COMBICOM

KEB Parametrierkanal-Treiber für Siemens S7
KEB Parameterizing Channel-Driver for Siemens S7

0S.58.00A-K000

KEB

10/2002

D **Seite D3 ..... D21**

GB **Page GB3 ..... GB21**

Table of Contents

1. GENERAL 4

2. REQUIRED DOCUMENTS, BIBLIOGRAPHY, CONTENTS OF THE DISK 4

3. INTRODUCTION 5

3.1 WHAT IS THE PARAMETERIZING CHANNEL 5

4. BASIC MODE OF OPERATION OF PARAMETERIZING CHANNEL DRIVER 6

5. PREPARATIONS FOR THE USE OF THE DRIVER 7

5.1 SETUP OF TEST ENVIRONMENTS 7
5.2 INSTALLATION OF THE DRIVER 8

6. THE USER INTERFACE 11
6.1 COMPATIBILITY OF APPLICATION PROGRAM AND PARAMETERIZING CHANNEL DRIVER 14
6.1.1 Configuration of Driver-Software 14
6.1.2 Cyclic Operation of Application-Software 14
6.2 SETUP OF THE SLAVE-DB 15
6.3 SETUP OF THE SLAVE-LISTS-DB 17
6.4 SETUP OF THE INIT-DB 18
6.5 TIMEOUT-MONITORING 19
6.6 PROFIBUS-DIAGNOSIS 19

7. SIDE NOTE 20
7.1 CONFIGURATION OF KEB-SLAVES 20
7.2 ACCESS TO CONSISTENT PERIPHERAL AREAS WITH SIMATIC S7 20

8. RESERVED PROGRAM RESOURCES OF THE DRIVER 21

1. General

The documentation as well as the hardware and software are developments of the Karl E. Brinkmann GmbH. Errors and ommissions excepted! The Karl E. Brinkmann GmbH has prepared the documentation, hardware and software to the best of their knowledge, however, no guarantee is given that the specifications will produce the benefits aimed at by the user. The Karl E. Brinkmann GmbH reserves the right to change the specifications without previous notice or information to any third parties.

2. Required Documents, Bibliography, Contents of the Disk

To carry out the test you require following documents:

[1]: This user description.
[2]: Instruction Manual for the employed KEB PROFIBUS Interface.
[3]: Application Instruction for the employed KEB frequency inverters/servos.

The contents of the driver-SW-disk:

S7_KEB2N.ARJ Archive of the example-project for Simatic-Manager including:
- **-** System data (Hardware configuration)
- **-** all modules (FCs, DBs, OBs)
- Symbol table
- List of variables VAT1 for variable control/observation
- This user description

KEB_DP1.GSD GSD-file for **„KEB-Gateway DP“**:
Is to be used for KEB PROFIBUS-DP-Gateway as well as for F4/S4-PROFIBUS-DP-Operator.

3. Introduction

KEB frequency inverters and servos can be fitted optionally with PROFIBUS-DP-interface. The PROFIBUS-DP is a process-controlled field bus for decentralized periphery. The PROFIBUS-DP is standardized in the european norm EN50170.
In the German-speaking area the Siemens Simatic PLC is very widespread in the automation technology. The changeover from the Simatic S5-series to the Simatic S7-series is becoming more and more perceptible. To support a large user group in the realization of the automation task, KEB has developed a S7-Driver Software. It concerns a software package which operates the so-called parameterizing channel of a KEB-PROFIBUS-DP-unit. The parameterizing channel descrbes the functionality of every KED-DP-slave, by which the user has access to any chosen parameter of a KEB-unit. Alternatively the user has read and write access to some previously defined parameters by way of the so-called process data. The access to the process data is very easy and does not require a SW-driver. For that reason the software described in the following is limited exclusively to the parameterizing channel.
To keep the extent of the description as short as possible, no detailed explanations to the sequence of the parameterizing channel protocol are listed. It is the aim of this driver to relieve the user from the necessity of such detailed knowledge. But of course all details concerning the parameterizing channel as well as all other specifications of a KEB-PROFIBUS-DP-unit can be looked up in the Instruction Manual for the employed KEB PROFIBUS-DPInterface [2].

3.1 What is the Parameterizing Channel

As already mentioned above the parameterizing channel is a part of the functionality of the KEB-PROFIBUS-DP-unit. With PROFIBUS-DP cyclic user data are exchanged between a master and the slave assigned to it after a short initialization phase. **At KEB-slaves** a part of this user data, that are always the first 8 bytes, contain the parameterizing channel. By way of these 8 bytes the master can query (read) or change (write) the value of a parameter to be addressed. At that the handling of a parameterzing order is done over at least two data cycles between master and slave. Furthermore, it is to note, that every parameterzing service is acknowledged. That means, the master requests the reading or writing of a specific parameter from the slave and receives an acknowledgement. Thus it is possible to recognize whether the slave has really carried out the service.

