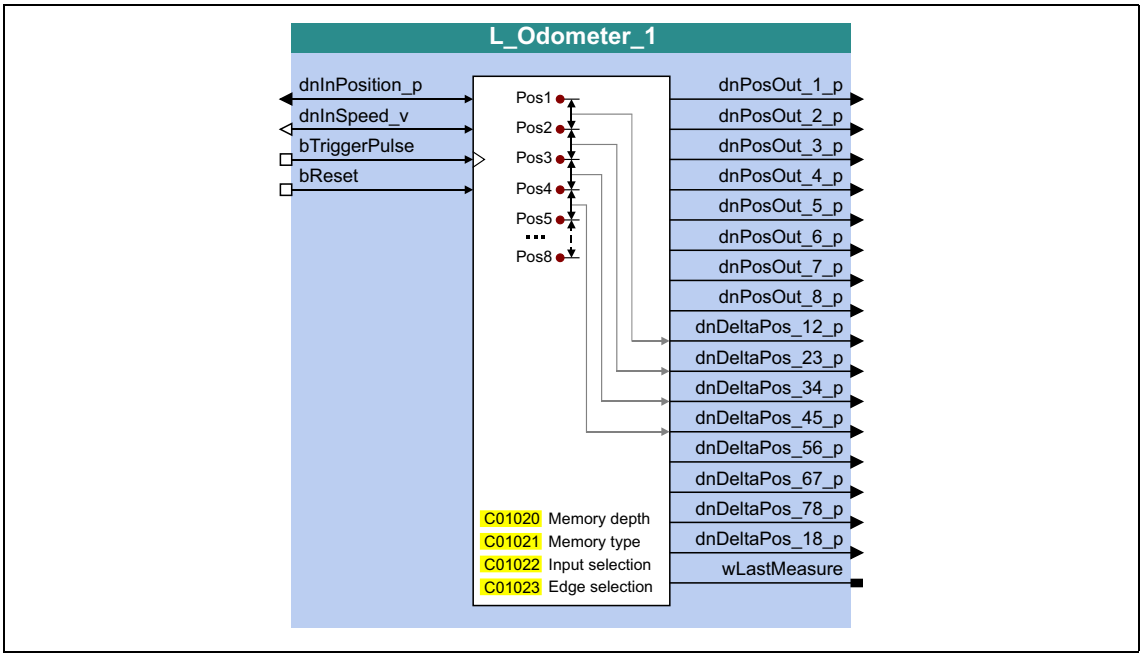
The motor control of the drive is provided with a function for automatically carrying along ramp function generators for "jerk-free" setpoint connection. For speed-controlled drive tasks, the LS_MotorInterface SB outputs the current actual speed value via the *nHlgSetValue_a* output (e.g. in case of a pulse inhibit, flying restart, controller inhibit).

- In case of a pulse inhibit, the main setpoint generator must be carried along with the current actual speed value to ensure jerk-free setpoint transfer.
- The actual speed value is carried along automatically if the following wiring is provided:
  - LS_MotorInterface.*nHlgSetValue_a* → L_NSet_1.*nCInhValue_a*
  - LS_MotorInterface.*bHlgLoad* → L_NSet_1.*bExternalCINH*

### 19.1.123 L_Odometer_1

This FB can be used to record positions and calculate distances.

- Depending on the input selection set in C01022, a position signal can be detected at the *dnInPosition_p* input or a speed can be detected at the *nInSpeed_v* input.
- The position is detected/accepted via the edge selected in C01023 at the *bTriggerPulse* input.
- If "Ring buffer" is selected in C01021, the measurement will start all over again after the number of measurements selected in C01020 has been performed and the old values will be overwritten. Otherwise, the measurement will stop.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnInPosition_p<br>DINT | Position measurement input<br>• Input is only evaluated if C01022 = "0: Pos input" has been set. |
| nInSpeed_v<br>INT | Speed measurement input<br>• The speed is internally integrated into a position.<br>• Input is only evaluated if C01022 = "1: V input" has been set. |
| bTriggerPulse<br>BOOL | Detect position/speed<br>• The tripping edge can be parameterised in C01023. |
| bReset<br>BOOL | Reset measurement results and internal integrator<br>TRUE: Reset measurement results and internal integrator. |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnPosOut_1_p<br>...<br>dnPosOut_8_p<br>DINT | Detected positions |
| dnDeltaPos_12_p<br>dnDeltaPos_23_p<br>dnDeltaPos_34_p<br>...<br>dnDeltaPos_18_p<br>DINT | Calculated distances between the detected positions<br>• *dnDeltaPos_12_p* = Distance between *dnPosOut_1_p* and *dnPosOut_1_p*<br>• *dnDeltaPos_23_p* = Distance between *dnPosOut_2_p* and *dnPosOut_3_p*<br>• *dnDeltaPos_34_p* = Distance between *dnPosOut_3_p* and *dnPosOut_4_p*<br>...<br>• *dnDeltaPos_18_p* = Distance between *dnPosOut_1_p* and *dnPosOut_8_p* |
| wLastMeasure<br>WORD | Memory location number (1 ... 8) of the last measurement |

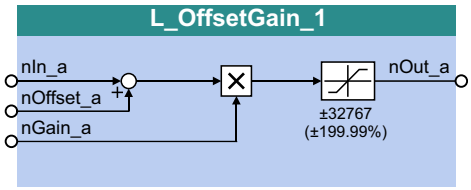
### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01020 | | | Memory length<br>• Lenze setting: 7 measurements |
| | 1 | 1 measurement | |
| | 2 | 2 measurements | |
| | ... | ... | |
| | 7 | 7 measurements | |
| C01021 | | | Memory type<br>• Lenze setting: No ring buffer |
| | 0 | No ring buffer | |
| | 1 | Ring buffer | |
| C01022 | | | Input selection<br>• Lenze setting: Pos input |
| | 0 | Pos input | |
| | 1 | V input | |
| C01023 | | | Edge selection<br>• Lenze setting: HIGH edge |
| | 0 | High edge | |
| | 1 | Low edge | |
| | 2 | High and low edge | |

### 19.1.124 L_OffsetGain_1

This FB can add an offset to an analog input signal and then amplify it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

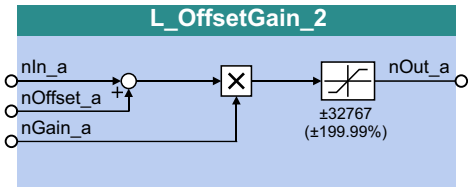
#### Function

$$\text{nOut_a} = (\text{nIn_a} + \text{Offset}) \cdot \text{Gain factor}$$

### 19.1.125 L_OffsetGain_2

This FB can add an offset to an analog input signal and then amplify it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

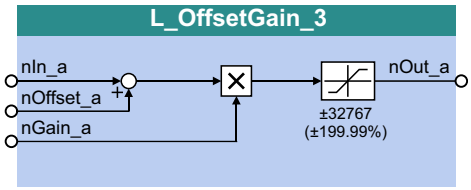
#### Function

$$nOut_a = (nIn_a + Offset) \cdot Gain\ factor$$

### 19.1.126 L_OffsetGain_3

This FB can add an offset to an analog input signal and then amplify it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

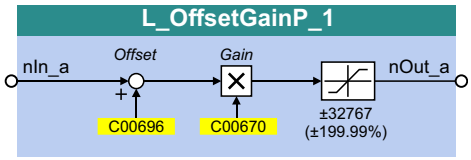
#### Function

$$nOut_a = (nIn_a + Offset) \cdot Gain\ factor$$

### 19.1.127 L_OffsetGainP_1

This FB can add an offset to an analog input signal and amplify it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00670 | -100.0000 | | 100.0000 | Gain factor<br>• High gain factor for further processing of smallest input signals.<br>• Please observe the difference with regard to the gain factors of other blocks in percent (±199.99 % ≈ 2).<br>• Lenze setting: 1.0000 |
| C00696 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

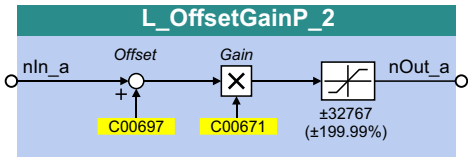
#### Function

$$nOut_a = (nIn_a + Offset) \cdot Gain\ factor$$

### 19.1.128 L_OffsetGainP_2

This FB can add an offset to an analog input signal and amplify it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00671 | -100.0000 | | 100.0000 | Gain factor<br>• High gain factor for further processing of smallest input signals.<br>• Please observe the difference with regard to the gain factors of other blocks in percent (±199.99 % ≈ 2).<br>• Lenze setting: 1.0000 |
| C00697 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

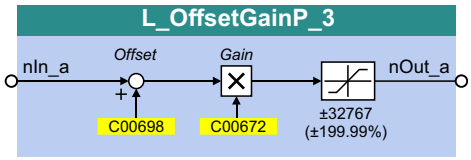
#### Function

$$nOut_a = (nIn_a + Offset) \cdot Gain\ factor$$

### 19.1.129 L_OffsetGainP_3

This FB can add an offset to an analog input signal and amplify it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00672 | -100.0000 | | 100.0000 | Gain factor<br>• High gain factor for further processing of smallest input signals.<br>• Please observe the difference with regard to the gain factors of other blocks in percent (±199.99 % ≈ 2).<br>• Lenze setting: 1.0000 |
| C00698 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

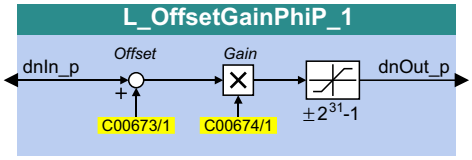
#### Function

$$nOut_a = (nIn_a + Offset) \cdot Gain\ factor$$

## 19.1.130 L_OffsetGainPhiP_1

This FB can add an offset to an angle signal and amplify it afterwards.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via parameters.
- The value provided at the *dnOut_p* output is internally limited to $\pm 2^{31}$-1.




### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00673/1 | -2147483647 | Incr. | 2147483647 | Offset<br>• Lenze setting: 0 incr. |
| C00674/1 | -2147483647 | | 2147483647 | Gain factor<br>• Scaling:<br>65535 ≡ gain factor 1<br>• Lenze setting: 65536 |

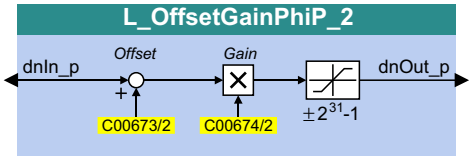
### Function

$$dnOut_p = (dnIn_p + Offset) \cdot Gain\ factor$$

### 19.1.131 L_OffsetGainPhiP_2

This FB can add an offset to an angle signal and amplify it afterwards.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via parameters.
- The value provided at the *dnOut_p* output is internally limited to $\pm 2^{31}$-1.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

#### Parameters

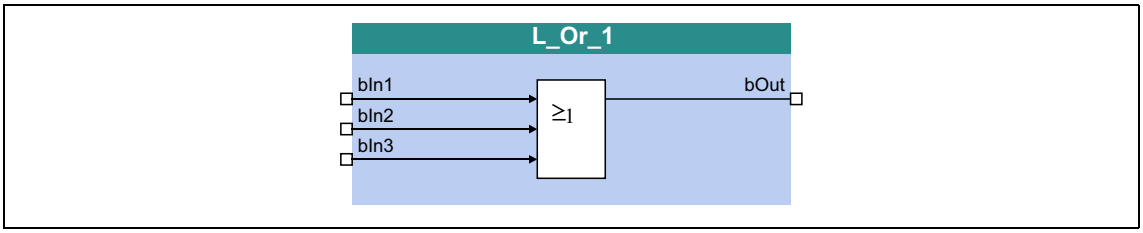
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00673/2 | -2147483647 | Incr. | 2147483647 | Offset<br>• Lenze setting: 0 incr. |
| C00674/2 | -2147483647 | | 2147483647 | Gain factor<br>• Scaling:<br>65535 ≡ gain factor 1<br>• Lenze setting: 65536 |

#### Function

$$dnOut_p = (dnIn_p + Offset) \cdot Gain\ factor$$

### 19.1.132 L_Or_1

This FB implements the OR operation of input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

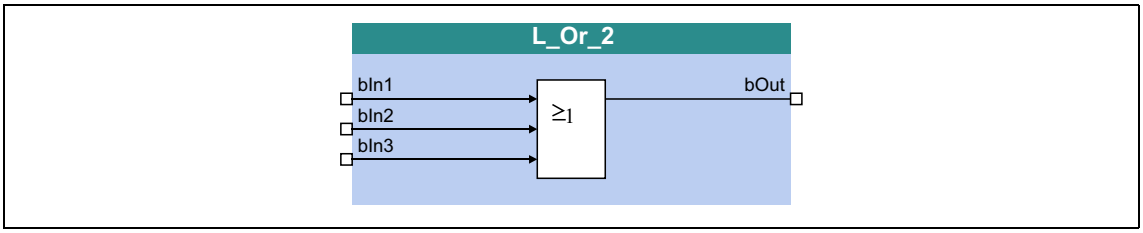
#### Function

| Inputs | | | Out |
|---|---|---|---|
| bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | TRUE |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | |

[19-54] Truth table of the FB L_Or_1

### 19.1.133 L_Or_2

This FB implements the OR operation of input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

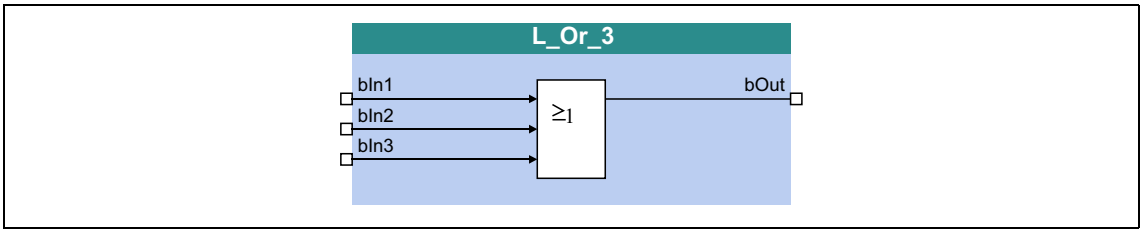
#### Function

| Inputs | | | Out |
|---|---|---|---|
| bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | TRUE |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | |

[19-55] Truth table of the FB L_Or_2

### 19.1.134 L_Or_3

This FB implements the OR operation of input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

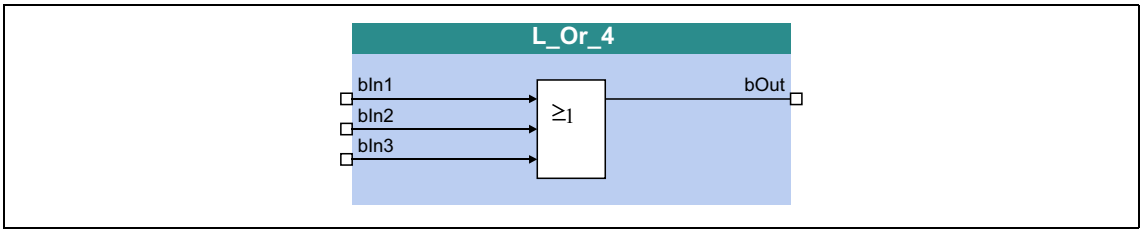
#### Function

| Inputs | | | Out |
|---|---|---|---|
| bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | TRUE |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | |

[19-56] Truth table of the L_Or_3 FB

### 19.1.135 L_Or_4

This FB implements the OR operation of input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

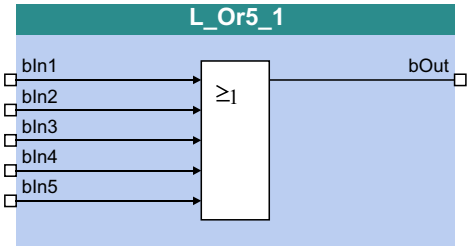
#### Function

| Inputs | | | Out |
|---|---|---|---|
| bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | TRUE |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | |

[19-57] Truth table of the L_Or_4 FB

### 19.1.136 L_Or5_1

This FB implements the OR operation of input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

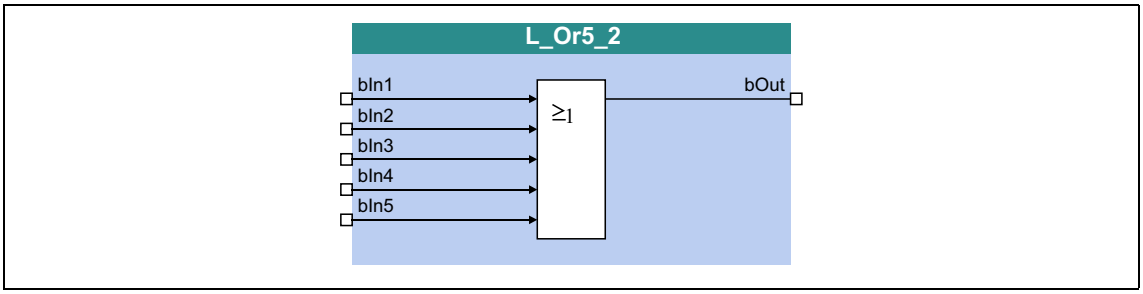
#### Function

| Inputs | | | | | Out |
|---|---|---|---|---|---|
| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | FALSE | FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE | TRUE | FALSE | |
| FALSE | FALSE | FALSE | TRUE | TRUE | |
| FALSE | FALSE | TRUE | FALSE | FALSE | |
| ... | | | | | |
| TRUE | TRUE | TRUE | FALSE | TRUE | |
| TRUE | TRUE | TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE | TRUE | |

[19-58] Truth table of the FB L_Or5_1

### 19.1.137 L_Or5_2

This FB implements the OR operation of input signals.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

#### Function

| Inputs | | | | | Out |
|---|---|---|---|---|---|
| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | FALSE | FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE | TRUE | FALSE | |
| FALSE | FALSE | FALSE | TRUE | TRUE | |
| FALSE | FALSE | TRUE | FALSE | FALSE | |
| ... | | | | | |
| TRUE | TRUE | TRUE | FALSE | TRUE | |
| TRUE | TRUE | TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE | TRUE | |

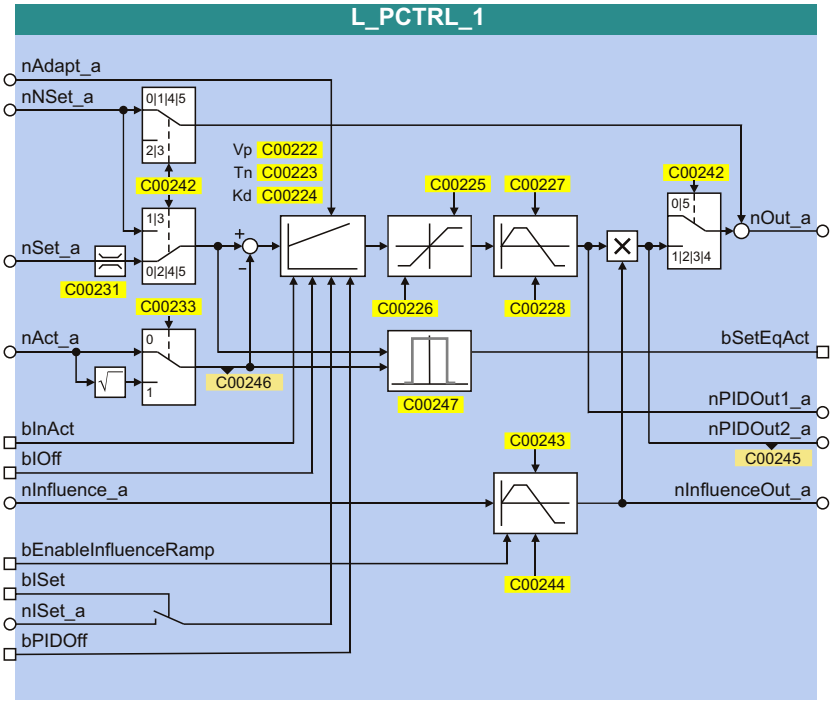
[19-59] Truth table of the FB L_Or5_2

### 19.1.138 L_PCTRL_1

This FB is a PID controller and can be used for various control tasks (e.g. as dancer position controller, tension controller, or pressure controller).

The FB is provided with the following functions:

- Adjustable control algorithm (P, PI, PID)
- Ramp function generator for preventing setpoint step-changes at the input
- Limitation of the controller output
- Factorisation of the output signal
- Vp adaptation
- Integral action component can be switched off
- Comparison function "Actual value = setpoint"




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nAdapt_a<br>INT | Percentage adaptation of the gain Vp set in C00222 and the reset time Tn set in C00223.<br>Then, the adapted reset time is as follows:<br>$Tn = \frac{Vp_{C222} \cdot nAdapt_a\ [\%] \cdot Tn_{C223}\ [ms]}{100\ [\%]}$<br>• Internal limitation to ± 199.99 %<br>• Changes can be done online.<br>• Display parameter: C00830/62 |
| nNset_a<br>INT | Speed setpoint<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Display parameter: C00830/89 |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nSet_a<br>INT | Sensor and process setpoint for operating modes 2, 4 and 5<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Display parameter: C00830/63 | |
| nAct_a<br>INT | Speed or actual sensor value (actual process value)<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Display parameter: C00830/61 | |
| bInAct<br>BOOL | Deactivate process controller temporarily (stop)<br>• Changes can be done online.<br>• Display parameter: C00833/76<br>Note: This input is not interconnected in the LA_NCtrl application block. | |
| | TRUE | • The current output value is frozen.<br>• The internal control algorithm is stopped.<br>• However, a setpoint selected via input *nNSet_a* is still provided in operating modes 0/1/4/5. |
| bIOff<br>BOOL | Switch off the I-component of the process controller<br>• Changes can be done online.<br>• Display parameter: C00833/77 | |
| | TRUE | The I component of the process controller is set to zero. |
| nInfluence_a<br>INT | Limitation of the influencing factor in percent<br>• *nInfluence_a* serves to limit the influencing factor of the PID controller contained in the FB to a required value (- 199.99 % ... + 199.99 %).<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Display parameter: C00830/64 | |
| bEnableInfluenceRamp<br>BOOL | Activate ramp for influencing factor<br>• Display parameter: C00833/106 | |
| | TRUE | Influencing factor of the PID controller is ramped up to the *nInfluence_a* value. |
| | FALSE | Influencing factor of the PID controller is ramped down to "0". |
| bISet<br>BOOL | Accept I component *nISet_a* in PID controller | |
| | TRUE | The value at the input *nISet_a* is accepted in the PID controller. |
| nISet_a<br>INT | Selection of I component of PID controller<br>• With a TRUE signal at *bISet*, the assigned value is accepted in the PID controller.<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |
| bPIDOff<br>BOOL | Reset the entire PID controller | |
| | TRUE | • The I component of the controller is set to zero.<br>• The controller output is set to zero.<br>• The internal control algorithm is stopped. |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %)<br>• Scaling: 16384 ≡ 100 % | |
| bSetEqAct<br>INT | Comparison function "Actual value = setpoint"<br>• The window for the comparison operation can be set in C00247. | |
| | TRUE | Setpoint and actual value are identical, i.e. no system deviation available. |
| nPIDOut1_a<br>INT | PID controller output without influencing factor *nInfluence_a*<br>• Inputs *bEnableInfluenceRamp* and *nInfluence_a* do not have any effect here, the limited PID output value influenced by the internal ramp times is output.<br>• There is no connection with the additive input *nNSet_a*.<br>• Scaling: 16384 ≡ 100 % | |
| nPIDOut2_a<br>INT | PID controller output with influencing factor *nInfluence_a*.<br>• There is no connection with the additive input *nNSet_a*.<br>• Scaling: 16384 ≡ 100 %<br>• Display parameter: C00245 | |
| nInfluenceOut_a<br>INT | Current influencing factor ("ramp status") on the PID output value<br>• Scaling: 16384 ≡ 100 % | |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00222 | 0.1 | 0.1 | 500.0 | Gain Vp<br>• Lenze setting: 1.0 |
| C00223 | 20 | ms | 6000 | Reset time Tn<br>• Lenze setting: 400 ms |
| C00224 | 0.0 | 0.1 | 5.0 | Differential component Kd<br>• Lenze setting: 0.0 |
| C00225 | -199.99 | % | +199.99 | MaxLimit<br>• Maximum value of the PID operating range<br>• Lenze setting: 199.99 % |
| C00226 | -199.99 | % | +199.99 | MinLimit<br>• Minimum value of the PID operating range<br>• Lenze setting: -199.99 % |
| C00227 | 0.000 | s | 999.999 | Acceleration time for the ramp at the PID output (should be set as steep as possible)<br>• Lenze setting: 0.010 s |
| C00228 | 0.000 | s | 999.999 | Deceleration time for the ramp at the PID output<br>• Lenze setting: 0.010 s |
| C00231/1 (Pos. Maximum)<br>C00231/2 (Pos. Minimum)<br>C00231/3 (Neg. Minimum)<br>C00231/4 (Neg. Maximum) | 0.00 | % | 199.99 | Operating range<br>• Determination of the operating range for the PID process controller by limiting the input signal *nSet_a*.<br>• Lenze setting: No limitation (-199.99 % ... +199.99 %) |

| Parameters | Possible settings | | | | Info |
|---|---|---|---|---|---|
| C00233 | | | | | Root function<br>• Lenze setting: "0: Off" |
| | 0 | Off | | | The actual value at *nAct_a* is not changed for further processing. |
| | 1 | On | | | The square root of the actual value at *nAct_a* is taken for further processing. |
| C00242 | | | | | Operating mode<br>• Lenze setting: "0: Off" |
| | 0 | Off | | | The input setpoint *nNSet_a* is output without any changes at the output *nOut_a*. |
| | 1 | nNSet + nNSet_PID | | | *nNSet_a* and *nAct_a* are used as PID input values. The arriving *nNSet_a* is additively linked to the value output by the PID element. |
| | 2 | nSet_PID | | | *nSet_a* and *nAct_a* are used as PID input values. The input *nNSet_a* is not considered. |
| | 3 | nNSet_PID | | | *nNSet_a* and *nAct_a* are used as PID input values. The input *nSet_a* is not considered. |
| | 4 | nNSet + nSet_PID | | | *nSet_a* and *nAct_a* are used as PID input values. The arriving *nNSet_a* setpoint is additively linked to the value output by the PID element. |
| | 5 | nNSet \|\| nSet_PID | | | *nSet_a* and *nAct_a* are used as PID input values. The setpoint *nNSet_a* is output at the output *nOut_a* . The PID output value is output at the output *nPIDOut_a*. |
| C00243 | 0.000 | | s | 999.999 | Influence acceleration time<br>• Acceleration time $T_{ir}$ for the influencing factor.<br>• Lenze setting: 5.000 s |
| C00244 | 0.000 | | s | 999.999 | Influence deceleration time<br>• Deceleration time $T_{if}$ for the influencing factor.<br>• Lenze setting: 5.000 s |
| C00245 | -199.99 | | % | +199.99 | Display of PID output value *nPIDOut_a* |
| C00246 | -199.99 | | % | +199.99 | Display of the internal PID input value *nAct_a* |
| C00247 | 0 | | % | 100 | Window for comparison function "Actual value = setpoint"<br>• Lenze setting: 2 %<br>• Hysteresis: 1 % (fixed) |

### 19.1.138.1 Control characteristic

The PI algorithm is active in the Lenze setting.

#### Gain (P component)

The input value is controlled by a linear characteristic. The slope of the characteristic is determined by the controller gain Vp.

The controller gain Vp is set under C00222.

- The controller gain can be adapted via the input *nAdapt_a* (also possible in online mode).
- The input value *nAdapt_a* has a direct effect on the controller gain:

$$P = nAdapt_a \cdot C00222$$

Example: With the parameterised controller gain Vp = 2.0 and *nAdapt_a* = 75 %, the resulting gain factor is as follows:

$$P = \frac{75\ [\%]}{100\ [\%]} \cdot 2.0 = 1.5$$

#### Integral action component (I component)

The I component can be selected via the input *nISet_a*. With a TRUE signal at *bISet*, the assigned value is accepted in the PID controller.

- Setting the adjustment time Tn to the maximum value of "6000 ms" deactivates the I component.
- The I component of the controller can also be deactivated by setting the input *bIOff* to TRUE.
- The I component can be switched on and off online.

#### Reset time

The adjustment time Tn is set under C00223.

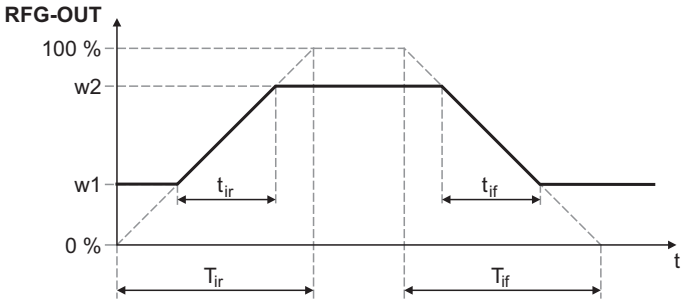
#### Differential component Kd (D component)

The differential component Kd is set under C00224.

- The setting "0.0 s" deactivates the D component (Lenze setting). In this way, the PID controller becomes a PI controller or P controller, if the I component has been deactivated as well.

### 19.1.138.2 Ramp function generator

The PID output is led via a ramp function generator with linear characteristic. This serves to transfer setpoint step-changes at the PID output into a ramp which should be as steep as possible.




w1, w2 = change of the main setpoint as a function of $t_{ir}$ and $t_{if}$
RFG-OUT = output of the ramp function generator

[19-60] Acceleration and deceleration times

- $t_{ir}$ and $t_{if}$ are the desired times for changing between w1 and w2.
- The ramps for acceleration and deceleration can be set individually.
  - C00227: Acceleration time $t_{ir}$
  - C00228: Deceleration time $t_{if}$
- The $t_{ir}/t_{if}$ values are converted into the required Ti times according to the following formula:

| | |
|---|---|
| $T_{ir} = t_{ir} \cdot \frac{100\,\%}{w2 - w1}$ | $T_{if} = t_{if} \cdot \frac{100\,\%}{w2 - w1}$ |

- The ramp function generator is immediately set to "0" by setting *bInAct* to TRUE.

### 19.1.138.3 Operating range of the PID process controller

The value range of the input signal *nSet_a* and thus the operating range of the PID process controller can be limited with the following parameters:

- C00231/1: Pos. maximum (default setting: 199.99 %)
- C00231/2: Pos. minimum (default setting: 0.00 %)
- C00231/3: Neg. minimum (default setting: 0.00 %)
- C00231/4: Neg. maximum (default setting: 199.99 %)

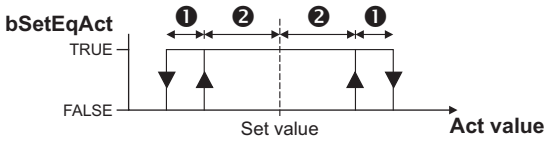
### 19.1.138.4 Evaluation of the output signal

After the limitation, the output signal is evaluated with the influencing factor *nInfluence_a*. The evaluation is activated/suppressed along a ramp when the *bEnableInfluenceRamp* input is set to TRUE. The ramp times are set with the parameters "Influence acceleration time" (C00243) and "Influence deceleration time" (C00244).

### 19.1.138.5 Comparison function "Actual value = setpoint"

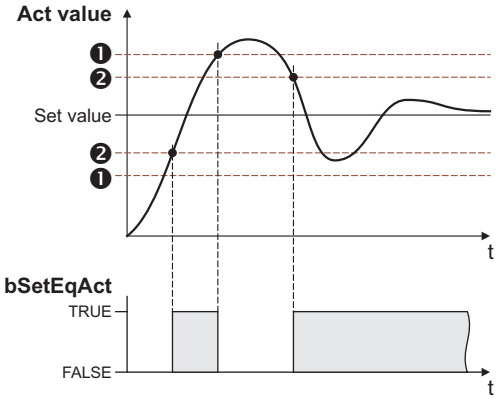
If setpoint and actual value are identical and there is no system deviation, the *bSetEqAct* status output is set to TRUE.

- The hysteresis of the comparison function has a fixed value of 1 %.
- The symmetrical window around the setpoint for the comparison function can be set in C00247 (Lenze setting: 2 %).




❶ Hysteresis (1 %, fixed value)
❷ C00247: Window

[19-61] Comparison function: Switching performance




❶ Hysteresis (1 %, fixed value)
❷ C00247: Window

[19-62] Comparison function: Example

### 19.1.138.6 Control functions

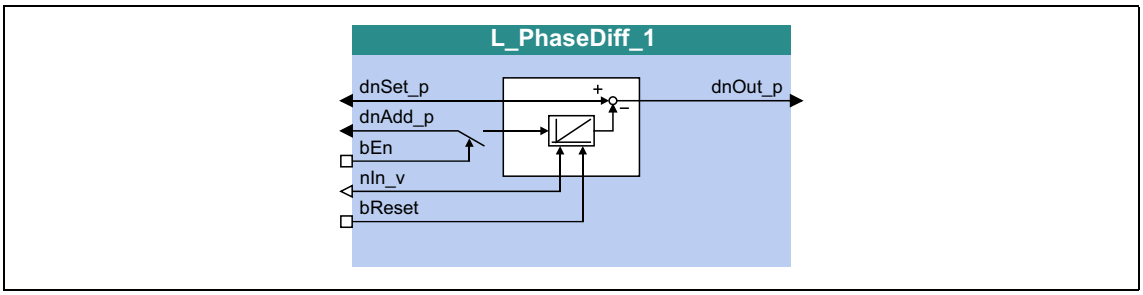
The process controller has various digital inputs for controlling the FB:

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bInAct<br>BOOL | Deactivate process controller temporarily (stop)<br>• Changes can be done online.<br>• Display parameter: C00833/76<br>Note: This input is not interconnected in the LA_NCtrl application block. | |
| | TRUE | • The current output value is frozen.<br>• The internal control algorithm is stopped.<br>• However, a setpoint selected via input *nNSet_a* is still provided in operating modes 0/1/4/5. |
| bIOff<br>BOOL | Switch off the I-component of the process controller<br>• Changes can be done online.<br>• Display parameter: C00833/77 | |
| | TRUE | The I component of the process controller is set to zero. |
| bPIDOff<br>BOOL | Reset the entire PID controller | |
| | TRUE | • The I component of the controller is set to zero.<br>• The controller output is set to zero.<br>• The internal control algorithm is stopped. |

### 19.1.139 L_PhaseDiff_1

This FB generates a position difference for the specified position setpoint from a position value and a speed signal.

- In an integrator, the *nIn_v* speed signal is integrated into a position value and subtracted from the *dnSet_p* position setpoint.
- In addition, the adaptive *dnAdd_p* position value can be added to the integrator content by setting *bEn* to TRUE.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnSet_p<br>DINT | Selection of a position setpoint | |
| dnAdd_p<br>DINT | Adaptive position value for the actual position | |
| bEn<br>BOOL | Activate addition of the adaptive position value | |
| | FALSE | 1. The speed signal at *nIn_v* is integrated by the angle integrator.<br>2. The result of the angle integrator is subtracted from the angle signal at *dnSet_p* and output at *dnOut_p* afterwards. |
| | TRUE | The adaptive position value selected via *dnAdd_p* is added:<br>1. The speed signal at *nIn_v* is integrated by the angle integrator.<br>2. The angle signal at *dnAdd_p* is added to the integrated speed signal in every task cycle.<br>3. The result of the angle integrator is subtracted from the angle signal at *dnSet_p* and output at *dnOut_p* afterwards. |
| nIn_v<br>INT | Selection of the actual speed to be converted into the position value | |
| bReset<br>BOOL | Reset actual angle integrator | |
| | TRUE | Actual angle integrator is set to "0". |

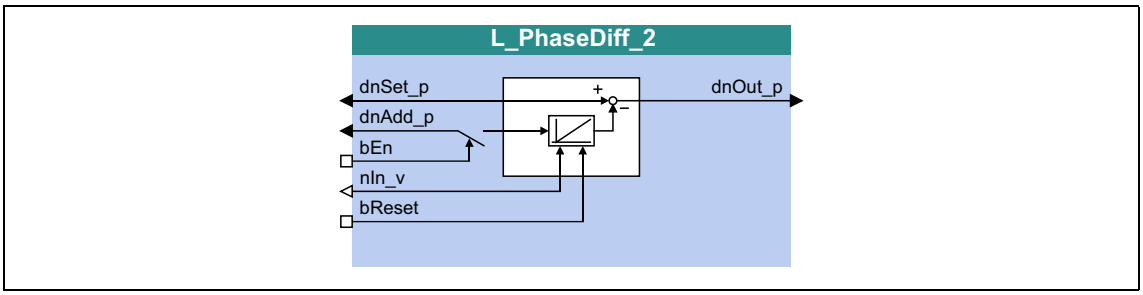
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output angle signal (position difference)<br>• Without limitation |

### 19.1.140 L_PhaseDiff_2

This FB generates a position difference for the specified position setpoint from a position value and a speed signal.

- In an integrator, the *nIn_v* speed signal is integrated into a position value and subtracted from the *dnSet_p* position setpoint.
- In addition, the adaptive *dnAdd_p* position value can be added to the integrator content by setting *bEn* to TRUE.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnSet_p<br>DINT | Selection of a position setpoint | |
| dnAdd_p<br>DINT | Adaptive position value for the actual position | |
| bEn<br>BOOL | Activate addition of the adaptive position value | |
| | FALSE | 1. The speed signal at *nIn_v* is integrated by the angle integrator.<br>2. The result of the angle integrator is subtracted from the angle signal at *dnSet_p* and output at *dnOut_p* afterwards. |
| | TRUE | The adaptive position value selected via *dnAdd_p* is added:<br>1. The speed signal at *nIn_v* is integrated by the angle integrator.<br>2. The angle signal at *dnAdd_p* is added to the integrated speed signal in every task cycle.<br>3. The result of the angle integrator is subtracted from the angle signal at *dnSet_p* and output at *dnOut_p* afterwards. |
| nIn_v<br>INT | Selection of the actual speed to be converted into the position value | |
| bReset<br>BOOL | Reset actual angle integrator | |
| | TRUE | Actual angle integrator is set to "0". |

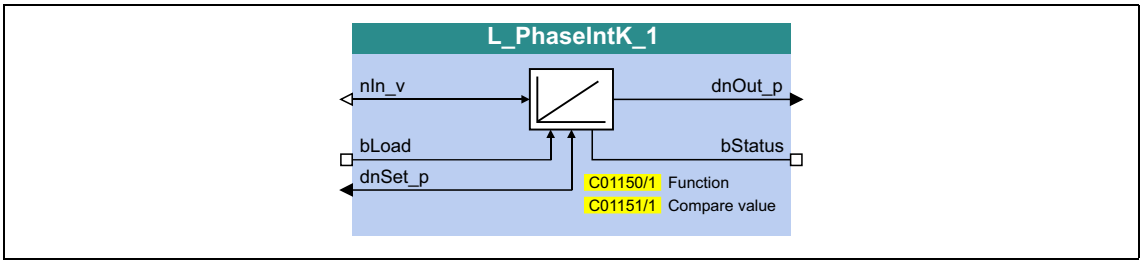
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output angle signal (position difference)<br>• Without limitation |

### 19.1.141 L_PhaseIntK_1

This FB can integrate a speed or a velocity to an angle (path). The FB can also detect a relatively completed distance.

- The integrator can take max. ±32000 encoder revolutions.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_v<br>INT | Selection of the actual speed<br>• 16384 ≡ 15000 rpm | |
| bLoad<br>BOOL | Load angle integrator with starting value and reset status signal | |
| | TRUE | Angle integrator is loaded with the value at *dnSet_p* and *bStatus* is reset to FALSE. |
| dnSet_p<br>DINT | Starting value for angle integrator | |

#### Outputs

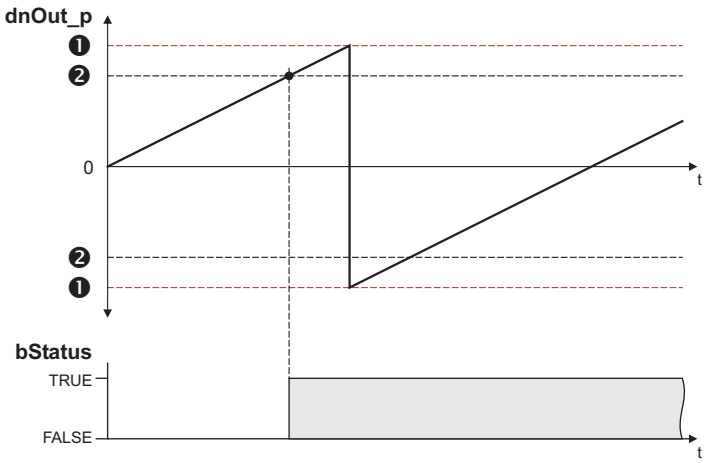
| Designator<br>Data type | Value/meaning | |
|---|---|---|
| dnOut_p<br>DINT | Angle output signal<br>• 65536 [incr.] ≡ 1 encoder revolution<br>• Overflow is possible (display via *bStatus*) | |
| bStatus<br>BOOL | Status signal "Overflow occurred/distance processed"<br>• Status signal can be reset via *bLoad*. | |
| | TRUE | Overflow has occurred or distance is processed. |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01150/1 | | | | Function |
| | 0 | Loading with level | | Load integrator with TRUE level at the input *bLoad*(Lenze setting) |
| | 1 | Loading with edge | | Load integrator with FALSE/TRUE edge at the input *bLoad*. |
| | 2 | Loading with level + reset | | Load integrator when reaching the comparison value or with TRUE level at the input *bLoad*. |
| C01151/1 | 0 | | 2000000000 | Comparison value<br>• Is valid for both the positive and the negative value range.<br>• Lenze setting: 0 |

### 19.1.141.1 Function at constant input value

Selection: C01150 = "0: Loading with level" or "1: Loading with edge"



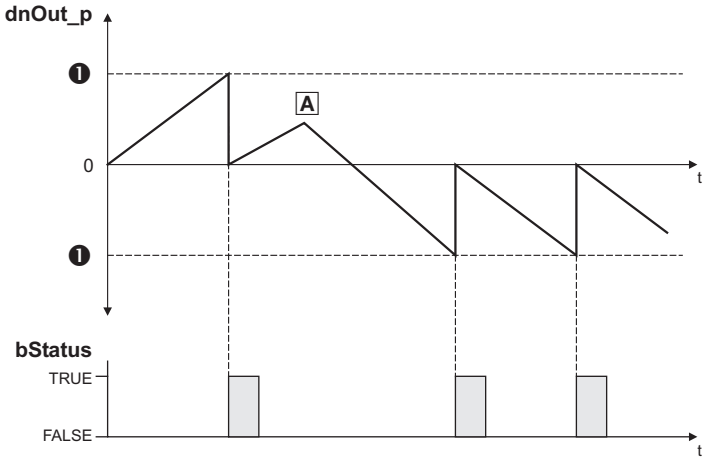

❶ Overflow at ±2147483648 incr. (±32767 encoder revolutions)
❷ C01151: Comparison value (applies to both positive and negative value range)

[19-63] Switching performance if the overflow is in the positive direction

- If "0: Loading with level" is selected in C01150, the *bLoad* input is status-controlled: In case of a TRUE signal, the integrator is loaded with the value at *dnSet_p* and the *bStatus* output is set to FALSE.
- If "1: Loading with edge" is selected in C01150, the *bLoad* input is edge-controlled: In case of a FALSE/TRUE edge, the integrator is loaded with the value at *dnSet_p* and then immediately continues to integrate, the *bStatus* output is set to FALSE.
- A positive signal at *nIn_v* is incremented (the counter content is increased with every function call).
- A negative signal at *nIn_v* is decremented (the counter content is reduced with every function call).
- *dnOut_p* outputs the counter content of the bipolar integrator.
  - If the counter content exceeds a value of +32767 encoder revolutions (corresponds to +2147483647 incr.), an overflow occurs and the counting process continues at a value of -32768 encoder revolutions.
  - If the counter content falls below a value of -32768 encoder revolutions (corresponds to -2147483648 incr.), an overflow occurs and the counting process starts at a value of +32767 encoder revolutions.
- *bStatus* will be set to TRUE if the comparison value set in C01151/x is reached.