Completely written is a parameterizing channel service through its parameters described in the following:

Index The index presets the parameter addressing together with the subindex.

Subindex The subindex presets the parameter addressing together with the index.

Dlen The data length of the addressed parameter in byte.

Data The value of the addressed parameter.

Service-Code Determines whether a write or read service shall be carried out.

Error-Code Indicates whether an error occurred during the execution of a service.

By limiting the parameterizing channel to 8 bytes only parameters with a data length of maximal 4 bytes can be changed or read. This means no restriction at all, since all parameters of KEB-units of the F4/S4-series and earlier generations have a data width of 2 bytes. All configuration parameters of the PROFIBUS-DP-connection as well as the DRIVECOM-Profile-Parameters are still usable with this restriction.

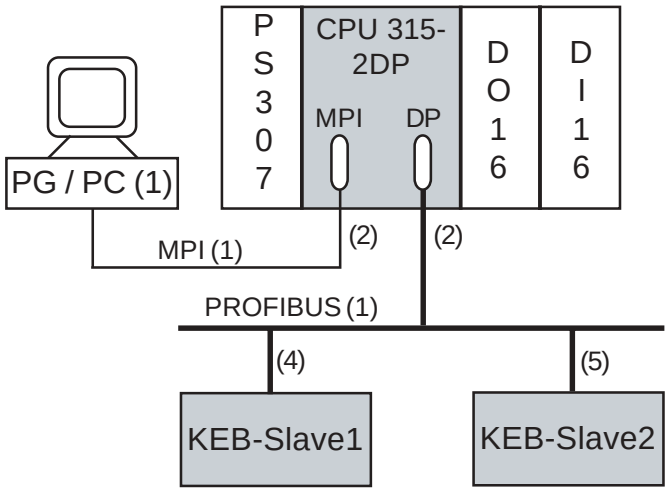
4. Basic Mode of Operation of Parameterizing Channel Driver

The S7-driver realized here, consists in principle of two parts. The first part represents the actual parameterizing channel driver. It essentially consists of a function to be called cyclically (FC2,DO_PARA) and of a data block for each KEB-PROFIBUS-slave, over which a service is executed.

On this basis the described SW provides another miminal application function. It consists of an administration data block (SL_LIST) in which a slave-list is entered. This list contains an entry "Slave-DB-No" for every KEB-slave and and entry for an "Initialization-DB-No". The second part of the driver ensures, that all slaves entered in the list are processed cyclically. After the PLC-restart the jobs, that the Init-DB contains for each slave that has an Init-DB-No entered (Init DBNR !=255), are written to this slave one-time.

During cyclic operation, i.e. when all slave are initialized, the application program can make entries for every slave with jobs in its Slave-DB. This is then processed by cyclical processing and the confirmation is entered in the Slave-DB as soon as it is available. The parameterizing channel driver also incorporates a Timeout-monitoring. This means that a parameterizing channel job from the driver is terminated with an error code, if no acknowledgment is received from the slave during the Timeout-time (see below).

5. Preparation for the Use of the Driver



5.1 Setup of Test Environments

To carry out the test of the driver the above shown test setup is necessary. Generally two possibilities for the PROFIBUS-coupling of KEB-units is possible. The following gives a short description of the necessary configuration for both:

PROFIBUS-coupling via PROFIBUS-DP-Operator (see [2]):

In this case a KEB-slave consists of one KEB-frequency inverter/servo plus snapped-on PROFIBUS-DP-Operator.

- Parameterizing channel always activated.
- Adjustment of PROFIBUS-address (Ts) by presetting of inverter address (ud.06), Ts = ud.06 .
- No adjustment of PROFIBUS-baud rate necessary as the standard operation includes automatic bit rate acknowledgment.

PROFIBUS-coupling via PROFIBUS-DP-Gateway (see [2]):

In this case a KEB-slave consists of one KEB-frequency inverter/servo plus Interface-Operator plus a PROFIBUS-DB-Gateway connected externally via the serial interface. To guarantee a perfect function the parameters Inverter Address and KEB-DIN66019-baud rate of frequency inverter and gateway must be matched. In the gateway S_addr = 1 and in KEB_DIN66019_Baud = 3(9600 Bits) are adjusted as default. The connected frequency inverter must also have the same values (ud.06=1,ud.07=3). Please also ensure that the connected frequency inverters/servos have the bus password (du.01) = application(440):

- Parameterizing channel activated by the position of the switch S2-1 on ON.
- Adjustment of PROFIBUS-address(Ts) by setting the 8-pole DIL-switch S1 at the gateway.
- Adjustment of PROFIBUS-transfer rate by setting the 4-pole DIL-switch S2 (S2-2...S2-4) at the gateway.