### 19.1.141.2 Function at input value with sign reversal

Selection: C01150 = "2: Loading with level + reset"




A Sign reversal of the value at *nIn_v*
❶ C01151/x: Comparison value (is valid for both the positive and the negative value range)

[19-64] Switching performance if the input signal changes signs

- If "2: Loading with level + reset" is selected in C01150, the *bLoad* input is status-controlled: In case of a TRUE signal, the integrator is loaded with the value at *dnSet_p* and the *bStatus* output is set to FALSE.
- A positive signal at *nIn_v* is incremented (the counter content is increased with every function call).
- A negative signal at *nIn_v* is decremented (the counter content is reduced with every function call).
- *dnOut_p* outputs the counter content of the bipolar integrator.
  - If the positive counter content is higher than the comparison value set in C01151/x, the comparison value will be subtracted from the counter content, and *bStatus* will be set to TRUE for one task cycle.
  - If the negative counter content is lower than the comparison value set in C01151/x, the comparison value will be added to the counter content, and *bStatus* will be set to TRUE for one task cycle.

### 19.1.141.3 Calculation of the output signal

The output value at *dnOut_p* can be detected according to the following formula:

$$\text{dnOut_p [incr.]} = \text{nIn_v [rpm]} \cdot \text{t [s]} \cdot 65535\ \text{[incr./rev.]}$$

t = integration time
16384 ≡ 15000 rpm
1 ≡ 1 incr.

**Example**

You want to determine the counter content of the integrator at a certain speed at the input and a certain integration time **t**.

Given values:

- *nIn_v* = 1000 rpm ≈ integer value 1092
- Integration time t = 10 s
- Starting value of the integrator = 0

Solution:

- Conversion of the input signal at *nIn_v*:

$$1000\ \text{rpm} = \frac{1000\ \text{rev.}}{60\ \text{s}}$$

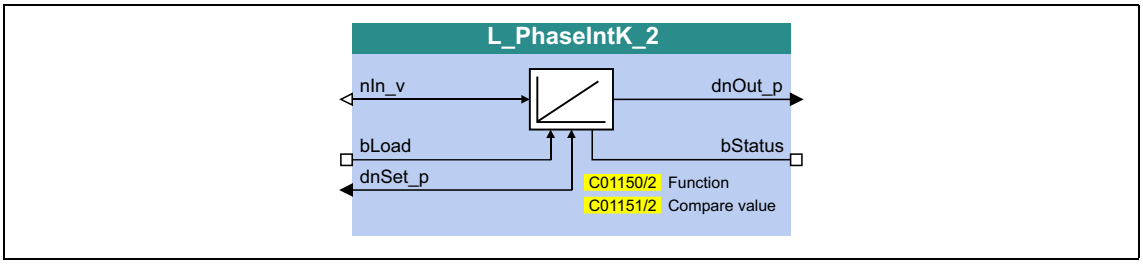
- Calculation of the output value:

$$\text{dnOut_p} = \frac{1000\ \text{rev.}}{60\ \text{s}} \cdot 10\ \text{s} \cdot \frac{65535\ \text{incr.}}{\text{Rev.}} = 10922666\ \text{incr.}$$

### 19.1.142 L_PhaseIntK_2

This FB can integrate a speed or a velocity to an angle (path). The FB can also detect a relatively completed distance.

- The integrator can take max. ±32000 encoder revolutions.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_v<br>INT | Selection of the actual speed<br>• 16384 ≡ 15000 rpm | |
| bLoad<br>BOOL | Load angle integrator with starting value and reset status signal | |
| | TRUE | Angle integrator is loaded with the value at *dnSet_p* and *bStatus* is reset to FALSE. |
| dnSet_p<br>DINT | Starting value for angle integrator | |

#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| dnOut_p<br>DINT | Angle output signal<br>• 65536 [incr.] ≡ 1 encoder revolution<br>• Overflow is possible (display via *bStatus*) | |
| bStatus<br>BOOL | Status signal "Overflow occurred/distance processed"<br>• Status signal can be reset via *bLoad*. | |
| | TRUE | Overflow has occurred or distance is processed. |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01150/2 | | | | Function |
| | 0 | Loading with level | | Load integrator with TRUE level at the input *bLoad*(Lenze setting) |
| | 1 | Loading with edge | | Load integrator with FALSE/TRUE edge at the input *bLoad*. |
| | 2 | Loading with level + reset | | Load integrator when reaching the comparison value or with TRUE level at the input *bLoad*. |
| C01151/2 | 0 | | 2000000000 | Comparison value<br>• Is valid for both the positive and the negative value range.<br>• Lenze setting: 0 |

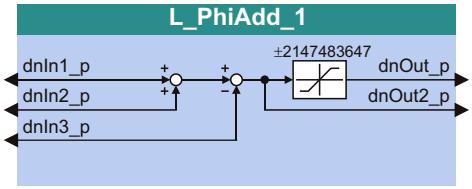


For a detailed functional description see L_PhaseIntK_1.

### 19.1.143 L_PhiAdd_1

This FB adds or subtracts angle signals depending on the input used.

- The value provided at the *dnOut_p* output is internally limited to ±2147483647 (±$2^{31}$ - 1).
- The value provided at the *dnOut2_p* output is not limited internally, this enables an internal overflow.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 (addition) |
| dnIn3_p<br>DINT | Input signal 3 (subtraction) |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |
| dnOut2_p<br>DINT | Output signal 2<br>• Without limitation/with overflow (which makes a wrong output possible) |

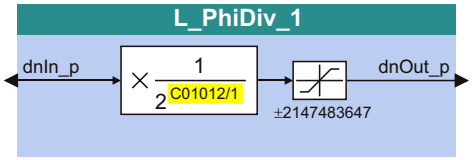
#### Function

$$dnOut_p = dnIn1_p + dnIn2_p - dnIn3_p$$

### 19.1.144 L_PhiDiv_1

This FB divides or multiplies angle signals in power of two format.

- A positive division factor (C01012/1) causes a division.
- A negative division factor (C01012/1) causes a multiplication.
- The value provided at the *dnOut_p* output is internally limited to ±2147483647 ($\pm 2^{31} - 1$).
- Division is not remainder considered.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01012/1 | -31 | | 31 | Division factor<br>• Lenze setting: 0 |

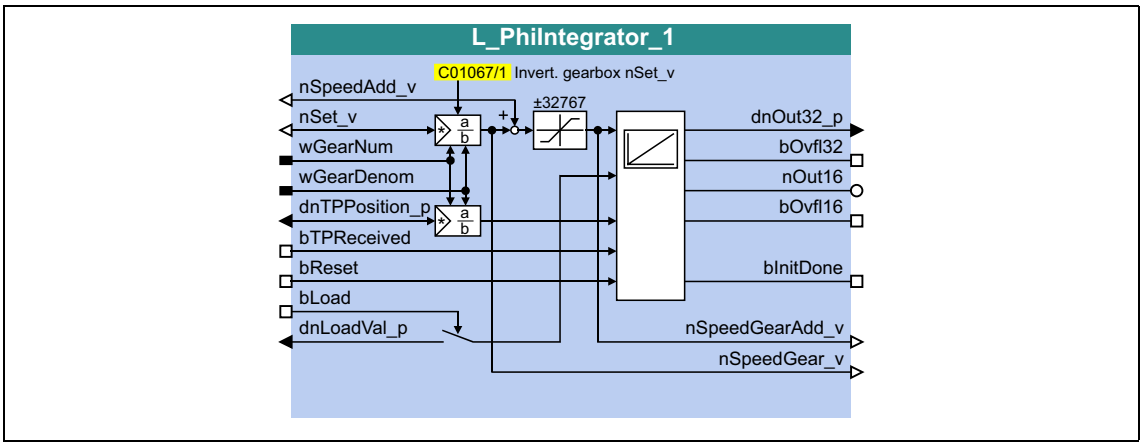
#### Function

$$dnOut_p = dnIn_p \cdot \frac{1}{2^{C01012/1}}$$

### 19.1.145 L_PhiIntegrator_1

This FB evaluates a speed with a gearbox factor and adds them in an integrator.

- The integrator value is output as 16-bit value and 32-bit value.
- An overflow of the prevailing output is signalled.
- The integrator can be loaded and initialised (e.g. with a touch probe signal).




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nSpeedAdd_v<br>INT | Additive speed | |
| nSet_v<br>INT | Speed input (angular difference) | |
| wGearNum<br>WORD | Gearbox factor (numerator) | |
| wGearDenom<br>WORD | Gearbox factor (denominator)<br>• When *wGearDenom* = 0, it is internally calculated with 1. | |
| dnTPPosition_p<br>DINT | Initialisation value | |
| bTPReceived<br>BOOL | Initialise integrator | |
| | FALSE↗TRUE | Initialise integrator |
| bReset<br>BOOL | Reset integrator | |
| | TRUE | Reset integrator<br>• The integrator is not integrated. |
| bLoad<br>BOOL | Load integrator | |
| | TRUE | Load integrator with *dnLoadVal_p*<br>• The integrator is not integrated.<br>• The *bInitDone* output is reset to FALSE with a delay of one call cycle after *bLoad* is reset to FALSE and a subsequent FALSE/TRUE edge to *bTPReceived*. |
| dnLoadVal_p<br>DINT | Value the integrator is to be loaded with. | |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| dnOut32_p<br>DINT | Integrator value (32 bits) | |
| bOvfl32<br>BOOL | Status signal "overflow (32 bits)" | |
| | TRUE | Overflow |
| nOut16<br>INT | Integrator value (16 bits) | |
| bOvfl16<br>BOOL | Status signal "overflow (16 bits)" | |
| | TRUE | Overflow |
| bInitDone<br>BOOL | Status signal "Initialisation completed" | |
| | TRUE | Integrator is initialised |
| nSpeedGearAdd_v<br>INT | Resulting speed with additive speed<br>(*nSet_v* * gearbox factor + *nSpeedAdd_v*) | |
| nSpeedGear_v<br>INT | Resulting speed without additive speed<br>(*nSet_v* * gearbox factor) | |

### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01067/1 | | | Invert. gearbox nSet_v |
| | 0 | not inverted | |
| | 1 | inverted | |
| | 2 | Automatically from MCK | |

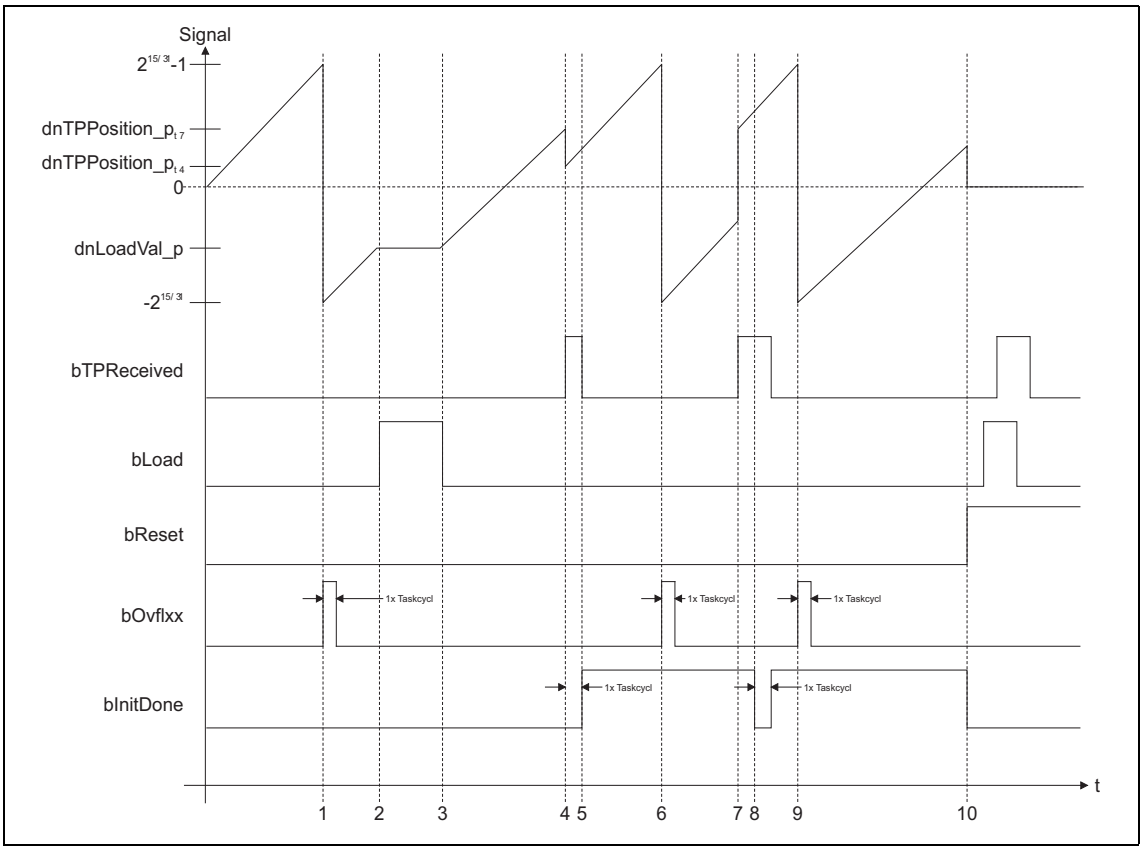
### 19.1.145.1 Function

The *nSet_v* input value is evaluated, integrated and output as 16-bit value ( *nOut16*) and 32-bit value *dnOut32_p*) with the gearbox factors *wGearNum* and *wGearDenom*.

$$nOut16 = nOut16 + nSet_v \cdot \frac{wGearNum}{wGearDenom}$$

$$dnOut32_p = dnOut32_p + nSet_v \cdot \frac{wGearNum}{wGearDenom}$$

- Evaluation of the gearbox factor is remainder considered.
- A positive or negative overflow of the integrator is signalised by a TRUE signal (for a task cycle) at the *bOvfl16* output for the *nOut16* output or at the *bOvfl32* output for the *dnOut32_p* output.




[19-65] Signal characteristic

### 19.1.145.2 Example

The current speed at *nSet_v* and the gearbox factors serve to create a specific machine measuring system from the motor measuring system.

- 216 increments in the motor measuring system are to correspond to 1 motor revolution.
- 216 increments in the machine measuring system (*nOut16*) are to correspond to 1 machine shaft revolution.

**Initialise integrator**

A FALSE/TRUE edge at *bTPReceived* initialises the integrator, i.e. a calculation with *dnTPPosition_p* and *dnLoadVal_p* is made:

$$\text{nOut16} = (\text{INT})(\text{dnLoadVal_p} + \text{dnTPPosition_p} \cdot \text{Gearbox factor})$$

$$\text{dnOut32_p} = \text{dnLoadVal_p} + \text{dnTPPosition_p} \cdot \text{Gearbox factor}$$

- After initialisation is completed, the *bInitDone* output is set to TRUE in the next call cycle.
- In case of a renewed initialisation without a previous reset (*bReset* = TRUE), the *bInitDone* output is set to FALSE for a task cycle.

**Reset integrator**

When *bReset* is set to TRUE, the integrator is set to 0. No calculations are made.

- The Boolean outputs are set to FALSE.
- The *bReset* input has the highest priority.

**Load integrator**

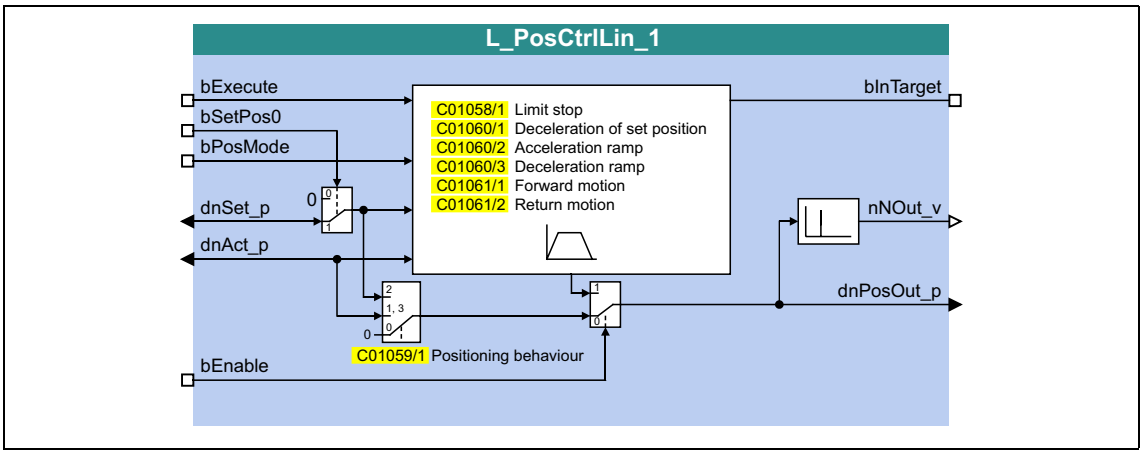
When *bLoad* is set to TRUE, the integrator is loaded with the value at *dnLoadVal_p* and the outputs are set.

- When *bLoad* is set to TRUE, the input is not integrated and no check for overflow is executed.
- When *bLoad* is reset to FALSE, the integrator continues from the loaded value. The *bInitDone* output is reset to FALSE after a FALSE/TRUE edge at *bTPReceived* with a delay of one call cycle.

### 19.1.146 L_PosCtrlLin_1

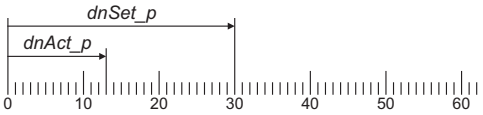
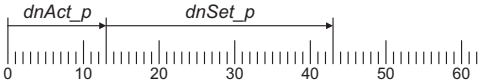
This FB serves to implement the following positioning functions:

- Bring drive to curve position (e.g. after mains connection, manual jog, homing)
- Release drive from curve and bring it to safe position (position override function)
- Positioning the X axis via X offset (higher-level positioning)
- Positioning the Y axis via Y offset (higher-level positioning)




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bExecute<br>BOOL | Execute positioning profile<br>• Only possible if the *bEnable* input is set to TRUE. | |
| | FALSE | No positioning/positioning is aborted.<br>• The speed at the *nNOut_v* output is braked to standstill via the deceleration ramp set in C01060/1 (*nNOut_v* = 0).<br>• At the same time, the *dnPosOut_p* output is stopped. |
| | FALSE↗TRUE | Positioning is executed/continued.<br>• In case of an activated limit stop (C01058/1 = TRUE), a renewed positive edge for a following positioning procedure is required. |
| bSetPos0<br>BOOL | Travel to zero position<br>• Only possible if the *bEnable* input is set to TRUE. | |
| | FALSE | The value at the *dnSet_p* input is used as setpoint position. |
| | TRUE | The value at the *dnSet_p* input is ignored. The setpoint position is internally set to "0" and the *dnPosOut_p* output can be travelled to zero position. |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bPosMode<br>BOOL | Positioning mode | |
| | FALSE | Absolute positioning<br>• The setpoint at *dnSet_p* is the absolute setpoint position (with regard to zero position):<br>  |
| | TRUE | Relative positioning<br>• With a FALSE/TRUE edge at the *bExecute* input, the outputs of the current position (*dnAct_p*) are traversed by the *dnSet_p* value:<br>  <br>**Note:**<br>Only to be used with activated limit stop (C01058/1 = TRUE)! |
| dnSet_p<br>DINT | Setpoint for positioning in [increments]<br>• Scaling: A revolution is displayed with 65536 increments or steps. | |
| dnAct_p<br>DINT | Actual position in [increments]<br>• Connect this input with e.g. *dnPosOut_p* of this FB or with another FB which outputs the actual position of the drive. | |
| bEnable<br>BOOL | Activate/deactivate positioning function<br>• This input has the highest priority. | |
| | FALSE | Positioning function deactivated.<br>• For behaviour see C01059/1. |
| | TRUE | Positioning function activated. |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bInTarget<br>BOOL | Status signal "Target position reached" | |
| | TRUE | Target position reached. |
| nNOut_v<br>INT | Speed output of the profile generator<br>• Scaling: 16384 ≡ 15000 rpm | |
| dnPosOut_p<br>DINT | Output position of the profile generator in [increments]<br>• Scaling: A revolution is displayed with 65536 increments or steps. | |

### Parameters

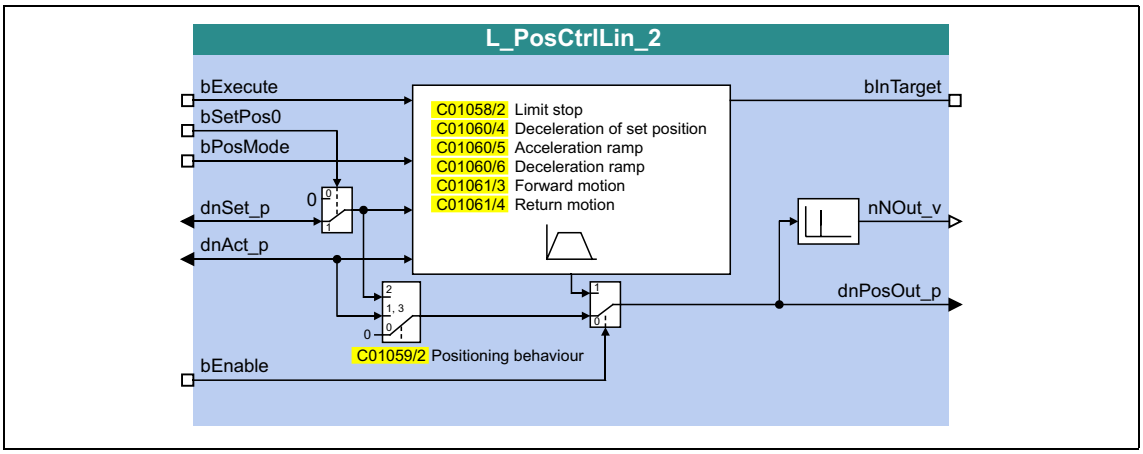
| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01058/1 | | | Limit stop |
| | 0 | deactivated<br>(Lenze setting) | The outputs continuously follow the setpoint at *dnSet_p*. |
| | 1 | activated | The outputs follow the setpoint at *dnSet_p* once.<br>• If the setpoint position changes, a new FALSE/TRUE edge is required at the *bExecute* input for another positioning process. |

| Parameters | Possible settings | | | | Info |
|---|---|---|---|---|---|
| C01059/1 | | | | | Positioning behaviour<br>• Behaviour with deactivated positioning function (*bEnable* = FALSE). |
| | 0 | *dnOut_p* = 0<br>(Lenze setting) | | | The *dnPosOut_p* output jumps to "0".<br>• The speed signal at the *nNOut_v* output maps this position step change as angle signal.<br>**Note:**<br>If the *nNOut_v* output has an impact on the setpoint generation for the drive, undefined motions (shocks, following errors) may occur. |
| | 1 | *dnOut_p/nNOut_v* follow *dnAct_p* | | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnAct_p* input (actual position). |
| | 2 | *dnOut_p/nNOut_v* follow *dnSet_p* | | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnSet_p* input (setpoint). |
| | 3 | *dnOut_p/nNOut_v* follow *dnAct_p*<br>(without limitation) | | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnAct_p* input (actual position), but without limitation. |
| C01060/1 | 0.010 | | s | 130.000 | Deceleration of set position<br>• Deceleration ramp for stop before reaching the setpoint position (*bExecute* = FALSE).<br>• Lenze setting: 1.000 s |
| C01060/2 | 0.010 | | s | 130.000 | Acceleration ramp<br>• Lenze setting: 1.000 s |
| C01060/3 | 0.010 | | s | 130.000 | Deceleration ramp<br>• Lenze setting: 1.000 s |
| C01061/1 | -15000 | | rpm | 15000 | Forward motion<br>• Positive speed<br>• Lenze setting: 200 rpm |
| C01061/2 | -15000 | | rpm | 15000 | Return motion<br>• Negative speed<br>• Lenze setting: 200 rpm |

### 19.1.147 L_PosCtrlLin_2

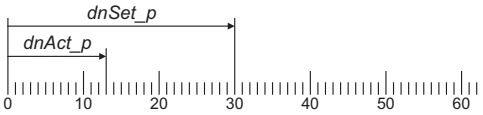
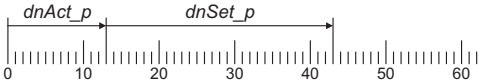
This FB serves to implement the following positioning functions:

- Bring drive to curve position (e.g. after mains connection, manual jog, homing)
- Release drive from curve and bring it to safe position (position override function)
- Positioning the X axis via X offset (higher-level positioning)
- Positioning the Y axis via Y offset (higher-level positioning)




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bExecute<br>BOOL | Execute positioning profile<br>• Only possible if the *bEnable* input is set to TRUE. | |
| | FALSE | No positioning/positioning is aborted.<br>• The speed at the *nNOut_v* output is braked to standstill via the deceleration ramp set in C01060/4 (*nNOut_v* = 0).<br>• At the same time, the *dnPosOut_p* output is stopped. |
| | FALSE↗TRUE | Positioning is executed/continued.<br>• In case of an activated limit stop (C01058/2 = TRUE), a renewed positive edge for a following positioning procedure is required. |
| bSetPos0<br>BOOL | Travel to zero position<br>• Only possible if the *bEnable* input is set to TRUE. | |
| | FALSE | The value at the *dnSet_p* input is used as setpoint position. |
| | TRUE | The value at the *dnSet_p* input is ignored. The setpoint position is internally set to "0" and the *dnPosOut_p* output can be travelled to zero position. |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bPosMode<br>BOOL | Positioning mode | |
| | FALSE | Absolute positioning<br>• The setpoint at *dnSet_p* is the absolute setpoint position (with regard to zero position):<br>  |
| | TRUE | Relative positioning<br>• With a FALSE/TRUE edge at the *bExecute* input, the outputs of the current position (*dnAct_p*) are traversed by the *dnSet_p* value:<br>  <br>**Note:**<br>Only to be used with activated limit stop (C01058/2 = TRUE)! |
| dnSet_p<br>DINT | Setpoint for positioning in [increments]<br>• Scaling: A revolution is displayed with 65536 increments or steps. | |
| dnAct_p<br>DINT | Actual position in [increments]<br>• Connect this input with e.g. *dnPosOut_p* of this FB or with another FB which outputs the actual position of the drive. | |
| bEnable<br>BOOL | Activate/deactivate positioning function<br>• This input has the highest priority. | |
| | FALSE | Positioning function deactivated.<br>• For behaviour see C01059/2. |
| | TRUE | Positioning function activated. |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bInTarget<br>BOOL | Status signal "Target position reached" | |
| | TRUE | Target position reached. |
| nNOut_v<br>INT | Speed output of the profile generator<br>• Scaling: 16384 ≡ 15000 rpm | |
| dnPosOut_p<br>DINT | Output position of the profile generator in [increments]<br>• Scaling: A revolution is displayed with 65536 increments or steps. | |

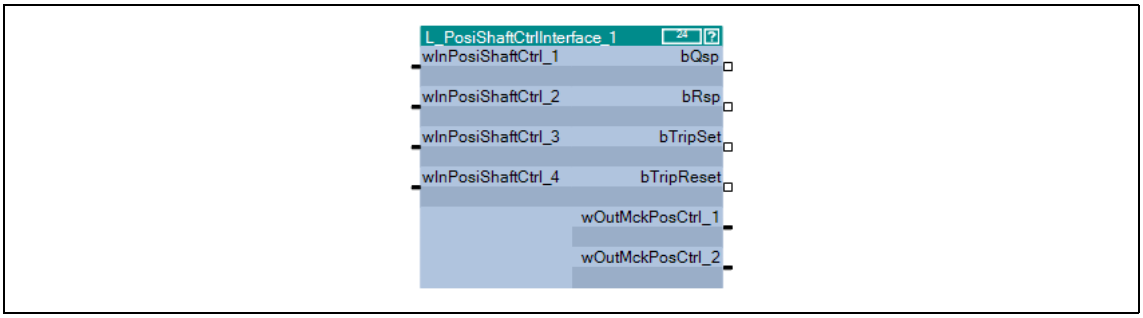
### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01058/2 | | | Limit stop |
| | 0 | deactivated<br>(Lenze setting) | The outputs continuously follow the setpoint at *dnSet_p*. |
| | 1 | activated | The outputs follow the setpoint at *dnSet_p* once.<br>• If the setpoint position changes, a new FALSE/TRUE edge is required at the *bExecute* input for another positioning process. |

| Parameters | Possible settings | | | | Info |
|---|---|---|---|---|---|
| C01059/2 | | | | | Positioning behaviour<br>• Behaviour with deactivated positioning function (*bEnable* = FALSE). |
| | 0 | *dnOut_p* = 0<br>(Lenze setting) | | | The *dnPosOut_p* output jumps to "0".<br>• The speed signal at the *nNOut_v* output maps this position step change as angle signal.<br>**Note:**<br>If the *nNOut_v* output has an impact on the setpoint generation for the drive, undefined motions (shocks, following errors) may occur. |
| | 1 | *dnOut_p/nNOut_v* follow *dnAct_p* | | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnAct_p* input (actual position). |
| | 2 | *dnOut_p/nNOut_v* follow *dnSet_p* | | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnSet_p* input (setpoint). |
| | 3 | *dnOut_p/nNOut_v* follow *dnAct_p*<br>(without limitation) | | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnAct_p* input (actual position), but without limitation. |
| C01060/4 | 0.010 | | s | 130.000 | Deceleration of set position<br>• Deceleration ramp for stop before reaching the setpoint position (*bExecute* = FALSE).<br>• Lenze setting: 1.000 s |
| C01060/5 | 0.010 | | s | 130.000 | Acceleration ramp<br>• Lenze setting: 1.000 s |
| C01060/6 | 0.010 | | s | 130.000 | Deceleration ramp<br>• Lenze setting: 1.000 s |
| C01061/3 | -15000 | | rpm | 15000 | Forward motion<br>• Positive speed<br>• Lenze setting: 200 rpm |
| C01061/4 | -15000 | | rpm | 15000 | Return motion<br>• Negative speed<br>• Lenze setting: 200 rpm |

### 19.1.148 L_PosiShaftCtrlInterface_1

FB in preparation!

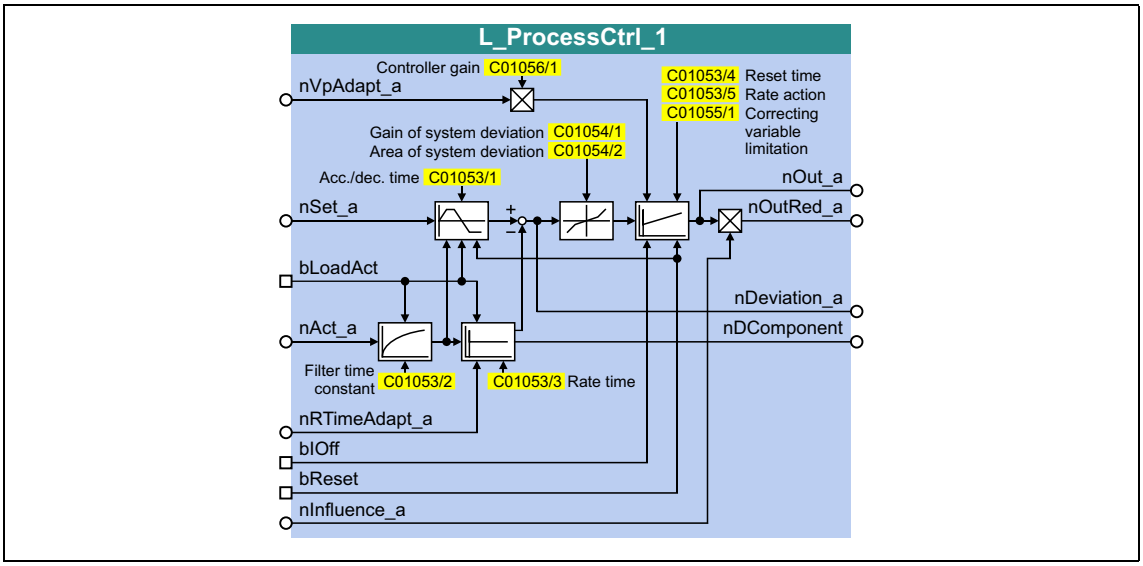


### 19.1.149 L_ProcessCtrl_1

This FB serves to implement a dancer position or tension control.

The FB is provided with the following functions:

- Adjustable control algorithm (P, PI, PID) with adaptable gain
- Reduced controller dynamics at low system deviation
- Setpoint ramp generator for preventing setpoint step-changes at the input
- Setpoint ramp generator can be loaded with actual value
- Low-pass filter and rate action in the actual value feedback
- Integral action component can be switched off
- Interruptible control




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nVpAdapt_a<br>INT | Proportional evaluation of the controller gain (Vp)<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to 0 ... 16384 (0 ... 100 %) |
| nSet_a<br>INT | Controller setpoint |
| bLoadAct<br>BOOL | Accept actual controller value<br>TRUE: The actual controller value *nAct_a* is taken over into the ramp generator, the low pass and the rate action. |
| nAct_a<br>INT | Actual controller value |
| nRTimeAdapt_a<br>INT | Proportional evaluation of the rate time in the actual value path<br>• Scaling: 16384 ≡ 100 % rate time (C01053)<br>• Internal limitation to 0 ... 16384 (0 ... 100 %) |
| bIOff<br>BOOL | Reset controller I component<br>TRUE: The controller I component is reset. |

| Designator<br>Data type | Information/possible settings |
|---|---|
| bReset<br>BOOL | Reset entire control<br>TRUE: All outputs are reset to 0. |
| nInfluence_a<br>INT | Proportional evaluation of the controller correcting variable *nOutRed_a*<br>• Scaling: 16384 ≡ 100 % controller correcting variable (*nOut_a*) |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Controller correcting variable<br>• Internal limitation to ±16384 (±100 %) |
| nOutRed_a<br>INT | Controller correcting variable (evaluated by *nInfluence_a*) |
| nDeviation_a<br>INT | System deviation<br>• Internal limitation to ±32767 (±199.99 %) |
| nDComponent<br>INT | D component of the rate action<br>• Internal limitation to ±16384 (±100 %) |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01053/1 | 0.000 | s | 30.000 | Acceleration/deceleration time of the setpoint ramp generator<br>• Lenze setting: 0.000 s |
| C01053/2 | 0.000 | s | 30.000 | Filter time constant for actual controller value<br>• Lenze setting: 0.000 s |
| C01053/3 | 0.000 | s | 30.000 | Rate time for actual controller value<br>• Lenze setting: 0.000 s |
| C01053/4 | 0.000 | s | 30.000 | Controller reset time<br>• Lenze setting: 1.000 s |
| C01053/5 | 0.000 | s | 30.000 | Controller rate action<br>• Lenze setting: 0.000 s |
| C01054/1 | 0.00 | % | 199.99 | Gain of the system deviation in the range of reduced sensitivity<br>• Lenze setting: 100.00 % |
| C01054/2 | 0.00 | % | 199.99 | Area of system deviation with reduced gain/sensitivity<br>• Lenze setting: 0.00 % |
| C01055/1 | | | | Limitation of the controller correcting variable to the positive area |
| | False | Limitation not active. (Lenze setting) | | |
| | True | Limitation active. | | |
| C01056/1 | 0.00 | | 100.00 | Controller gain<br>• Lenze setting: 0.10 |

### 19.1.149.1 Control characteristic

The dancer position or tension controller can be optionally operated as P, PI or PID controller. In the Lenze setting, the PI algorithm is active.

#### Gain (P component)

The input value is controlled by a linear characteristic. The slope of the characteristic is determined by the controller gain Vp.

The controller gain Vp is set in C01056/1.

- The controller gain can be adapted via the *nVpAdapt_a* input (also possible in online mode).
- The *nVpAdapt_a* input value has a direct effect on the controller gain:

$$P = nVpAdapt_a \cdot C01056/1$$

Example: With the parameterised controller gain Vp = 2.0 and *nVpAdapt_a* = 75 %, the resulting gain factor is as follows:

$$P = \frac{75\ [\%]}{100\ [\%]} \cdot 2.0 = 1.5$$

#### Reset time Tn (I component)

The adjustment time Tn is set under C01053/4.

- The I component of the controller can be deactivated by setting the input *bIOff* to TRUE.
- The I component can be switched on and off online.

#### Differential component Kd (D component)

The differential component Kd is set under C01053/5.

- The setting "0.0 s" deactivates the D component (Lenze setting). In this way, the PID controller becomes a PI controller or P controller, if the I component has been deactivated as well.

#### Evaluation of the output signal (controller influence)

If the motor speed or motor torque is precontrolled, a low influence is sufficient for the controller to comply with the setpoint.

Use the *nInfluence_a* input to select the influencing factor the controller correcting variable (*nOut_a*) is to be evaluated with. The evaluated controller correcting variable is output at *nOutRed_a*.

**Note!**

The controller influence evaluates the output signal multiplicatively. A change of the *nInfluence_a* influencing factor also changes the dynamics of the controller!

#### Switching on/off the controller

By setting the *bReset* input to TRUE, the process controller can be switched off.

#### Loading the setpoint ramp generator with the actual value

If the actual value is loaded into the setpoint ramp generator in switched-off state, it has the advantage that in the moment of the renewed controller enable, the system deviation is zero first. Thus, compensation processes can be mostly prevented.

When the *bLoadAct* input is set to TRUE, the setpoint ramp generator can be loaded with the actual value. This keeps the system deviation equal to zero and I component of the controller.

#### Low pass and rate action in the actual value path

In order to filter signal interferences more effectively, you can activate a low pass in the actual value path.

- The filter time constant for the low pass is set in C01053/2.
- The rate time constant in the feedback path can be set in C01053/3. This serves to compensate interfering decelerations.

#### Reduced controller dynamics at low system deviation

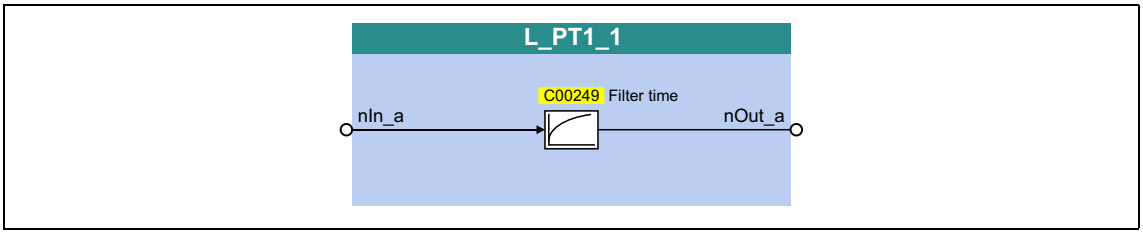
Due to a reduced controller dynamics with low system deviations, the dampening behaviour of the control loop is mostly influenced favourably.

- C01054/2 serves to determine the tolerance zone in which the system deviation is transmitted to the controller with a slight gain.
- C01054/1 serves to determine the amount of percent the gain in the defined tolerance zone is to be reduced to.

### 19.1.150 L_PT1_1

This FB filters and delays analog signals.

- The filter time constant T can be set under C00249.
- The gain is defined with $V_p = 1$.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

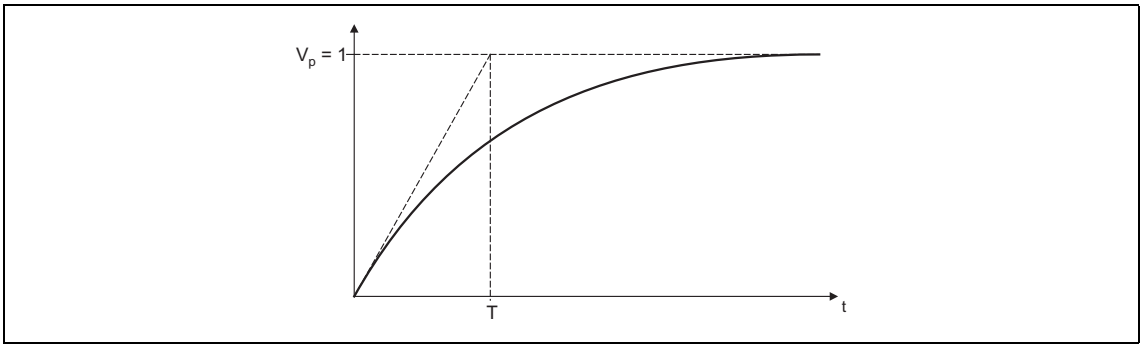
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00249 | 0 | ms | 5000 | Filter time constant<br>• The filter is not active with a setting of "0 ms". The input signal is passed through one-to-one to the output.<br>• Lenze setting: 2000 ms |

#### Function

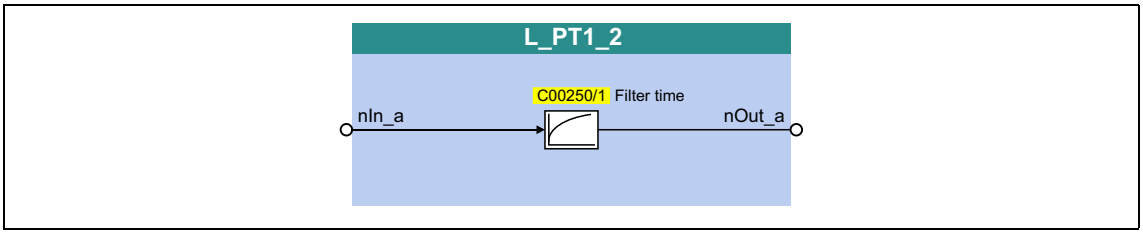



[19-66] Filter time constant T of the first-order delay element

### 19.1.151 L_PT1_2

This FB filters and delays analog signals.

- Filter time constant T can be set in C00250/1.
- The gain is defined with $V_p = 1$.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

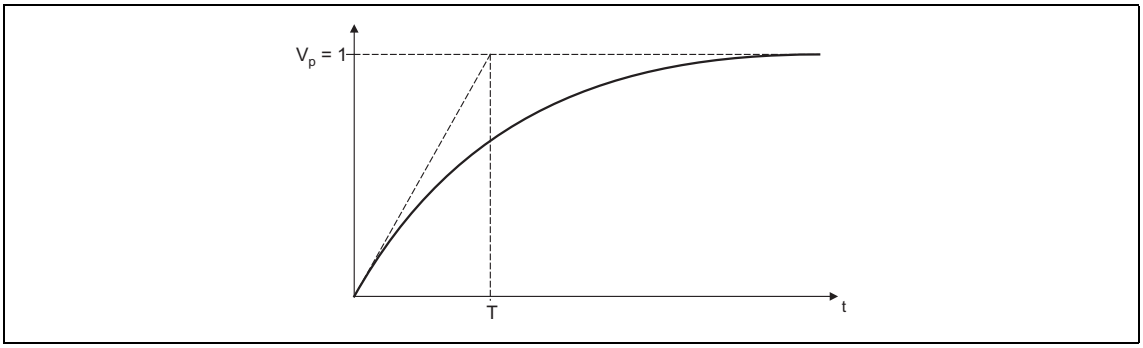
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00250/1 | 0 | ms | 5000 | Filter time constant<br>• The filter is not active with a setting of "0 ms". The input signal is passed through one-to-one to the output.<br>• Lenze setting: 2000 ms |

#### Function

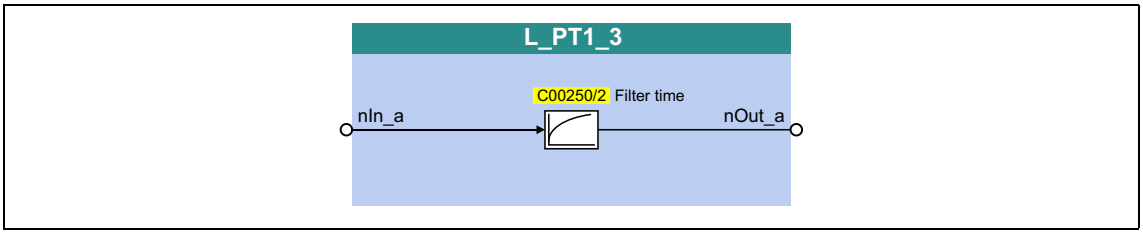



[19-67] Filter time constant T of the first-order delay element

### 19.1.152 L_PT1_3

This FB filters and delays analog signals.

- Filter time constant T can be set in C00250/2.
- The gain is defined with $V_p$ = 1.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

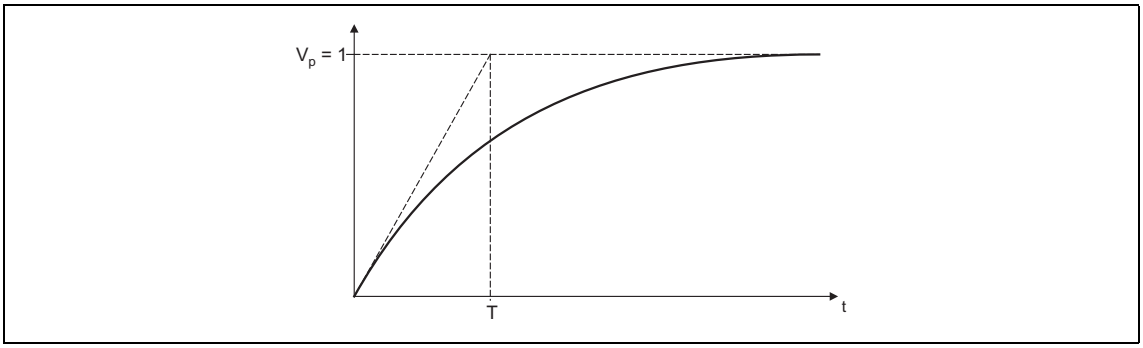
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00250/2 | 0 | ms | 5000 | Filter time constant<br>• The filter is not active with a setting of "0 ms". The input signal is passed through one-to-one to the output.<br>• Lenze setting: 2000 ms |

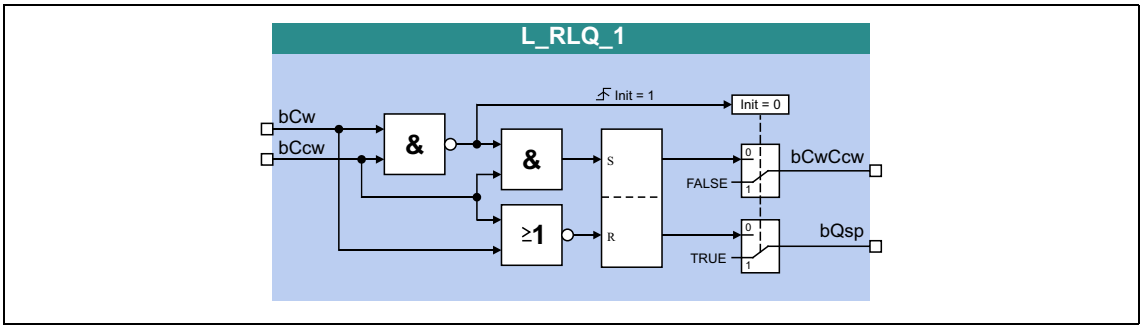
#### Function




[19-68] Filter time constant T of the first-order delay element

### 19.1.153 L_RLQ_1

This FB links a selected direction of rotation to the quick stop function with wire-break protection.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bCw<br>BOOL | Input<br>• TRUE = CW rotation |
| bCCw<br>BOOL | Input<br>• TRUE = CCW rotation |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bQSP<br>BOOL | Output signal for quick stop (QSP) |
| bCwCcw<br>BOOL | Output signal for CW/CCW rotation<br>• TRUE = CCW rotation |

#### Function

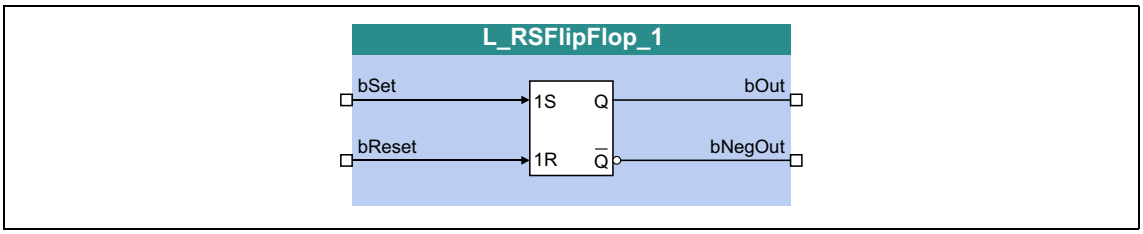
| Inputs | | Outputs | | Notes |
|---|---|---|---|---|
| bCw | bCCw | bCwCcw | bQSP | |
| TRUE | TRUE | FALSE | TRUE | The inputs have this status only if a TRUE signal is being applied to both inputs at the moment of switch-on!<br>See also FB illustration above, "Init" = 1. |
| If *one* of the inputs has the TRUE status, the following truth table applies: | | | | |
| FALSE | FALSE | FALSE | TRUE | See also FB illustration above, "Init" = 0. |
| TRUE | FALSE | FALSE | FALSE | |
| FALSE | TRUE | TRUE | FALSE | |
| TRUE | TRUE | X (save) | | |

[19-69] Truth table of the FB L_RLQ, 0 = FALSE, 1 = TRUE

### 19.1.154 L_RSFlipFlop_1

The functionality of this FB corresponds to that of an RS flipflop:

- An input signal at *bSet* serves to set the *bOut* output permanently.
- An input signal at *bReset* serves to reset the output.
- Additional, inverted output *bNegOut*




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bSet<br>BOOL | Setting input | |
| | TRUE | • The *bOut* output is set to TRUE.<br>• The *bNegOut* output is set to FALSE. |
| bReset<br>BOOL | Reset input | |
| | TRUE | • The *bOut* output is set to FALSE.<br>• The *bNegOut* output is set to TRUE. |

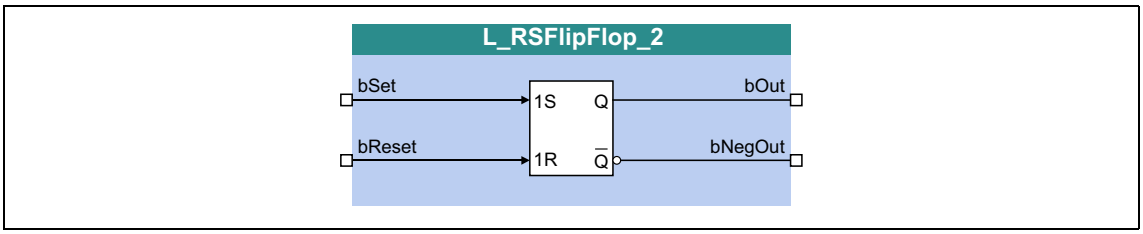
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |
| bNegOut<br>BOOL | Output signal, inverted |

### 19.1.155 L_RSFlipFlop_2

The functionality of this FB corresponds to that of an RS flipflop:

- An input signal at *bSet* serves to set the *bOut* output permanently.
- An input signal at *bReset* serves to reset the output.
- Additional, inverted output *bNegOut*




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bSet<br>BOOL | Setting input | |
| | TRUE | • The *bOut* output is set to TRUE.<br>• The *bNegOut* output is set to FALSE. |
| bReset<br>BOOL | Reset input | |
| | TRUE | • The *bOut* output is set to FALSE.<br>• The *bNegOut* output is set to TRUE. |

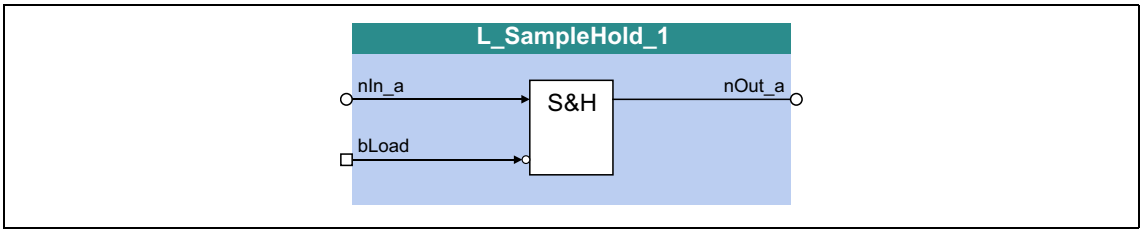
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |
| bNegOut<br>BOOL | Output signal, inverted |

### 19.1.156 L_SampleHold_1

This FB can save a signal.

- The saved value is also available after mains switching.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_a<br>INT | Input signal | |
| bLoad<br>BOOL | Save input signal | |
| | FALSE | The last-valid value at *nIn* is saved and output to *nOut*. A signal change at *nIn* does not cause a change at *nOut*. |
| | TRUE | The *nOut* output provides *dnIn*. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal |

#### Function

- When *bLoad* = TRUE, the *nIn_a* signal is switched to *nOut_a*.
- When *bLoad* = FALSE, the last-valid value is saved and output at *nOut_a*. A signal change at *nIn_a* does not cause a change at *nOut_a*.

#### Behaviour after mains switching

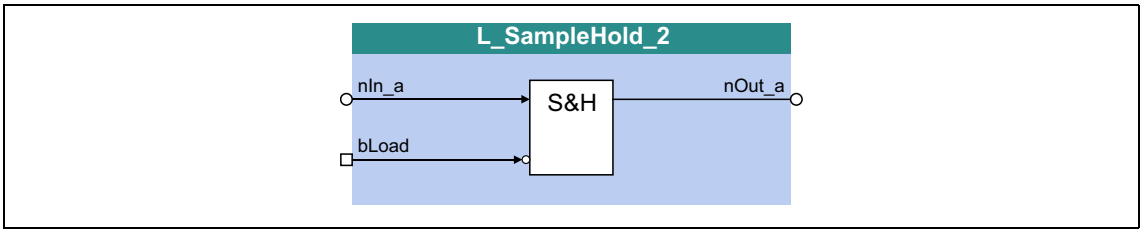
The last-loaded value is permanently stored after switching of the supply voltage and reloaded after restart.

- In order that the saved value is not immediately overwritten with the current input signal at *nIn* after restart, *bLoad* must be set to FALSE at restart.

### 19.1.157 L_SampleHold_2

This FB can save a signal.

- The saved value is also available after mains switching.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_a<br>INT | Input signal | |
| bLoad<br>BOOL | Save input signal | |
| | FALSE | The last-valid value at *nIn* is saved and output to *nOut*. A signal change at *nIn* does not cause a change at *nOut*. |
| | TRUE | The *nOut* output provides *dnIn*. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal |

#### Function

- When *bLoad* = TRUE, the *nIn_a* signal is switched to *nOut_a*.
- When *bLoad* = FALSE, the last-valid value is saved and output at *nOut_a*. A signal change at *nIn_a* does not cause a change at *nOut_a*.

#### Behaviour after mains switching

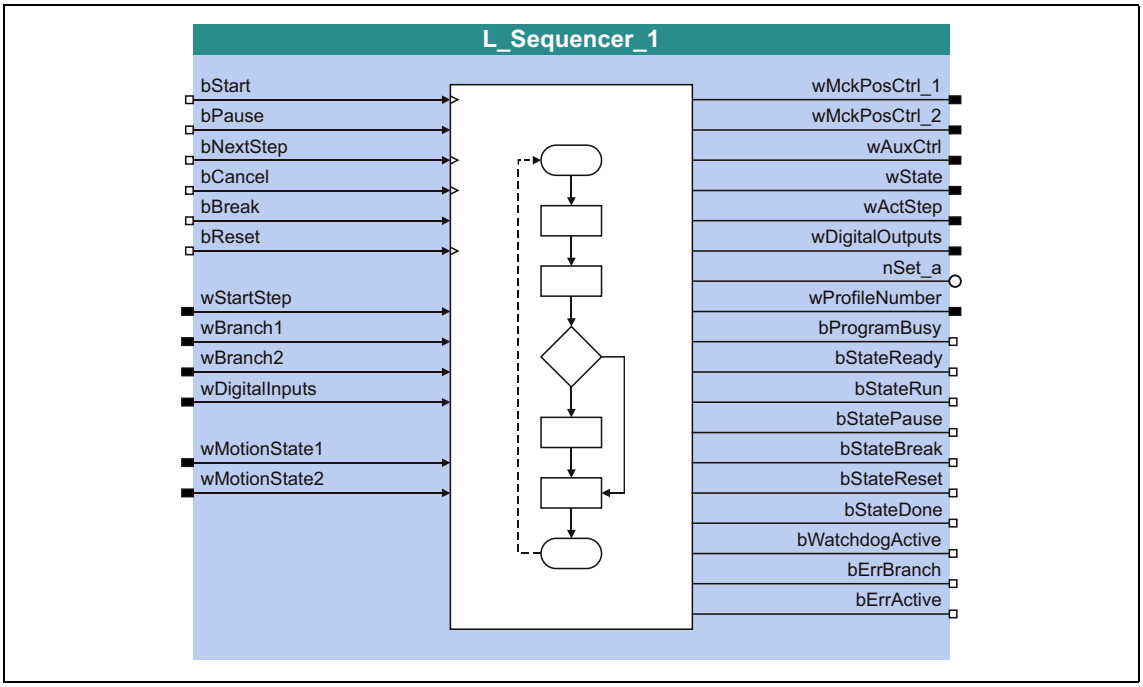
The last-loaded value is permanently stored after switching of the supply voltage and reloaded after restart.

- In order that the saved value is not immediately overwritten with the current input signal at *nIn* after restart, *bLoad* must be set to FALSE at restart.

### 19.1.158 L_Sequencer_1

This FB processes a positioning program by means of a sequence table.

- The sequence table is a central component of this FB and is represented by a code with 100 subcodes where each subcode references to a program segment which is called "action".
- Different action types are available which serve to realise, for instance, program branching, switching operation, waiting times and counters. The actions themselves are represented by codes as well.
  - The following action types are passive: "Branching", "variable branching", "switching", "setting the counter", "counting", "time waiting" and "program end". When these action types are being processed, the "Stop" operating mode is active and a brake, if available, will be applied.
  - The action types "positioning" and "homing", however, are active as these two action types trigger the execution of the respective basic function via the corresponding control inputs.
  - The "Stand-by" action type is active when a setpoint has been set for the speed follower or a setpoint position is transferred to the SB LS_MotionControlKernel when "position follower" has been selected.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bStart<br>BOOL | Start/continue positioning program | |
| | FALSE↗TRUE | The positioning program is started/continued.<br>• Counters and outputs are not reset automatically through this.<br>• A positioning program previously interrupted by *bBreak* is continued with the sequence step.<br>• The started positioning program is processed up to its program end even if *bStart* is reset to FALSE again. |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bPause<br>BOOL | Interrupt the current step of the positioning program | |
| | TRUE | • An active positioning is interrupted.<br>  • The drive is stopped with the current profile deceleration.<br>  • A deceleration override is considered.<br>• The sequence of an active timing element is stopped.<br>• An action of "Standby" type is not interrupted. |
| | FALSE | • The interrupted positioning is completed.<br>  • The distance of the profile already covered is considered here.<br>  • Positionings with final speed are not continued when the final speed cannot be reached anymore.<br>• The sequence of the stopped timing element is continued with the remaining time. |
| bNextStep<br>BOOL | Cancel the current step and go to the next step | |
| | FALSE↗TRUE | The current step of the positioning program is cancelled immediately and the positioning program is continued with the next step in the sequence table. |
| bCancel<br>BOOL | Cancel the current step and go to the step defined in C01404 | |
| | FALSE↗TRUE | The current step of the positioning program is cancelled immediately and the positioning program is continued wit the step defined in C01404. |
| bBreak<br>BOOL | Interrupt positioning program<br>• This input is intended for being connected to drive-relevant status signals (e.g. for controller inhibit or quick stop).<br>• If the interruption is deactivated again by a reset to FALSE, a new FALSE/TRUE edge is required at *bStart* in order that the positioning program is continued with the sequence step. | |
| | FALSE | Condition for program start via *bStart*. |
| | TRUE | The active positioning program is interrupted.<br>• If a positioning is active, the drive is brought to a standstill with the delay time for stop (C01251/1).<br>• The MCK operating modes Speed follower, Homing, Positioning and Position follower are deactivated.<br>• The digital output signals keep their states. |
| bReset<br>BOOL | Reset positioning program.<br>• Also possible with interrupted positioning program. | |
| | FALSE↗TRUE | The positioning program is reset.<br>• If a positioning is active, the drive is brought to a standstill with the delay time for stop (C01251/1) without considering an acceleration override.<br>• The program flow is cancelled ("program end").<br>• The digital output signals, counters and timing elements are reset.<br>• A possibly active action of the "Standby" type is aborted. |
| wStartStep<br>WORD | Step of the sequence table the positioning program is to start with. | |
| | 0 | Positioning program starts with step 1 of the sequence table. |
| | 1 ... 100 | First step |
| | Others | Positioning program does not start. |
| | not assigned | Positioning program starts with step 1 of the sequence table. |

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wBranch1<br>WORD | Input signal for action 1 of "Variable branching" type<br>• When the positioning program contains an action 1 of "Variable branching" type, branching is carried out in the corresponding step depending on the value available at this input: | |
| | 0 | Branching to the next step. |
| | 1 | Branching to the step defined in C01418/1. |
| | 2 | Branching to the step defined in C01419/1. |
| | ... | ... |
| | 20 | Branching to the step defined in C01437/1. |
| | >20 | Branching to the next step. |
| wBranch2<br>WORD | Input signal for action 2 of "Variable branching" type<br>• When the positioning program contains an action 2 of "Variable branching" type, branching is carried out in the corresponding step depending on the value available at this input: | |
| | 0 | Branching to the next step. |
| | 1 | Branching to the step defined in C01418/2. |
| | 2 | Branching to the step defined in C01419/2. |
| | ... | ... |
| | 20 | Branching to the step defined in C01437/2. |
| | >20 | Branching to the next step. |
| wDigitalInputs<br>WORD | Sequencer inputs 1 ... 16 (bit coded)<br>• The action types "positioning", "branching", "waiting" and "stand-by" have the "Input for..." parameter. If this is non-zero, it names the number of the sequencer input where the positioning program expects the respectively defined level before action is executed.<br>• For the transfer of individual control signals, e.g. the converter FB L_ConvBitsToWord_1 can be connected to this input. | |
| | Bit 0 | Sequencer input 1 |
| | Bit 1 | Sequencer input 2 |
| | ... | ... |
| | Bit 15 | Sequencer input 16 |
| wMotionState1<br>wMotionState2<br>WORD | Interface to the basic drive functions (MCK)<br>• Connect these inputs to the outputs with the same name of the SB LS_MotionControlKernel.<br>▸ Connection to the basic drive functions (MCK) | |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wMckPosCtrl1<br>wMckPosCtrl2<br>wAuxCtrl<br>WORD | Interface to the basic drive functions (MCK)<br>• Connect the outputs *wMckPosCtrl1* and *wMckPosCtrl2* to the inputs *wInMckPosCtrl1* and *wInMckPosCtrl2* des FB L_MckCtrlInterface_1.<br>• Connect the *wAucCtrl* output to the input with the same name of the SB LS_MotionControlKernel.<br>▸ Connection to the basic drive functions (MCK) |

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| wState<br>WORD | Status (bit coded) | |
| | Bit 1 | Positioning program ready to start/program end reached. |
| | Bit 2 | Positioning program is running. |
| | Bit 3 | Positioning program started, break active. |
| | Bit 4 | Positioning program stopped. |
| | Bit 5 | Positioning program reset. |
| | Bit 6 | Positioning program completed. |
| | Bit 7 | Time monitoring for "positioning" action has been triggered. |
| | Bit 8 | Wrong value at branch input. |
| | Bit 15 | Fault is active. |
| wActStep<br>WORD | Current step being processed by the positioning program. | |
| | 0 | Positioning program not started. |
| | 1 ... 100 | Current step |
| wDigitalOutputs<br>WORD | Sequencer outputs 1 ... 16 (bit coded)<br>• The sequencer outputs can be set to "0" or "1" by the "switching" action type during the runtime of the positioning program.<br>• For providing individual control signals, e.g. the converter FB L_ConvWordToBits_1 can be connected to this output. | |
| | Bit 0 | Sequencer output 1 |
| | Bit 1 | Sequencer output 2 |
| | ... | ... |
| | Bit 15 | Sequencer output 16 |
| nSet_a<br>INT | Speed setpoint which is output in the "Stand-by" action.<br>• If no "Stand-by" action is active, the value "0" is output. | |
| wProfileNumber<br>WORD | Display of the profile data set active in the current step | |
| bProgramBusy<br>BOOL | Status signal "Positioning program is running" | |
| | FALSE | Positioning program not started or stopped. |
| | TRUE | Positioning program is running. |
| bStateReady<br>BOOL | TRUE | Positioning program ready to start/program end reached. |
| bStateRun<br>BOOL | TRUE | Positioning program is running. |
| bStatePause<br>BOOL | TRUE | Positioning program started, break active. |
| bStateBreak<br>BOOL | TRUE | Positioning program stopped. |
| bStateReset<br>BOOL | TRUE | Positioning program reset. |
| bStateDone<br>BOOL | TRUE | Positioning program completed. |
| bWatchdogActive<br>BOOL | TRUE | Time monitoring for "positioning" action has been triggered. |
| bErrBranch<br>BOOL | TRUE | Wrong value at branch input. |
| bErrActive<br>BOOL | TRUE | Fault is active. |

### General parameters

**Note!**

The parameters of the different actions for the sequence table are described in the corresponding action subchapter.

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01400/1...100 | | | | Action in sequence step n<br>• In the subcodes, the calls of the actions required for the positioning program are stored. In this way, the basic sequence (with the exception of the branches) is defined. |
| | 0 | Program end | | |
| | xxyyy | Action call<br>• xx ≡ Action type<br>• yyy ≡ Action number | | |
| C01401 | 1 | | 100 | Current step<br>• Read only |
| C01402 | | | | Sequence control: Status<br>• Read only |
| | 0 | READY: Positioning program ready to start/program end reached. | | |
| | 1 | RUN: Positioning program is running. | | |
| | 2 | PAUSE: Current step interrupted, break is active. | | |
| | 3 | BREAK: Positioning program interrupted. | | |
| | 4 | RESET: Positioning program reset. | | |
| C01403 | | | | Sequence control: Control word<br>• Bit coded code for controlling the sequencer by a PC or a master control via parameter channel.<br>• By setting a bit to "1", the respective function is activated. |
| | Bit 0 | Start | | |
| | Bit 1 | Break | | |
| | Bit 2 | Break | | |
| | Bit 3 | Cancel | | |
| | Bit 4 | Reset | | |
| | Bit 5 | NextStep | | |
| C01404 | | | | Step takes place when *bCancel* = TRUE<br>• This step is processed if the *bCancel* input is set to TRUE. |
| | **0** | **Sequence step** | | |
| | 1...100 | Step 1 ... 100 | | |
| | 101 | Program end | | |

### 19.1.158.1 Connection to the basic drive functions (MCK)

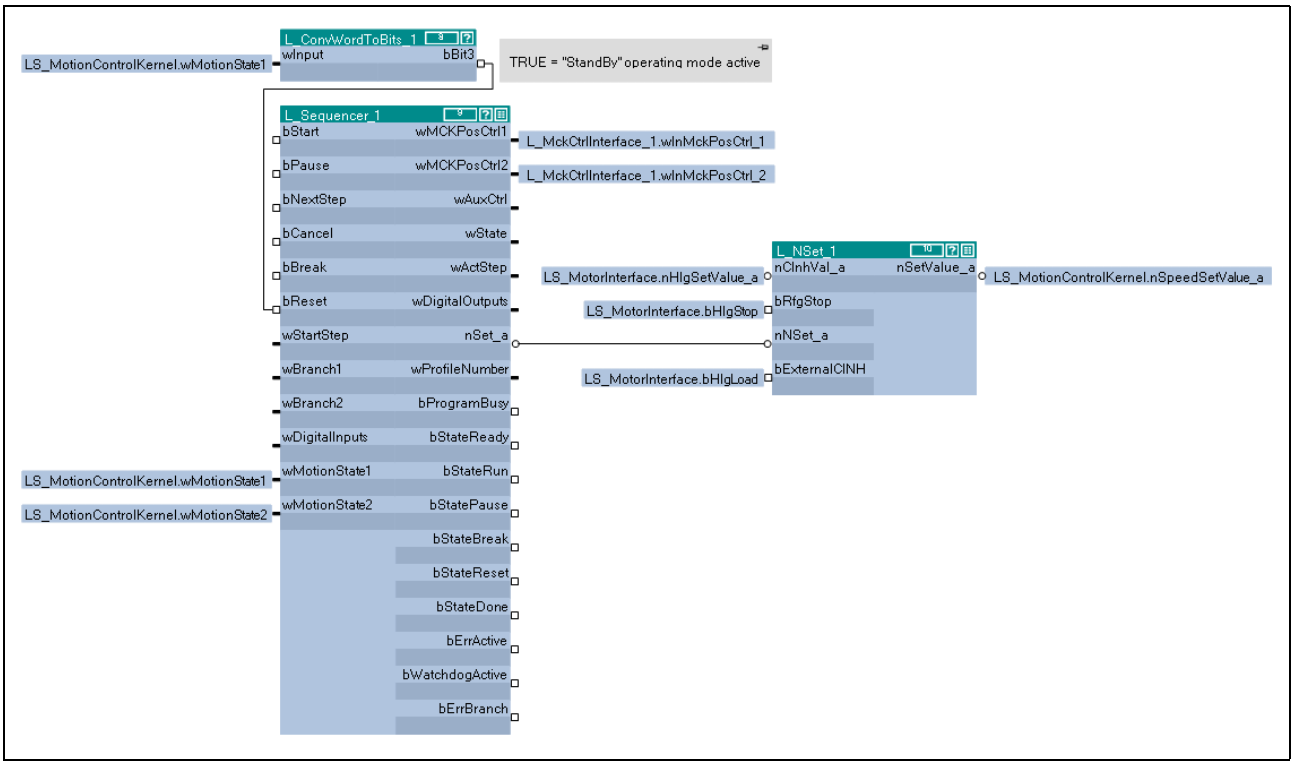
**Required connections in the FB Editor**

'The recommended starting basis for a connection with the FB **L_Sequencer_1** is the use of the TA "Table positioning".

The following configuration shows the required connections in order that the execution of the prevailing basic function is triggered in LS_MotionControlKernel for the two action types "Positioning" and "Referencing".

With a corresponding parameter setting, the *nSet_a* output serves to transfer a speed setpoint to the basic function "speed follower" if an action of Standby" type is active.

The process is reset in the MCK "Standby" operating mode. The internal "StandBy" operating mode is assumed implicitly if controller inhibit, pulse inhibit, quick stop and/or DC-injection braking are activated. ▸ "StandBy" operating mode (📖 578)



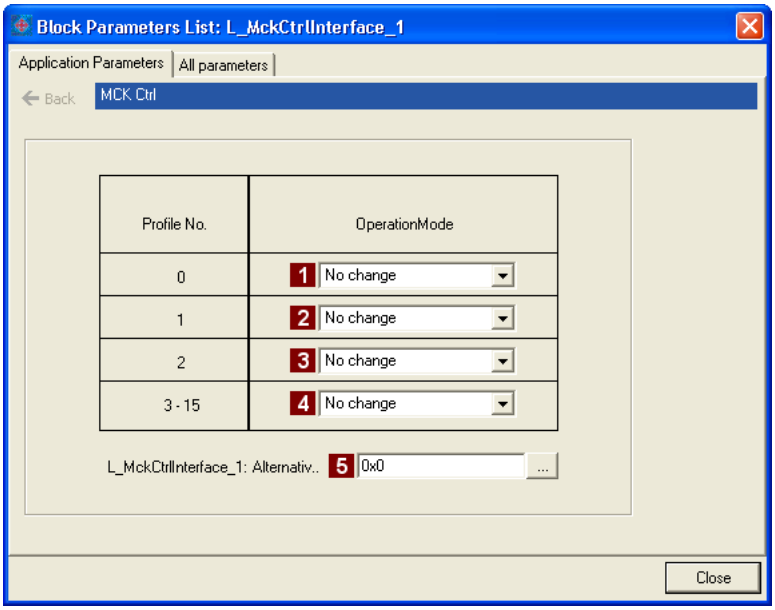

[19-70] Basic integration of the sequence control into the application

**Note!**

The FB **L_Sequencer_1** has to be called in the processing order before the FB L_MckCtrlInterface_1!

### Required parameter settings

For an error-free function of the sequence control, the following parameter settings are required for the FB L_MckCtrlInterface_1:




| Parameters | | Info |
|---|---|---|
| 1 | **MCK operating mode at profile no. 0** (C01298/1) | Selection of the operating mode for profile 0 ... 15 at L_MckCtrlInterface_1.<br>**For these four parameters, set the selection "0: No change" (**no assignment of an operating mode to a profile). |
| 2 | **MCK operating mode at profile no. 1** (C01298/2) | |
| 3 | **MCK operating mode at profile no. 2** (C01298/3) | |
| 4 | **MCK operating mode at profile no. 3...15** (C01298/4) | |
| 5 | **L_MckCtrlInterface_1: Alternative function** (C01297) | Bit coded selection of the Alternative functions for the bit 16 (PosExecute) in the MCK control word.<br>**For this parameter, set the hexadecimal value "0x0" to deactivate all alternative functions.** |

## 19.1.158.2 Priorities of the control signals

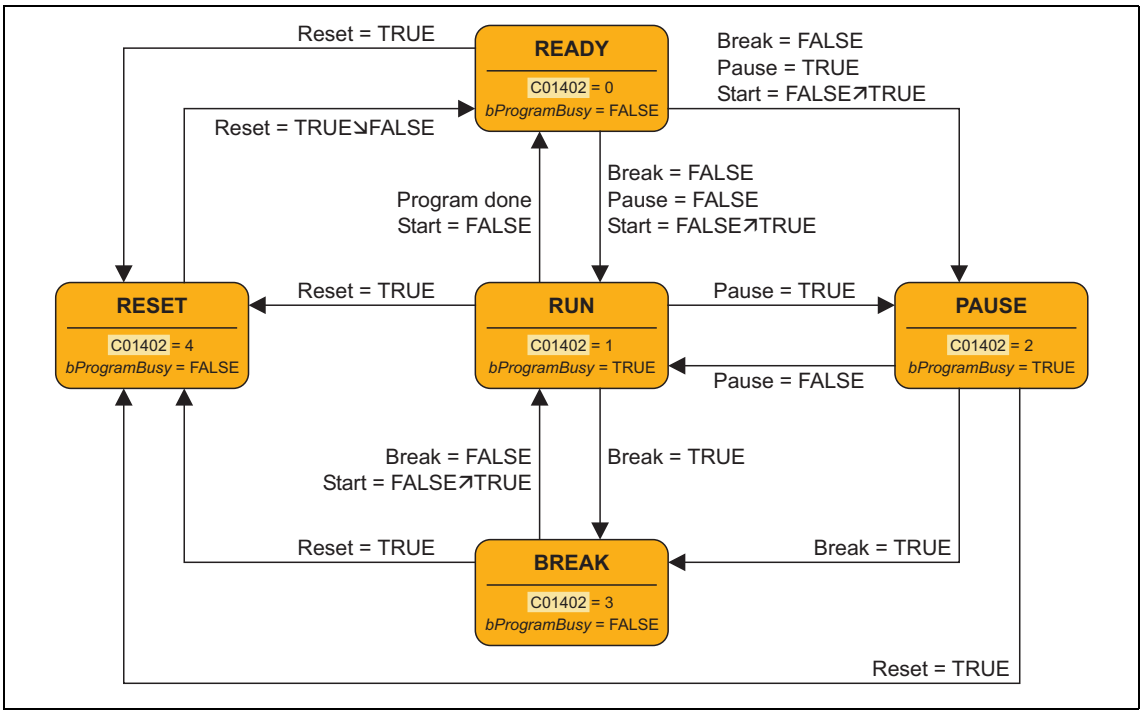
The boolean control signals have the following priorities (1 = highest priority):

1. *bReset* 2. *bBreak* 3. *bPause* 4. *bStart* 5. *bCancel*

### 19.1.158.3 States of the sequencer

The internal state machine of the sequencer distinguishes the following five states:

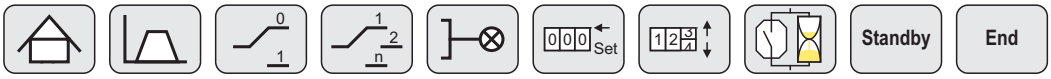
| Status | Display C01402 | Impact/meaning |
|---|---|---|
| **Ready** | 0 | When there is no interruption by *bBreak*, the program can be started. |
| **RUN** | 1 | The program is processed. |
| **Break** | 2 | The program is paused and the traversing of a profile is stopped.<br>• The outputs keep their states<br>• An active positioning is interrupted.<br>  • The drive is stopped with the current profile deceleration.<br>  • A acceleration override is not considered.<br>• After deactivating the break, the positioning program and a previously active positioning are continued at the same position. |
| **Break** | 3 | The traversing of a profile is cancelled.<br>• The drive with the delay time for stop (C01251/1) is brought to a standstill.<br>• The digital output signals keep their states.<br>• After the interruption has been deactivated (*bBreak* = FALSE), another start signal is required to continue the program with the sequence step. |
| **Reset** | 4 | The program is reset.<br>• An active positioning is cancelled.<br>  • The drive with the delay time for stop (C01251/1) is brought to a standstill.<br>  • A acceleration override is not considered.<br>• The program flow is cancelled ("program end").<br>• The digital output signals and the counter are reset.<br>• If a standby operation is active, it will be cancelled. |




[19-71] Internal state machine of the sequencer

### 19.1.158.4 Overview: Available action types

The program flow of the sequential positioning control is defined by means of a sequence table which can contain up to 100 references to "actions".




- An action comprises a clear functionality which is described with a few parameters.
- Different action types are available which serve to realise, for instance, program branching, switching operation, waiting times and counters.
- A certain number of actions are available from every action type which can be parameterised individually.
- An action can be called from several positions in the sequence table.
- After an action has been processed, the action entered in the sequence step of the sequence table is automatically processed unless a branching causes a jump to another step in the sequence table.
- Maximally one action can be processed per calculation cycle.
- The sequence table and the actions themselves are represented by parameters (codes with subcodes).
- Action calls are entered in the sequence table as a decimal number that contains the action type [A] and the action number [B]:




**Overview**

| Action | Action type [A] | Action number [B] | Number of available actions | Parameter range |
|---|---|---|---|---|
| Program end | 0 | 000 | 1 | - |
| Positioning | 1 | 001 ... 050 | 50 | C01405 ... C01410 |
| Switching | 2 | 001 ... 016 | 16 | C01411 ... C01414 |
| Branching | 3 | 001 ... 016 | 16 | C01415 ... C01417 |
| Variable branching | 4 | 001 ... 002 | 2 | C01418 ... C01437 |
| Homing | 5 | 000 | 1 | - |
| Time waiting | 7 | 001 ... 008 | 8 | C01438 ... C01440 |
| Counter setting | 8 | 001 ... 005 | 5 | C01441 ... C01442 |
| Counting | 9 | 001 ... 008 | 8 | C01443 ... C01448 |
| Stand-by | 10 | 001 ... 005 | 5 | C01449 ... C01452 |

You can find detailed some information with regard to the action types in the following subchapters.

### 19.1.158.5 Action type "Program end"




In order to define the program end within the sequence table, an action of "program end" type is available.

**Action call (entry in the sequence table)**

**0** (Leading zeros can be omitted.)

- When the *bStart* input is reset to FALSE while the positioning program is running, processing is only continued until program end is reached.
- In the Lenze setting, all entries of the sequence table are set to "0" and thus to "program end".

### 19.1.158.6 Action type "positioning"



For starting a positioning, 50 actions of "positioning" type are available. The Profile entry is made via the parameters of the basic function "Positioning".

#### Action call (entry in the sequence table)

| 1 | x | x | x |
|---|---|---|---|

with action number xxx = 001 ... 050

#### Parameters

The available actions 1 ... 50 are represented via the subcodes 1 ... 50.

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01405/1...50 | | | | Pos. action n: Starting with |
| | **0** | **Waiting function deactivated** | | In the default setting, execution of the profile is started immediately. |
| | 1 | Sequencer input 1 (Bit 0 of *wDigitalInputs*) | | "Wait for level": The execution of the profile is only started when the selected sequencer input has the polarity set in C01406/x. |
| | 2 | Sequencer input 2 (Bit 1 of *wDigitalInputs*) | | |
| | ... | ... | | |
| | 16 | Sequencer input 16 (Bit 15 of *wDigitalInputs*) | | |
| C01406/1...50 | | | | Pos. action n: Polarity input<br>• State which the sequencer input selected for the profile start must have. |
| | **0** | **Condition is bit state "0"** | | |
| | 1 | Condition is bit state "1" | | |
| C01407/1...50 | | | | Pos. action n: Profile number<br>• A sequence profile can be set in the corresponding profile parameter. |
| | **0** | **No profile executed** | | |
| | 1...15 | Execute profile no. 1 ... 15 | | |
| C01408/1...50 | | | | Pos. action n: Jump destination<br>• Step within the sequence table which is processed after the profile has been executed. |
| | **0** | **Sequence step** | | |
| | 1...100 | Step 1 ... 100 | | |
| C01409/1...50 | 0.000 | s | 2147480.000 | Pos. action n: Monitoring time<br>• If the positioning process lasts longer than the monitoring time set here, it is branched to the jump destination set in C01410/x. |
| | When "0.000 s" is set (Lenze setting), the time monitoring function is deactivated. | | | |
| C01410/1...50 | | | | Pos. action n: Jump destination monitoring<br>• Step which is executed after the monitoring time has been exceeded. |
| | **0** | **Sequence step** | | |
| | 1...100 | Step 1 ... 100 | | |

### 19.1.158.7 Action type "switching"



For switching digital output signals, 16 actions of the "Switch" type are provided. Each action can set two bits of the *wDigitalOutputs* output signal, which can be selected, to "0" or "1", alternatively and independently of each other.

#### Action call (entry in the sequence table)

| 2 | x | x | x |
|---|---|---|---|

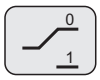
with action number xxx = 001 ... 016

#### Parameters

The available actions 1 ... 16 are represented via the subcodes 1 ... 16.

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01411/1...16 | | | Switching act. n: Output switching A<br>• Selection of the sequencer output which is to be set to the polarity set in C01412/x. |
| | **0** | **Deactivated** | |
| | 1 | Sequencer output 1 (Bit 0 of *wDigitalOutputs*) | |
| | 2 | Sequencer output 2 (Bit 1 of *wDigitalOutputs*) | |
| | ... | ... | |
| | 16 | Sequencer output 16 (Bit 15 of *wDigitalOutputs*) | |
| C01412/1...16 | | | Switching act. n: Pol. switching A<br>• State to which the sequencer output selected in C01411/x is to be set. |
| | **0** | **Set output bit to "0"** | |
| | 1 | Set output bit to "1" | |
| C01413/1...16 | | | Switching act. n: Output switching B<br>• Selection of the sequencer output which is to be set to the polarity set in C01414/x. |
| | **0** | **Deactivated** | |
| | 1 | Sequencer output 1 (Bit 0 of *wDigitalOutputs*) | |
| | 2 | Sequencer output 2 (Bit 1 of *wDigitalOutputs*) | |
| | ... | ... | |
| | 16 | Sequencer output 16 (Bit 15 of *wDigitalOutputs*) | |
| C01414/1...16 | | | Switching act. n: Pol. switching B<br>• State to which the sequencer output selected in C01413/x is to be set. |
| | **0** | **Set output bit to "0"** | |
| | 1 | Set output bit to "1" | |

### 19.1.158.8 Action type "Branching"




For conditional and unconditional branching (jumps), 16 actions of "Branching" Type are available.

#### Action call (entry in the sequence table)

| 3 | x | x | x |
|---|---|---|---|

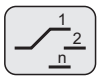
with action number xxx = 001 ... 016

#### Parameters

The available actions 1 ... 16 are represented via the subcodes 1 ... 16.

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01415/1...16 | | | Branching action n: Input for jump |
| | **0** | **Unconditional branch** | "Unconditional branch": It is always branched to the jump destination set in C01417/x. |
| | 1 | Sequencer input 1<br>(Bit 0 of *wDigitalInputs*) | "Conditional branch": It is only branched to the jump destination set in C01417/x if the selected sequencer input has the polarity set in C01416/x.<br>• Otherwise, the following step in the sequence table will be processed. |
| | 2 | Sequencer input 2<br>(Bit 1 of *wDigitalInputs*) | |
| | ... | ... | |
| | 16 | Sequencer input 16<br>(Bit 15 of *wDigitalInputs*) | |
| C01416/1...16 | | | Branching action n: Polarity input<br>• State of the selected sequencer input required for a conditional branch to the jump destination. |
| | **0** | **Condition is bit state "0"** | |
| | 1 | Condition is bit state "1" | |
| C01417/1...16 | | | Branching act. n: Jump destination<br>• In case of unconditional branch, it is always branched to the jump destination set here.<br>• In case of conditional branch, it is only branched to the jump destination set here if the sequencer input selected in C01415/x has the polarity set in C01416/x. Otherwise, the following step in the sequence table will be processed. |
| | **0** | **Sequence step** | |
| | 1...100 | Step 1 ... 100 | |

### 19.1.158.9 Action type "Variable branching"




For variable branching (jumps), two actions of "variable branching" type are available.

- Branching to one of 20 possible steps is carried out depending on the *wBranch1* input signal or *wBranch2* input signal at the time of processing.
  - *wBranch1* input signal defines branching for action 1.
  - *wBranch2* input signal defines branching for action 2.
- Example: If the *wBranch2* input signal has the value "15" at the time of processing of the action no. 2, it is branched to the step entered in the parameter "Var. branching n: Jump destination 15" for action no. 2 (C01432/2).

#### Action call (entry in the sequence table)

| 4 | x | x | x |
|---|---|---|---|

with action number xxx = 001 ... 002

#### Parameters

The available actions 1 ... 2 are represented via the subcodes 1 ... 2.

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C01418/1...2<br>...<br>C01437/1...2 | | | Var. branch n: Jump destination 1 ... 20<br>• Branch destinations 1 ... 20 for input values 1 ... 20 of *wBranch1...2*. |
| | **0** | **Branching deactivated**<br>(The following step in the sequence table is processed.) | |
| | 1...100 | Step 1 ... 100 | |

### 19.1.158.10Action type "Homing"



For starting a homing, an action of "homing" type is available.

- The settings for homing (e.g. homing mode) can be made via the parameters of the basic function "Homing".
- After homing is completed (*bHomingDone* = TRUE), the program flow is continued with the next step.

#### Action call (entry in the sequence table)

| | | | |
|---|---|---|---|
| 5 | 0 | 0 | 0 |

#### Parameters

The "homing" action has no own parameters.

### 19.1.158.11Action type "Time waiting"



For inserting waiting times into the program flow, 8 actions of "Time waiting" are available. The sequence step will only be processed after a waiting time has elapsed or when a selectable sequencer input has a certain bit state ("0" or "1").