If there is only one KEB-unit with PROFIBUS-DP-connection, then nothing needs to be changed on the hardware-configuration of the driver. However, in this case the red LEB **BUSF** on the CPU 315-2DP flashes cyclically, as not all configured slaves are available. But the test with a KEB-slave can still be carried out.

5.2 Installation of the Driver

The creation of the driver was done with the **Simatic Manager STEP7 S7/ M7 Version 4.02.1**. The development took place on the module level. Thus no sources exist. The driver consists of a complete project which incorporates all necessary data. Before the user can start, the project must be installed in the user-PLC. The project is supplied as archive, which you must first **de-archive**. This is done in the Simatic Manager under menu **File->De-archive**.

Following hardware-configuration is preset in the project:

- 1 Simatic 300-Station, consisting off:
  - 1 Load current supply PS307 5A in slot 1,
  - 1 CPU315-2DP as PROFIBUS-DP-master(Ts=2) on subnet PROFIBUS(1) in slot 2. The CPU is still connected as MPI-user with PG/PC(1), MPI-address = 2
  - 1 Digital output module DO16*DC24V/0.5A in slot 4
  - 1 Digital input module DI16*DC24V in slot 5

- 2 KEB Gateway DP as PROFIBUS-DP-slaves(Ts=4,5) on subnet PROFIBUS(1).

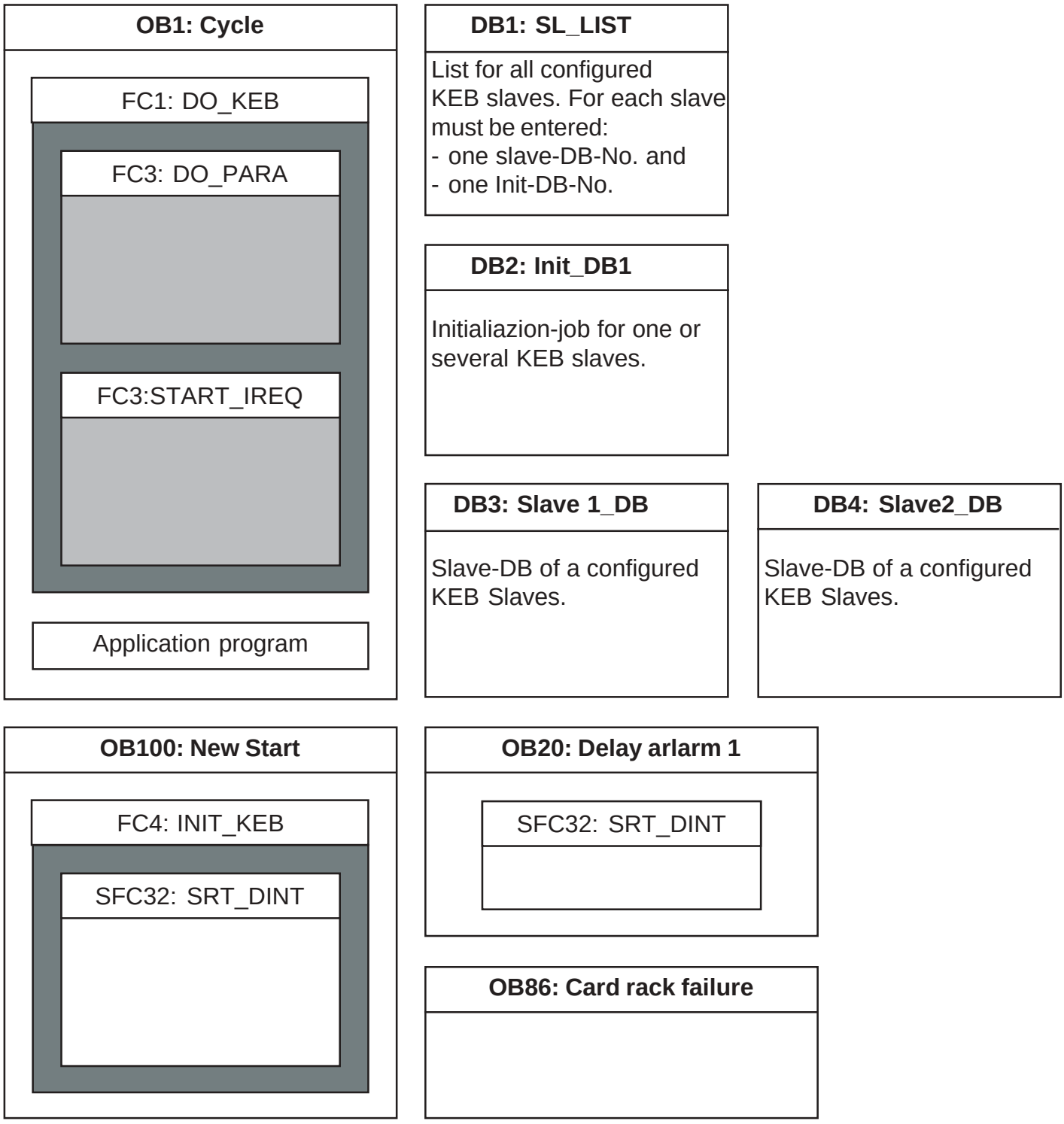
- 1 PG/PC(1) as MPI-user on subnet MPI(1), MPI-address = 0