#### Action call (entry in the sequence table)

| 7 | x | x | x |
|---|---|---|---|

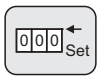
with action number xxx = 001 ... 008

#### Parameters

The available actions 1 ... 8 are represented via the subcodes 1 ... 8.

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01438/1...8 | 0.000 | s | 2127480.000 | Waiting act. n: Waiting time<br>• Initialisation: 1.000 s |
| C01439/1...8 | | | | Waiting act. n: Input for "Next" |
| | **0** | **Input deactivated** | | "Wait": Sequence step is only processed after the waiting time set has elapsed. |
| | 1 | Sequencer input 1<br>(Bit 0 of *wDigitalInputs*) | | "Wait for level": Sequence step will only be processed when the selected sequencer input has the polarity set in C01440/x.<br>**Note!**<br>However, if a waiting time > 0 s is set, the sequence step is processed at the latest after the waiting time set has elapsed. |
| | 2 | Sequencer input 2<br>(Bit 1 of *wDigitalInputs*) | | |
| | ... | ... | | |
| | 16 | Sequencer input 16<br>(Bit 15 of *wDigitalInputs*) | | |
| C01440/1...8 | | | | Waiting act. n: Polarity input<br>• State which the sequencer input selected must have so that the sequence step is processed. |
| | **0** | **Condition is bit state "0"** | | |
| | 1 | Condition is bit state "1" | | |

### 19.1.158.12Action type "Counter setting"




For setting one of the 5 available counters to a certain starting value, 5 actions of "Counter setting" type are available.

- The 5 actions of "counter setting" type are not permanently assigned to the 5 counters.
- You can, e.g., set a counter to a value using an action of "counter setting" type and at a later program time set the same counter to a different value using another action of "counter setting" type.
- The current counter content of the counters 1 ... 5 is displayed in C01443/1...5.

#### Action call (entry in the sequence table)

| 8 | x | x | x |
|---|---|---|---|

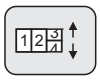
with action number xxx = 001 ... 005

#### Parameters

The available actions 1 ... 5 are represented via the subcodes 1 ... 5.

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01441/1...5 | | | | Counter setting act. n: Counter no. |
| | 0 | Counter setting deactivated | | |
| | **1** | **Counter 1** | | |
| | ... | ... | | |
| | 5 | Counter 5 | | |
| C01442/1...5 | -2147483647 | | 2147483647 | Counter setting act. n: Starting value |

### 19.1.158.13Action type "Counting"



For counting processes, 8 actions of "counting" type are available.

- Every processing of the action causes an increase or decrease of the counter content of the corresponding counter by the set step value (counting up or down).
- If the comparison condition for comparing the counter content to an adjustable comparison value is fulfilled, it can be branched to any step.
- For setting a counter to a starting value, 5 actions of "counter setting" type are available.
- The current counter content of the counters 1 ... 5 is displayed in C01443/1...5.

#### Action call (entry in the sequence table)

| 9 | x | x | x |
|---|---|---|---|

with action number xxx = 001 ... 008

#### Parameters

The available actions 1 ... 8 are represented via the subcodes 1 ... 8.

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01443/1...5 | -2147483647 | | 2147483647 | Counter n: Current counter content<br>• Read only |
| C01444/1...8 | | | | Counting act. n: Counter no. |
| | 0 | Counting deactivated<br>(Sequence step is processed.) | | |
| | **1** | **Counter 1** | | |
| | ... | ... | | |
| | 5 | Counter 5 | | |
| C01445/1...8 | -2147483647 | | 2147483647 | Counting act. n: Step value<br>• Value by which the counter is increased or decreased.<br>• Initialisation: 1 |
| C01446/1...8 | -2147483647 | | 2147483647 | Counting act. n: Comparison value<br>• Value to which the counter is compared. |
| C01447/1...8 | | | | Counting act. n: Jump destination<br>• If the set comparison condition is fulfilled, it is branched to the step set here.<br>• If the condition is not fulfilled or "0" setting, the step followed in the sequence table is processed. |
| | **0** | **Sequence step** | | |
| | 1...100 | Step 1 ... 100 | | |
| C01448/1...8 | | | | Counting act. n: Comparison operation |
| | **1** | **Counter content = comparison value** | | |
| | 2 | Counter content > comparison value | | |
| | 3 | Counter content ≥ comparison value | | |
| | 4 | Counter content < comparison value | | |
| | 5 | Counter content ≤ comparison value | | |

### 19.1.158.14"Stand-by" action type




For a temporary activation of a setpoint follower, 5 actions of "stand-by" type are available.

- The corresponding setpoint follower remains enabled until the condition for completing stand-by is fulfilled.
- If the "stand-by" step is active, the setpoint set in C01452 is provided at the *nSet_a* output (Lenze setting: 0 %) .

#### Action call (entry in the sequence table)

| 1 | 0 | x | x | x |
|---|---|---|---|---|

with action number xxx = 001 ... 005

#### Parameters

The available actions 1 ... 5 are represented via the subcodes 1 ... 5.

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01449/1...5 | | | | Stand-by act. n: Input for "End"<br>• The setpoint follower selected in C01451/x remains enabled until the sequencer input selected here has the polarity set in C01450/x. |
| | **1** | **Sequencer input 1** (Bit 0 of *wDigitalInputs*) | | |
| | 2 | Sequencer input 2 (Bit 1 of *wDigitalInputs*) | | |
| | ... | ... | | |
| | 16 | Sequencer input 16 (Bit 15 of *wDigitalInputs*) | | |
| C01450/1...5 | | | | Stand-by act. n: Polarity input<br>• State which the sequencer input selected must have so that "Standby" is exited and the sequence step is processed. |
| | **0** | **Condition is bit state "0"** | | |
| | 1 | Condition is bit state "1" | | |
| C01451/1...5 | | | | Stand-by act. n: Follower mode<br>• Selection of the follower that is to be activated in standby mode. |
| | **0** | Operation Speed follower | | |
| | 1 | Operation Position follower | | |
| C01452/1...5 | -200 | % | 200 | Stand-by act. n: Setpoint<br>• Setpoint for the Speed follower (provided at the output *nSet_a*).<br>• Initialisation: 0 % |

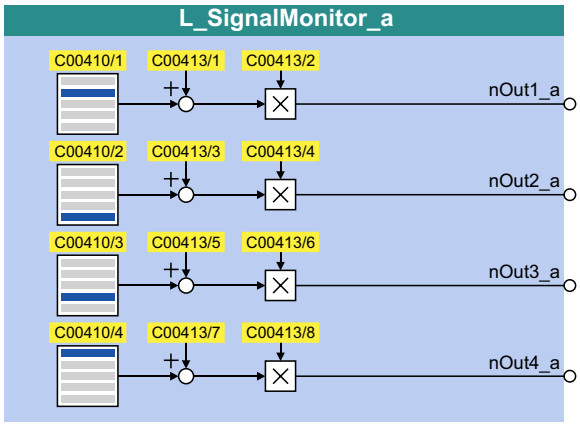
### 19.1.158.15Example: Sequence table

| Parameters | Entry | ≡ Action type/action no. | | Meaning |
|---|---|---|---|---|
| C01400/1 | 5000 |  | - | Execute homing |
| C01400/2 | 3001 |  | 1 | Process action 1 of "branching" (3) type |
| C01400/3 | 9005 |  | 5 | Process action 5 of "Counting" type (9) |
| C01400/4 | 1001 |  | 1 | Process action 1 of "positioning" type (1) |
| C01400/5 | 7003 |  | 3 | Process action 3 of "Time waiting" type (7) |
| C01400/6 | 1002 |  | 2 | Process action 2 of "positioning" type (1) |
| C01400/7 | 3002 |  | 2 | Process action 2 of "branching" (3) type |
| C01400/8 | 9005 |  | 5 | Process action 5 of "Counting" type (9) |
| C01400/9 | 3004 |  | 4 | Process action 4 of "branching" (3) type |
| C01400/... | yyxxx | yy | xxx | Process action *xxx* of *yy* type |
| C01400/24 | 1003 |  | 3 | Process action 3 of "positioning" type (1) |
| C01400/25 | 0 |  | - | Program end |
| C01400/26 | 0 |  | - | Program end |
| C01400/... | ... | ... | ... | ... |
| C01400/100 | 0 |  | - | Program end |

### 19.1.159 L_SignalMonitor_a

This FB outputs four analog signals which can be selected from a list of analog output signals of all function blocks provided in the device.

- Offset and gain of the source signals are adjustable.




#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nOut1_a<br>INT | Output signal<br>• Internal limitation to ±32767 |
| nOut2_a<br>INT | Output signal<br>• Internal limitation to ±32767 |
| nOut3_a<br>INT | Output signal<br>• Internal limitation to ±32767 |
| nOut4_a<br>INT | Output signal<br>• Internal limitation to ±32767 |

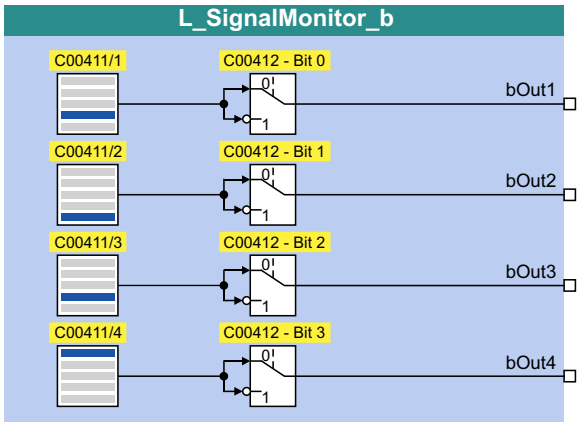
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00410/1<br>...<br>C00410/4 | See selection list - analog signals | | | Selection of the signal sources for *nOut1_a ... nOut4_a* |
| C00413/1<br>C00413/3<br>C00413/5<br>C00413/7 | -199.99 | % | +199.99 | Offset |
| C00413/2<br>C00413/4<br>C00413/6<br>C00413/8 | -199.99 | % | +199.99 | Gain |

### 19.1.160 L_SignalMonitor_b

This FB outputs four binary signals which can be selected from a list of binary output signals of all function blocks provided in the device.

- Inversion of the output signals can be set.




#### Outputs

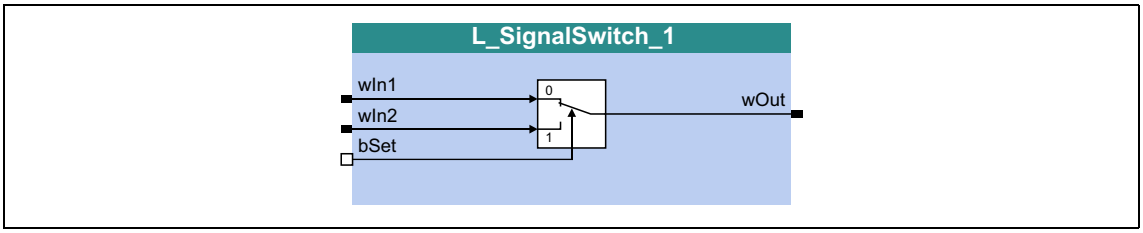
| Designator<br>Data type | Value/meaning |
|---|---|
| bOut1<br>...<br>bOut4<br>BOOL | Output signal<br>FALSE / TRUE |

#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00411/1<br>...<br>C00411/4 | See selection list - digital signals | | Selection of the signal sources for *bOut1 ... bOut4* |
| C00412 | Value is bit-coded: | | Inversion<br>• Bit set = inversion active |
| | Bit 0 | bOut1 inverted | |
| | Bit 1 | bOut2 inverted | |
| | Bit 2 | bOut3 inverted | |
| | Bit 3 | bOut4 inverted | |
| | Bit 4 | Reserved | |
| | Bit 5 | Reserved | |
| | Bit 6 | Reserved | |
| | Bit 7 | Reserved | |

### 19.1.161 L_SignalSwitch_1

This FB switches between two input signals of the "WORD" data type. The switch-over is controlled by means of a boolean input signal.




#### Inputs

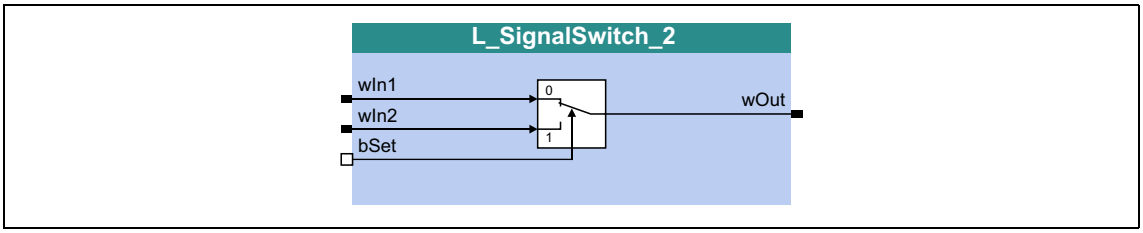
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wIn1<br>WORD | Input signal 1 | |
| wIn2<br>WORD | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *wOut* | |
| | FALSE | *wIn1* |
| | TRUE | *wIn2* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### 19.1.162 L_SignalSwitch_2

This FB switches between two input signals of the "WORD" data type. The switch-over is controlled by means of a boolean input signal.




#### Inputs

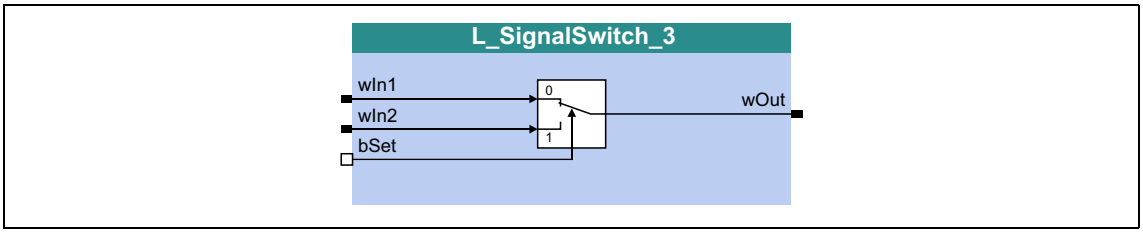
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wIn1<br>WORD | Input signal 1 | |
| wIn2<br>WORD | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *wOut* | |
| | FALSE | *wIn1* |
| | TRUE | *wIn2* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### 19.1.163 L_SignalSwitch_3

This FB switches between two input signals of the "WORD" data type. The switch-over is controlled by means of a boolean input signal.




#### Inputs

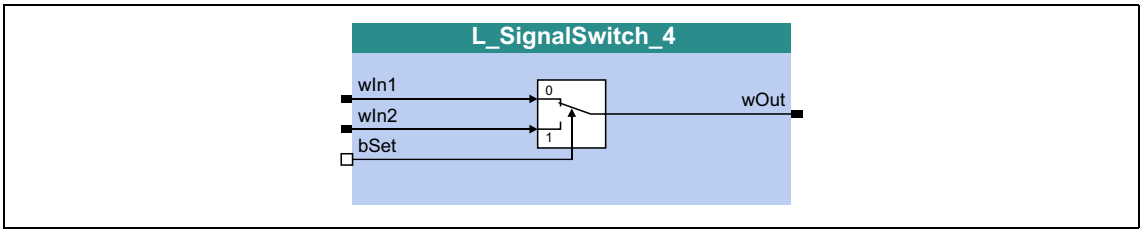
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wIn1<br>WORD | Input signal 1 | |
| wIn2<br>WORD | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *wOut* | |
| | FALSE | *wIn1* |
| | TRUE | *wIn2* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### 19.1.164 L_SignalSwitch_4

This FB switches between two input signals of the "WORD" data type. The switch-over is controlled by means of a boolean input signal.




#### Inputs

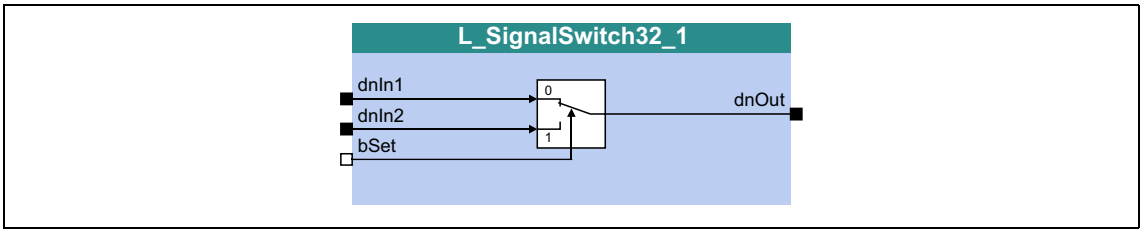
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| wIn1<br>WORD | Input signal 1 | |
| wIn2<br>WORD | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *wOut* | |
| | FALSE | *wIn1* |
| | TRUE | *wIn2* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### 19.1.165 L_SignalSwitch32_1

This FB switches between two input signals of the "DINT" data type. The switch-over is controlled by means of a boolean input signal.




#### Inputs

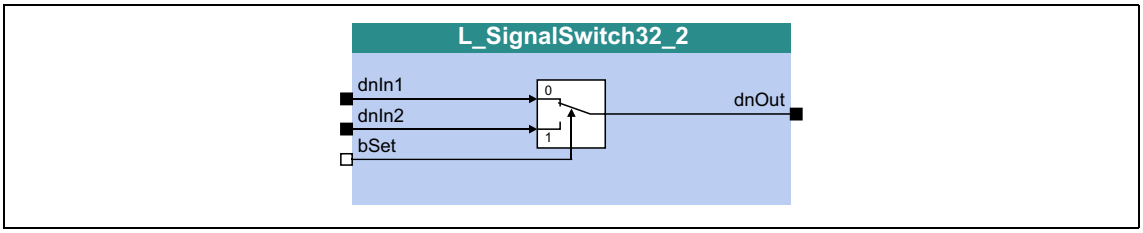
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn1<br>DINT | Input signal 1 | |
| dnIn2<br>DINT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *dnOut* | |
| | FALSE | *dnIn1* |
| | TRUE | *dnIn2* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut<br>DINT | Output signal |

### 19.1.166 L_SignalSwitch32_2

This FB switches between two input signals of the "DINT" data type. The switch-over is controlled by means of a boolean input signal.




#### Inputs

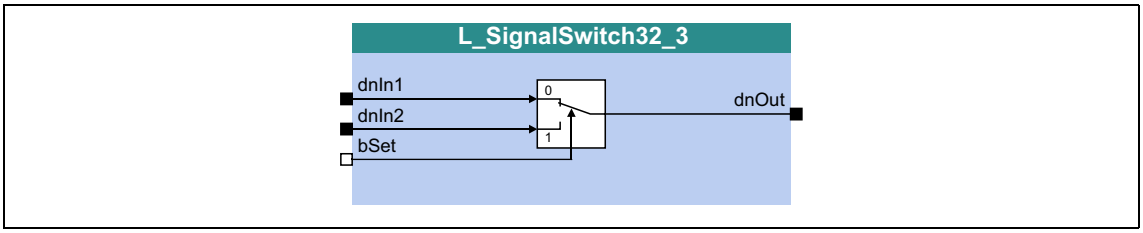
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn1<br>DINT | Input signal 1 | |
| dnIn2<br>DINT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *dnOut* | |
| | FALSE | *dnIn1* |
| | TRUE | *dnIn2* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut<br>DINT | Output signal |

### 19.1.167 L_SignalSwitch32_3

This FB switches between two input signals of the "DINT" data type. The switch-over is controlled by means of a boolean input signal.




#### Inputs

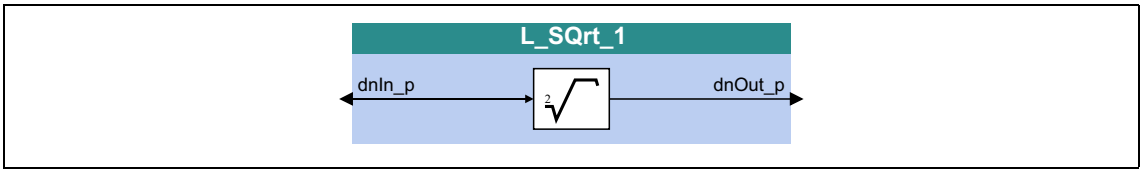
| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn1<br>DINT | Input signal 1 | |
| dnIn2<br>DINT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *dnOut* | |
| | FALSE | *dnIn1* |
| | TRUE | *dnIn2* |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut<br>DINT | Output signal |

### 19.1.168 L_SQrt_1

This FB outputs the square root of the DINT input value.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

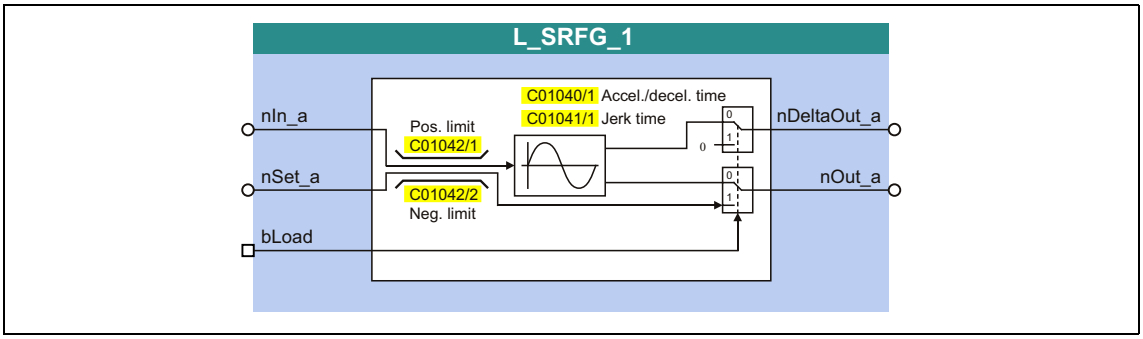
#### Function

$$dnOut_p = \sqrt{dnIn_p}$$

### 19.1.169 L_SRFG_1

This FB is a ramp function generator with S-shaped ramps for limiting the increase over time of analog signals. The S-shape of the ramps is a result of trapezoidal acceleration.

- The ramp function generator is provided with a setting function so that a value can directly be loaded into the internal ramp generator.
- The balanced acceleration/deceleration time is set in C01040.
- An S-ramp time can be set in C01041 for jerk-free acceleration until maximum acceleration is reached.
- At the *nDeltaOut_a* output, the dy/dt slope of the *nOut_a* output signal is output.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_a<br>INT | Input signal | |
| nSet_a<br>INT | Starting value for the ramp function generator<br>• Will be accepted if *bLoad* = TRUE | |
| bLoad<br>BOOL | Initialise ramp function generator | |
| | FALSE | With the set acceleration/deceleration time, the ramp function generator switches over from the value loaded via *nSet_a* to the value at *nIn_a*. |
| | TRUE | At the *nOut_a* output, *nSet_a* is output.<br>• *nDeltaOut_a* remains at 0 %. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nDeltaOut_a<br>INT | Acceleration of the ramp function generator<br>• Internal limitation to ±100 % (100 % ≡ 16384) |
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01040/1 | 0.001 | s | 999.999 | Acceleration/Deceleration time<br>• Initialisation: 100.000 s |
| C01041/1 | 0.001 | s | 50.000 | S-ramp time<br>• Initialisation: 0.200 s |

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01042/1 | -199.99 | s | 199.99 | Pos. limit<br>• Initialisation: 100.00 % |
| C01042/2 | -199.99 | s | 199.99 | Neg. limit<br>• Initialisation: -100.00 % |

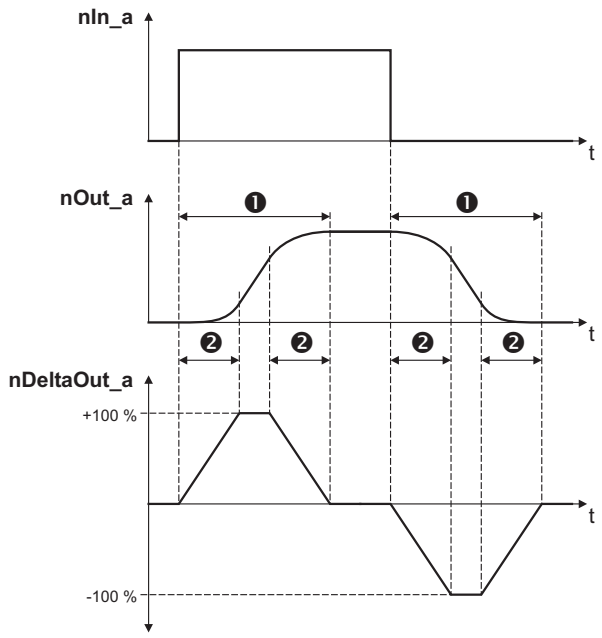
### Loading the ramp function generator

When *bLoad* is set to TRUE, the ramp function generator is loaded with the signal at *nSet_a*. This value is accepted immediately and output at *nOut_a*. There is no acceleration or deceleration via an S shape. As long as *bLoad* = TRUE, the ramp function generator remains inhibited.

### Acceleration/Deceleration time and S-ramp time

The acceleration/deceleration time and the S-ramp time for jerk-free acceleration can be set separately.

- Acceleration/Deceleration time = Time until the *nOut_a* output value has reached the *nIn_a* input value.
- S-ramp time = Time until the ramp function generator operates at maximum acceleration.



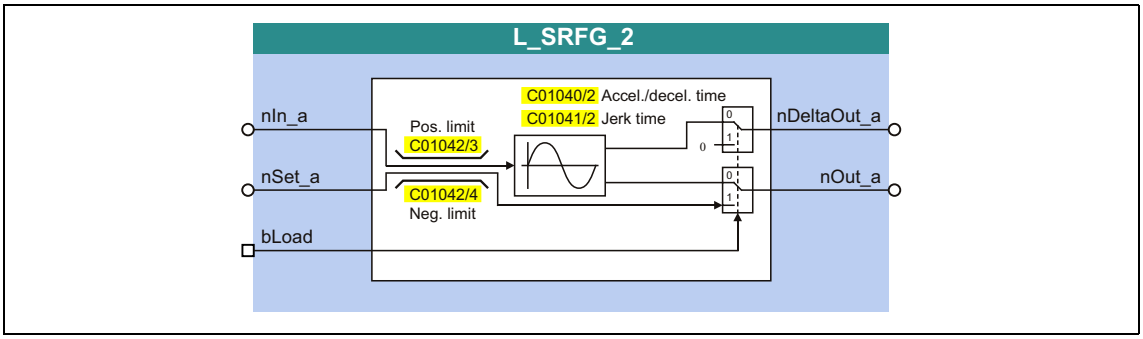

❶ C01040/x: Acceleration/Deceleration time
❷ C01041/x: S-ramp time

[19-72] Signal flow

### 19.1.170 L_SRFG_2

This FB is a ramp function generator with S-shaped ramps for limiting the increase over time of analog signals. The S-shape of the ramps is a result of trapezoidal acceleration.

- The ramp function generator is provided with a setting function so that a value can directly be loaded into the internal ramp generator.
- The balanced acceleration/deceleration time is set in C01040.
- An S-ramp time can be set in C01041 for jerk-free acceleration until maximum acceleration is reached.
- At the *nDeltaOut_a* output, the dy/dt slope of the *nOut_a* output signal is output.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_a<br>INT | Input signal | |
| nSet_a<br>INT | Starting value for the ramp function generator<br>• Will be accepted if *bLoad* = TRUE | |
| bLoad<br>BOOL | Initialise ramp function generator | |
| | FALSE | With the set acceleration/deceleration time, the ramp function generator switches over from the value loaded via *nSet_a* to the value at *nIn_a*. |
| | TRUE | At the *nOut_a* output, *nSet_a* is output.<br>• *nDeltaOut_a* remains at 0 %. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| nDeltaOut_a<br>INT | Acceleration of the ramp function generator<br>• Internal limitation to ±100 % (100 % ≡ 16384) |
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01040/2 | 0.001 | s | 999.999 | Acceleration/Deceleration time<br>• Initialisation: 100.000 s |
| C01041/2 | 0.001 | s | 50.000 | S-ramp time<br>• Initialisation: 0.200 s |

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01042/3 | -199.99 | s | 199.99 | Pos. limit<br>• Initialisation: 100.00 % |
| C01042/4 | -199.99 | s | 199.99 | Neg. limit<br>• Initialisation: -100.00 % |

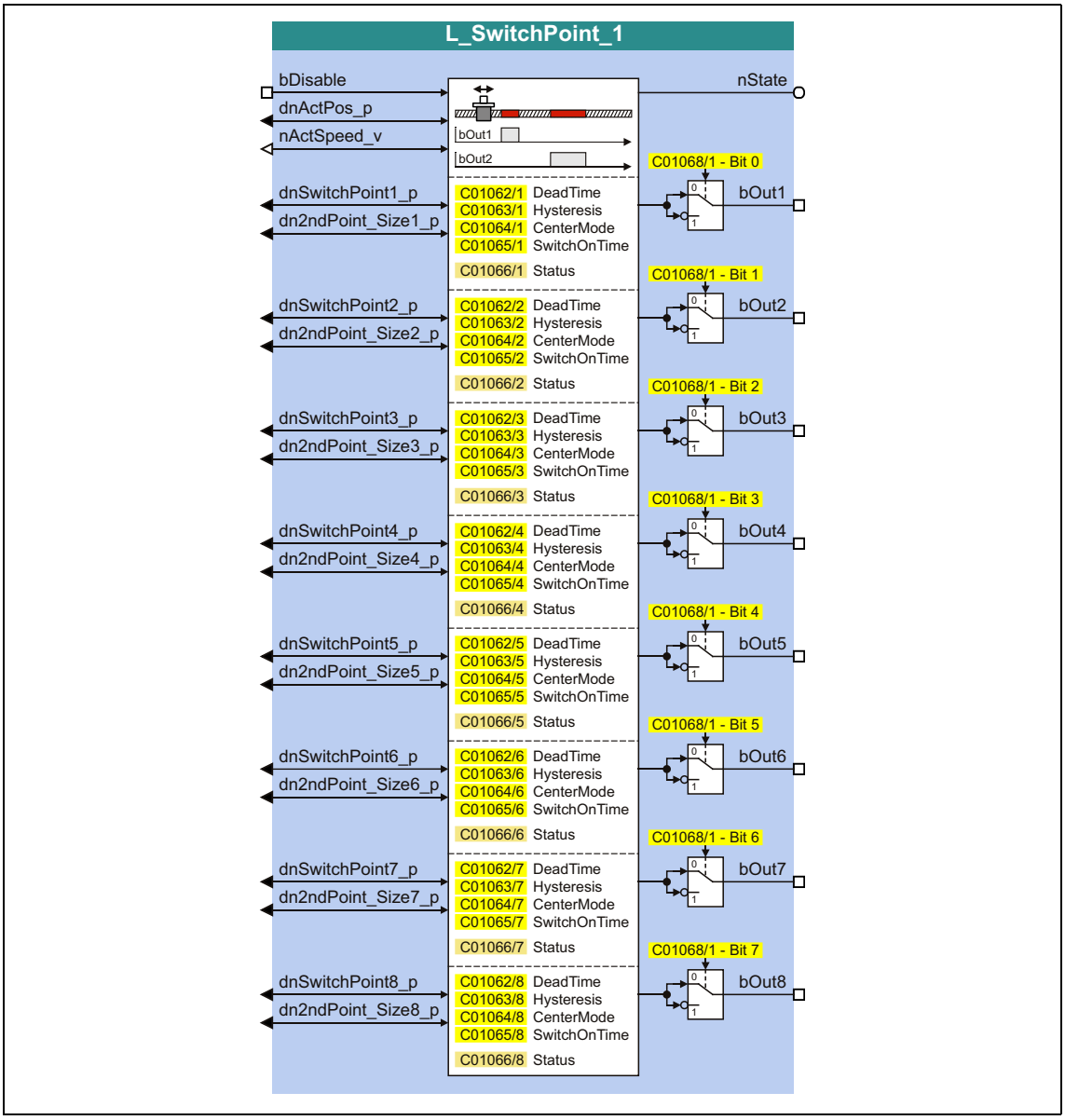


For a detailed functional description see L_SRFG_1.

### 19.1.171 L_SwitchPoint_1

This FB provides position switch points, i.e. digital switches the binary statuses (FALSE/TRUE) of which depend on the actual position.

- From version 12.00.00, eight instead of four position switch points are available.
- A position switch point serves to start peripherals as paint nozzles or knifes depending on the tool position.
- Moreover the FB supports the compensation of delay times of external switching elements (dead time compensation).
- By setting a running time, position/time-based cams can be realised as well.



### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bDisable<br>BOOL | Deactivate position switch points<br>• This input has the highest priority. | |
| | TRUE | Position switch points are deactivated.<br>• Outputs *bOut1* ... *bOut8* = FALSE |
| dnActPos_p<br>DINT | Actual position in [increments] | |
| nActSpeed_v<br>INT | Actual speed in [increments/ms]<br>• Scaling: 16384 ≡ 15000 rpm | |
| dnSwitchPoint**x**_p<br>DINT | Position switch point **x**: Position of the first switching point in [increments] | |
| dn2ndPoint_Size**x**_p<br>DINT | Position switch point **x**: Position of the second switching point or size of the switching window depending on the CenterMode set in C01064/x: | |
| | When CenterMode = FALSE:<br>Position of the second switching point in [increments]<br>• *dn2ndPoint_Size_p* must be higher than *dnSwitchPoint_p*. | |
| | When CenterMode = TRUE:<br>Size of the switching window in [increments]<br>• Due to the symmetrical arrangement of the window around the first switching point, uneven values are rounded to even values.<br>• Only positive values are permissible. | |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut1 ... bOut8<br>BOOL | Switching output 1 ... 8 | |
| | TRUE | The actual position is inside the defined switching window. |
| nState<br>INT | Status | |
| | 1 | FB is not active |
| | 2 | OK |
| | 3 | The data form resulting from the switching points are not plausible (*dnSwitchPoint_p*, *dn2ndPoint_Size_p*, parameterised hysteresis). |

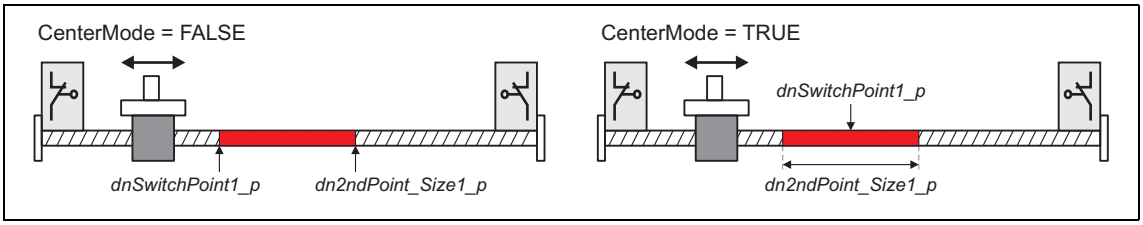
### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01062/1...8 | 0 | µs | 65535 | Dead time for dead time compensation<br>• The resulting switching position is not subject to a plausibility check.<br>• Lenze setting: 0 µs |
| C01063/1...8 | 0 | Incr. | 65535 | Switching hysteresis<br>• Lenze setting: 0 incr. |
| C01064/1...8 | | | | CenterMode<br>• Definition how the *dn2ndPoint_Size_p* selection is interpreted. |
| | False | *dn2ndPoint_Size_p* defines the second switching point.<br>(Lenze setting) | | |
| | True | *dn2ndPoint_Size_p* defines the size of the switching window. | | |

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01065/1...8 | 0 | ms | 60000 | Running time for position/time-based cams<br>• Lenze setting: 0 ms |
| | 0 ms = position-based cam | | | |
| C01066/1...8 | | | | Status 1 ... 4<br>• Read only. |
| | 0 | OK | | |
| | 10 | FB is not active | | |
| | 100 | Switching points not plausible | | |
| C01068/1<br>From version 12.00.00 | Setting is bit coded: | | | Inversion of outputs<br>• Lenze setting: 0x0000 |
| | Bit 0 | Inversion of output 1 | | |
| | Bit 1 | Inversion of output 2 | | |
| | Bit 2 | Inversion of output 3 | | |
| | Bit 3 | Inversion of output 4 | | |
| | Bit 4 | Inversion of output 5 | | |
| | Bit 5 | Inversion of output 6 | | |
| | Bit 6 | Inversion of output 7 | | |
| | Bit 7 | Inversion of output 8 | | |
| | Bit 8 | Reserved | | |
| | ... | | | |
| | Bit 15 | | | |

### 19.1.171.1 Definition of the switching range

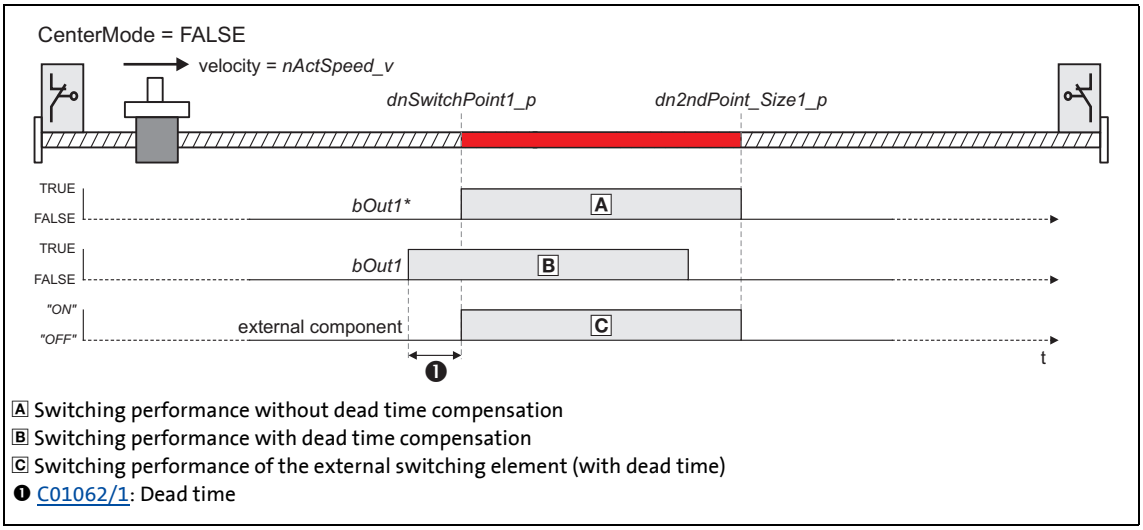
Depending on the setting of the CenterMode (C01064/x), the switching range can be defined via a start and end position or via the data of the center point/size of switching range:




[19-73] Definition of the switching range

### 19.1.171.2 Dead time compensation

The delay time of external switching elements can be compensated by setting the corresponding delay time in C01062/x. From this input and the current speed ,the FB calculates a preliminary stop, i.e. the *bOut* switching output is set correspondingly earlier.

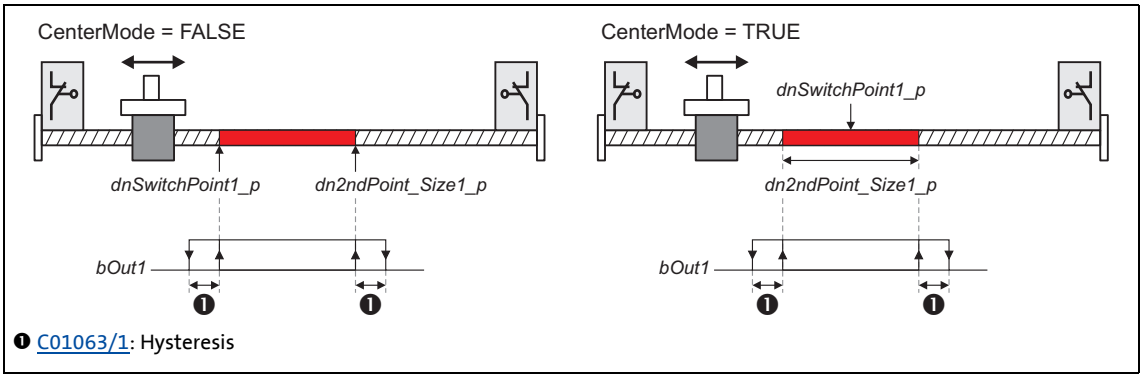



A Switching performance without dead time compensation
B Switching performance with dead time compensation
C Switching performance of the external switching element (with dead time)
❶ C01062/1: Dead time

[19-74] Dead time compensation

### 19.1.171.3 Switching hysteresis

Setting a hysteresis in C01063/x serves to prevent a permanent state change of the *bOut* switching output that may be caused by actual position value changes due to a mechanical irregularity at the axis.




❶ C01063/1: Hysteresis

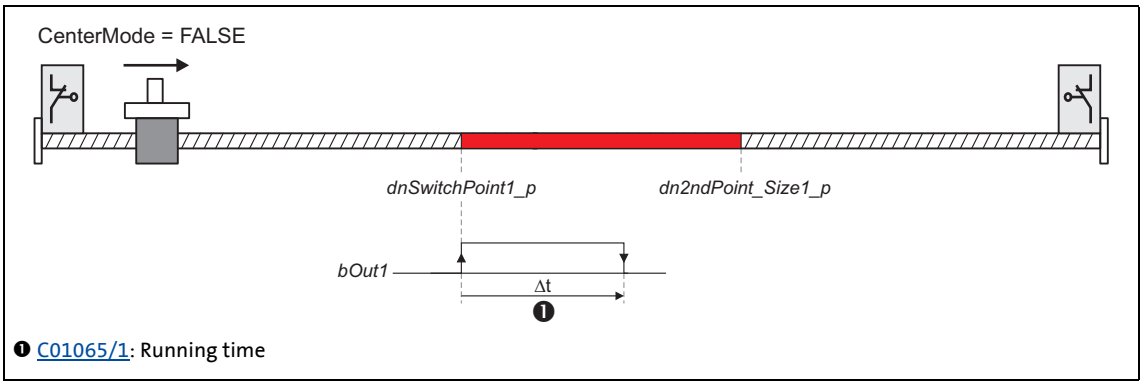
[19-75] Switching hysteresis

**Tip!**

Please note that a delay time set in C01062/x shifts the switching points including the hysteresis.

### 19.1.171.4 Position/time-based cams

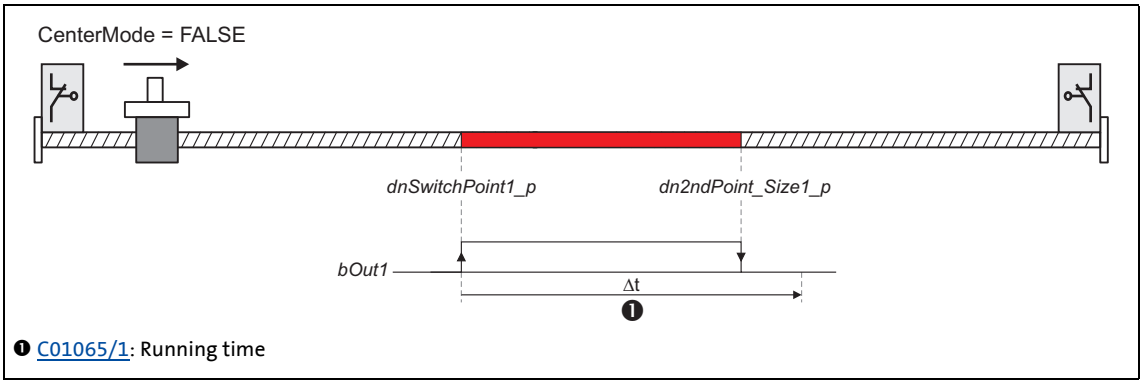
The setting of a running time in C01065/x higher 0 ms serves to realise "position/time-based cams": If the drive reaches the switching range, the *bOut* switching output is set to TRUE and reset to FALSE after the set switching time has elapsed.




❶ C01065/1: Running time

[19-76] Position/time-based cams (here without switching hysteresis and dead time compensation)

If the switching range is left before the running time has expired, the *bOut* switching output is reset to FALSE.




❶ C01065/1: Running time

[19-77] Position/time-based cams: Behaviour when the switching range is left before the running time has expired

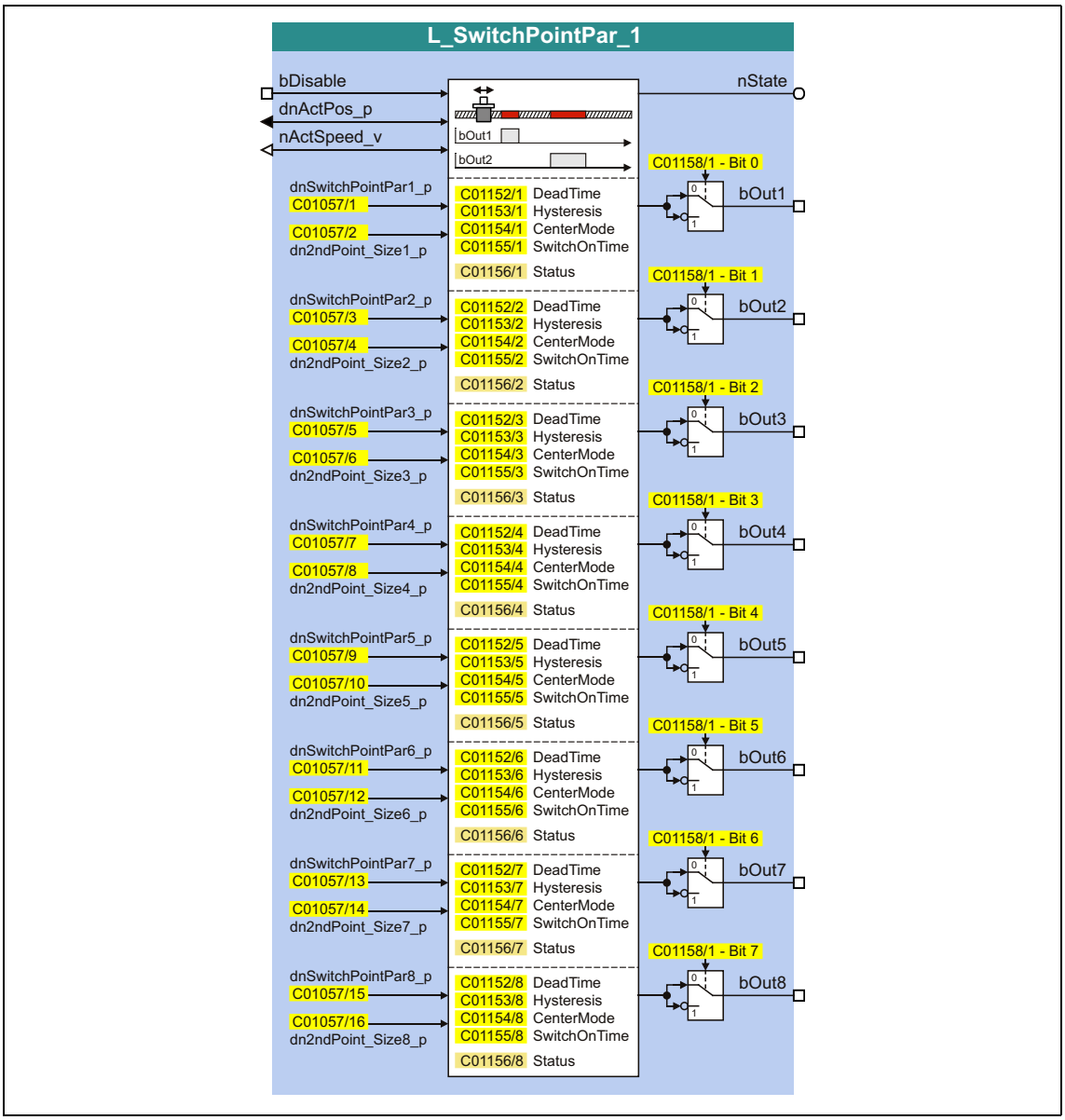
**Tip!**

Please note that for a resetting of the switching output the switching range must be left first.

### 19.1.172 L_SwitchPointPar_1

This FB provides position switch points, i.e. digital switches the binary statuses (FALSE/TRUE) of which depend on the actual position.

- This FB has the same function as the FB L_SwitchPoint_1. However, for this FB the position switch point positions must be specified via parameters instead of FB inputs.
- A position switch point serves to start peripherals as paint nozzles or knifes depending on the tool position.
- Moreover the FB supports the compensation of delay times of external switching elements (dead time compensation).
- By setting a running time, position/time-based cams can be realised as well.



### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bDisable<br>BOOL | Deactivate position switch points<br>• This input has the highest priority. | |
| | TRUE | Position switch points are deactivated.<br>• Outputs *bOut1* ... *bOut8* = FALSE |
| dnActPos_p<br>DINT | Actual position in [increments] | |
| nActSpeed_v<br>INT | Actual speed in [increments/ms]<br>• Scaling: 16384 ≡ 15000 rpm | |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bOut1 ... bOut8<br>BOOL | Switching output 1 ... 8 | |
| | TRUE | The actual position is inside the defined switching window. |
| nState<br>INT | Status | |
| | 1 | FB is not active |
| | 2 | OK |
| | 3 | The data form resulting from the switching points are not plausible (*dnSwitchPoint_p*, *dn2ndPoint_Size_p*, parameterised hysteresis). |

### Parameters

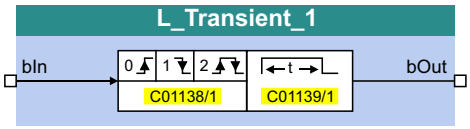
| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01152/1...8 | 0 | µs | 65535 | Dead time for dead time compensation<br>• The resulting switching position is not subject to a plausibility check.<br>• Lenze setting: 0 µs |
| C01153/1...8 | 0 | Incr. | 65535 | Switching hysteresis<br>• Lenze setting: 0 incr. |
| C01154/1...8 | | | | CenterMode<br>• Definition how the *dn2ndPoint_Size_p* selection is interpreted. |
| | False | *dn2ndPoint_Size_p* defines the second switching point. (Lenze setting) | | |
| | True | *dn2ndPoint_Size_p* defines the size of the switching window. | | |
| C01155/1...8 | 0 | ms | 60000 | Running time for position/time-based cams<br>• Lenze setting: 0 ms |
| | 0 ms = position-based cam | | | |
| C01156/1...8 | | | | Status 1 ... 4<br>• Read only. |
| | 0 | OK | | |
| | 10 | FB is not active | | |
| | 100 | Switching points not plausible | | |

<table>
<tr><th>Parameters</th><th colspan="4">Possible settings</th><th>Info</th></tr>
<tr><td rowspan="2">C01157/1...16</td><td colspan="2">0.0000</td><td>units</td><td>214748.0000</td><td>Position switch point positions<br>• Lenze setting: 0.0000 units</td></tr>
<tr><td colspan="4">dnSwitchPointx_p = position of the first switching point for position switch point x.<br>dn2ndPoint_Sizex_p = position of the second switching point or size of the switching window for position switch point x, depending on the CenterMode set in C01154/x:<br>• When CenterMode = FALSE: Position of the second switching point<br>• dn2ndPoint_Sizex_p must be higher than dnSwitchPointx_p.<br>• When CenterMode = TRUE: Size of the switching window<br>• Due to the symmetrical arrangement of the window around the first switching point, uneven values are rounded to even values.<br>• Only positive values are permissible.</td></tr>
<tr><td rowspan="12">C01158/1</td><td colspan="4">Setting is bit coded:</td><td rowspan="12">Inversion of outputs<br>• Lenze setting: 0x0000</td></tr>
<tr><td>Bit 0</td><td colspan="3">Inversion of output 1</td></tr>
<tr><td>Bit 1</td><td colspan="3">Inversion of output 2</td></tr>
<tr><td>Bit 2</td><td colspan="3">Inversion of output 3</td></tr>
<tr><td>Bit 3</td><td colspan="3">Inversion of output 4</td></tr>
<tr><td>Bit 4</td><td colspan="3">Inversion of output 5</td></tr>
<tr><td>Bit 5</td><td colspan="3">Inversion of output 6</td></tr>
<tr><td>Bit 6</td><td colspan="3">Inversion of output 7</td></tr>
<tr><td>Bit 7</td><td colspan="3">Inversion of output 8</td></tr>
<tr><td>Bit 8</td><td colspan="3" rowspan="3">Reserved</td></tr>
<tr><td>...</td></tr>
<tr><td>Bit 15</td></tr>
</table>

For a detailed functional description see L_SwitchPoint_1.

### 19.1.173 L_Transient_1

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01138/1. |

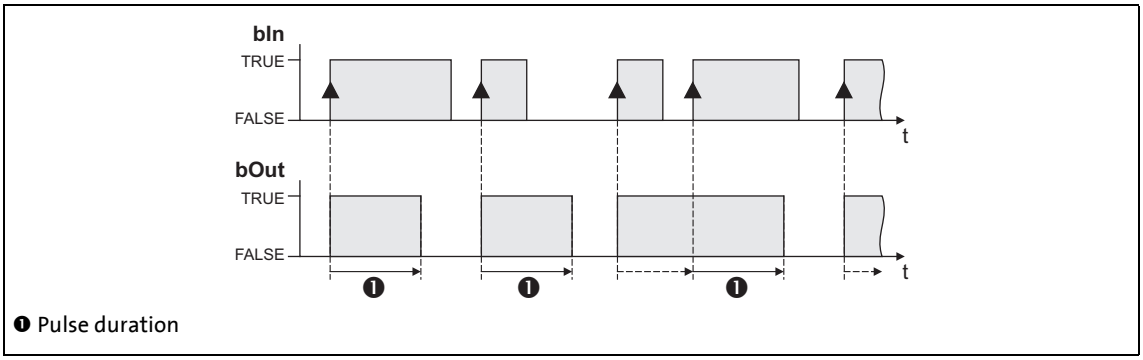
#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01138/1 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01139/1 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

### 19.1.173.1 Function 0: Evaluate rising signal edges

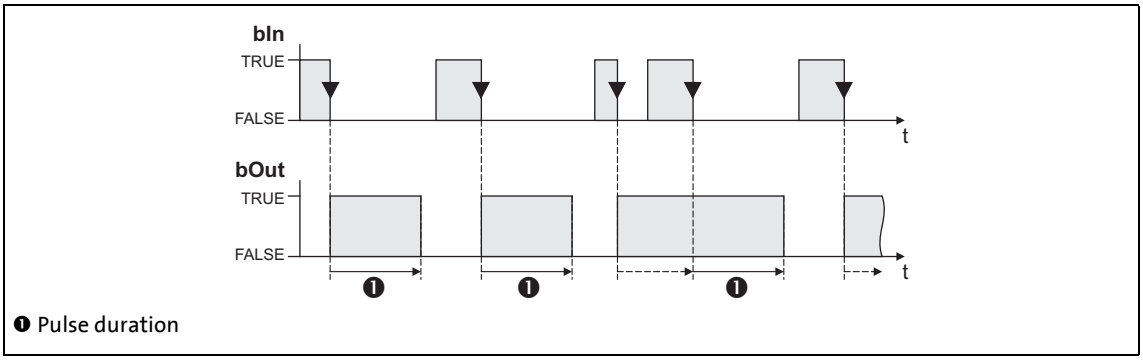



[19-78] Switching performance for function selection "0: High edge"

**Functional sequence**

1. A FALSE-TRUE edge at the *bIn* input sets the *bOut* output to TRUE.
2. After the parameterised pulse duration has elapsed, the *bOut* output is reset to FALSE unless another FALSE/TRUE edge has been set at the *bIn* input.
   - If an additional FALSE-TRUE edge occurs at the *bIn* input, the pulse duration starts again from the beginning, i.e. the *bOut* output can be retriggered.

### 19.1.173.2 Function 1: Evaluate falling signal edges

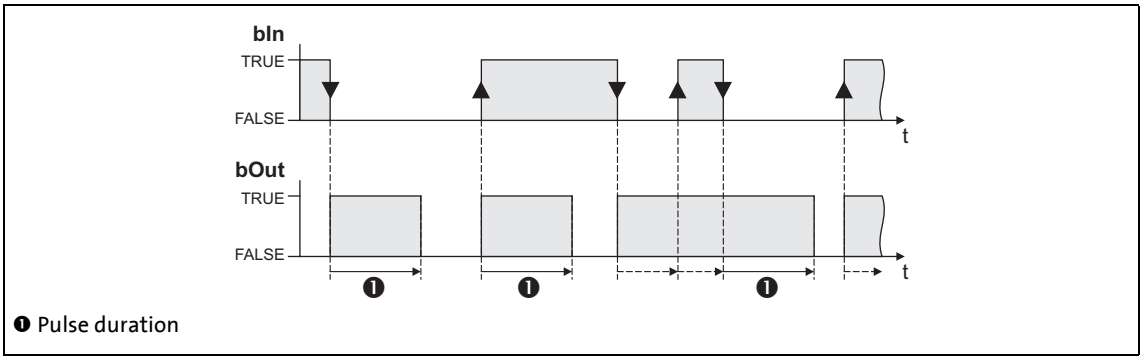



[19-79] Switching performance for function selection "1: Low edge"

**Functional sequence**

1. A TRUE-FALSE edge at the *bIn* inputs sets the *bOut* output to TRUE.
2. After the parameterised pulse duration has elapsed, the *bOut* output is reset to FALSE unless another TRUE/FALSE edge has been set at the *bIn* input.
   - If an additional TRUE-FALSE edge occurs at the *bIn* input, the pulse duration starts again from the beginning, i.e. the *bOut* output can be retriggered.

### 19.1.173.3 Function 2: Evaluate rising and falling signal edges



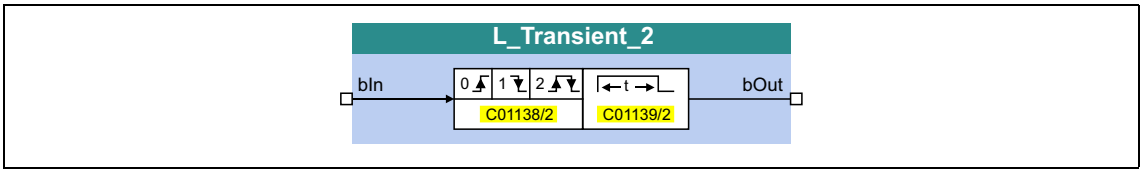

[19-80] Switching performance for function selection "2: High and low edge"

**Functional sequence**

1. A signal change (FALSE/TRUE edge or TRUE/FALSE edge) at the *bIn* input sets the *bOut* output to TRUE.
2. After the parameterised pulse duration has elapsed, the *bOut* output is reset to FALSE unless another signal change has taken place at the *bIn* input.
   - In case of another signal change at the input *bIn*, the pulse time restarts to elapse, i.e. the output *bOut* can be retriggered.

### 19.1.174 L_Transient_2

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01138/2. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

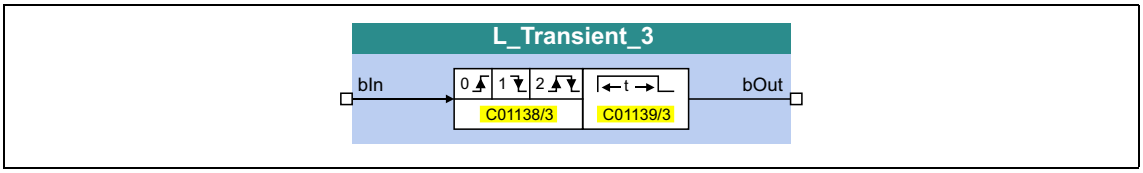
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01138/2 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01139/2 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

### 19.1.175 L_Transient_3

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01138/3. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

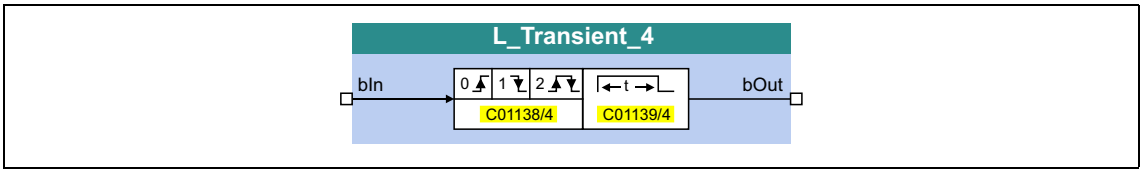
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01138/3 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01139/3 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

### 19.1.176 L_Transient_4

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01138/4. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

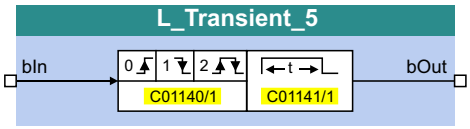
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01138/4 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01139/4 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

### 19.1.177 L_Transient_5

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01140/1. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

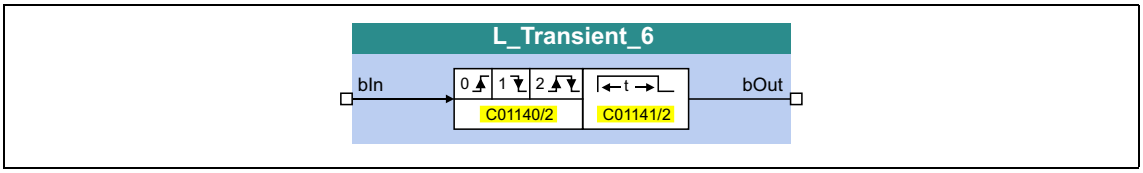
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01140/1 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01141/1 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

### 19.1.178 L_Transient_6

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01140/2. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

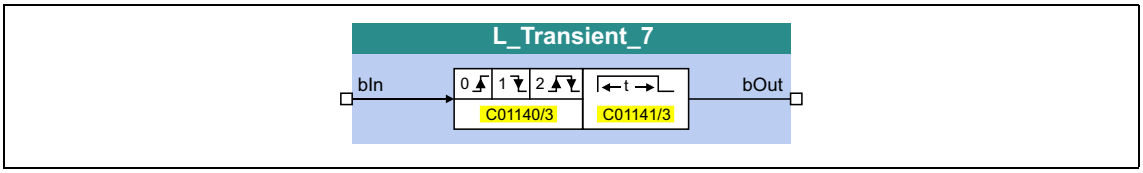
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01140/2 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01141/2 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

### 19.1.179 L_Transient_7

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01140/3. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

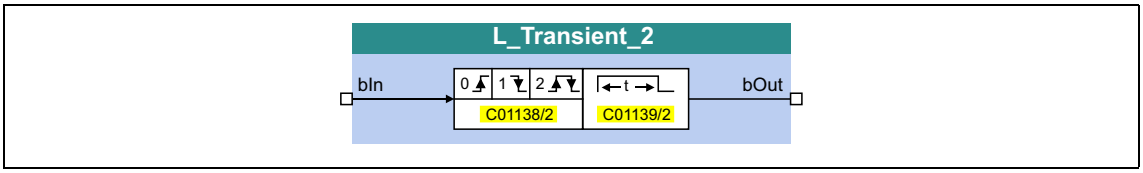
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01140/3 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01141/3 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

### 19.1.180 L_Transient_8

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01140/4. |

#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01140/4 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01141/4 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

## 19.2 System blocks

This chapter describes the system blocks which are available for the inverter in the FB Editor.



The function blocks are described in the chapter "Function blocks". (📖 1425)

### Overview of system blocks available

| System block | Function | can be inserted into level: | |
|---|---|---|---|
| | | I/O | Appl. |
| LS_AnalogInput | Interface to the analog input terminals<br>▸ Analog terminals (📖 409) | ● | |
| LS_AnalogOutput | Interface to the analog output terminals<br>▸ Analog terminals (📖 409) | ● | |
| LS_AxisBusAux | Interface to the axis bus<br>▸ Axis bus (📖 859) | ● | ● |
| LS_AxisBusIn | | ● | ● |
| LS_AxisBusIO | | ● | ● |
| LS_AxisBusOut | | ● | ● |
| LS_CANManagement | Control of internal functions of the CAN driver and display of the "Operational" status as well as the node address<br>▸ System bus "CAN on board" (📖 772) | ● | ● |
| LS_DataAccess | *Lenze internal only* | | ● |
| LS_DeviceMonitor | Motor control status signals<br>▸ Motor control (MCTRL) (📖 138) | | ● |
| LS_DFOut | Interface to the digital frequency output (multi-encoder interface X8)<br>• This SB is available from version 12.00.00.<br>▸ Digital frequency coupling (📖 353) | ● | |
| LS_DigitalInput | Interface to the digital input terminals<br>▸ Digital input terminals (📖 385) | ● | |
| LS_DigitalOutput | Interface to the digital output terminals<br>▸ Digital output terminals (📖 406) | ● | |
| LS_DisFree | Display of 8 arbitrary 16-bit signals of the application on display codes | ● | ● |
| LS_DisFree_a | Display of 8 arbitrary analog signals of the application on display codes | ● | ● |
| LS_DisFree_b | Display of 16 arbitrary digital signals of the application on a bit coded display code | ● | ● |
| LS_DisFree_p | Display of 8 arbitrary position signals of the application on display codes | ● | ● |
| LS_DriveInterface | Interface to drive control (DCTRL)<br>▸ Device control (DCTRL) (📖 102) | | ● |
| LS_Keypad | Control via keypad | ● | |
| LS_MotionControlKernel | Interface to the basic drive function implemented in the Motion Control Kernel (MCK)<br>▸ Basic drive functions (MCK) (📖 560) | | ● |
| LS_MotorInterface | Interface to motor control (MCTRL)<br>▸ Motor control (MCTRL) (📖 138) | | ● |
| LS_MultiEncoder | Interface to the Multi-Encoder<br>▸ Multi-Encoder (X8) (📖 332) | ● | ● |

| System block | Function | can be inserted into level: | |
|---|---|---|---|
| | | I/O | Appl. |
| LS_ParFix<br>LS_ParFix_2 | Output of frequently used constants (TRUE, FALSE, 100 %, etc.) to be used in the interconnection<br>• LS_ParFix_2 is available from version 02.00.00. | ● | ● |
| LS_ParFree<br>LS_ParFree_2 | Output of 32 parameterisable 16-bit signals<br>• LS_ParFree_2 is available from version 02.00.00. | ● | ● |
| LS_ParFree_a<br>LS_ParFree_a_2 | Output of 16 parameterisable analog signals<br>• LS_ParFree_a_2 is available from version 02.00.00. | ● | ● |
| LS_ParFree_b | Output of 32 parameterisable digital signals | ● | ● |
| LS_ParFree_p | Output of 8 parameterisable position signals | ● | ● |
| LS_ParFree_v<br>LS_ParFree_v_2 | Output of 8 parameterisable speed signals<br>• LS_ParFree_v_2 is available from version 02.00.00. | ● | ● |
| LS_ParFree32 | Output of 8 parameterisable 32-bit signals<br>• This SB is available from version 02.00.00. | ● | ● |
| LS_ParFreeUnit<br>LS_ParFreeUnit_2 | Output of 16 parameterisable position signals with internal conversion of [unit] in [increments]<br>• These SBs are available from version 02.00.00. | ● | ● |
| LS_ParReadWrite_1<br>...<br>LS_ParReadWrite_6 | Reading/Writing of local parameters | ● | ● |
| LS_PulseGenerator | Output of 9 fixed frequencies and 1 parameterisable frequency | ● | ● |
| LS_Resolver | Interface to the resolver<br>▸ Resolver (X7) (📖 326) | ● | ● |
| LS_RetainData | Selection and saving of retain data<br>• This SB is available from version 02.00.00. | ● | ● |
| LS_SetError_1 | Parameterisable responses to user-defined events are tripped<br>▸ Diagnostics & error management (📖 691) | ● | ● |
| LS_SetError_2 | | ● | ● |
| LS_SyncManagement | Output of status information for synchronising the internal time base<br>▸ Synchronisation of the internal time base (📖 880) | ● | ● |
| LS_TouchProbe | Interface for touch probe detection<br>▸ Touch probe detection (📖 419) | ● | ● |
| LS_WriteParamList | Interface to the basic "Parameter change-over" function<br>▸ Parameter change-over (📖 882) | ● | |

**Related topics:**

- ▸ Overview of function blocks available (📖 1425)
- ▸ Working with the FB Editor (📖 1373)

### 19.2.1 LS_AnalogInput

Interface to the analog input terminals.



For a detailed description see the main chapter "I/O terminals":
▸ Internal interfaces | System block "LS_AnalogInput" (📖 417)

### 19.2.2 LS_AnalogOutput

Interface to the analog output terminals.



For a detailed description see the main chapter "I/O terminals":
▸ Internal interfaces | System block "LS_AnalogInput" (📖 417)

### 19.2.3 LS_AxisBusAux

Interface to the axis bus.



For a detailed description see main chapter "Axis bus":
▸ Internal interfaces | System block "LS_AxisBusAux" (📖 872)

### 19.2.4 LS_AxisBusIn

Interface to the axis bus.



For a detailed description see main chapter "Axis bus":
▸ Internal interfaces | System block "LS_AxisBusIn" (📖 871)

### 19.2.5 LS_AxisBusIO

Interface to the axis bus.



For a detailed description see main chapter "Axis bus":
▸ Internal interfaces | System block "LS_AxisBusIO" (📖 879)

### 19.2.6 LS_AxisBusOut

Interface to the axis bus.



For a detailed description see main chapter "Axis bus":

▸ Internal interfaces | System block "LS_AxisBusOut" (📖 870)

### 19.2.7 LS_CANManagement

Control of internal functions of the CAN driver and display of the "Operational" status as well as the node address.



For a detailed description see the main chapter "System bus CAN on board":

▸ Internal interfaces | System block "LS_CANManagement" (📖 851)

### 19.2.8 LS_DataAccess

Only for Lenze-internal use.

### 19.2.9 LS_DeviceMonitor

Motor control status signals.



For a detailed description see the main chapter "Motor control (MCTRL)":

▸ Internal status signals | System block "LS_DeviceMonitor" (📖 318)

### 19.2.10 LS_DFOut

Interface to the digital frequency output (multi-encoder interface X8).



For a detailed description see main chapter "Encoder-/feedback system":

▸ Internal interfaces | System block "LS_DFOut" (📖 356)

### 19.2.11 LS_DigitalInput

Interface to the digital input terminals.

For a detailed description see the main chapter "I/O terminals":
▸ Internal interfaces | System block "LS_DigitalInput" (📖 398)

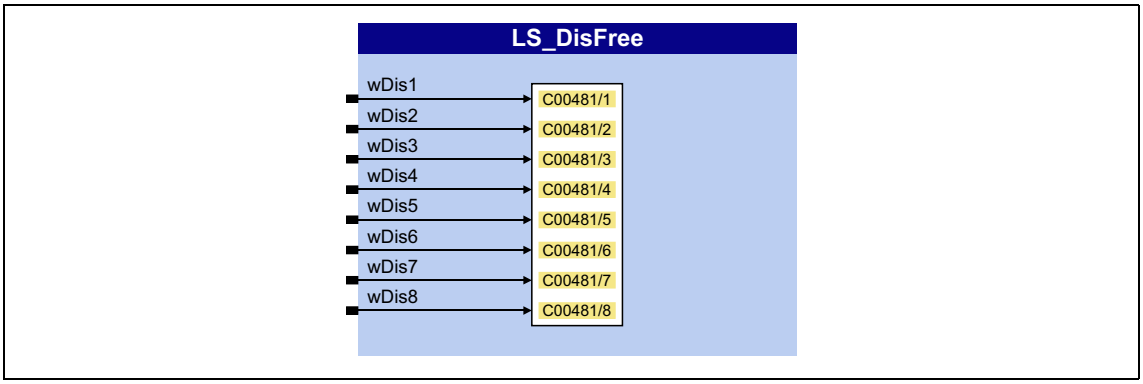
### 19.2.12 LS_DigitalOutput

Interface to the digital output terminals.

For a detailed description see the main chapter "I/O terminals":
▸ Internal interfaces | System block "LS_DigitalOutput" (📖 408)

### 19.2.13 LS_DisFree

This system block displays 8 arbitrary 16-bit signals of the application on display codes.




#### Inputs

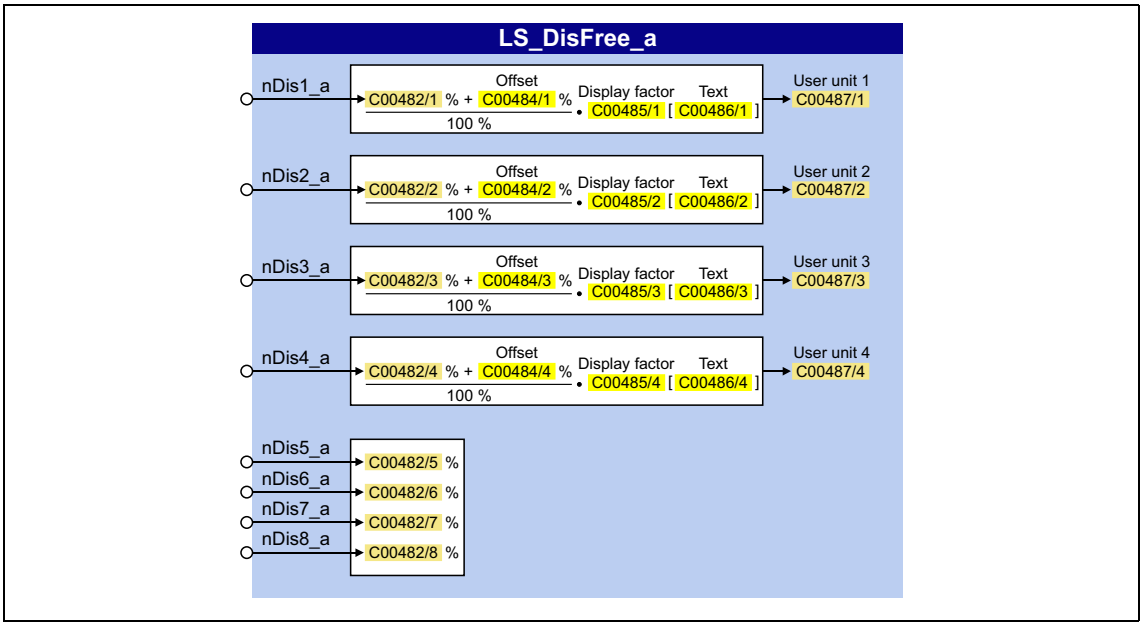
| Designator<br>Data type | Information/possible settings |
|---|---|
| wDis1 ... wDis8<br>WORD<br>From version 02.00.00:<br>wC481_1 ... wC481_8_a | Inputs for any 16-bit signals of the application<br><br>**Note:** From version 02.00.00 the inputs are named according to the display parameter for an easier allocation. |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00481/1...8 | 0x0000 | | 0xFFFF | Display of the 16-bit signals which are applied at the *wDis1* ... *wDis8* inputs |

### 19.2.14 LS_DisFree_a

This system block displays 8 arbitrary analog signals of the application on display codes.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| nDis1_a ... nDis8_a<br>INT<br>From version 02.00.00:<br>nC482_1_a ... nC482_8_a | Inputs for arbitrary analog signals of the application<br><br>**Note:** From version 02.00.00 the inputs are named according to the display parameter for an easier allocation. |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00482/1...8 | -199.99 | % | 199.99 | Display of the analog signals which are applied at the *nDis1_a ... nDis8_a* inputs |
| C00484/1...4<br>...<br>C00487/1...4 | ▸ Display of internal process variables in application units | | | |

### 19.2.14.1 Display of internal process variables in application units

In addition to the display in percent in C00482/1...8, for the first four analog signals *nDis1_a* ... *nDis4_a* the configurable display parameters C00487/1...4 are provided. Via these display parameters, internal process variables can be displayed, e.g. on the keypad, with an individual scaling and an individual unit (from keypad version 02.01).

**Configuration of the display parameters (C00487/1...4):**

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00484/1...4 | -199.99 | % | 199.99 | Offset 1 ... 4<br>• See formula [19-81].<br>• Lenze setting: 0.00 % |
| C00485/1...4 | -65536.0000 | | 65536.0000 | Display factor 1 ... 4<br>• Scaling of the input variable for the display.<br>• See formula [19-81].<br>• Lenze setting: 1.0000 |
| C00486/1...4 | String<br>(max. 7 digits) | | | Text 1 ... 4<br>• For each display value, an individual unit (e.g. "parts") can be set. |

$$\text{User unit 1} = \frac{\text{nDis1_a [\%]} + \text{Offset 1 [\%]}}{\text{100 [\%]}} \cdot \text{Display factor 1 [text 1]}$$

[19-81] Formula for scaling the display

**Example 1:**

- Input variable *nDis1_a* = 100 %
- Offset 1 (C00484/1) = 0 %
- Display factor 1 (C00485/1) = 123.45
- Text 1 (C00486/1) = "parts"

$$\text{User unit 1} = \frac{\text{100 [\%]} + \text{0 [\%]}}{\text{100 [\%]}} \cdot \text{123.45 [parts]} = \text{123.45 parts}$$

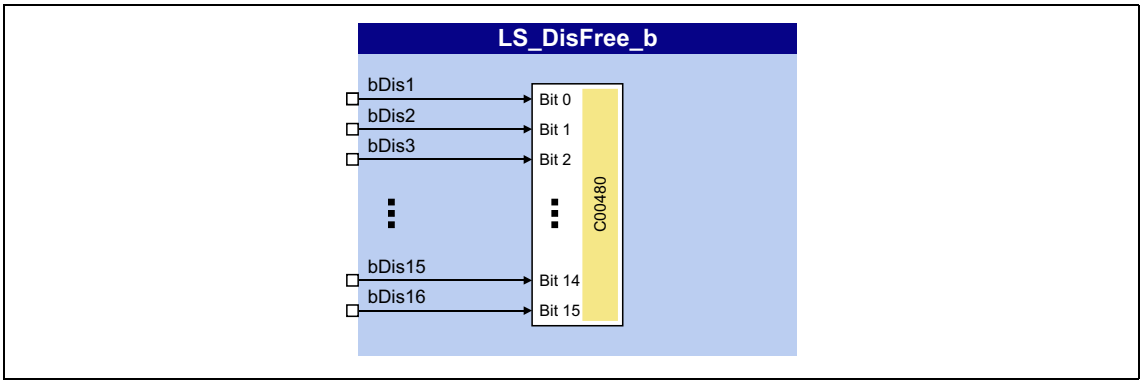
**Example 2:**

- Input variable *nDis2_a* = 40 %
- Offset 2 (C00484/2) = 35 %
- Display factor 2 (C00485/2) = 20
- Text 2 (C00486/2) = "kg"

$$\text{User unit 2} = \frac{\text{40 [\%]} + \text{35 [\%]}}{\text{100 [\%]}} \cdot \text{20 [kg]} = \text{15.00 kg}$$

### 19.2.15 LS_DisFree_b

This system block displays 16 arbitrary digital signals of the application on a bit coded display code.




#### Inputs

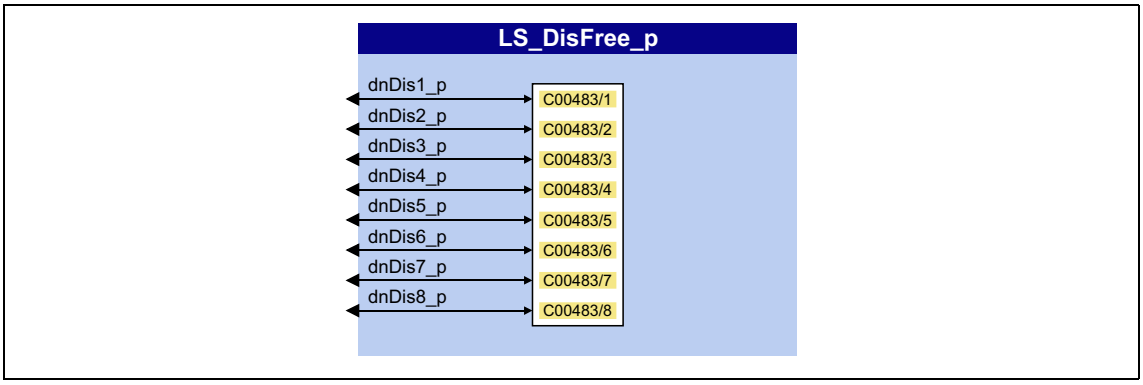
| Designator<br>Data type | Information/possible settings |
|---|---|
| bDis1 ... bDis16<br>BOOL<br>From version 02.00.00:<br>bC480_B0 ... bC480_B16 | Inputs for arbitrary digital signals of the application<br><br>**Note:** From version 02.00.00 the inputs are named according to the display parameter for an easier allocation. |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00480 | 0x0000 | | 0xFFFF | Display of the digital signals as hexadecimal values which are applied at the *bDis1* ... *bDis16* inputs |
| | Bit 0 | Signal level at the *bDis1* input | | |
| | Bit 1 | Signal level at the *bDis2* input | | |
| | Bit 2 | Signal level at the *bDis3* input | | |
| | ... | ... | | |
| | Bit 15 | Signal level at the *bDis16* input | | |

### 19.2.16 LS_DisFree_p

This system block displays 8 arbitrary position signals of the application on display codes.




#### Inputs

| Designator<br>Data type | Information/possible settings |
|---|---|
| dnDis1_p ... dnDis8_p<br>DINT<br>From version 02.00.00:<br>dnC483_1_p ... dnC483_8_p | Inputs for arbitrary position signals of the application<br><br>**Note:** From version 02.00.00 the inputs are named according to the display parameter for an easier allocation. |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00483/1...8 | -2147483647 | Incr. | 2147483647 | Display of the position signals which are applied at the *dnDis1_p ... dnDis8_p* inputs |

### 19.2.17 LS_DriveInterface

Interface to internal device control.

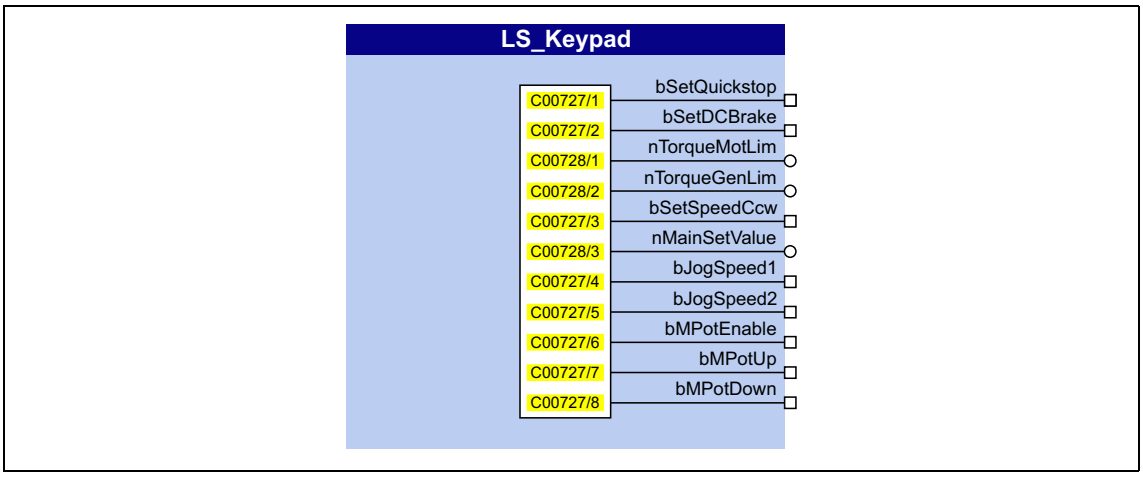
For a detailed description see main chapter "Device control (DCTRL)":

▸ Internal interfaces | "LS_DriveInterface" system block (📖 130)

### 19.2.18 LS_Keypad

This system block is used on I/O interconnection level if the "Keypad" control mode has been selected in C00007.

In the "Keypad" control mode, the **LS_Keypad** system block passes on various setpoints and control commands to the technology application which can be selected/activated via codes using the keypad.




#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bSetQuickstop<br>BOOL | C00727/1 = "1" ≡ Request quick stop |
| bSetDCBrake<br>BOOL | C00727/2 = "1" ≡ Request DC-injection braking |
| nTorqueMotLim<br>INT | Torque limit in motor mode set in C00728/1<br>• Lenze setting: 100.00 % |
| nTorqueGenLim<br>INT | Torque limit in generator mode set in C00728/2<br>• Lenze setting: 100.00 % |
| bSetSpeedCcw<br>BOOL | C00727/3 = "1" ≡ Request reversal |
| nMainSetValue<br>INT | Setpoint speed set in C00728/3<br>• Lenze setting: 0.00 % |
| bJogSpeed1<br>BOOL | C00727/4 = "1" ≡ Request fixed speed setpoint 1 |
| bJogSpeed2<br>BOOL | C00727/5 = "1" ≡ Request fixed speed setpoint 2 |
| bMPotEnable<br>BOOL | C00727/6 = "1" ≡ Motor potentiometer: Request activation |
| bMPotUp<br>BOOL | C00727/7 = "1" ≡ Motor potentiometer: Request positive acceleration |
| bMPotDown<br>BOOL | C00727/8 = "1" ≡ Motor potentiometer: Request negative acceleration |

**Parameters**

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00727/1...8 | 0 | | 1 | Keypad digital values<br>• Execution of control commands for keypad operation<br>• See the "Outputs" table for the meaning of the individual subcodes |
| C00728/1...3 | -199.99 | % | 199.99 | Analog values - keypad<br>• Selection of various setpoints for operation via keypad<br>• See the "Outputs" table for the meaning of the individual subcodes |

### 19.2.19 LS_MotionControlKernel

Interface to the basic drive functions implemented in **Motion Control Kernel** (MCK).

For a detailed description see the main chapter "Basic drive functions":
▸ Internal interfaces | System block "LS_MotionControlKernel" (📖 562)

### 19.2.20 LS_MotorInterface

Interface to internal motor control.

For a detailed description see the main chapterl "Motor control (MCTRL)":
▸ Internal interfaces | System block "LS_MotorInterface" (📖 312)

### 19.2.21 LS_MultiEncoder

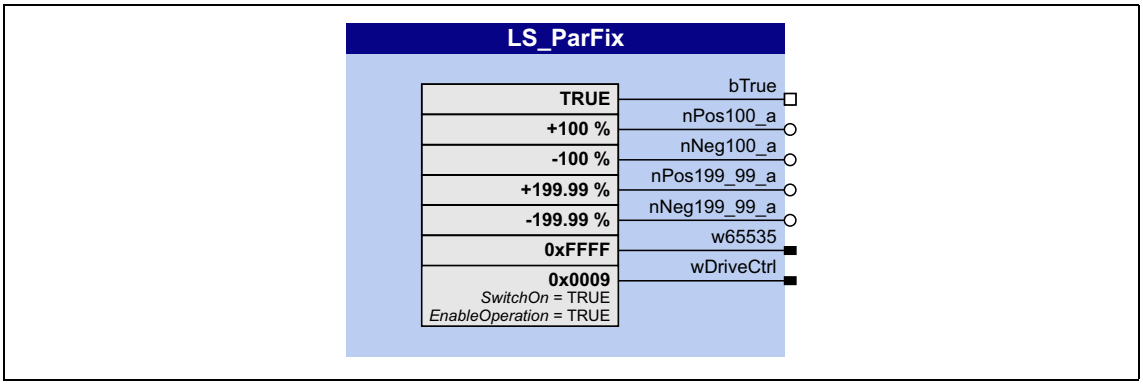
Interface to the Multi-Encoder.

For detailed information please see the main chapter "Encoder/Feedback system":
▸ Internal interfaces | "LS_MultiEncoder" system block (📖 357)

### 19.2.22 LS_ParFix

This system block outputs various fixed values (constants) to be used in the interconnection.