Perhaps the hardware-configuration needs to be adapted to your requirements. For a first test of the SW it would be of an advantage to use the standard hardware-configuration. The hardware-configuration must then be loaded into the PLC.

Subsequently the S7-program must be loaded in the PLC. For that you load, for example the view of the modules, OBs, FCs and DBs into the PLC. The listed SFCs must already exist in your PLC.

The modules of the drivers are in detail:

OB1, Cycle	Contains the cyclic main program of the complete application. On delivery it comprises only the call of the function FC1 (DO_KEB).
OB100, New Start	Contains after delivery only the call of the function FC4 (INIT_KEB).
OB20, Delay alarm 1	With the delay alarm a cyclic alarm is realized, which is called with a cycle time of 100ms. It is used for the Timeout-detection of the parameterizing channel driver.
FC1, DO_KEB	User part of the driver. It guarantees that at the start all configured KEB-slaves are initialized with an appropriate Init_DB. After the initialization the FC1 ensures that all slaves are processed from the parameterizing channel driver.
FC2,DO_PARA	Is called by the FC1. Completely processes the parameterizing jobs.
FC3, START_IREQ	Is called by the FC1 and starts a new initialization job.
FC4,INIT_KEB	Takes over the complete initialization of the driver (see OB100).
DB1,SL_LIST	Administration block for the driver, besides administration information it contains a slave-list with one entry each of Slave-DB-No and Init-DB-No for every slave.
DB2,INIT1_DB	Contains exemplary a list of initialization jobs, which are written to both KEB-slaves during the initialization phase. At that it concerns jobs which the standard process data reservation writes. Contains exemplary a list of initialization jobs, which are written to both KEB slaves during the initialization phase. It concerns jobs, which the standard process data assignment writes. The initialization data block DB2, that is contained in the project, serves as example. It initializes profile parameters as process data and is thus only suitable for KEB COMBIVERT F4-C units. For the other types (KEB COMBIVERT F4-S, F4-F / S4) the DB2 must be modified accordingly or a new init DB must be written.
DB3,Slave1_DB	Slave-data block of the 1. KEB-Slave.
DB4,Slave2_DB	Slave-data block of the 2. KEB-Slave.


OB1: Cycle
FC1: DO_KEB
FC3: DO_PARA
FC3:START_IREQ
Application program
DB1: SL_LIST
List for all configured KEB slaves. For each slave must be entered:
- one slave-DB-No. and
- one Init-DB-No.
DB2: Init_DB1
Initialiazion-job for one or several KEB slaves.
DB3: Slave 1_DB
Slave-DB of a configured KEB Slaves.
DB4: Slave2_DB
Slave-DB of a configured KEB Slaves.
OB100: New Start
FC4: INIT_KEB
SFC32: SRT_DINT
OB20: Delay arlarm 1
SFC32: SRT_DINT
OB86: Card rack failure

6. The User Interface

In this chapter you learn what you need to do to use the parameterizing channel driver in your application program. The first test should be carried out from the PG/PC by means of **Variable Obersation/Control**. At this point we assume that the complete SW is loaded in the PLC and that the CPU is in status STOP. Likewise we presuppose that the general PROFIBUS-communication is running. This can be recognized by the two red LEDs **SF DP** and **BUSF** on the CPU which do not light up or flash.

In the view under modules double-click on **VAT1** to reach the menu Variable Observation/Control directly. The variables listed there show for one thing the current processing status of the driver (DB1). For another thing the most important variables of the two slave-data blocks (DB3,DB4) are displayed.

Please make sure, that in the menu **Variable->Trigger** the trigger conditions are adjusted as follows:

- Trigger point for observation; cycle begin,
  trigger condition for obersation: permanent.

- Trigger point for control: cycle begin,
  trigger condition for control: one-time.

Activate the variable observation.