- From version 02.00.00, another **LS_ParFix_2** system block is available which can be used e.g. in the application level.

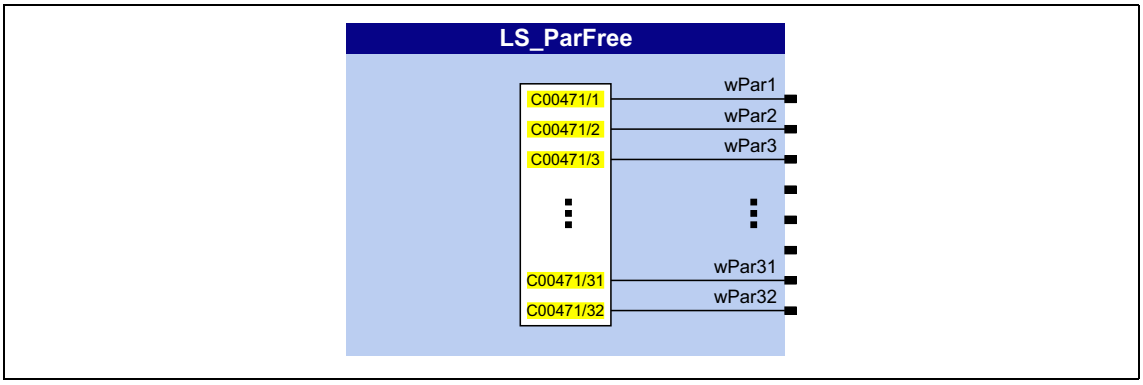



#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| bTrue<br>BOOL | 1 ≡ TRUE |
| nPos100_a<br>INT | 16384 ≡ + 100 % |
| nNeg100_a<br>INT | -16384 ≡ - 100 % |
| nPos199_99_a<br>INT | 32767 ≡ + 199.99 % |
| nNeg199_99_a<br>INT | -32767 ≡ - 199.99 % |
| w65535<br>WORD | 65535 ≡ 0xFFFF |
| wDriveCtrl<br>WORD | 9 ≡ 0x0009<br>• Bit 0, SwitchOn = TRUE<br>• Bit 3, EnableOperation = TRUE<br>• All others: FALSE<br>See also:<br>▸ wCANControl/wMCIControl control words (📖 133) |

### 19.2.23 LS_ParFree

This system block outputs 32 parameterisable 16-bit signals.




#### Outputs

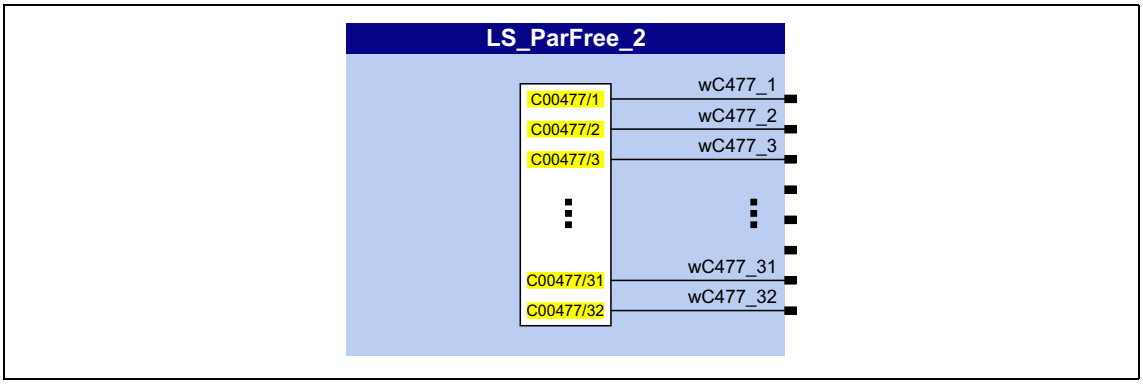
| Designator<br>Data type | Value/meaning |
|---|---|
| wPar1 ... wPar32<br>WORD<br>From version 02.00.00:<br>wC471_1 ... wC471_32 | Output of the 16-bit signals parameterised in C00471/1...32<br><br>**Note:** From version 02.00.00 the outputs are named according to the respective setting parameter for an easier allocation. |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00471/1...32 | 0x0000 | | 0xFFFF | Setting of the 16-bit signals to be output |

### 19.2.24 LS_ParFree_2

This system block outputs 32 parameterisable 16-bit signals.




#### Outputs

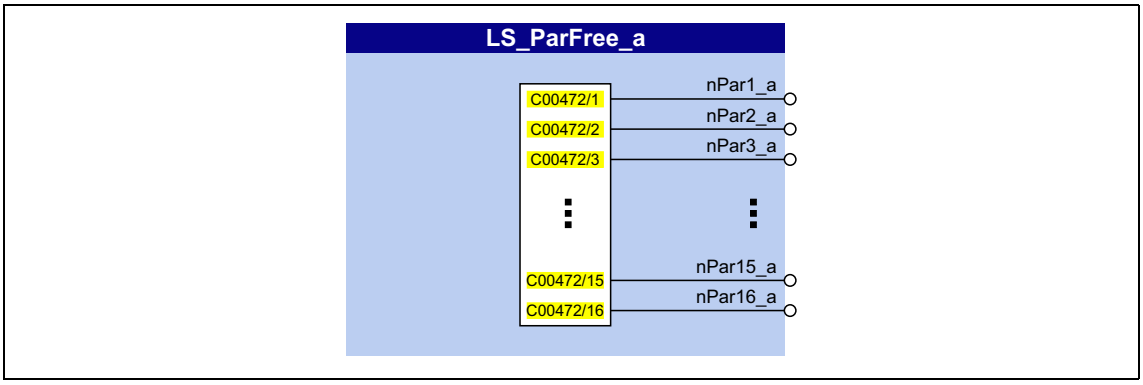
| Designator<br>Data type | Value/meaning |
|---|---|
| wC477_1 ... wC477_32<br>WORD | Output of the 16-bit signals parameterised in C00477/1...32 |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00477/1...32 | 0x0000 | | 0xFFFF | Setting of the 16-bit signals to be output |

### 19.2.25 LS_ParFree_a

This system block outputs 16 parameterisable analog signals.




#### Outputs

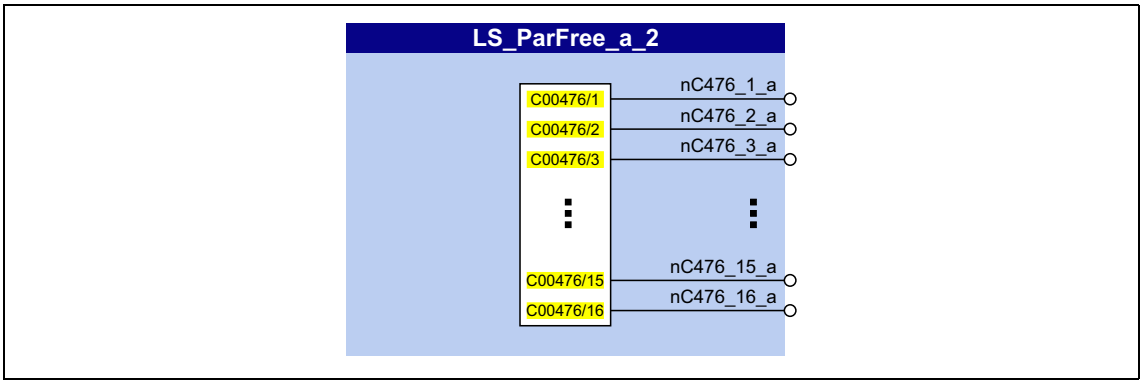
| Designator<br>Data type | Value/meaning |
|---|---|
| nPar1_a ... nPar16_a<br>INT<br>From version 02.00.00:<br>nC472_1_a ... nC472_16_a | Output of the analog signals parameterised in C00472/1...16<br><br>**Note:** From version 02.00.00 the outputs are named according to the respective setting parameter for an easier allocation. |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00472/1...16 | -199.99 | % | +199.99 | Selection of analog signals to be output |

### 19.2.26 LS_ParFree_a_2

This system block outputs 16 parameterisable analog signals.




#### Outputs

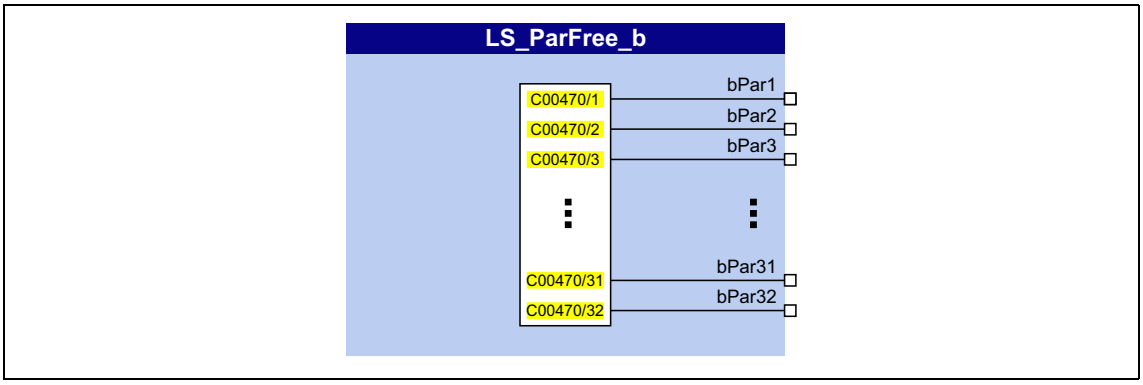
| Designator<br>Data type | Value/meaning |
|---|---|
| nC476_1_a ... nC476_16_a<br>INT | Output of the analog signals parameterised in C00476/1...16 |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00476/1...16 | -199.99 | % | +199.99 | Selection of analog signals to be output |

### 19.2.27 LS_ParFree_b

This system block outputs 32 parameterisable digital signals.




#### Outputs

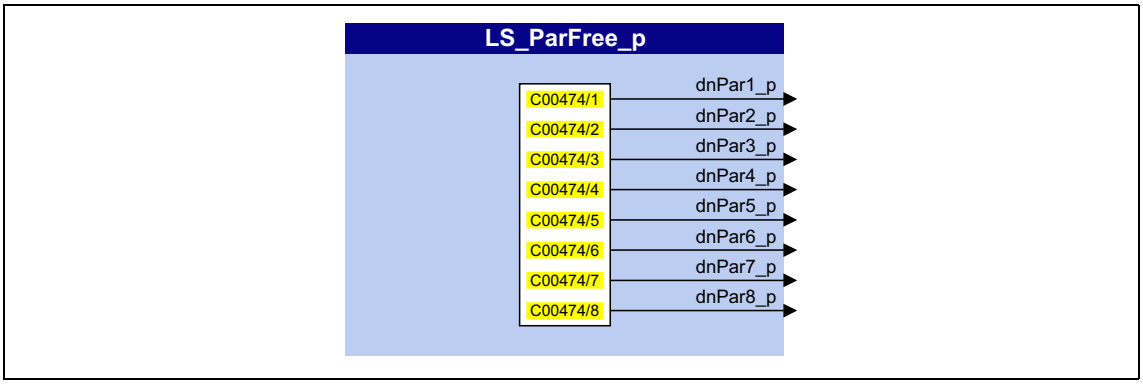
| Designator<br>Data type | Value/meaning |
|---|---|
| bPar1 ... bPar32<br>BOOL<br>From version 02.00.00:<br>bC470_1 ... bC470_32 | Output of the signal levels (FALSE/TRUE) parameterised in C00470/1...32<br><br>**Note:** From version 02.00.00 the outputs are named according to the respective setting parameter for an easier allocation. |

#### Parameters

| Parameters | Possible settings | | Info |
|---|---|---|---|
| C00470/1...32 | | | Selection of signal levels to be output<br>• Bit 0 ... 31 = *bPar1* ... *bPar32* |
| | 0 | "FALSE" signal is output | |
| | 1 | "TRUE" signal is output | |

### 19.2.28 LS_ParFree_p

This system block outputs 8 parameterisable position signals.




#### Outputs

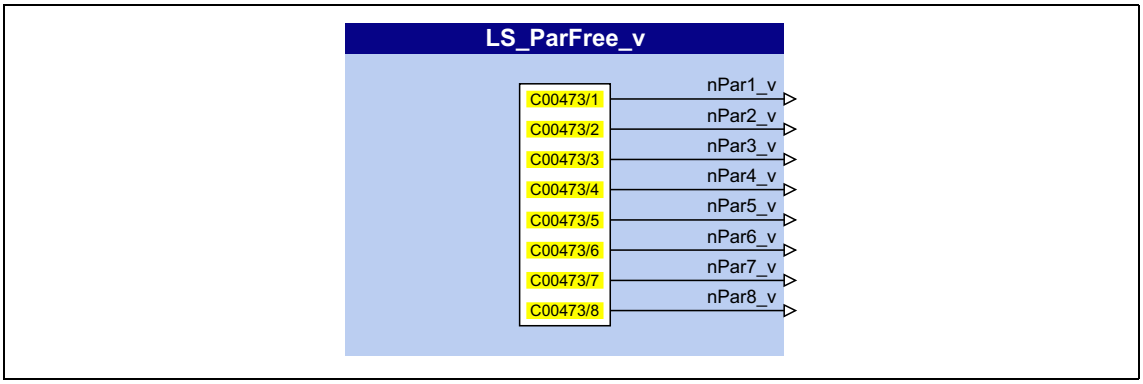
| Designator<br>Data type | Value/meaning |
|---|---|
| dnPar1_p ... dnPar8_p<br>DINT<br>From version 02.00.00:<br>dnC474_1_p ... dnC474_8_p | Output of the position signals parameterised in C00474/1...8<br><br>**Note:** From version 02.00.00 the outputs are named according to the respective setting parameter for an easier allocation. |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00474/1...8 | -2147483647 | Incr. | 2147483647 | Setting of the position signals to be output |

### 19.2.29 LS_ParFree_v

This system block outputs 8 parameterisable speed signals.




#### Outputs

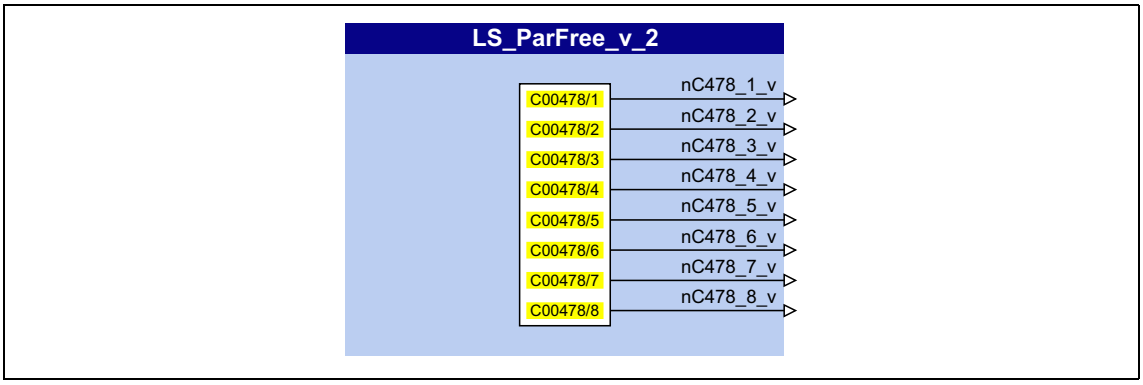
| Designator<br>Data type | Information/possible settings |
|---|---|
| nPar1_v ... nPar8_v<br>INT<br>From version 02.00.00:<br>nC473_1_v ... nC473_8_v | Output of the speed signals parameterised in C00473/1...8<br><br>**Note:** From version 02.00.00 the outputs are named according to the respective setting parameter for an easier allocation. |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00473/1...8 | -32767 | Incr/ms | +32767 | Selection of speed signals to be output |

### 19.2.30 LS_ParFree_v_2

This system block outputs 8 parameterisable speed signals.




#### Outputs

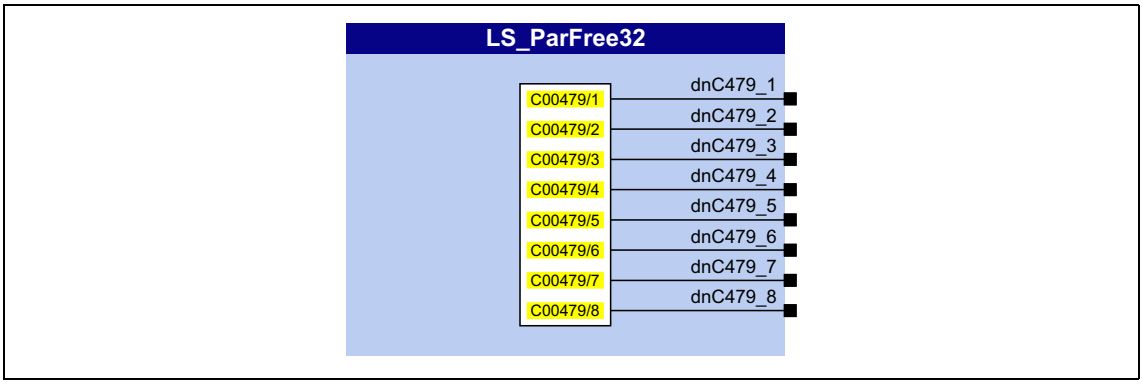
| Designator<br>Data type | Information/possible settings |
|---|---|
| nC478_1_v ... nC478_8_v<br>INT | Output of the speed signals parameterised in C00478/1...8 |

#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00478/1...8 | -32767 | Incr/ms | +32767 | Selection of speed signals to be output |

### 19.2.31 LS_ParFree32

This system block outputs 8 parameterisable 32-bit signals.




#### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| dnC479_1 ... dnC479_8<br>DINT | Output of the 32-bit signals parameterised in C00479/1...8 |

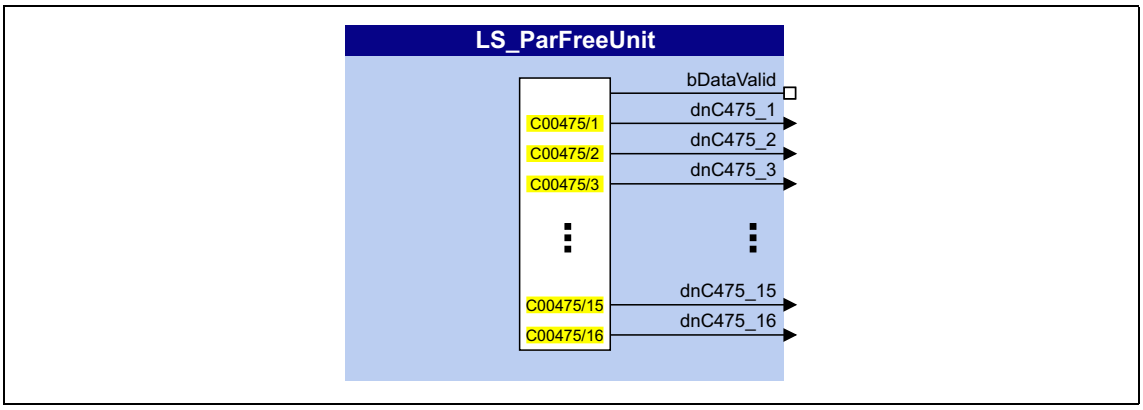
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00479/1...8 | -2147483647 | | 2147483647 | Setting of the 32-bit signals to be output |

### 19.2.32 LS_ParFreeUnit

This system block outputs 16 parameterisable position signals.

- In contrast to the SB LS_ParFree_p, the positions are set in the application unit [unit] instead of in [increments].
- The *bDataValid* status signal indicates that the internal position conversion is completed and the output data are consistent.




#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bDataValid<br>BOOL | Status signal "Position conversion completed, data consistent" | |
| | TRUE | The conversion of the positions from [unit] to [increments] is completed. |
| dnC475_1 ... dnC475_16<br>DINT | Output of the position signals parameterised in C00475/1...16 in [increments]<br>• Observe *bDataValid* status signal! | |

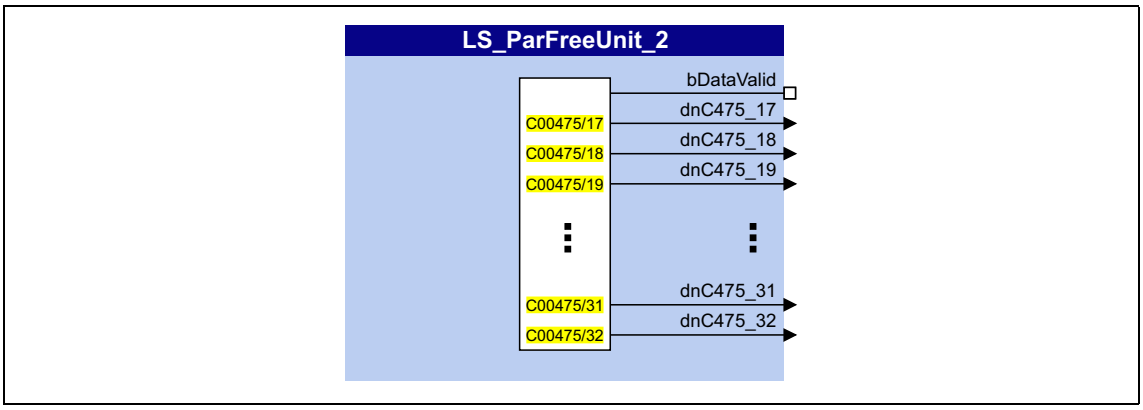
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00475/1...16 | -214748.3647 | Unit | 214748.3647 | Setting of the position signals to be output |

### 19.2.33 LS_ParFreeUnit_2

This system block outputs 16 parameterisable position signals.

- In contrast to the SB LS_ParFree_p, the positions are set in the application unit [unit] instead of in [increments].
- The *bDataValid* status signal indicates that the internal position conversion is completed and the output data are consistent.




#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bDataValid<br>BOOL | Status signal "Position conversion completed, data consistent" | |
| | TRUE | The conversion of the positions from [unit] to [increments] is completed. |
| dnC475_17 ... dnC475_32<br>DINT | Output of the position signals parameterised in C00475/17...32 in [increments]<br>• Observe *bDataValid* status signal! | |

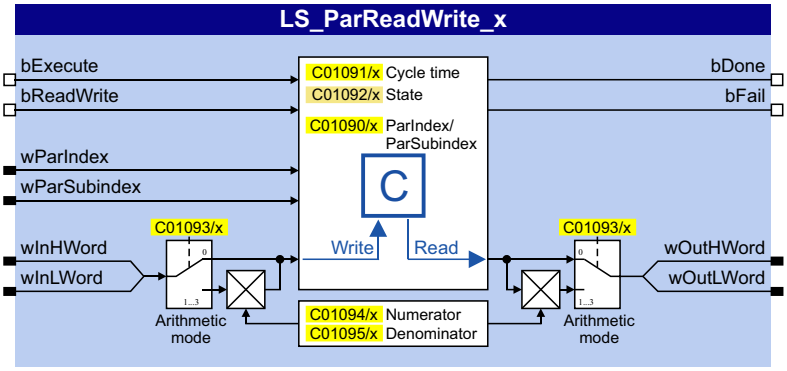
#### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00475/17...32 | -214748.3647 | Unit | 214748.3647 | Setting of the position signals to be output |

### 19.2.34 LS_ParReadWrite_1-6

The **LS_ParReadWrite_1** ... **LS_ParReadWrite_6** system blocks are used for reading and writing local parameters.

- If several system blocks are activated at the same time, the blocks will be processed one at a time every time the main program is executed.
- The SB supports one-time and cyclic reading/writing in an adjustable time interval.




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bExecute<br>BOOL | Trip read/write request | |
| | FALSE↗TRUE | If cycle time (C01091) = "0 ms":<br>One-time reading/writing of the parameter value which has been addressed via the*wParIndex* and *wParSubindex* inputs. |
| | | If cycle time (C01091) > "0 ms":<br>Cyclic reading/writing of the parameter value which has been addressed via the *wParIndex* and *wParSubindex* inputs. |
| | TRUE↘FALSE | Deactivate cyclic reading/writing again. |
| bReadWrite<br>BOOL | Selection: Read or write request | |
| | FALSE | Read request |
| | TRUE | Write request |
| wParIndex<br>WORD | Code to be read or written.<br>• The selection can be made alternatively via C01090. | |
| wParSubindex<br>WORD | Subcode to be read or written.<br>• The selection can be made alternatively via C01090. | |
| wInHWord<br>wInLWord<br>WORD | Value to be written (DataHigh/DataLow portion) | |

### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| bDone<br>BOOL | "Read/Write request successfully completed" status signal<br>• The output is automatically reset to FALSE if a new request is activated via *bExecute* or the cycle time (C01091) expires. | |
| | TRUE | Read/Write request successfully completed. |
| | FALSE | The FALSE status can have the following meanings:<br>1. There is no active read/write request.<br>2. The read/write request has not been completed yet.<br>3. An error has occurred (if *bFail* = TRUE). |
| bFail<br>BOOL | "Error" status | |
| | TRUE | An error has occurred (group signal).<br>• See display parameter (C01092) for details. |
| wOutHWord<br>wOutLWord<br>WORD | Value which was read (DataHigh/DataLow portion) after read request | |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01090/1...6 | 0,000 | | 16000,000 | Parameter to be read or written.<br>• For a setting of "0,000", inputs *wParIndex* and *wParSubindex* are effective for addressing purposes instead.<br>• Lenze setting: 0.000 |
| | Format:<br><Code number>,<subcode number> | | | |
| C01091/1...6 | | | | Cycle time<br>• Subcode 1 = **LS_ParReadWrite_1**<br>Subcode 2 = **LS_ParReadWrite_2**<br>...<br>Subcode 6 = **LS_ParReadWrite_6**<br>• Lenze setting: 0 |
| | 0 | One-time reading/writing at *bExecute* in case of a FALSE/TRUE edge | | |
| | Cyclic reading/writing: | | | |
| | 20 | 20 ms | | |
| | 50 | 50 ms | | |
| | 100 | 100 ms | | |
| | 200 | 200 ms | | |
| | 500 | 500 ms | | |
| | 1000 | 1 s | | |
| | 2000 | 2 s | | |
| | 5000 | 5 s | | |
| | 10000 | 10 s | | |

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01092/1...6 | | | | Error status<br>• If *bFail* = TRUE:<br>Error status is displayed.<br>• Subcode 1 = **LS_ParReadWrite_1**<br>Subcode 2 = **LS_ParReadWrite_2**<br>**...**<br>Subcode 6 = **LS_ParReadWrite_6** |
| | 0 | No error | | |
| | 33803 | Invalid data type (e.g. STRING) | | |
| | 33804 | Limit violation | | |
| | 33806 | Invalid code | | |
| | 33813 | No element of the selection list | | |
| | 33815 | Writing of the parameter not permitted | | |
| | 33816 | Writing of the parameter only permitted if controller is inhibited | | |
| | 33829 | Invalid subcode | | |
| | 33865 | No parameter with subcodes | | |
| C01093/1...6 | | | | Arithmetic mode<br>• Lenze setting:"0: No arithmetic"<br>▸ Arithmetic function |
| | 0 | No arithmetic | | |
| | 1 | In16Bit: LW=+/-32767 | | |
| | 2 | In16Bit: HW=+/-; LW=0..65535 | | |
| | 3 | In32Bit: HW_LW=+/-2147483647 | | |
| C01094/1...6 | -32767 | | 32767 | Counter<br>• For internal conversion in arithmetic modes 1 ... 3.<br>• Lenze setting: 1 |
| C01095/1...6 | 1 | | 32767 | Denominator<br>• For internal conversion in arithmetic modes 1 ... 3.<br>• Lenze setting: 1 |
| C01098/1...6 | 1 | | 32767 | Configuration |

### 19.2.34.1 Arithmetic function

The implemented arithmetic function enables easy arithmetic conversion of the process values to be written or read via parameterisable factors into the format of the target parameter without the need of an additional arithmetic function block.

- In C01093, the interpretation of the *wInHWord* and *wInLWord* inputs can be set to be able to write to parameters:

| Arithmetic mode | | wInHWord | wInLWord | Internal conversion |
|---|---|---|---|---|
| 0 | No arithmetic (Lenze setting) | **INTEGER_32** (4 bytes with sign) | | No (same behaviour as before) |
| | | DataHigh portion | DataLow portion | |
| 1 | In16Bit: LW=+/-32767 | - | **INTEGER_16** (2 bytes with sign) | Yes (see the following section) |
| 2 | In16Bit: HW=+/-; LW=0..65535 | **Sign** (0 ≡ positive value) | **UNSIGNED_16** (2 bytes without sign) | |
| 3 | In32Bit: HW_LW= +/-2147483647 | **INTEGER_32** (4 bytes with sign) | | |
| | | DataHigh portion | DataLow portion | |

#### Internal conversion

If arithmetic modes 1 ... 3 are selected in C01093, the input value / read parameter value is internally converted via parameterisable factors.

- Division is not remainder considered.

$$\text{parameter value to be written} = \text{Input value}_{[32]} \cdot \frac{\text{Counter}_{[16]}}{\text{Denominator}_{[16]}}$$

C01094: Numerator
C01095: Denominator

[19-82] Internal conversion with write access

$$\text{Output value}_{[32]} = \text{Read parameter value} \cdot \frac{\text{Counter}_{[16]}}{\text{Denominator}_{[16]}}$$

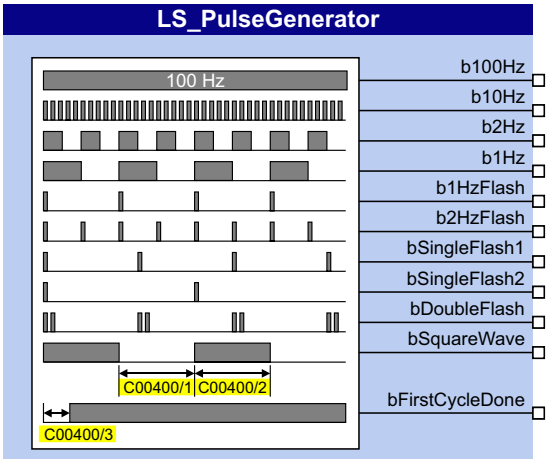
C01094: Numerator
C01095: Denominator

[19-83] Internal conversion with read access

### 19.2.35 LS_PulseGenerator

This system block outputs 9 different fixed frequencies and 1 frequency with parameterisable pulse/dead time.

The SB provides a TRUE signal at the *bFirstCycleDone* output when the first 1-ms processing cycle is completed and the time set in 1C00400/3 has expired. This status signal can e.g. be used for the delayed enable of peripheral devices or motor control setpoints so that all required initial values are calculated first after the inverter switch-on.




#### Outputs

| Designator<br>Data type | Value/meaning | |
|---|---|---|
| b100Hz<br>BOOL | Rectangular signal 100 Hz | |
| b10Hz<br>BOOL | Rectangular signal 10 Hz | |
| b2Hz<br>BOOL | Rectangular signal 2 Hz | |
| b1Hz<br>BOOL | Rectangular signal 1 Hz | |
| b1HzFlash<br>BOOL | 80 ms-pulse, repetition rate every second | |
| b2HzFlash<br>BOOL | 80 ms-pulse, repetition rate every 0.5 seconds | |
| bSingleFlash1<br>BOOL | 80 ms pulse, repetition rate every 1.25 seconds | |
| bSingleFlash2<br>BOOL | 80 ms pulse, repetition rate every 3 seconds | |
| bDoubleFlash<br>BOOL | 80 ms-double pulse, repetition rate every 1.25 seconds | |
| bSquareWave<br>BOOL | Output frequency with pulse/dead time set in C00400/1...2 | |
| bFirstCycleDone<br>BOOL | Status signal "First processing cycle completed" | |
| | TRUE | The first 1-ms processing cycle has been completed and the time set in C00400/3 has expired (i.e. all FBs have been called at least once). |

**Parameters**

**Note!**

The real length of the low or high level at the *bSquareWave* output is always 1 ms higher than set in C00400/1 or C00400/2. If "0 ms" is set, a level with a length of 1 ms is created length, for instance, and if "1000 ms" is set, a level with a length of 1001 ms is created.

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C00400/1 | 0 | ms | 60000 | Length of the low level (break) + 1 ms<br>• For output *bSquareWave*<br>• Lenze setting: 1000 ms |
| C00400/2 | 0 | ms | 60000 | Length of the high level + 1 ms<br>• For output *bSquareWave*<br>• Lenze setting: 1000 ms |
| C00400/3 | 0 | ms | 60000 | Delay of the status *bFirstCycleDone*<br>• Lenze setting: 100 ms |

## 19.2.36 LS_Resolver

Interface to the resolver.

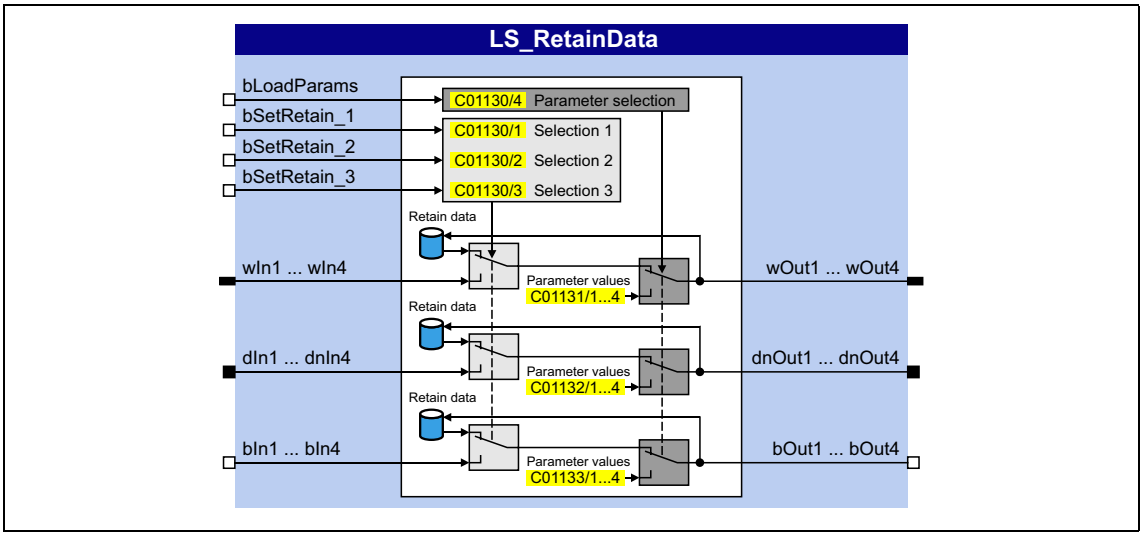
For detailed information please see the main chapter "Encoder/Feedback system":

▸ Internal interfaces | "LS_Resolver" system block (📖 331)

### 19.2.37 LS_RetainData

This system block serves to select and save retain data similar to the FBs L_SampleHold_1 and L_SampleHold_2.

- Retain data are values that are saved automatically at mains disconnection or disconnection of the 24-V supply in the inverter and are restored when the 24-V supply is switched on again.
- A total of 12 retain data can be stored in the retain memory:
  - 4 WORD signals (16 bits, 0 ... 65535)
  - 4 DINT signals (32 bits, -2147483647 ... 2147483647)
  - 4 BOOL signals (FALSE or TRUE)




#### Inputs

| Designator<br>Data type | Information/possible settings | |
|---|---|---|
| bLoadParams<br>BOOL | Set selected outputs & retain values to parameter values.<br>• This input has priority over the *bSetRetain* inputs.<br>• Typical application: Initialisation of the outputs & retain values. | |
| | FALSE | The input or retain values are output, depending on the *bSetRetain* input. |
| | TRUE | The outputs selected in C01130/4 and the respective retain values are set to the parameter values and held as long as *bLoadParams* is TRUE.<br>If the parameter values change during this time, the respective outputs and retain values change as well. |
| bSetRetain_1...3<br>BOOL | Transfer selected input values to the retain memory ("latch") | |
| | FALSE | No transfer. Retain values or parameter values (when *bLoadParams* = TRUE) are provided at the outputs. |
| | TRUE | The input values selected in C01130/1...3 are stored in the retain memory and provided at the outputs. |
| wIn1 ... wIn4<br>WORD | Input values | |
| dnIn1 ... dnIn4<br>DINT | | |
| bIn1 ... bIn4<br>BOOL | | |

### Outputs

| Designator<br>Data type | Value/meaning |
|---|---|
| wOut1 ... wOut4<br>WORD | Output of the input or retain values, depending on the *bSetRetain* input |
| dnOut1 ... dnOut4<br>DINT | |
| bOut1 ... bOut4<br>BOOL | |

### Parameters

| Parameters | Possible settings | | | Info |
|---|---|---|---|---|
| C01130/1...4 | Setting is bit coded: | | | Subcodes 1 ... 3: Selection 1 ... 3<br>• Select which input values are to be stored in the retain memory when setting the respective *bSetRetain*-input to TRUE.<br>• Lenze setting: 0x0000<br><br>Subcode 4: Parameter selection<br>• Select which parameter values are to be stored in the retain memory when setting *bLoadParams* to TRUE.<br>• Lenze setting: 0x0000 |
| | | Subcodes 1 ... 3: | Subcode 4: | |
| | Bit 0 | wIn1 | C01131/1 | |
| | Bit 1 | wIn2 | C01131/2 | |
| | Bit 2 | wIn3 | C01131/3 | |
| | Bit 3 | wIn4 | C01131/4 | |
| | Bit 4 | dnIn1 | C01132/1 | |
| | Bit 5 | dnIn2 | C01132/2 | |
| | Bit 6 | dnIn3 | C01132/3 | |
| | Bit 7 | dnIn4 | C01132/4 | |
| | Bit 8 | bIn1 | C01133/1 | |
| | Bit 9 | bIn2 | C01133/2 | |
| | Bit 10 | bIn3 | C01133/3 | |
| | Bit 11 | bIn4 | C01133/4 | |
| | Bit 12 | Reserved | | |
| | Bit 13 | | | |
| | Bit 14 | | | |
| | Bit 15 | | | |
| C01131/1...4 | 0 | | 65536 | Parameter value for retain memory (outputs *wOut1* ... *wOut4*)<br>• Lenze setting: 0 |
| C01132/1...4 | -2147483647 | | 2147483647 | Parameter values for retain memory (outputs *dnOut1* ... *dnOut4*)<br>• Lenze setting: 0 |
| C01133/1...4 | | | | Parameter values for retain memory (outputs *bOut1* ... *bOut4*)<br>• Lenze setting: 0: FALSE |
| | 0 | FALSE | | |
| | 1 | TRUE | | |

### 19.2.38 LS_SetError_1

Parameterisable responses to user-defined events are tripped.



For a detailed description see the main chapter "Diagnostics & error management":

▸ "LS_SetError_1" system block (📖 752)

### 19.2.39 LS_SetError_2

Parameterisable responses to user-defined events are tripped.



For a detailed description see the main chapter "Diagnostics & error management":

▸ "LS_SetError_2" system block (📖 753)

### 19.2.40 LS_SyncManagement

Output of status information for synchronising the internal time base.



For a detailed description see main chapter "Synchronisation of the internal time base of the inverter":

▸ Internal interfaces | System block "LS_SyncManagement" (📖 881)

### 19.2.41 LS_TouchProbe

Interface for touch probe detection.



For a detailed description see the main chapter "I/O terminals":

▸ Touch probe detection (📖 419)

### 19.2.42 LS_WriteParamList

Writing to a configurable list which contains up to 32 local parameters.



For a detailed description see the main chapterl "Parameter change-over". (📖 882)

## 19.3 Application blocks

### 19.3.1 Technology application "Actuating drive speed"

This technology application serves to solve speed-controlled drive tasks, e.g. conveyor drives (interconnected), extruders, test benches, vibrators, travelling drives, presses, machining systems, metering units.



For a detailed description see TA "Actuating drive speed". (📖 438)

### 19.3.2 Technology application "actuating drive speed (AC Drive profile)"

This technology application available from version 13.00.00 provides a speed and torque control by means of "AC Drive Profile". The fieldbuses EtherNet/IP™ and system bus (CANopen) are supported.



For a detailed description see TA "actuating drive speed (AC Drive Profile)". (📖 465)

### 19.3.3 Technology application "Table positioning"

This technology application serves to solve position-controlled drive tasks which are normally controlled by a higher-level control via a fieldbus, e.g. transport facilities, rotary tables, storage and retrieval units, feed drives, metering units, hoists.



For a detailed description see TA "Table positioning". (📖 498)

### 19.3.4 "Switch-off positioning" technology application

This technology application is used to solve speed-controlled drive tasks which require a pre-switch off or stopping at certain positions, e.g. roller conveyors and conveying belts. The pre-switch off is implemented by connecting switch-off sensors.



For a detailed description see TA "Switch-off positioning". (📖 527)

## 19.4 Port blocks

### 19.4.1 Port block "LP_CanIn1"

The **LP_CanIn1** port block maps process data object RPDO1 in the FB Editor.



For a detailed description see RPDO1 | Port block "LP_CanIn1". (📖 789)

### 19.4.2 Port block "LP_CanIn2"

The **LP_CanIn2** port block maps process data object RPDO2 in the FB Editor.



For a detailed description see RPDO2 | "LP_CanIn2" port block. (📖 791)

### 19.4.3 Port block "LP_CanIn3"

The **LP_CanIn3** port block maps process data object RPDO3 in the FB Editor.



For a detailed description see RPDO3 | "LP_CanIn3" port block. (📖 793)

### 19.4.4 Port block "LP_CanIn4"

The **LP_CanIn4** port block maps process data object RPDO4 in the FB Editor.



For a detailed description see RPDO4 | "LP_CanIn4" port block. (📖 795)

### 19.4.5 Port block "LP_CanOut1"

The **LP_CanOut1** port block maps process data object TPDO1 in the FB Editor.



For a detailed description see TPDO1 | "LP_CanOut1" port block. (📖 797)

### 19.4.6 Port block "LP_MciIn"

The **LP_MciIn** port block maps the received MCI-PDOs in the FB Editor.



For a detailed description see Port block "LP_MciOut". (📖 857)

### 19.4.7 Port block "LP_MciOut"

The **LP_MciOut** port block maps the sent MCI-PDOs in the FB Editor.



For a detailed description see Port block "LP_MciIn". (📖 856)

### 19.4.8 Port block "LP_CanOut2"

The **LP_CanOut2** port block maps process data object TPDO2 in the FB Editor.



For a detailed description see TPDO2 | "LP_CanOut2" port block. (📖 798)

### 19.4.9 Port block "LP_CanOut3"

The **LP_CanOut3** port block maps process data object TPDO3 in the FB Editor.



For a detailed description see TPDO3 | "LP_CanOut3" port block. (📖 799)

### 19.4.10 Port block "LP_CanOut4"

The **LP_CanOut4** port block maps process data object TPDO4 in the FB Editor.