If you switch the CPU into **RUN-P**, the CPU will start and carry out the initialization of the slaves. The initialization of the slaves is completed when the variable DB1.DBB0 contains the value 3. By controlling the variable-values of DB3,DB4 you can now send jobs to the 1st or 2nd slave or to both slaves simultaneously. The following example should make the execution clear:

Example In the example you will try to read from both slaves the value of the parameter with the index= 4606h and the subindex = 0. In the KEB-units of the series F4 or S4 the parameter with the name **inverter address** is located at this address. For all other unit types it may be necessary to change the index accordingly in order to read another parameter value. Please refer to the Application Manual of the employed KEB-unit. Please observe, that the **Index** is a result of **the parameter-address + 2000h** Offset. Consequently you must add 2000(hex) to the parameter-address from the Application Manual, to get the corresponding index, which is entered in the Slave-DB.

Example 1 **Reading of the parameter** with index=4606(hex) and subindex = 0: **(ud.06, inverter address):** Make the following entries in the **control value**-column of the variables:

Operand	Symbol	Control Value
DB3.DBB0	„Slave1_DB".State	B#16#**01**
DB3.DBD2	„Slave1_DB".Error	Preset no control value
DB3.DBW8	„Slave1_DB".Index	W#16#**4606**
DB3.DBB10	„Slave1_DB".Subindex	B#16#**00**
DB3.DBB11	„Slave1_DB".Dlen	Preset no control value
DB3.DBD12	„Slave1_DB".Data	Preset no control value
DB4.DBB0	„Slave2_DB".State	B#16#**01**
DB4.DBD2	„Slave2_DB".Error	Preset no control value
DB4.DBW8	„Slave2_DB".Index	W#16#**4606**
DB4.DBB10	„Slave2_DB".Subindex	B#16#**00**
DB4.DBB11	„Slave2_DB".Dlen	Preset no control value
DB4.DBD12	„Slave2_DB".Data	Preset no control value

Send the control values prepared in this manner by **Variable->Control** to the CPU. If the status values of „Slave1_DB".State and „Slave2_DB".State show the value 0, both jobs have been processed. You can then evaluate the errror code in „Slave1_DB".Error and „Slave2_DB".Error. They should contain the value 0. If this is not the case you can read the value in „Slave1_DB".Data and „Slave2_DB".Data. The status values of the variables „Slave1_DB".Dlen und „Slave2_DB".Dlen indicate how many of the data bytes in the data-field are relevant. In our example the status value should contain following values:

Operand	Symbol	Status value
DB3.DBB0	„Slave1_DB".State	B#16#00
DB3.DBD2	„Slave1_DB".Error	DW#16#**00000000**
DB3.DBW8	„Slave1_DB".Index	W#16#4606
DB3.DBB10	„Slave1_DB".Subindex	B#16#00
DB3.DBB11	„Slave1_DB".Dlen	B#16#**02**
DB3.DBD12	„Slave1_DB".Data	DW#16#**0004**XXXX
DB4.DBB0	„Slave2_DB".State	B#16#00
DB4.DBD2	„Slave2_DB".Error	DW#16#**00000000**
DB4.DBW8	„Slave2_DB".Index	W#16#4606
DB4.DBB10	„Slave2_DB".Subindex	B#16#00
DB4.DBB11	„Slave2_DB".Dlen	B#16#**02**
DB4.DBD12	„Slave2_DB".Data	DW#16#**0005**XXXX

Example 2 **Writing of the parameter** with index = 4100 (hex) and subindex = 0: **(op.00,Setpoint source):** Make the following entries in the **control value**-column of the variables:

Operand	Symbol	Control value
DB3.DBB0	„Slave1_DB".State	B#16#**02**
DB3.DBD2	„Slave1_DB".Error	Preset no control value
DB3.DBW8	„Slave1_DB".Index	W#16#**4100**
DB3.DBB10	„Slave1_DB".Subindex	B#16#**00**
DB3.DBB11	„Slave1_DB".Dlen	B#16#**02**
DB3.DBD12	„Slave1_DB".Data	DW#16#**0001**XXXX
DB4.DBB0	„Slave2_DB".State	B#16#**02**
DB4.DBD2	„Slave2_DB".Error	Preset no control value
DB4.DBW8	„Slave2_DB".Index	W#16#**4100**
DB4.DBB10	„Slave2_DB".Subindex	B#16#**00**
DB4.DBB11	„Slave2_DB".Dlen	B#16#**02**
DB4.DBD12	„Slave2_DB".Data	DW#16#**1000**XXXX

Send the control values prepared in this manner by **Variable->Control** to the CPU. If the status values of „Slave1_DB".State and „Slave2_DB".State show the value 0, both jobs have been processed. You can then evaluate the errror code in „Slave1_DB".Error and „Slave2_DB".Error. In this example „Slave1_DB".Error should contain the value 0. But „Slave2_DB".Error should contain the value DW#08000030. This means that slave2 has not processed the write job, because it was attempted to preset an invalid value (see Error-Codes in **[2]**).