For a detailed description see TPDO4 | "LP_CanOut4" port block. (📖 800)

# Index


## Symbols

"DRV-RDY" LED 693
"LA_TabPos" application block 502
»Engineer« motor catalogue 143

## Figures

120 % operation 267
16-bit inputs (C00831) 1103
16-bit inputs (Set2) (C00836) 1116
16-bit inputs [%] (C00830) 1100
16-bit inputs [%] (Set2) (C00835) 1112
16-bit inputs [Incr./ms] (C00832) 1106
16-bit inputs [Incr./ms] (Set2) (C00837) 1120
16-bit inputs I/O level (C00841) 1134
16-bit inputs I/O level [%] (C00840) 1131
24 V supply voltage (C00065) 923
32-bit inputs [incr] (C00834) 1112
32-bit inputs [incr] (Set2) (C00839) 1129
32-bit inputs I/O level [incr] (C00844) 1142
87-Hz operation 169

## A

Ab01: Axis bus time-out (error message) 739
Ab02: Axis bus IO error (error message) 739
AB-STATE (LED) 860
Accel. time - main setpoint (C00012) 912
Acceleration override 671
Access protection 46
Accessories for commissioning 37
ACDrive
  Control word (C01351) 1242
  Drive mode (C01350) 1242
  Setpoint scaling (C01353) 1244
  Status word (C01352) 1243
Acknowledgement error 820, 868
Activating the bus terminating resistor 776, 863
Active COBID (C00355) 982
Actual values (C00055) 921
Adapt keypad menus 40
Add. accel. time x (C00101) 930
Add. decel. time x (C00103) 931
Additive speed specification 287
AIN1
  Characteristic (C00010) 912
AIN2
  Characteristic (C00020) 915
AINx
  Configuration (C00034) 918
  Gain (C00027) 917
  Input current (C00029) 917
  Input voltage (C00028) 917
  Offset (C00026) 916
  Output value (C00033) 918
An01: AIN1_I < 4 mA (error message) 739
An02: AIN2_I < 4 mA (error message) 739
Analog terminals 409
  For electrical data see the hardware manual
  User-defined terminal assignment 429
Angle controller overflow 1553
Angle correction in case of transmission errors 1585
Angular drift monitoring (encoder) 379
AOutx
  Decoupling value (C00442) 1002
Appl.
  Reference frequency C11 (C00059) 922
  Reference speed (C00011) 912
Application (C00005) 906
Application block "LA_NCtrl" 441, 470
Application block "LA_SwitchPos" 530
Application notes 30
Application units
  Configuration of the display parameters 1748
  Display factor (C00485) 1018
  Offset (C00484) 1018
  Text (C00486) 1018
Application units (C00487) 1019
Assignment of the process data objects 788
Auto-DCB 275
  Hold time (C00106) 933
  Threshold (C00019) 914
AutoFailReset configuration (C00188) 957
AutoFailReset function 711
AutoFailReset repetition time (C00184) 956
AutoFailReset residual runtime (C00185) 956
Automatic DC-injection braking (auto DCB) 275
Automatic motor data identification 145
Auto-start option 127
Auto-start option (C00142) 941
Axis bus 859
Axis bus address and no. of nodes (C02430) 1302
Axis bus error status (C02436) 1303
Axis bus MessageError (C02437) 1304
Axis bus status (C02435) 1303
Axis bus time settings (C02431) 1303
AxisBusIO slave/master (C02440) 1304

## B

Basic drive functions 560
Basic functions 560
Behaviour after mains connection 127
Binary inputs (C00833) 1108
Binary inputs (Set2) (C00838) 1124
Binary inputs I/O level (C00843) 1137
Binding ID 51
Bit error 820, 868
Blocks 1425
Brake chopper 288
  Master-slave operation 296

Brake control 673
Brake energy management (C00175) 954
Brake resistance value (C00129) 937
Brake resistor 288
Brake resistor monitoring (I2xt) 305
Brake resistor overload threshold (C00572) 1034
Brake resistor utilisation (C00133) 937
Braking operation/brake energy management 288
Braking procedures 291

## C

C1 902
C10 912
C100 930
C1000 1163
C1001 1163
C101 930, 1624
C1010 1164, 1447, 1448, 1449
C1011 1164, 1450, 1451, 1452
C1012 1164, 1666
C1020 1165, 1634
C1021 1165, 1634
C1022 1165, 1634
C1023 1165, 1634
C1025 1166, 1524
C1026 1166, 1524
C1028 1166, 1525
C1029 1167, 1525
C103 931, 1624
C1030 1167, 1527
C1031 1167, 1527
C1033 1168, 1527
C1034 1168, 1527
C1035 1168, 1528
C104 932
C1040 1169, 1720
C1041 1169, 1720
C1042 1169, 1721
C1045 1170, 1479, 1480, 1481
C1046 1170, 1488, 1489, 1490
C1047 1170, 1580
C1048 1171, 1580
C1049 1171, 1454
C105 933
C1050 1171, 1455, 1456
C1051 1171, 1454, 1455, 1456
C1052 1172, 1455, 1456
C1053 1172, 1679
C1054 1172, 1679
C1055 1172, 1679
C1056 1173, 1679
C1057 1173, 1455
C1058 1173, 1672, 1675
C1059 1173, 1673, 1676
C106 933
C1060 1174, 1673, 1676
C1061 1174, 1673, 1676
C1062 1174, 1725
C1063 1175, 1725
C1064 1175, 1725
C1065 1176, 1726
C1066 1176, 1726
C1067 1177, 1455, 1544, 1668
C1068 1177, 1726
C1069 1178, 1544
C107 933
C1070 1178, 1544, 1545
C1071 1178, 1544
C1072 1178, 1544
C1073 1179
C1074 1179, 1545
C1075 1180, 1546
C1076 1180, 1534
C1077 1180, 1534
C1078 1181, 1534
C1079 1181, 1534
C1080 1181, 1534
C1081 1181, 1535
C1082 893, 1182
C1083 1182
C1084 1182
C1085 1182
C1086 1183
C1087 1183
C1088 1183
C1089 1184
C1090 1184, 1766
C1091 1185, 1766
C1092 1185, 1767
C1093 1186, 1767
C1094 1186, 1767
C1095 1187, 1767
C1098 1187
C11 912
C1100 1188, 1514, 1517, 1519
C1101 1188, 1515, 1517, 1519
C1108 1189
C1109 1189
C1110 1190
C1111 1190
C1112 1190
C1119 1191
C1120 1191
C1121 1191
C1122 1192
C1123 1192

C1124 1192
C1130 1193, 1771, 1772
C1131 1193, 1772
C1132 1194, 1772
C1133 1194, 1772
C1134 1194
C1135 1195
C1136 1195
C1138 1195, 1732, 1735, 1736, 1737
C1139 1196, 1732, 1735, 1736, 1737
C114 934
C1140 1196, 1738, 1739, 1740, 1741
C1141 1196, 1738, 1739, 1740, 1741
C115 934
C1150 1197, 1659, 1663
C1151 1197, 1659, 1663
C1152 1197, 1730
C1153 1198, 1730
C1154 1198, 1730
C1155 1199, 1730
C1156 1199, 1730
C1157 1200, 1731
C1158 1201, 1731
C117 935
C118 935
C1190 1201
C1191 1202
C1192 1202
C1193 1202
C12 912, 1624
C120 935
C1201 1202
C1202 1203
C1203 1203
C1204 1203
C1205 1203
C1206 1204
C121 936
C1210 1204
C1211 1205
C1213 1205
C1215 1205
C1216 1206
C1218 1207
C1219 1207
C122 936
C1220 1208
C1221 1209
C1222 1209
C1223 1210
C1224 1210
C1225 1210
C1226 1211
C1227 1211
C1228 1211
C1229 1212
C123 936
C1230 1212
C1231 1213
C1232 1213
C1233 1213
C1234 1214
C1235 1214
C1236 1214
C1237 1215
C1238 1215
C1239 1215
C124 936
C1240 1215
C1241 1217
C1242 1219
C1243 1219
C1244 1219
C1245 1220
C1246 1220
C1247 1220
C1248 1222
C1251 1223
C1252 1223
C129 937
C1293 1224
C1294 1224, 1494, 1495, 1496, 1498
C1295 1225
C1296 501, 507, 510, 521, 522, 586, 587, 590, 591, 592, 1225
C1297 1226
C1298 661, 1227
C1299 1228
C13 913, 1624
C130 937
C1300 1228
C1301 1229
C1302 1230
C1303 1230
C1304 1230
C1305 1231
C1306 1231
C1307 1231
C1308 1232
C1309 1232
C131 937
C1310 1233
C1311 1234
C1312 1235
C1313 1236
C1314 1237

C1315 1238
C1320 1239
C1321 1239
C1322 1240
C1323 1240
C1324 1241
C1325 1241
C133 937
C134 937, 1624
C1350 1242
C1351 1242
C1352 1243
C1353 1244
C136 938
C137 939
C138 940
C1400 1244, 1694
C1401 1244, 1694
C1402 1245, 1694
C1403 1245, 1694
C1404 1245, 1694
C1405 1246, 1700
C1406 1246, 1700
C1407 1247, 1700
C1408 1247, 1700
C1409 1247, 1700
C1410 1248, 1700
C1411 1248, 1701
C1412 1249, 1701
C1413 1249, 1701
C1414 1250, 1701
C1415 1250, 1702
C1416 1251, 1702
C1417 1251, 1702
C1418 1251, 1703
C1419 1252
C142 941
C1420 1252
C1421 1252
C1422 1252
C1423 1253
C1424 1253
C1425 1253
C1426 1253
C1427 1254
C1428 1254
C1429 1254
C1430 1254
C1431 1255
C1432 1255
C1433 1255
C1434 1255
C1435 1256
C1436 1256
C1437 1256, 1703
C1438 1257, 1705
C1439 1257, 1705
C144 941
C1440 1258, 1705
C1441 1258, 1706
C1442 1258, 1706
C1443 1259, 1707
C1444 1259, 1707
C1445 1259, 1707
C1446 1260, 1707
C1447 1260, 1707
C1448 1260, 1707
C1449 1261, 1708
C1450 1261, 1708
C1451 1262, 1708
C1452 1262, 1708
C1453 1263
C1454 1263
C1455 1264
C1460 1264, 1477
C1461 1264, 1477
C1462 1265, 1477
C1463 1265, 1477
C1464 1265, 1477
C1465 1265, 1477
C1466 1266, 1477
C1467 1266, 1477
C1468 1266, 1477
C1469 1266, 1594
C1470 1267, 1594
C148 942
C15 913
C150 943
C1501 1267
C155 944
C158 945
C159 946
C16 913
C160 946
C161 947
C162 947
C163 947
C164 950
C165 952
C166 952
C167 952
C1670 1268, 1467, 1472, 1473, 1474, 1475
C1671 1268, 1467, 1472, 1473, 1474, 1475
C1672 1269, 1467, 1472, 1473, 1474, 1475
C168 952
C169 953

C170 953
C171 953
C173 954
C174 954
C175 954
C1751 1269
C1752 1269
C1755 1269
C176 954
C1763 1269
C1764 1269
C1765 1269
C177 955
C1770 1270
C1771 1270
C1772 1270
C1773 1270
C1774 1270
C1775 1270
C178 955
C179 955
C18 914
C180 955
C181 956
C182 956, 1624
C184 956
C185 956
C186 956
C187 957
C188 957
C189 957
C19 914
C190 958, 1624
C1902 1271
C1903 1271
C1905 1271
C191 958
C192 958
C193 959
C199 959
C2 902
C20 915
C200 959
C201 960
C203 960
C204 961
C205 961
C206 961
C21 915
C210 961
C219 961
C22 915
C220 962, 1624
C2200 1271
C221 962, 1625
C2210 1272
C2212 1272
C2213 1273
C2214 1273
C2215 1274
C2216 1274
C2217 1274
C2218 1275
C2219 1275
C222 962, 1651
C2220 1275
C2221 1276
C2222 1276
C2223 1277
C2224 1277
C2225 1277
C2226 1278
C2227 1278
C2228 1278
C2229 1279
C223 962, 1651
C2230 1279
C2231 1280
C2232 1280
C2233 1280
C2234 1281
C2236 1281
C2237 1281
C2238 1282
C2239 1282
C224 962, 1651
C2240 1282
C2241 1283
C2242 1283
C2244 1283
C2245 1284
C2246 1284
C2249 1284
C225 963, 1651
C2250 1285
C2251 1285
C2252 1285
C2256 1286
C226 963, 1651
C2260 1286
C2261 1286
C2262 1287
C2263 1287
C2264 1287
C227 963, 1651
C2272 1288

C2273 1288
C2274 1289
C2275 1289
C2276 1289
C2278 1290
C2279 1290
C228 963, 1651
C2280 1290
C2281 1291
C2284 1291
C2285 1291
C2286 1292
C2287 1292
C2288 1292
C2289 1293
C2290 1293
C2291 1293
C2292 1294
C2293 1294
C2294 1294
C2295 1295
C2296 1295
C2297 1295
C2298 1296
C2299 1296
C23 915
C2300 1296
C2301 1297
C2302 1297
C2303 1297
C2304 1298
C2305 1298
C2306 1300
C2307 1300
C231 963, 1651
C2311 1300
C2312 1301
C2313 1301
C2314 1302
C2315 1302
C233 964, 1652
C234 964
C235 964
C236 964
C24 916
C241 964, 1625
C242 965, 1652
C243 965, 1652
C2430 1302
C2431 1303
C2435 1303
C2436 1303
C2437 1304
C2438 1304
C244 965, 1652
C2440 1304
C2442 1305
C2443 1306
C2444 1307
C245 965, 1652
C246 966, 1652
C247 966, 1652
C249 966, 1682
C25 916
C250 966, 1683, 1684
C251 966, 1568
C252 967, 1568
C253 967, 1568
C254 967
C2556 1307
C2580 1308
C2581 1308
C2582 1308
C2589 1310
C2593 1310
C26 916
C2607 1311
C2610 1312
C2611 1312
C265 967
C2652 1313
C27 917
C270 967
C271 968
C272 968
C273 968
C274 968
C275 968
C276 969
C28 917
C280 969
C2810 1314
C2811 1314
C2812 1315
C2813 1315
C2814 1316
C2815 1316
C2816 1317
C2817 1317
C2830 1318
C2840 1319
C2841 1319
C2842 1319
C2843 1320
C2844 402, 1320
C2845 402, 1320

C2853 1321
C2855 1321
C2859 1321
C2862 1322
C2863 1322
C2864 1322
C2865 1324
C2866 1325
C2867 1326
C2868 1327
C2870 1328
C2871 1328
C2872 1329
C2873 1329
C2874 1330
C2875 1330
C2876 1331
C2877 1331
C2878 1332
C2879 1333
C29 917
C290 969
C291 969
C295 969
C296 969
C297 969
C2993 1333
C2994 1333
C2995 1333
C2996 1334
C2998 1334
C3 905
C30 918
C301 970
C302 970
C304 970
C305 970
C306 970
C307 970
C308 970
C309 970
C310 970
C311 971
C312 971
C313 971
C314 971
C315 971
C316 971
C317 971
C320 971
C321 972
C322 972
C323 973
C324 973
C33 918
C338 974, 1442
C339 974, 1443
C34 918
C341 975
C342 976
C343 977
C344 978
C345 978
C347 979
C349 979
C350 980
C351 980
C352 980
C353 981
C354 981
C355 982
C356 982
C357 983
C358 983
C359 983
C36 919
C360 984
C364 984
C366 985
C367 985
C368 986
C369 986
C370 987
C371 987
C372 987
C381 987
C385 988
C386 988
C387 988
C39 919, 1624
C400 989, 1770
C401 989
C407 990
C408 990
C409 991
C410 991, 1710
C411 992, 1711
C412 992, 1711
C413 992, 1710
C417 993
C420 993
C421 993
C422 994
C423 994
C424 995
C425 995

C426 996
C427 996
C428 997
C430 997
C431 998
C432 999
C434 999
C435 1000
C436 1000
C437 1000
C439 1001
C440 1001
C441 1002
C442 1002
C443 1003
C444 1004
C445 1004
C446 1005
C447 1005
C448 1006
C449 1006
C451 1007
C452 1008
C453 1008
C455 1008
C456 1009
C458 1009
C459 1009
C460 1009
C461 1009
C462 1009
C463 1010
C464 1011
C465 1011
C466 1011
C467 1011
C468 1011
C469 1012
C470 1012, 1758
C471 1012, 1754
C472 1013, 1756
C473 1013, 1760
C474 1014, 1759
C475 1014, 1763, 1764
C476 1015, 1757
C477 1015, 1755
C478 1016, 1761
C479 1016, 1762
C480 1016, 1749
C481 1017, 1746
C482 1017, 1747
C483 1017, 1750
C484 1018, 1748
C485 1018, 1748
C486 1018, 1748
C487 1019, 1748
C488 1019
C490 1020
C491 1020
C492 1020
C493 1021
C494 1021
C495 1021
C496 1022
C497 1022
C498 1023
C499 1024
C5 906
C50 919
C505 1024
C506 1025
C507 1026
C508 1027
C509 1028
C51 920
C510 1029
C516 1029
C517 1030
C52 920
C53 920
C54 920
C540 1031
C545 1031
C55 921
C56 921
C560 1031
C561 1032
C563 1032
C565 1032
C566 1033
C567 1033
C569 1033
C57 921
C570 1034
C571 1034
C572 1034
C574 1035
C575 1035
C576 1035
C577 1035
C578 1035
C579 1036
C58 921
C580 1036
C581 1036
C582 1037

C583 1037
C584 1038
C585 1038
C586 1038
C588 1038
C59 922
C590 1039
C591 1039
C592 1040
C593 1040
C594 1041
C595 1042
C597 1043
C598 1043
C599 1043
C6 906
C60 922
C600 1044
C601 1044
C602 1044
C603 1045
C604 1045
C605 1046
C606 1046
C607 1046
C608 1046
C609 1047
C61 922
C610 1047
C611 1047
C612 1047
C613 1047
C615 1047
C617 1047
C62 922
C620 1048
C621 1052
C622 1059
C63 922
C630 1060, 1588, 1589
C631 1060, 1590, 1591, 1592
C632 1061, 1625
C633 1061, 1625
C634 1062, 1625
C635 1062, 1625
C636 1062, 1625
C637 1063, 1625
C638 1063, 1625
C639 1063, 1625
C64 923
C640 1063, 1625
C643 1063
C644 1064
C645 1064
C646 1064
C647 1064
C65 923
C650 1065, 1444, 1445, 1446
C653 1065
C654 1066
C658 1066
C659 1066
C66 923
C660 1066, 1569
C661 1067, 1570
C662 1067, 1571
C670 1067, 1638
C671 1067, 1639
C672 1068, 1640
C673 1068, 1641, 1642
C674 1068, 1641, 1642
C677 1068, 1575, 1576, 1577
C678 1069, 1578, 1579
C679 1069, 1609
C680 1069, 1458
C681 1069, 1458
C682 1070, 1458
C685 1070, 1463
C686 1070, 1463
C687 1070, 1463
C690 1071, 1464
C691 1071, 1464
C692 1071, 1464
C693 1071, 1465, 1466
C694 1072, 1465, 1466
C695 1072, 1465, 1466
C696 1072, 1638
C697 1072, 1639
C698 1072, 1640
C699 1073, 1608
C7 908
C70 924
C700 1073
C701 1074
C705 1076
C706 1077
C71 924
C710 1077
C711 1078
C712 1081
C715 1081
C716 1081
C717 1081
C72 924
C720 1082, 1554
C721 1082, 1556, 1557

C725 1083
C726 1083
C727 1084, 1752
C728 1084, 1752
C729 1084
C73 924
C730 1084
C731 1085
C732 1085
C734 1085
C735 1085
C736 1085
C74 925
C740 1085
C741 1085
C742 1085
C743 1085
C744 1086
C746 1086
C747 1086
C748 1086
C749 1086
C75 925
C750 1086
C751 1086
C753 1086
C76 925
C760 1087
C761 1088
C762 1090
C765 1090
C766 1090
C767 1090
C77 925
C78 926
C79 926
C8 910
C80 926
C800 1090, 1605
C801 1091, 1605
C802 1091, 1605
C803 1091, 1605
C804 1091, 1605
C805 1091
C806 1092, 1605
C807 1092, 1614
C808 1092, 1614
C809 1093, 1617
C81 927
C810 1093, 1617
C811 1093
C812 1094
C82 927
C820 1094, 1558
C821 1095, 1559
C822 1095, 1560
C823 1096, 1561
C824 1096, 1564
C825 1097, 1564
C826 1098, 1566
C827 1098, 1566
C828 1099, 1562
C829 1100, 1563
C83 927
C830 1100
C831 1103
C832 1106
C833 1108
C834 1112
C835 1112
C836 1116
C837 1120
C838 1124
C839 1129
C84 927
C840 1131
C841 1134
C843 1137
C844 1142
C85 927
C866 1143
C868 1144
C87 928
C876 1145
C877 1146
C88 928
C89 928
C890 1146
C90 928
C905 1147
C909 1147
C91 929
C910 1147
C915 1148
C916 1148
C917 1148
C918 1148
C919 1149
C92 929
C920 1149
C922 1149
C925 1149
C926 1150
C927 1150
C93 929
C937 1150

C938 1150
C939 1151
C940 1151, 1500, 1502, 1503, 1504
C941 1151, 1500, 1502, 1503, 1504
C942 1151, 1500, 1502, 1503, 1504
C95 929
C950 1152, 1583
C951 1152, 1583
C952 1152, 1583
C953 1152, 1583
C959 1153, 1520, 1524, 1527
C960 1153, 1520
C961 1153, 1520
C963 1154, 1521
C964 1154, 1521
C965 1154
C966 1154
C967 1155
C968 1155
C969 1156
C97 929
C971 1156
C972 1156
C973 1156
C975 1157
C976 1157
C977 1157
C978 1157
C979 1157
C98 930
C980 1158
C981 1158
C982 1158
C983 1158
C985 1159
C986 1159
C987 1159
C988 1159
C989 1159
C99 930
C990 1160
C991 1160
C992 1160
C993 1160
C994 1161
C995 1161
C996 1161
C997 1161
C998 1162
C999 1162
CA06: CAN CRC error (error message) 740
CA07: CAN bus warning (error message) 740
CA08: CAN bus stopped (error message) 741
CA0b: CAN HeartBeatEvent (error message) 741
CA0F: CAN control word (error message) 741
CAB configuration (C02444) 1307
CAB decoupling inputs (C02442) 1305
CAB decoupling value (C02443) 1306
CAB_Tx_Rx_diagnosis (C02438) 1304
Calculating mass inertia (SC) 248
Calculating mass inertia (SLVC) 207
CAN baud rate (C00351) 980
CAN data telegram 780
CAN decoupling PDOInOut (C00342) 976
CAN error status (C00345) 978
CAN ErrorCode (C00371) 987
CAN HeartBeat ConsumerTime (C00386) 988
CAN Heartbeat producer time (C00381) 987
CAN IN/OUT COBID source (C00353) 981
CAN input words (C00866) 1143
CAN management - error configuration (C00341) 975
CAN MessageError (C00364) 984
CAN monitoring times (C00357) 983
CAN node addr. HeartBeat producer (C00385) 988
CAN node address (C00350) 980
CAN on board 772
CAN output words (C00868) 1144
CAN reset node 779
CAN setting - DIP switch (C00349) 979
CAN slave/master (C00352) 980
CAN start remote node 786
CAN status (C00359) 983
CAN status HeartBeat producer (C00347) 979
CAN SYNC Rx identifier (C00367) 985
CAN sync transmission cycle time (C00369) 986
CAN SYNC Tx identifier (C00368) 986
CAN telegram counter (C00360) 984
CAN time settings (C00356) 982
CAN transmission blocking time (C00324) 973
CAN_Tx_Rx_Error (C00372) 987
CAN-ERR (LED) 778
CAN-GatewayAddr (C00387) 988
CAN-RUN (LED) 778
CANx_OUT data length (C00358) 983
CANxInOut
  Inversion (C00401) 989
Cause of controller inhibit (C00158) 945
Cause of quick stop QSP (C00159) 946
CE04: MCI communication error (error message) 740
CE0F: MCI control word (error message) 740
CE1: CAN RPDO1 (error message) 741
CE2: CAN RPDO2 (error message) 741
CE3: CAN RPDO3 (error message) 742
CE4: CAN bus off (error message) 740
CE5: CAN RPDO4 (error message) 742
Change of the operating mode 577

CI01: Module missing/incompatible (error message) 742
Ck01: Pos. HW limit switch (error message) 747
Ck02: Neg. HW limit switch (error message) 747
Ck03: Pos. SW limit position (error message) 747
Ck04: Neg. SW limit position (error message) 748
Ck05: Following error 1 (error message) 748
Ck06: Following error 1 (error message) 748
Ck07: Travel range limit exceeded (error message) 749
Ck08: Home position unknown (error message) 749
Ck09: Positioning mode invalid (error message) 749
Ck10: Implausible profile data (error message) 749
Ck11: Invalid operating mode (error message) 750
Ck12: Invalid profile number (error message) 750
Ck13: Error - MCKCtrlInterface function block (error message) 750
Ck14: Target position outside SW limit position (error message) 748
Ck15: Error status sign. brake (error message) 747
Ck16: Time overflow - manual operation (error message) 748
COB-ID 780
COBID (C00354) 981
COB-ID EMCY (I-1014) 831
COB-ID SYNC message (I-1005) 830
Commissioning wizard 8400 59
Communication control words (C00136) 938
Communication cycle period (I-1006) 831
Communication time 775
Compare FB interconnections 1420
Comparing applications 1420
Comparing interconnections 1420
Comparison L_Counter 1-3 (C01101) 1188
Configuring exception handling of the CAN PDOs 806
Configuring exception handling of the output terminals 428
Consumer heartbeat time (I-1016) 832
Control mode (C00007) 908
Control type 153
Control USER-LED 408
Control word (MCK) 569
Conventions used 27
Conventions used for variable identifiers 1378
Copy function block interconnection 1423
Copying a complete interconnection 1423
Copying an interconnection 1423
Copying elements 1414
Copying interconnection elements 1414
Copying objects 1414
Correction of the leakage inductance 269
Cosine phi (C00979) 1157
CountInx
  Counter content (C02841) 1319
  Parameter (C02840) 1319
CRC error 820, 868
Current AutoFailReset processes (C00187) 957
Current error (C00170) 953
Current monitoring
  Breaking current (C00124) 936
  Delay time (C00563) 1032
Current password protection (C00507) 1026
Current setpoint filter (band-stop filter) 251
Current switching frequency (C00725) 1083

## D

Data type 897
Data type entry 1378
DC braking
  Current (C00036) 919
  Hold time (C00107) 933
DCB (DC-injection braking) 275
DC-bus system
  Control of multiple brake choppers 296
DC-bus voltage (C00053) 920
DC-injection braking 274
Deactivate resolver error comp. (C00417) 993
Decel. time - main setpoint (C00013) 913
Decel. time - quick stop (C00105) 933
Decoupling AnalogOut (C00441) 1002
Defining a user-defined V/f characteristic 176
Defining the current limits 158
Defining the speed limits 158
Del.resp. to fault
  DC-bus overvoltage (C00601) 1044
Delay (C00983) 1158
Description data (C00199) 959
Device access protection 46
Device commands (C00002) 902
Device overload monitoring (Ixt) 300
Device personalisation 51
Device search function 115
Device status (C00137) 939
Device statuses (LED status display) 693
Device type (I-1000) 828
Device utilisat. threshold (Ixt) (C00123) 936
Device utilisation (Ixt) (C00064) 923
dF01: FW updated (error message) 745
dF10: AutoTrip reset (error message) 746
dF14: SW-HW invalid (error message) 745
dF15: DCCOM CU2 error (error message) 745
dF18: BU RCOM error (error message) 745
dF21: BU watchdog (error message) 746
dF22: CU watchdog (error message) 746
dF25: CU RCOM error (error message) 745
dF26: Appl. watchdog (error message) 746
dF50: Retain error (error message) 746
dF51: CuCcr error (error message) 746
dF52: BuCcr error (error message) 747
dH09: EEPROM power section (error message) 750
dH10: Fan failure (error message) 750
dH68: Adjustment data error CU (error message) 750

dH69: Adjustment data error BU (error message) 751
dH70: ControlUnit is unequal to BaseUnit (error message) 751
DI 1/2 & 6/7
  Function (C00115) 934
Diagnosis terminal X400 37
Diagnostic interface (DIAG) 39
Diagnostics X6
  Change baud rate (C01903) 1271
  current baud rate (C01905) 1271
  Max. baud rate (C01902) 1271
DigInX
  Inversion (C00114) 934
Digital frequency coupling 353
Digital terminals
  For electrical data see the hardware manual
  Input terminals 385
  Output terminals 406
  Touch probe detection 419
  User-defined terminal assignment 429
DigOut decoupling (C00447) 1005
DigOut decoupling value (C00448) 1006
DigOutX
  Inversion (C00118) 935
DIP switch settings 776
Direction of rotation check (manual control) 61
Display details of the current error 696
Display error details 696
Display of internal process factors in application units 1748
Display process factors on the keypad 1748
DIx
  Debounce time (C02830) 1318
  Level (C00443) 1003
DOx
  Delay times (C00423) 994
  Level (C00444) 1004
Drifting of the motor shaft 578
Drive interface 102
Drive-based safety 579
DRV-ERR (LED) 693
DRV-RDY (LED) 693

## E

EASY Starter 36
Elapsed-hour meter (C00178) 955
Electrical data
  see hardware manual
E-mail to Lenze 1806
Emergency 824
Emergency operation in case of encoder open circuit 190, 231, 320
Encoder 363, 365
Encoder - angular drift monitoring 379
Encoder evaluation method 363
Encoder evaluation method DigIn12 (C00496) 1022
Encoder scanning time (C00425) 995
Encoder/feedback system 321
Energy display (C00981) 1158
Energy-saving V/f characteristic control (VFCplusEco) 180
Engineer 36
Error detection 820
Error ID 717, 719
Error information (C00165) 952
Error information text (C00166) 952
Error messages 715
Error messages (short overview) 721
Error messages (system bus) 813
Error number 715, 718
  xx.0000.00980 724
  xx.0000.00981 724
  xx.0000.00982 724
  xx.0000.00983 724
  xx.0000.00984 725
  xx.0000.00985 725
  xx.0000.00986 725
  xx.0000.00987 725
  xx.0111.00002 725
  xx.0111.00003 726
  xx.0111.00004 726
  xx.0111.00006 726
  xx.0111.00007 726
  xx.0119.00000 727
  xx.0119.00001 727
  xx.0119.00002 727
  xx.0119.00003 727
  xx.0119.00012 727
  xx.0119.00015 728
  xx.0119.00020 728
  xx.0119.00021 728
  xx.0119.00022 728
  xx.0119.00050 729
  xx.0123.00001 729
  xx.0123.00007 729
  xx.0123.00014 729
  xx.0123.00015 730
  xx.0123.00016 730
  xx.0123.00017 730
  xx.0123.00024 731
  xx.0123.00026 731
  xx.0123.00027 731
  xx.0123.00028 731
  xx.0123.00029 731
  xx.0123.00030 732
  xx.0123.00031 732
  xx.0123.00032 732
  xx.0123.00033 732
  xx.0123.00034 732
  xx.0123.00056 733
  xx.0123.00057 733
  xx.0123.00058 733
  xx.0123.00059 733
  xx.0123.00060 734
  xx.0123.00061 734

xx.0123.00062 734
xx.0123.00065 734
xx.0123.00071 734
xx.0123.00074 735
xx.0123.00075 735
xx.0123.00090 735
xx.0123.00093 735
xx.0123.00094 736
xx.0123.00095 736
xx.0123.00096 736
xx.0123.00097 736
xx.0123.00098 736
xx.0123.00099 737
xx.0123.00105 737
xx.0123.00145 737
xx.0123.00200 737
xx.0123.00201 737
xx.0123.00205 738
xx.0123.00206 738
xx.0123.00207 738
xx.0123.00208 738
xx.0125.00001 739
xx.0125.00002 739
xx.0126.00001 739
xx.0126.00002 739
xx.0127.00002 740
xx.0127.00015 740
xx.0131.00000 740
xx.0131.00006 740
xx.0131.00007 740
xx.0131.00008 741
xx.0131.00011 741
xx.0131.00015 741
xx.0135.00001 741
xx.0135.00002 741
xx.0135.00003 742
xx.0135.00004 742
xx.0140.00013 742
xx.0144.00001 742
xx.0144.00002 743
xx.0144.00003 743
xx.0144.00004 743
xx.0144.00007 744
xx.0144.00008 744
xx.0144.00009 744
xx.0144.00010 744
xx.0144.00011 744
xx.0144.00012 745
xx.0144.00013 745
xx.0145.00001 745
xx.0145.00014 745
xx.0145.00015 745
xx.0145.00024 745
xx.0145.00025 745
xx.0145.00026 746
xx.0145.00033 746
xx.0145.00034 746
xx.0145.00035 746
xx.0145.00050 746
xx.0145.00051 746
xx.0145.00052 747
xx.0184.00001 747
xx.0184.00002 747
xx.0184.00005 747
xx.0184.00007 747
xx.0184.00008 748
xx.0184.00015 748
xx.0184.00064 748
xx.0184.00153 748
xx.0184.00154 748
xx.0184.00155 749
xx.0184.00156 749
xx.0184.08005 749
xx.0184.08007 749
xx.0184.08009 750
xx.0184.08014 750
xx.0184.08015 750
xx.0400.00009 750
xx.0400.00016 750
xx.0400.00104 750
xx.0400.00105 751
xx.0400.00106 751
Error number masked (C00162) 947
Error register (I-1001) 828
Error subject area 716, 718
Error type 716
Export error texts 720
Exporting logbook entries 704
Extended status word (C00155) 944

## F

Failure indication (C00561) 1032
FB Editor 1373
  Functional range 1373
FC01: Switching frequency reduction (error message) 736
FC02: Maximum speed for Fchop (error message) 736
FC03: Field controller limitation (error message) 737
Feed constant 596
Feedback to Lenze 1806
Feedforward control of the current controller 262
FI brake 291
Field weakening for synchronous motors 283
Field weakening oscillation damping (C00236) 964
Fieldbus interface 852
Field-oriented motor currents (C00937) 1150
Filter setpoint feedforward control (C00275) 968
Filter time - earth-fault detect. is running (C01770) 1270
Filter time constant (C00497) 1022
Firmware (C00201) 960
Firmware product type (C00200) 959
Firmware update 117
Firmware version (C00099) 930
Firmware version (C00100) 930

Fixed setpoint x (L_NSet_1 n-Fix) (C00039) 919
Flying restart function 271
  Activation (C00990) 1160
  Current (C00994) 1161
  Integration time (C00993) 1160
  Process (C00991) 1160
  Restart on the fly fd_add (C00989) 1159
  Start frequency (C00992) 1160
Following error 609
Following error monitoring system 609
Format error 820, 868
FreqIn12
  Function (C02844) 1320
  PosIn comparison value (C02845) 1320
FreqInxx
  Gain (C02843) 1320
  Offset (C02842) 1319
FreqInxx_dnOut_p (C00449) 1006
FreqInxx_nOut_a (C00446) 1005
FreqInxx_nOut_v (C00445) 1004
Frequency limitation (C00910) 1147
Function block editor 1373
Function block interconnection 35
Function blocks 1425
Function L_Counter 1-3 (C01100) 1188
Function library 1425

## G

Gearbox ratio 595
General data (CAN on board) 773
General purpose functions 554
GP functions (GeneralPurpose) 554

## H

Hardware limit switches 604
Heartbeat protocol 821
Heatsink temperature (C00061) 922
Hiperface
  Detected TypCode (C00492) 1020
  Resolutions (C00494) 1021
  Settings (C00499) 1024
  TypCode (C00493) 1021
Hiperface messages (C00491) 1020
Holding brake
  Activation time (C02593) 1310
  Operating mode (C02580) 1308
  Setting (C02582) 1308
  Speed thresholds (C02581) 1308
  Status (C02607) 1311
  Time system (C02589) 1310
Holding brake control 673
Homing 618
Homing on the fly 637
HomPosAvailable (status bit) 573

## I

I-1000 828
I-1001 828
I-1003 829
I-1005 830
I-1006 831
I-1014 831
I-1016 832
I-1017 833
I-1018 833
I-1200 834
I-1201 835
I-1400 837
I-1401 838
I-1402 839
I-1403 840
I-1600 841
I-1601 842
I-1602 842
I-1603 843
I-1800 844
I-1801 846
I-1802 847
I-1803 848
I-1A00 849
I-1A01 849
I-1A02 850
I-1A03 850
Id1: Motor data identification error (error message) 733
Id2: Motor data identification error (error message) 733
Id3: CINH identification (error message) 733
Id4: Resistance identification error (error message) 733
Id5: Pole position identification error (error message) 735
Id6: Resolver ident. error (error message) 735
Id7: Motor control does not match motor data (error message) 734
Id8: Speed sensor has not been set (error message) 734
Identification procedure (C02867) 1326
Identifier (CAN) 780
Identifiers of the parameter data objects 809
Identifiers of the process data objects 801
Identity (C00219) 961
Identity object (I-1018) 833
Imax controller 166
Imax in generator mode (C00023) 915
Imax in motor mode (C00022) 915
Imax/M controller gain (C00073) 924
Inching mode 638
Information regarding the validity 25
Initial value motor overload (I²xt) (C00122) 936
Insert options for copied elements 1416
Inserting complete interconnection from reference project 1424

Inserting copied elements 1416
Inserting FB interconnection from reference project 1424
Integrated error detection 820
Internal control signals (C00138) 940
Inversion of gearbox stages (C01067) 1177
Inverter motor brake 291
  nAdd (C00987) 1159
  PT1 filter time (C00988) 1159

## J

Jerk limitation 659

## K

Keypad 37, 39
  Default manual jog (C00453) 1008
  Default parameter (C00466) 1011
  Default welcome screen (C00467) 1011
  Display of internal process factors 1748
  Fct. STOP key (C00469) 1012
  LCD display 697
  Set language 41
  Timeout welcome screen (C00465) 1011
  User level 40
Keypad analog values (C00728) 1084
Keypad language 41
Keypad Userlevel (C00001) 902
Kp position controller (C00254) 967
KTY motor temperature compensation (C02878) 1332

## L

L_Absolut 1430
L_Absolute_1 1430
L_Absolute_2 1430
L_AddSub 1431
L_AddSub_1 1431
L_AnalogSwitch 1432
L_AnalogSwitch_1 1432
L_AnalogSwitch_2 1433
L_AnalogSwitch_3 1434
L_AnalogSwitch_4 1435
L_AnalogSwitch_5 1436
L_And 1437
L_And_1 1437
L_And_2 1438
L_And_3 1439
L_And5 1440
L_And5_1 1440
L_And5_2 1441
L_Arithmetik 1442
L_Arithmetik 3-5
  Function (C00650) 1065
L_Arithmetik_1 1442
  Function (C00338) 974
L_Arithmetik_2 1443
  Function (C00339) 974
L_Arithmetik_3 1444
L_Arithmetik_4 1445
L_Arithmetik_5 1446
L_ArithmetikPhi 1447
L_ArithmetikPhi 1-3
  Function (C01010) 1164
L_ArithmetikPhi 4-6
  Function (C01011) 1164
L_ArithmetikPhi_1 1447
L_ArithmetikPhi_2 1448
L_ArithmetikPhi_3 1449
L_ArithmetikPhi_4 1450
L_ArithmetikPhi_5 1451
L_ArithmetikPhi_6 1452
L_CalcDiameter 1453
L_CalcDiameter_1 1453
  Current diameter (C01057) 1173
  Diameter recalculation (C01050) 1171
  Filter time constant (C01051) 1171
  Status (C01049) 1171
  Web break monitoring (C01052) 1172
L_Compare 1458
L_Compare 4-5
  Fct. (C00693) 1071
  Hysteresis (C00694) 1072
  Window (C00695) 1072
L_Compare_1 1458
  Fct. (C00680) 1069
  Hysteresis (C00681) 1069
  Window (C00682) 1070
L_Compare_2 1463
  Fct. (C00685) 1070
  Hysteresis (C00686) 1070
  Window (C00687) 1070
L_Compare_3 1464
  Fct. (C00690) 1071
  Hysteresis (C00691) 1071
  Window (C00692) 1071
L_Compare_4 1465
L_Compare_5 1466
L_ComparePhi 1467
L_ComparePhi 1-5
  Function (C01670) 1268
  Hysteresis (C01671) 1268
  Window (C01672) 1269
L_ComparePhi_1 1467
L_ComparePhi_2 1472
L_ComparePhi_3 1473
L_ComparePhi_4 1474
L_ComparePhi_5 1475
L_ConvActPos 1476
  Damping (C01468) 1266
  DeadBand (C01467) 1266
  DelayTime (C01465) 1265
  Length (C01462) 1265

MaxPos/MinPos (C01461) 1264
TConst (C01460) 1264
TConstVAdd (C01466) 1266
UpperRef/LowerRef (C01463) 1265
WindowSetPos (C01464) 1265
L_ConvActPos_1 1476
L_ConvAP 1479
L_ConvAP 1-3
Numerator/denominator (C01045) 1170
L_ConvAP_1 1479
L_ConvAP_2 1480
L_ConvAP_3 1481
L_ConvBitsToWord 1482
L_ConvBitsToWord_1 1482
L_ConvBitsToWord_2 1483
L_ConvBitsToWord_3 1484
L_ConvDIntToWords 1485
L_ConvDIntToWords_1 1485
L_ConvDIntToWords_2 1486
L_ConvDIntToWords_3 1487
L_ConvPA 1488
L_ConvPA 1-3
byDivision (C01046) 1170
L_ConvPA_1 1488
L_ConvPA_2 1489
L_ConvPA_3 1490
L_ConvPP 1491
L_ConvPP_1 1491
L_ConvPP_2 1492
L_ConvPP_3 1493
L_ConvUnitsToIncr 1494
L_ConvUnitsToIncr_1 1494
L_ConvUnitsToIncr_2 1496
L_ConvUnitsToIncr_3 1498
L_ConvW 1500
L_ConvW conversion method (C00942) 1151
L_ConvW denominator (C00941) 1151
L_ConvW numerator (C00940) 1151
L_ConvW_1 1500
L_ConvW_2 1502
L_ConvW_3 1503
L_ConvW_4 1504
L_ConvWordsToDInt 1505
L_ConvWordsToDInt_1 1505
L_ConvWordsToDInt_2 1506
L_ConvWordsToDInt_3 1507
L_ConvWordToBits 1508
L_ConvWordToBits_1 1508
L_ConvWordToBits_2 1509
L_ConvWordToBits_3 1510
L_ConvX 1511
L_ConvX_1 1511
L_ConvX_2 1512
L_ConvX_3 1513
L_Counter 1514
L_Counter_1 1514
L_Counter_2 1516
L_Counter_3 1518
L_Curve 1520
Current output value (C00959) 1153
L_Curve_1 1520
Input limitation (C00961) 1153
Selected curve type (C00960) 1153
Table X-values (C00963) 1154
Table Y-values (C00964) 1154
L_Curve_2 1524
Input limitation (C01026) 1166
Selected curve type (C01025) 1166
Table X-values (C01028) 1166
Table Y-values (C01029) 1167
L_Curve_3 1526
Input limitation (C01031) 1167
SelectCurve (C01035) 1168
Selected curve type (C01030) 1167
Table X-values (C01033) 1168
Table Y-values (C01034) 1168
L_DFlipFlop 1531
L_DFlipFlop_1 1531
L_DFlipFlop_2 1532
L_DFRFG 1533
L_DFRFG_1 1533
Following error (C01078) 1181
Max. speed-up (C01077) 1180
Offset (C01080) 1181
Sync. direction / TP function (C01081) 1181
Synchronisation window (C01079) 1181
Times (C01076) 1180
L_DFSET 1542
L_DFSET_1 1542
Angular trimming (C01070) 1178
Following error limit (C01071) 1178
Multiplier - angular trimming (C01072) 1178
Ramp settings (C01069) 1178
Setting (C01073) 1179
Synchronisation mode (C01075) 1180
Zero pulse divider (C01074) 1179
L_DigitalDelay 1554
L_DigitalDelay 2..3
Delay (C00721) 1082
L_DigitalDelay_1 1554
Delay (C00720) 1082
L_DigitalDelay_2 1556
L_DigitalDelay_3 1557
L_DigitalLogic 1558
L_DigitalLogic_1 1558
Function (C00820) 1094
Truth table (C00821) 1095
L_DigitalLogic_2 1560
Function (C00822) 1095
Truth table (C00823) 1096