Operand	Symbol	Status Value
DB3.DBB0	„Slave1_DB".State	B#16#00
DB3.DBD2	„Slave1_DB".Error	DW#16#**00000000**
DB3.DBW8	„Slave1_DB".Index	W#16#4100
DB3.DBB10	„Slave1_DB".Subindex	B#16#00
DB3.DBB11	„Slave1_DB".Dlen	B#16#02
DB3.DBD12	„Slave1_DB".Data	DW#16#0001XXXX
DB4.DBB0	„Slave2_DB".State	B#16#00
DB4.DBD2	„Slave2_DB".Error	DW#16#**08000030**
DB4.DBW8	„Slave2_DB".Index	W#16#4100
DB4.DBB10	„Slave2_DB".Subindex	B#16#00
DB4.DBB11	„Slave2_DB".Dlen	B#16#02
DB4.DBD12	„Slave2_DB".Data	DW#16#1000XXXX

6.1 Compatibility of Application Program and Parameterizing Channel Driver

To integrate the parameterizing channel driver into your application program the following procedure is to be recommended:

- Configuration of the Driver-Software.
- Starting of the Application Program.
- Wait until „SL_LIST“.State (DB1.DBB0) = 3.
- Cyclic operation of the application software: Processing of the desired parameterizing channel jobs.

6.1.1 Configuration of Driver-Software

First of all carry out the configuration of the driver. For that the Slave-List-DB must be prepared in accordance with the number of KEB-slaves connected to the PROFIBUS. A Slave-DB-No (uneven 255) must be entered for each configured slave. Please note, that the value 255 as Slave-DB-No means the end of the list. All slaves that might come after it would not be processed. Moreover, enter for each slave to be initialized the desired Init-DB-No (uneven 255).

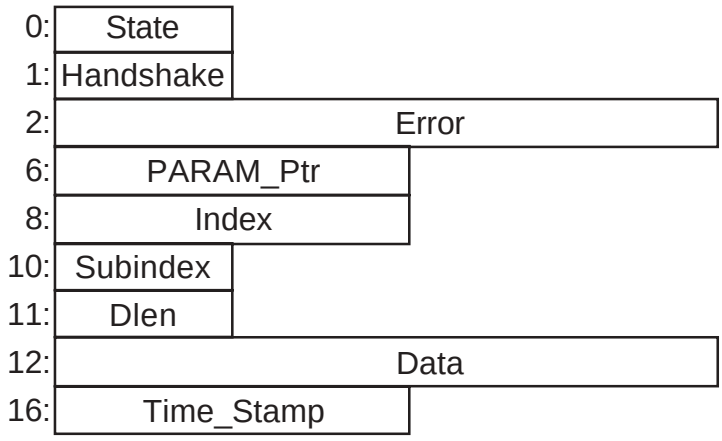
For every configured KEB-slave a Slave-DB must be set up with a length of 18 bytes (see below). In each Slave-DB the parameter **Param_PTR** with the peripheral address is to be reserved where the parameterizing channel of this slave begins. Use the hardware-configuration of the individual slaves as information source. Any intended initialization-data blocks are to be generated according to the standard of DB2 in accordance with your requests (see below).

6.1.2 Cyclic Operation of Application-Software

When the initialization phase of the parameterizing channel driver is completed, you can issue jobs to the individual slaves. Each KEB-slave can process only one job at the same time. But you can issue simultaneously jobs to several or all KEB-slaves. Following step sequence exists for the processing of the parameterizing channel job:

- Check whether the parameter **State** in the relevant Slave-DB is **= 0**. Only then a new job can be issued.
- Enter the complete job in the Slave-DB (State, Index, Subindex, *Dlen (only at writing), Data (only at writing)*), see below.
- While you wait for the completion of the order, make sure, that the cyclic polling of the parameterizing channel driver (DO_KEB) is guaranteed. Only if that is the case the job can be processed. The job is completed when the parameter **State** in the Slave-DB has been reset to value **0**.
- Analyze the result.

6.2 Setup of the Slave-DB



byte State **Processing state:**

0 = Idle (set by the driver after completed processing)

1 = reading requested (set by the applicaiton program)

2 = writing requested (set by the application program)

3 = reading runs (set by the driver)

4 = writing runs (set by the driver)

dword Error Result:

0 = no error occurred

FFFFXXXXh: error message from S7-operating system

with XXXXh = error code of operating system function, see description of the Return Code of SFC14, SFC15

FFFEYYYYh: error message from the driver

FFFE0001h: Invalid Dlen at writing

FFFE0002h: no answer from the slave (Timeout)

KKCCAAAAh: Error-Code from the slave

with KKh: Error-Class

CCh: Error-Code

AAAAh: Additional-Code

word PARAM_Ptr	Parameter-Pointer:	Pointer on the 1. periphery byte of parameterizing channel of this DP-Slave.
byte Handshake	Handshake-Marker:	Updated by the driver.
word Index	Parameter-Index:	Addressing of the parameter (application program).
byte Subindex	Parameter-Subindex:	Addressing of the parameter (application program).
byte Dlen	Data length in byte:	Set at writing by application program, set at reading by driver.
dword Data	Value of the parameter:	Set at writing by application program, set at reading by driver.
word Time_Stamp	Time stamp:	Set by the driver after sending the parameterizing channel-request and is used for the Timeout monitoring.

6.3 Setup of the Slave-Lists-DB

0:	State
1:	Slave_Index
2:	Init_Index
3:	Slave_DBNR**1**
4:	Init_DBNR**1**
5:	Slave_DBNR**2**
6:	Init_DBNR**2**
(N*2)+1:	Slave_DBNR**N**
(N*2)+2:	Init_DBNR**N**
(N*2)+3:	**ENDE = FFh**

byte State Processing state

0 = Start
1 = Starts current Init-job
2 = Init-job runs
3 = Idle (initializition completed)

byte Slave_Index Index of the current processing slave (1, 2, ...).

byte Init_Index Index of the current processing Init-job (1, 2, ...).

byte SLAVE_DBNR1 Data block-No. for slave 1.

byte INIT_DBNR1 Data block-No. of initialization list for Slave 1.

byte SLAVE_DBNR2 Data block-No. for Slave 2.

byte INIT_DBNR2 Data block-No. of initialization list for Slave 2.

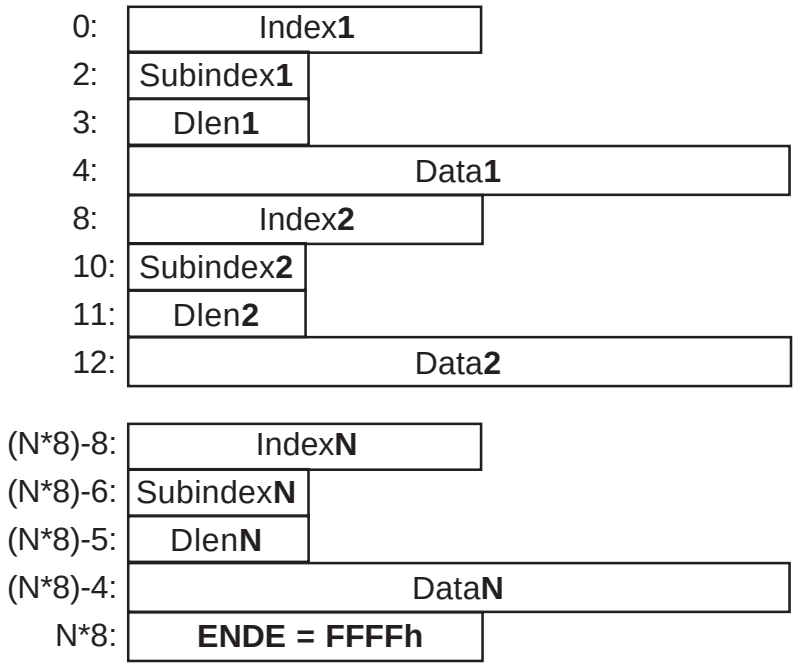
byte SLAVE_DBNRN Data block-No. for Slave N.

byte INIT_DBNRN Data block-No. of initialization list for Slave N.

byte ENDE_KENNUNG FFh

6.4 Setup of the Init-DB

The Init-DB contains a list of write jobs, which was written to one or several slaves during the initialization phase:



word Index1	Write job 1, Index of parameter.
byte Subindex1	Write job 1, Subindex of parameter.
byte Dlen1	Write job 1, Data length in byte.
dword Data1	Write job 1, Data of parameter.
word Index2	Write job 2, Index of parameter.
byte Subindex2	Write job 2, Subindex of parameter.
byte Dlen2	Write job 2, Data length in byte.
dword Data2	Write job 2, Data of parameter.
word IndexN	Write job N, Index of parameter.
byte SubindexN	Write job N, Subindex of parameter.
byte DlenN	Write job N, Data length in byte.
dword DataN	Write job N, Data of parameter.
word END_IDENT	FFFFh

6.5 Timeout-Monitoring

The parameterizing channel driver contains a timeout-monitoring. It monitores the response time for parameterizing channel job. If the response time exceeds a specific value (default approx. 1 s), the driver terminates the job with the corresponding error message.