L_DigitalLogic_3 1562
Function (C00828) 1099
Truth table (C00829) 1100
L_DigitalLogic5 1564
L_DigitalLogic5_1 1564
Function (C00824) 1096
Truth table (C00825) 1097
L_DigitalLogic5_2 1566
Function (C00826) 1098
Truth table (C00827) 1098
L_DT1 1568
L_DT1_1 1568
Gain (C00252) 967
Sensitivity (C00253) 967
Time constant (C00251) 966
L_FixSet_a 1569
L_FixSet_a_1 1569
Analog values (C00660) 1066
L_FixSet_w 1570
L_FixSet_w_1 1570
Fixed values (C00661) 1067
L_FixSet_w_2 1571
Fixed values (C00662) 1067
L_GainOffset 1572
L_GainOffset_1 1572
L_GainOffset_2 1573
L_GainOffset_3 1574
L_GainOffsetP 1575
L_GainOffsetP 1-3
Parameter (C00677) 1068
L_GainOffsetP_1 1575
L_GainOffsetP_2 1576
L_GainOffsetP_3 1577
L_GainOffsetPhiP 1578
L_GainOffsetPhiP 1-2
Parameter (C00678) 1069
L_GainOffsetPhiP_1 1578
L_GainOffsetPhiP_2 1579
L_GearComp 1580
L_GearComp_1 1580
Num_Denom (C01048) 1171
Offset (C01047) 1170
L_Interpolator 1582
L_Interpolator_1 1582
Activation FB functions (C00950) 1152
Limit value - error cycles (C00952) 1152
No. of interpolation steps (C00951) 1152
Speed-up (C00953) 1152
L_JogCtrlEdgeDetect_1 (C00488) 1019
L_JogCtrlExtension 1586
L_JogCtrlExtension_1 1586
L_Limit 1588
L_Limit 1-2
Min/Max (C00630) 1060
L_Limit_1 1588
L_Limit_2 1589
L_LimitPhi 1590
L_LimitPhi 1-3
Min/Max (C00631) 1060
L_LimitPhi_1 1590
L_LimitPhi_2 1591
L_LimitPhi_3 1592
L_MckCtrlInterface 583
L_MckCtrlInterface_1 583
Alternative function (C01297) 1226
L_MckStateInterface 590
L_MckStateInterface_1 590
Pos. selection (C01295) 1225
Status (C01293) 1224
L_MFail 1593
Tn (C01470) 1267
Vp (C01469) 1266
L_MFail_1 1593
L_MPot 1604
L_MPot_1 1604
Acceleration time (C00802) 1091
Deceleration time (C00803) 1091
Inactive fct. (C00804) 1091
Init fct. (C00805) 1091
Lower limit (C00801) 1091
Upper limit (C00800) 1090
Use (C00806) 1092
L_MulDiv 1608
L_MulDiv_1 1608
Parameter (C00699) 1073
L_MulDiv_2 1609
Parameter (C00679) 1069
L_Mux 1610
L_Mux_1 1610
L_Negation 1611
L_Negation_1 1611
L_Negation_2 1612
L_NLim 1613
Current status (C00812) 1094
L_NLim_1 1613
Current output value (C00811) 1093
Max.SkipFrq. (C00807) 1092
Min.SkipFrq. (C00808) 1092
L_NLim_2 1616
Max.SkipFrq. (C00809) 1093
Min.SkipFrq. (C00810) 1093
L_Not 1618
L_Not_1 1618
L_Not_2 1618
L_Not_3 1619
L_Not_4 1619
L_Not_5 1620
L_Not_6 1620
L_Not_7 1621
L_NSet 1622

L_NSet_1 1622
Accel. time - add. setpoint (C00220) 962
Additional value output (C00639) 1063
Decel. time - add. setpoint (C00221) 962
Hyst. NSet reached (C00241) 964
Max.SkipFrq. (C00632) 1061
Min.SkipFrq. (C00633) 1061
nMaxLimit (C00635) 1062
nMinLimit (C00636) 1062
nNOut_a (C00640) 1063
Output blocking zones (C00637) 1063
Output ramp rounding (C00638) 1063
Ramp smoothing (C00134) 937
Setpoint arithmetic (C00190) 958
S-ramp time PT1 (C00182) 956
wState (C00634) 1062
L_Odometer 1633
L_Odometer_1 1633
Edge selection (C01023) 1165
Input selection (C01022) 1165
Memory length (C01020) 1165
Memory type (C01021) 1165
L_OffsetGain 1635
L_OffsetGain_1 1635
L_OffsetGain_2 1636
L_OffsetGain_3 1637
L_OffsetGainP 1638
L_OffsetGainP_1 1638
Gain (C00670) 1067
Offset (C00696) 1072
L_OffsetGainP_2 1639
Gain (C00671) 1067
Offset (C00697) 1072
L_OffsetGainP_3 1640
Gain (C00672) 1068
Offset (C00698) 1072
L_OffsetGainPhiP 1641
L_OffsetGainPhiP 1-2
Gain (C00674) 1068
Offset (C00673) 1068
L_OffsetGainPhiP_1 1641
L_OffsetGainPhiP_2 1642
L_Or 1643
L_Or_1 1643
L_Or_2 1644
L_Or_3 1645
L_Or_4 1646
L_Or5 1647
L_Or5_1 1647
L_Or5_2 1648
L_PCTRL 1649
L_PCTRL_1 1649
Acceleration time (C00227) 963
Acceleration time influence (C00243) 965
Deceleration time (C00228) 963
Deceleration time influence (C00244) 965
Internal actual value nAct_a (C00246) 966
Kd (C00224) 962
MaxLimit (C00225) 963
MinLimit (C00226) 963
Operating mode (C00242) 965
Operating range (C00231) 963
PID output value (C00245) 965
Root function (C00233) 964
Tn (C00223) 962
Vp (C00222) 962
Window setpoint reached (C00247) 966
L_PhaseDiff 1657
L_PhaseDiff_1 1657
L_PhaseDiff_2 1658
L_PhaseIntK 1659
Compare (C01151) 1197
Function (C01150) 1197
L_PhaseIntK_1 1659
L_PhaseIntK_2 1663
L_PhiAdd 1665
L_PhiAdd_1 1665
L_PhiDiv 1666
byDivision (C01012) 1164
L_PhiDiv_1 1666
L_PhiIntegrator 1667
L_PhiIntegrator_1 1667
L_PosCtrlLin 1671
L_PosCtrlLin 1-2
Limit stop (C01058) 1173
Positioning behaviour (C01059) 1173
Ramps (C01060) 1174
Traversing speeds (C01061) 1174
L_PosCtrlLin_1 1671
L_PosCtrlLin_2 1674
L_PosiShaftCtrlInterface 1677
L_PosiShaftCtrlInterface_1 1677
L_ProcessCtrl 1678
L_ProcessCtrl_1 1678
Controller gain (C01056) 1173
Controller times (C01053) 1172
Correcting variable limitation (C01055) 1172
System deviation (C01054) 1172
L_PT1 1682
L_PT1 2-3
Time constant (C00250) 966
L_PT1_1 1682
Time constant (C00249) 966
L_PT1_2 1683
L_PT1_3 1684
L_RLQ 1685
L_RLQ_1 1685
L_RSFlipFlop 1686
L_RSFlipFlop_1 1686
L_RSFlipFlop_2 1687
L_SampleHold 1688

L_SampleHold_1 1688
L_SampleHold_2 1689
L_Sequencer 1690
L_Sequencer_1 1690
Aktueller Zählerstand (C01443) 1259
Bool signal (C01453) 1263
BranchIn (C01415) 1250
BranchLevel (C01416) 1251
BranchStep (C01417) 1251
BranchStep1 (C01418) 1251
BranchStep10 (C01427) 1254
BranchStep11 (C01428) 1254
BranchStep12 (C01429) 1254
BranchStep13 (C01430) 1254
BranchStep14 (C01431) 1255
BranchStep15 (C01432) 1255
BranchStep16 (C01433) 1255
BranchStep17 (C01434) 1255
BranchStep18 (C01435) 1256
BranchStep19 (C01436) 1256
BranchStep2 (C01419) 1252
BranchStep20 (C01437) 1256
BranchStep3 (C01420) 1252
BranchStep4 (C01421) 1252
BranchStep5 (C01422) 1252
BranchStep6 (C01423) 1253
BranchStep7 (C01424) 1253
BranchStep8 (C01425) 1253
BranchStep9 (C01426) 1253
Current step (C01401) 1244
INT signal (C01454) 1263
PS_WD_Step (C01410) 1248
PS_WD_Time (C01409) 1247
PSInput (C01405) 1246
PSLevel (C01406) 1246
PSProfil (C01407) 1247
PSStep (C01408) 1247
SchrittwertZähler (C01445) 1259
Sequence control
Control word (C01403) 1245
Status (C01402) 1245
Sequence step (C01400) 1244
SetZählernummer (C01441) 1258
SetZählerStartwert (C01442) 1258
Standby Eingang f. Ende (C01449) 1261
Standby Modusfolger (C01451) 1262
Standby PolaritätEingang (C01450) 1261
StandbySollwert (C01452) 1262
Step for bCancel = TRUE (C01404) 1245
SwitchLevel1 (C01412) 1249
SwitchLevel2 (C01414) 1250
SwitchOut1 (C01411) 1248
SwitchOut2 (C01413) 1249
Wait Eingang f. Weiter (C01439) 1257
WaitLevel (C01440) 1258
Wartezeit (C01438) 1257
Word signal (C01455) 1264
ZählerNummer (C01444) 1259
ZählerSprungziel (C01447) 1260
ZählerVergleichsfunktion (C01448) 1260
ZählerVergleichswert (C01446) 1260
L_SignalMonitor_a 1710
Offs./gain (C00413) 992
Signal sources (C00410) 991
L_SignalMonitor_b 1711
Inversion (C00412) 992
Signal sources (C00411) 992
L_SignalSwitch 1712
L_SignalSwitch_1 1712
L_SignalSwitch_2 1713
L_SignalSwitch_3 1714
L_SignalSwitch_4 1715
L_SignalSwitch32 1716
L_SignalSwitch32_1 1716
L_SignalSwitch32_2 1717
L_SignalSwitch32_3 1718
L_SQrt 1719
L_SQrt_1 1719
L_SRFG 1720
L_SRFG_1 1720
L_SRFG_1..2 limitations of output values (C01042) 1169
L_SRFG_1..2 linear ramp time (C01040) 1169
L_SRFG_1..2 S-ramp time (C01041) 1169
L_SRFG_2 1722
L_SwitchPoint 1724
L_SwitchPoint_1 1724
CenterMode (C01064) 1175
Dead time (C01062) 1174
Dead time factor (C01108) 1189
Hysteresis (C01063) 1175
inversion (C01068) 1177
Running time (C01065) 1176
Status (C01066) 1176
L_SwitchPointPar 1729
L_SwitchPointPar_1 1729
CenterMode (C01154) 1198
Dead time (C01152) 1197
Dead time factor (C01109) 1189
Hysteresis (C01153) 1198
Inversion (C01158) 1201
Position (C01157) 1200
Running time (C01155) 1199
Status (C01156) 1199
L_Transient 1732
L_Transient 1-4
Function (C01138) 1195
Pulse duration (C01139) 1196
L_Transient 5-8
Function (C01140) 1196
L_Transient 5-8 pulse duration (C01141) 1196
L_Transient_1 1732
L_Transient_2 1735
L_Transient_3 1736

L_Transient_4 1737
L_Transient_5 1738
L_Transient_6 1739
L_Transient_7 1740
L_Transient_8 1741
LA_NCtrl 441, 470
  Analog connection list (C00700) 1073
  Application block 441, 470
  Digital connection list (C00701) 1074
LA_NCtrl_In 441, 470
LA_NCtrl_Out 441, 470
LA_SwitchPos 530
  Analog connection list (C00760) 1087
  Application block 530
  Digital connection list (C00761) 1088
LA_SwitchPos_In 530
LA_SwitchPos_Out 530
LA_TabPos 502
  Analog connection list (C00710) 1077
  Application block 502
  Digital connection list (C00711) 1078
  phi connection list (C00712) 1081
LA_TabPos_In 502
LA_TabPos_Out 502
Laser position encoder 349
Layout of the safety instructions 30
LCD display (keypad) 697
LED "AB-STATE" 692, 860
LED "CAN-ERR" 778
LED "CAN-RUN" 778
LED "DRV-ERR" 693
LED "USER" 692
LED status displays 692
L-force »EASY Starter« 36
L-force »Engineer« 36
Library 1425
Limit position monitoring 601
Limit switches 604
Lock bFail at TroubleQSP 133
Logbook - analog elements (C00164) 950
Logbook - binary elements (C00163) 947
Logbook access index User (C00191) 958
Logbook data User (C00192) 958
Logbook element User (C00193) 959
Logbook setting (C00169) 953
LP_CanIn decoupling value (C00343) 977
LP_CanIn mapping (C00407) 990
LP_CanIn mapping (C00409) 991
LP_CanIn mapping selection (C00408) 990
LP_CanIn1 789
LP_CanIn2 791
LP_CanIn3 793
LP_CanIn4 795
LP_CanOut decoupling value (C00344) 978
LP_CanOut1 797
LP_CanOut2 798
LP_CanOut3 799
LP_CanOut4 800
LP_MciIn 856
LP_MciOut 857
LP1: Motor phase failure (error message) 737
LS_AnalogIn1
  PT1 time constant (C00440) 1001
LS_AnalogInput 417
LS_AnalogOutput 418
LS_AxisBusAux 872
LS_AxisBusIn 871
LS_AxisBusIO 879
LS_AxisBusOut 870
LS_Brake 674
LS_CANManagement 851
LS_DeviceMonitor 318
LS_DFOut 356
  Angular offset (C00545) 1031
  Const. (C00030) 918
  Function (C00540) 1031
LS_DigitalInput 398
LS_DigitalOutput 408
LS_DisFree 1746
LS_DisFree (C00481) 1017
LS_DisFree_a 1747
LS_DisFree_a (C00482) 1017
LS_DisFree_b 1749
LS_DisFree_b (C00480) 1016
LS_DisFree_p 1750
LS_DisFree_p (C00483) 1017
LS_DriveInterface 130
  bNActCompare (C00024) 916
  bNActHysterese (C00025) 916
  Error message config. (C00148) 942
LS_Keypad 1751
LS_Keypad digital values(C00727) 1084
LS_MotionControlKernel 562
LS_MotorInterface 312
LS_MultiEncoder 357
  bFail information (C00451) 1007
  bFail selection (C00431) 998
  Current position (C01119) 1191
  Encoder constant (C01111) 1190
  Encoder type (C00422) 994
  Max. initialization time (C00430) 997
  Position values (C01112) 1190
  Solid measure (C01110) 1190
  Supply voltage (C00421) 993
LS_ParFix 1753
LS_ParFix_2 1753
LS_ParFree 1754
LS_ParFree (C00471) 1012
LS_ParFree_2 1755

LS_ParFree_2 (C00477) 1015
LS_ParFree_a 1756
LS_ParFree_a (C00472) 1013
LS_ParFree_a_2 1757
LS_ParFree_a_2 (C00476) 1015
LS_ParFree_b 1758
LS_ParFree_b (C00470) 1012
LS_ParFree_p 1759
LS_ParFree_p (C00474) 1014
LS_ParFree_v 1760
LS_ParFree_v (C00473) 1013
LS_ParFree_v_2 1761
LS_ParFree_v_2 (C00478) 1016
LS_ParFree32 1762
LS_ParFree32 (C00479) 1016
LS_ParFreeUnit 1763
LS_ParFreeUnit_1_2 (C00475) 1014
LS_ParFreeUnit_2 1764
LS_ParReadWrite 1765
LS_ParReadWrite 1-6
  Arithmetic mode (C01093) 1186
  Configuration (C01098) 1187
  Cycle time (C01091) 1185
  Denominator (C01095) 1187
  FailState (C01092) 1185
  Index (C01090) 1184
  Numerator (C01094) 1186
LS_ParReadWrite_1 1765
LS_ParReadWrite_2 1765
LS_ParReadWrite_3 1765
LS_ParReadWrite_4 1765
LS_ParReadWrite_5 1765
LS_ParReadWrite_6 1765
LS_PulseGenerator 1769
LS_PulseGenerator (C00400) 989
LS_Resolver 331
  bFail information (C00452) 1008
  bFail selection (C00432) 999
  Number of pole pairs (C00925) 1149
LS_RetainData 1771
  16Bit data (C01131) 1193
  16Bit data (C01134) 1194
  32Bit data (C01132) 1194
  32bit data (C01135) 1195
  Bool data (C01133) 1194
  Bool data (C01136) 1195
  Selection (C01130) 1193
LS_SetError_1 752
LS_SetError_2 753
LS_SetError_x
  Error number (C00161) 947
LS_SyncManagement 881
LS_TouchProbe 425
LS_WriteParamList 893
  Auto-DCB
  Hold time (C02244) 1283
  Threshold (C02215) 1274
DC braking
  Current (C02219) 1275
  Hold time (C02245) 1284
Empfindlichkeit Sollwertvorsteuerung (C02264) 1287
Error line (C01084) 1182
Error status (C01083) 1182
Execute Mode (C01082) 1182
Field weakening oscillation damping (C02251) 1285
Flying restart function
  Activation (C02295) 1295
  Current (C02299) 1296
  Integration time (C02298) 1296
  Process (C02296) 1295
  Start frequency (C02297) 1295
Frequency limitation (C02274) 1289
Function (C02200) 1271
Imax in generator mode (C02218) 1275
Imax in motor mode (C02217) 1274
Imax/M controller gain (C02223) 1277
Index (C01085) 1182
Inverter motor brake
  nAdd (C02293) 1294
  PT1 filter time (C02294) 1294
Kp position controller (C02252) 1285
Manual entry of motor type (C02315) 1302
Max. motor speed (C02280) 1290
Moment of inertia (C02256) 1286
Motor cable cross-section (C02276) 1289
Motor cable length (C02275) 1289
Motor control (C02210) 1272
Motor cosine phi (C02240) 1282
Motor magnetising current (C02242) 1283
Motor magnetising inductance (C02241) 1283
Motor phase direction of rotation (C02272) 1288
Motor rotor resistance (C02232) 1280
Motor stator leakage inductance (C02234) 1281
Motor stator resistance (C02233) 1280
Oscillation damping filter time (C02250) 1285
Oscillation damping influence (C02249) 1284
Override point of field weakening (C02230) 1279
PLI without motion
  Adaptation of ident angle (C02313) 1301
  Adaptation of time duration (C02311) 1300
PLI without motion (C02312) 1301
PSM
  Activate Lss sat. char. (C02307) 1300
  Imax Lss saturation characteristic (C02306) 1300
  Lss saturation characteristic (C02305) 1298
  Maximum motor current field weakening (C02278) 1290
Rated motor current (C02237) 1281
Rated motor frequency (C02238) 1282
Rated motor power (C02231) 1280
Rated motor speed (C02236) 1281
Rated motor voltage (C02239) 1282
Reset time Imax/M controller (C02224) 1277
SC

Field feedforward control (C02261) 1286
Settings (C02229) 1279
Tdn speed controller (C02222) 1276
Tn field controller (C02228) 1278
Tn field weakening controller (C02263) 1287
Vp field controller (C02227) 1278
Vp field weakening controller (C02262) 1287
Setting of motor overload (I²xt) (C02246) 1284
Slip comp. (C02216) 1274
SLPSM
Controlled current setpoint (C02300) 1296
Filter cutoff frequency (C02302) 1297
Filter time rotor position (C02303) 1297
PLL gain (C02304) 1298
Switching speed (C02301) 1297
SLVC
Cross current controller gain (C02292) 1294
Field current controller gain (C02291) 1293
Speed limitation (C02273) 1288
Speed sensor selection (C02260) 1286
Switching frequency (C02214) 1273
Ti current controller (C02226) 1278
Ti speed controller (C02221) 1276
Ultimate motor current (C02279) 1290
VFC
Limitation V/f + sensor (C02284) 1291
Ti V/f +sensor (C02286) 1292
Time const. slip comp. (C02281) 1291
V/f base frequency (C02212) 1272
Vmin boost (C02213) 1273
Vp V/f +sensor (C02285) 1291
VFC-ECO
Minimum voltage V/f (C02289) 1293
Ti CosPhi controller (C02288) 1292
Voltage reduction ramp (C02290) 1293
Vp CosPhi controller (C02287) 1292
Vp current controller (C02225) 1277
Vp speed controller (C02220) 1275
WriteValue_1 (C01086) 1183
WriteValue_2 (C01087) 1183
WriteValue_3 (C01088) 1183
WriteValue_4 (C01089) 1184
LU: DC bus undervoltage (error message) 730


## M


Machine parameters 593
Main program runtime (C00321) 972
Mains connection (behaviour) 127
Mains phase failure monitoring 310
Mains voltage (C00173) 954
Manual control 61
Manual DC-injection braking (DCB) 275
Manual entry of motor type (C01001) 1163
Manual jog 638
Setting (C01230) 1212
Manual jog to software limit position 644
Manual motor direction of rotation check 61
Master functionality (CAN) 786
MasterPin 53
Max. motor speed (C00965) 1154
Max. number of AutoFailReset processes (C00186) 956
Maximum current monitoring 311
Maximum torque (C00057) 921
Maximum torque monitoring 311
Maximum travel distance 608
MCI input words (C00876) 1145
MCI output words (C00877) 1146
MCI_InOut
Inversion (C00890) 1146
MCK 560
Accel./decel. times (C02610) 1312
Control word (C01240) 1215
Current operating mode (C01243) 1219
Current pos profile number (C01242) 1219
Current positions (C01210) 1204
Cycle (C01201) 1202
Diagnostic word (C01247) 1220
Feed constant (C01204) 1203
Follower accelerations (C01237) 1215
Follower speeds (C01236) 1214
Follower S-ramp times (C01238) 1215
Following error (C01215) 1205
iG motor/position encoder (C01203) 1203
iM motor/process (C01202) 1203
Limitations (C02611) 1312
Manual jog accelerations (C01232) 1213
Manual jog breakpoints (C01234) 1214
Manual jog speeds (C01231) 1213
Manual jog S-ramp time (C01233) 1213
Manual jog waiting times (C01235) 1214
Max. travel distance 608
Max. traversing distance (C01213) 1205
Mounting direction (C01206) 1204
Operating mode change at profile no. (C01298) 1227
Position follower setting (C01218) 1207
Position limiting values (C01229) 1212
Position resolution (C01205) 1203
Positioning setting (C01216) 1206
Ref speeds (C01224) 1210
Ref. accelerations (C01225) 1210
Ref. M limit mode 14/15 (C01222) 1209
Ref. mode (C01221) 1209
Ref. positions (C01227) 1211
Ref. sequence profile (C01228) 1211
Ref. setting (C01220) 1208
Ref. S-ramp time (C01226) 1211
Ref. step (C01248) 1222
Ref. waiting time mode 14/15 (C01223) 1210
Resp. to MCK error (C00595) 1042
Select signal source (C01246) 1220
Speed (C01211) 1205
Speed follower setting (C01219) 1207
Status word (C01241) 1217
Stop accelerations (C01251) 1223

Stop S-ramp times (C01252) 1223
Target detection - positions (C01245) 1220
Target detection - times (C01244) 1219
MCK control word 569
MCK diagnostic word 576
MCK interface 581
MCK state machine 577
MCK status bit "bReadyToOperate" 575
MCK status bit "HomPosAvailable" 573
MCK status bit "S_ShapingActive" 574
MCK status word 571
MCKI
Status MCKInterface (C01299) 1228
MCTRL
Actual speed value (C00051) 920
Optimisations (C02864) 1322
Special settings (C02865) 1324
Special settings (C02866) 1325
Speed setpoint (C00050) 919
Status (C01000) 1163
Memory module 43
Binding ID 51
Min/max speed 600
Mode
Position calculation (C01294) 1224
Position calculation (C01296) 1225
Modulo measuring system 597
Moment of inertia from load (C00919) 1149
Moment of inertia motor (C00273) 968
Monitoring 299, 708
Monitoring of the maximum travel distance 608
Motion Control Kernel (MCK) 560
Motor cable cross-section (C00916) 1148
Motor cable length (C00915) 1148
Motor cable resistance (C00917) 1148
Motor catalogue 143
Motor control 138
87-Hz operation 169
DC-injection braking 274
Energy-saving V/f characteristic control (VFCplusEco) 180
Flying restart function 271
Oscillation damping 279
Selection help 157
Selection of switching frequency 263
Selection of the control type 153
Sensorless vector control (SLVC) 197
Servo control (SC) 231
Slip compensation 278
V/f characteristic control (VFCplus) 161
V/f control (VFCplus + encoder) 190
Motor control (C00006) 906
Motor cosine phi (C00091) 929
Motor current (C00054) 920
Motor data 139
Motor data changeover 882
Motor data parameter list 882
Motor holding brake 673
Motor load monitoring (I2xt) 301
Motor magnetising current (C00095) 929
Motor magnetising inductance (C00092) 929
Motor overcurrent monitoring 303
Motor parameter identification 145
Motor parameter identification is active 119
Motor phase direction of rotation (C00905) 1147
Motor phase error monitoring before operation 308
Motor phase failure monitoring 307
Motor phase failure threshold (C00599) 1043
Motor rotor position (C00060) 922
Motor rotor position (C00927) 1150
Motor rotor resistance (C00082) 927
Motor rotor time constant (C00083) 927
Motor selection 139
Motor speed monitoring 311
Motor stator leakage inductance (C00085) 927
Motor stator resistance (C00084) 927
Motor temp. feedback system (C01193) 1202
Motor temperature (C00063) 922
Motor temperature monitoring (PTC) 304
Motor thermal sensor (C01190) 1201
Motor voltage (C00052) 920
Motorparameter (C00969) 1156
Multilingualism 41


## N


Network management telegram (NMT) 785
NMT (network management) 785
Node address 781
Node ID 781
Number of CAN SDO channels (C00366) 985
Number of encoder increments (C00420) 993


## O


oC1: Power section - short circuit (error message) 730
oC10: Maximum current reached (error message) 732
oC11: Clamp operation active (error message) 734
oC12: I2xt overload - brake resistor (error message) 734
oC13: Maximum current for Fch exceeded (error message) 735
oC14: Direct-axis current controller limitation (error message) 736
oC15: Cross current controller limitation (error message) 736
oC16: Torque controller limitation (error message) 736
oC17: Clamp sets pulse inhibit (error message) 732
oC18: Current monitoring overload (error message) 732
oC2: Power section - earth fault (error message) 730
oC5: Ixt overload (error message) 729
oC6: I2xt overload - motor (error message) 737
oC7 303
oC7: Motor overcurrent (error message) 729
oH1: Heatsink overtemperature (error message) 727

oH12: Motor overtemperature MultiEncoder (error message) 728
oH3: Motor temperature (X106) triggered (error message) 728
oH4: Heatsink temp. > shutdown temp. -5°C (error message) 727
oH6: Motor temperature MultiEncoder >= C121 (error message) 728
oH7: Motor temperature resolver >= C121 (error message) 727
oH9: Motor overtemperature resolver (error message) 727
Open-circuit monitoring
  HTL encoder 378
Open-circuit monitoring (C00498) 1023
Operating conditions (CAN on board) 773
Operating mode
  Homing 618
  Manual jog 638
  Position follower 666
  Positioning 646
  Speed follower 616
  StandBy 578
  Stop 665
Operating mode change with profile number 589
Operation with increased rated power 267
Operation with safety module 579
Optical tracking 115
Optimising the response to setpoint changes (SC) 248
Optimising the response to setpoint changes (SLVC) 207
Option "Lock bFail at TroubleQSP" 133
Original application|control source (C00008) 910
oS1: Maximum speed limit reached (error message) 732
oS2: Max. motor speed (error message) 732
Oscillation damping 279
Oscillation damping filter time (C00235) 964
Oscillation damping influence (C00234) 964
Oscilloscope function 754
ot1: Maximum torque reached (error message) 729
ot2: Speed controller output limited (error message) 735
oU: DC bus overvoltage (error message) 729
Output frequency (C00058) 921
Output power (C00980) 1158
Overchange 653
Override 669
Override point of field weakening (C00080) 926
OxI
  Current (C00437) 1000
OxU
  Voltage (C00436) 1000
OxU/I
  Gain (C00434) 999
  Input value (C00439) 1001
  Offset (C00435) 1000

## P

Parameter change-over 882
Parameterisable function blocks 1376
Password data (C00505) 1024
Password protection 47
PC manual control 93
PDO mapping 789, 791, 793, 795
PDO synchronisation 805
Peak current limitation 158
Phase controller 287
Phase sequence reversal 282
Plant parameters 144
PLI current amplitude (C00646) 1064
PLI max. permissible deflection (C00645) 1064
PLI ramp time (C00647) 1064
PLI traversing direction (C00644) 1064
PLI without motion
  Adaptation of ident angle (C02875) 1330
  Adaptation of time duration (C02872) 1329
  Ident. el. rotor displ. angle (C02873) 1329
  Optimisation factor (C02870) 1328
  Running time (C02871) 1328
PLI without motion (C02874) 1330
Pole position (C00926) 1150
Pole position identification (PPI) 366
Pole position identification without motion 372
Port block "LP_CanIn1" 789
Port block "LP_CanIn2" 791
Port block "LP_CanIn3" 793
Port block "LP_CanIn4" 795
Port block "LP_CanOut1" 797
Port block "LP_CanOut2" 798
Port block "LP_CanOut3" 799
Port block "LP_CanOut4" 800
PosExecute (control bit) 588
PosExecute control bit 588
position control 287
Position controller
  Limitation (C02556) 1307
Position encoder selection (C00490) 1020
Position follower 666
Position teaching 664
Positioning 646
Positioning modes 657
Power section ID (C00093) 929
Power-on time meter (C00179) 955
Pre-defined error field (I-1003) 829
Printing the interconnection 1419
Process data objects, identifiers 801
Processing time 775
Producer heartbeat time (I-1017) 833
Product type code (C00203) 960
Profile data
  Acceleration (C01303) 1230
  Acceleration (C01322) 1240
  Deceleration (C01304) 1230
  Deceleration (C01323) 1240

Final speed (C01305) 1231
Final speed (C01324) 1241
PI accel. (C01312) 1235
PI decel. (C01313) 1236
PI final speed (C01314) 1237
PI position (C01310) 1233
PI speed (C01311) 1234
PI S-ramp time (C01315) 1238
Position (C01301) 1229
Position (C01320) 1239
Positioning mode (C01300) 1228
Sequence profile (C01307) 1231
Speed (C01302) 1230
Speed (C01321) 1239
S-ramp time (C01306) 1231
S-ramp time (C01325) 1241
TP profile (C01308) 1232
TP signal source (C01309) 1232
Profile generator (L_DFRFG) 1533
Profile Input 651
Profile linkage with velocity changeover (overchange) 653
PS01: No memory module (error message) 742
PS02: Par. set invalid (error message) 743
PS03: Par. set device invalid (error message) 743
PS04: Par. set Mci invalid (error message) 743
PS07: Par. memory module invalid (error message) 744
PS08: Par. device invalid (error message) 744
PS09: Par. format invalid (error message) 744
PS10: Memory module binding invalid (error message) 744
PS11: Lenze setting loaded (error message) 744
PS12: Parameter sets loaded (error message) 745
PS13: Parameter sets saved (error message) 745
PSM
Activate Lss sat. char. (C02859) 1321
Imax Lss saturation characteristic (C02855) 1321
Lss saturation characteristic (C02853) 1321
Max. motor temperature (C02876) 1331
Maximum motor current field weakening (C00938) 1150
PSM temperature coefficient (C02877) 1331
PTC 304
PTC characteristic
Resistance 1/2 (C01192) 1202
Temperature 1/2 (C01191) 1202
Pulse form TTL-Encoder (C00424) 995
PW protection config MCI (C00510) 1029
PW protection config X1(CAN) (C00509) 1028
PW protection config X6(DIAG) (C00508) 1027
PW protection internal config (C00506) 1025


## Q


Quick stop setting (C00104) 932


## R


Ramp function generator (L_DFRFG) 1533
Rated device current (C00098) 930
Rated device currents (C00920) 1149
Rated heat amount of brake resist. (C00131) 937
Rated motor current (C00088) 928
Rated motor frequency (C00089) 928
Rated motor power (C00081) 927
Rated motor speed (C00087) 928
Rated motor torque (C00097) 929
Rated motor voltage (C00090) 928
Rated power - brake resistor (C00130) 937
Reduc. brake chopper threshold (C00174) 954
Reference setting 618
Referencing mode 622
Referential project 1424
Remote
Acceleration/deceleration time (C00461) 1009
Control (C00462) 1009
MCK control (C00463) 1010
Monitoring timeout (C00464) 1011
Setpoint selection (C00729) 1084
Reset error message 719
Reset node (CAN) 779
Reset reference information 573
Reset time Imax/M controller (C00074) 925
Resolver
Phase error (C02863) 1322
Resolver gain (C02862) 1322
Resp. to axis bus error (C00591) 1039
Resp. to brake resist. overtemp. (C00574) 1035
Resp. to CAN bus connection (C00592) 1040
Resp. to CANx_IN monitoring (C00593) 1040
Resp. to communication error with MCI (C01501) 1267
Resp. to control word error (C00594) 1041
Resp. to controller limitations (C00570) 1034
Resp. to current monitoring (C00584) 1038
Resp. to DC-bus voltage (C00600) 1044
Resp. to device overload (Ixt) (C00604) 1045
Resp. to earth fault (C00602) 1044
Resp. to fan failure (C00566) 1033
Resp. to feedback (C00603) 1045
Resp. to feedback (C00605) 1046
Resp. to heatsink temp. > shutdown temp. -5°C (C00582) 1037
Resp. to LS_SetError_x (C00581) 1036
Resp. to mains phase failure (C00565) 1032
Resp. to max. freq. feedb. DIG12/67 (C00607) 1046
Resp. to max. speed at switching freq. (C00588) 1038
Resp. to max. speed/output freq. reached (C00579) 1036
Resp. to maximum current (C00609) 1047
Resp. to maximum torque (C00608) 1046
Resp. to motor overload (I²xt) (C00606) 1046
Resp. to motor overtemp. PTC (C00585) 1038
Resp. to motor phase failure (C00597) 1043
Resp. to motor temperature KTY (C00583) 1037
Resp. to open circuit AINx (C00598) 1043
Resp. to open circuit HTL encoder (C00586) 1038

Resp. to operating system error (C00580) 1036
Resp. to peak current (C00569) 1033
Resp. to PLI monitoring (C00643) 1063
Resp. to speed controller limited (C00567) 1033
Resp. to switching frequency reduction (C00590) 1039
Resp. to too frequent AutoFailReset (C00189) 957
Resp. to wrong controller setting (C00571) 1034
Retracting from limit switches 645
Reversing the phase sequence 282
Rotary table application 597
Rotor displacement angle detection after controller enable 372
RPDO1 communication parameter (I-1400) 837
RPDO1 mapping parameter (I-1600) 841
RPDO2 communication parameter (I-1401) 838
RPDO2 mapping parameter (I-1601) 842
RPDO3 communication parameter (I-1402) 839
RPDO3 mapping parameter (I-1602) 842
RPDO4 communication parameter (I-1403) 840
RPDO4 mapping parameter (I-1603) 843
Running time (C00180) 955

## S

S_ShapingActive (status bit) 574
Safe stop 1 579
Safety functions 579
Safety instructions 30
Safety module 579
Saturation characteristic 269
SC
  Depth of current setpoint filter (C00272) 968
  Filter time const. DC detection (C00280) 969
  Freq. current setpoint filter (C00270) 967
  Max. acceleration change (C00274) 968
  max. output voltage (C00276) 969
  Optimisation of field feedforward control (C00576) 1035
  Settings (C00079) 926
  Start motor magnetising current (C00918) 1148
  Tdn speed controller (C00072) 924
  Tn field controller (C00078) 926
  Tn field weakening controller (C00578) 1035
  Vp field controller (C00077) 925
  Vp field weakening controller (C00577) 1035
  Width of current setpoint filter (C00271) 968
Scaling of physical units 1379
Sd10: Speed limit for feedback system 12 (error message) 737
Sd11: Speed limit for feedback system 67 (error message) 737
Sd12: Error thermal detector MultiEncoder (error message) 728
Sd13: Inaccuracy SinCos (error message) 738
Sd14: Position invalid Hiperface (error message) 731
Sd15: Open circuit HTL 4-fold (error message) 738
Sd16: Hiperface-SinCos deviation (error message) 738
Sd2: Open circuit resolver (error message) 731
Sd3: Open circuit HTL 2-fold or 4-fold (error message) 738
Sd4: Open circuit MultiEncoder (error message) 731
Sd6: Error thermal detector resolver (error message) 727
Sd7: Encoder communication (error message) 731
Sd8: Encoder angular drift monit. (error message) 734
Sd9: Status message Hiperface (error message) 731
SDO1 server parameter (I-1200) 834
SDO2 server parameter (I-1201) 835
Selection help for motor control 157
Selection of switching frequency 263
Selection of the control type 153
Sensibility - Setpoint feedforward control (C00653) 1065
Sensorless control for synchronous machines (SLPSM) 155
Sensorless vector control (SLVC) 155, 197
Sequence control 1690
Serial number (C00204) 961
Servo control (SC) 155, 231
Setpoint holding (C01239) 1215
Setpoint synchronisation ramp 600
Setting of compatibility (C02868) 1327
Setting of motor overload (I²xt) (C00120) 935
Setting the baud rate 777
Setting the error response 710
Setting the node address 777
Settings of measuring system (C02652) 1313
Short overview of error messages 721
Signal flow
  Energy-saving V/f characteristic control (VFCplusEco) 181, 215
  Sensorless vector control (SLVC) 199
  Servo control (SC) for ASM 236
  Servo control (SC) for PSM 233
  V/f characteristic control (VFCplus) 162, 163
  V/f control (VFCplus + encoder) 191, 192, 198
Signal type entry 1379
Slip calculation from equivalent circuit diagram (C02879) 1333
Slip comp. (C00021) 915
Slip compensation 278
Slip regulator 194
SLPSM
  Controlled current setpoint (C00995) 1161
  Filter cutoff frequency (C00997) 1161
  Filter time rotor position (C00998) 1162
  PLL gain (C00999) 1162
  Switching speed (C00996) 1161
SLVC
  Gain of cross current controller (C00986) 1159
  Gain of field current controller (C00985) 1159
Software limit positions 601
Source of diff. setpoint feedforward control (C00654) 1066
Speed control with torque limitation (SC) 239
Speed control with torque limitation (SLVC) 200
Speed feedback 156
Speed follower 616
Speed limitation (C00909) 1147
Speed override 670
Speed sensor selection (C00495) 1021
Speed setpoint generation 600

S-ramp smoothing override 672
S-ramp time for jerk limitation 659
SS1 579
SSI encoder
  Bit rate (C00427) 996
  Coding (C00428) 997
  Data bits (C00426) 996
SSI laser position encoder 349
StandBy (MCK operating mode) 578
State machine (MCK) 577
Status "Drive in target" 606
Status "Reference known" 573
Status determining error (16-bit) (C00160) 946
Status determining error (C00168) 952
Status displays 692
Status of brake output BD (C00117) 935
Status of last device command (C00003) 905
Status word (C00150) 943
Status word (MCK) 571
Stop 665
Stop of the ramp function generator 291
Stop the ramp function generator 291
Stuff-bit error 820, 868
Su02: One mains phase is missing (error message) 725
Su03: Too frequent mains switching (error message) 726
Su04: CU insufficiently supplied (error message) 726
Su06: Mains input overload (error message) 726
Su07: 24V supply off (error message) 726
Switching cycles (C00177) 955
Switching freq. reduct. (Temp.) (C00144) 941
Switching frequency 263
Switching frequency (C00018) 914
Switching status of the fans (C00560) 1031
Switch-on inhibit 127
Sync correction width (C01124) 1192
Sync cycle time setpoint (C01121) 1191
Sync phase position (C01122) 1192
Sync signal source (C01120) 1191
Sync telegram 805
Sync window (C01123) 1192
Synchronisation mode 1550
Synchronous motor
  Field weakening 283
SyncTxRxTimes (C00370) 987
System blocks 1742
System bus 772
System connection list
  16-bit (C00620) 1048
  Angle (C00622) 1059
  Bool (C00621) 1052
System error messages 715
System runtimes (C00312) 971

## T

Target group 25
Target position monitoring 606
Task selection 1383
Teaching of the position 664
Technology applications 33
Temperature inside the controller (C00062) 922
Terminals 384
  Analog terminals 409
  Digital input terminals 385
  Digital output terminals 406
  For electrical data see the hardware manual
  User-defined terminal assignment 429
Thermal motor load (I²xt) (C00066) 923
Ti current controller (C00076) 925
Ti speed controller (C00071) 924
Time settings (C00181) 956
Torque (C00056) 921
Torque control with speed limitation (SC) 241
Torque control with speed limitation (SLVC) 201
Torque feedforward control (SC) 248
Torque feedforward control (SLVC) 207
Torque limitation 174
Touch probe detection 419
TP
  Edge selection (C02810) 1314
  Pos. window end (C02814) 1316
  Pos. window start (C02813) 1315
  Position offset (C02812) 1315
  Position source (C02815) 1316
  Sensor delay (C02811) 1314
  Signal counter (C02816) 1317
  TouchProbe position (C02817) 1317
TP (touch probe) 419
TPDO1 communication parameter (I-1800) 844
TPDO1 mapping parameter (I-1A00) 849
TPDO2 communication parameter (I-1801) 846
TPDO2 mapping parameter (I-1A01) 849
TPDO3 communication parameter (I-1802) 847
TPDO3 mapping parameter (I-1A02) 850
TPDO4 communication parameter (I-1803) 848
Transmission mode CAN Rx PDOs (C00323) 973
Transmission mode CAN TxPDOs (C00322) 972
Transmission type 803
Travel range monitoring 604

## U

Ultimate motor current 303
Ultimate motor current (C00939) 1151
USB diagnostic adapter 37
User data 809
User error 1 (error message) 724
User error 2 (error message) 724
User error 3 (error message) 724

User error 4 (error message) 724
User error 5 (error message) 725
User error 6 (error message) 725
User error 7 (error message) 725
User error 8 (error message) 725
User level 40
User menu (C00517) 45, 1030
User-definable V/f characteristic 176

## V

V/f base frequency 168
V/f characteristic control (VFCplus) 154, 161
V/f control (VFCplus + encoder) 190
V/f emergency operation in case of encoder open circuit 190, 231, 320
Validity of the reference measuring system 573
VFC
  Frequency interpol. point n (C00967) 1155
  Limitation V/f + sensor (C00971) 1156
  Ti V/f +sensor (C00973) 1156
  Time const. slip comp. (C00966) 1154
  V/f base frequency (C00015) 913
  Vmin boost (C00016) 913
  Voltage interpol. point n (C00968) 1155
  Vp V/f +sensor (C00972) 1156
VFC-ECO
  Minimum voltage V/f (C00977) 1157
  Ti CosPhi controller (C00976) 1157
  Voltage reduction (C00978) 1157
  Voltage reduction ramp (C00982) 1158
  Vp CosPhi controller (C00975) 1157
Vmin boost 170
Vp current controller (C00075) 925
Vp speed controller (C00070) 924

## W

Warning threshold motor temperature (C00121) 936

FEEDBACK

## Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

If you have suggestions for improvement, please e-mail us to:

feedback-docu@Lenze.de

Thank you for your support.

*Your Lenze documentation team*

CE

Lenze Drives GmbH
Breslauer Straße 3
D-32699 Extertal
Germany
+49 5154 82-0
lenze@lenze.com
www.lenze.com

**Service**
Lenze Service GmbH
Breslauer Straße 3
D-32699 Extertal
Germany
008000 24 46877 (24 h helpline)
+49 5154 82-1112
service@lenze.com

Lenze