The timeout-monitoring is realized by a cyclic timer (cycle time = 100ms), that consists of the delay alarm OB20. It is started during the initialization of the driver (FC4,INIT_KEB) by call of SFC32. Then the delay alarm (OB20) starts itself again and again. In the delay alarm a global variable “TTICK” is increased. If a parameterizing channel job was started, the current value of “TTICK” is saved in the Slave-DB under the cell “Time_Stamp”. As long as the job has not been confirmed the difference between “Time-Stamp” and the current “TTICK”-value is used as waiting time. In the available driver-SW the timeout-time is programmed fixed to 10 “TTICK”-cycles (see FC2, network 4 and network 5).

6.6 PROFIBUS-Diagnosis

The S7-application program provides several ways to check whether all of the configured slaves work correctly or if any problems exist. But with the parameterizing channel driver this is not necessary additonally, since during access to the parameterizing channel it is always worked via the system functions SFC14 and SFC15. When a problem arises they return an error-code, thus the failure of a slave is always striking during access to the parameterizing channel.

Nevertheless, an additional alarm-module **OB86 (card rack failure)** has been programmed exemplary in the driver software. If a configured slave failes the PROFIBUS-address of this slave is output on the output byte **AB0**. Furthermore, a counter is incorporated in **AB1**, which counts the number of occurred failures.

At this point we only want to point out the alaram-module **OB82 (diagnosis alarm)** and the system functions **SFC13 (DPNRM_DG)**.

7. Side Notes

7.1 Configuration of KEB Slaves

In the present example project the two configured KEB-slaves are programmed differently (see **SIMATIC 300(1) -> Hardware**). This has been made very consciously so, to introduce both possibilities. Generally the module-definitions from the GSD-file (keb_dpl.gsd) can be adopted for the configuration of a KEB-slave or they can be entered manually over the so-called **universal module**. The presetting via the universal module has the advantage, that subsequent changes in the slave-configuration can be made relatively easy. This may become necessary for example, when you want to preset the KEB-slave/slaves with other processing data reservations which results in a change of the configuration. If at all, the presetting via the GSD-file with the module name **parameterizing channel**, **process-output data** and **process-input data** is quite complicated. That means that if you should intend to change the length of the process data in your application, you should best use the **universal module** for the two process data modules at the configuration of the KEB-slaves. At that it is important that the **unit: „Byte“** and **Consistent over: „entire length“** are preset.

7.2 Access to Consistent Peripheral Areas with Simatic S7

Please note, during access to the periphery of a DP-slave, that with a length of three or more than four bytes the access must be exclusively made via the sytem functions SFC14, SFC15. For all other cases the direct access is to be chosen. That is the reason why in the present driver the access to the parameterizing channel periphery has been realized over the so-called system functions.

8. Reserved Program-Resources of the Driver

- In the marker area the marker-bytes 100 to 125 are used (26 marker bytes). They are not to be changed by the application software.
- For each configured KEB-slave 1 data block (DB) with a length of 18 bytes.
- For each initialization block one data block with a length of (N*8)+2 bytes is needed , at that N represents the number of the initialization jobs.
- 4 functions (FC1, FC2, FC3, FC4). With regard to the length of the functions please refer to the specification to the individual functions in the object characteristics of the Simatic-manager.
- 1 delay alarm-OB (OB20). It can also be used by the application program. From the parameterizing channel driver it is operated as cyclic timer with a cycle time of 100ms and is used for the timeout-recognition.
- 1 card rack alarm-OB (OB86). It is not necessary for the function of the driver and can be deleted or reprogrammed without negative effects.

Karl E. Brinkmann GmbH
Försterweg 36 - 38 • D - 32683 Barntrup
Telefon 00 49 / 52 63 / 4 01 - 0 • Fax 00 49 / 52 63 / 4 01 - 1 16
Internet: www.keb.de • E-mail: info@keb.de

KEB Antriebstechnik GmbH & Co. KG
Wildbacher Str. 5 • **D** - 08289 Schneeberg
Telefon 0049 / 37 72 / 67 - 0 • Telefax 0049 / 37 72 /67 - 2 81
E-mail: info@keb-combidrive.de

KEB Antriebstechnik Austria GmbH
Ritzstraße 8 • **A** - 4614 Marchtrenk
Tel.: 0043 / 7243 / 53586 - 0 • FAX: 0043 / 7243 / 53586 - 21
Kostelni 32/1226 • **CZ** - 370 04 Ceské Budejovice
Tel.: 00420 / 38 / 731 92 23 • FAX: 00420 / 38 / 733 06 97
E-mail: info@keb.at

KEB Antriebstechnik
Herenveld 2 • **B** - 9500 Geraadsbergen
Tel.: 0032 / 5443 / 7860 • FAX: 0032 / 5443 / 7898
E-mail: koen.detaeye@keb.de

KEB China
Xianxia Road 299 • **CHN** - 200051 Shanghai
Tel.: 0086 / 21 / 62350922 • FAX: 0086 / 21 / 62350015
Internet: www.keb-cn.com • E-mail: info@keb-cn.com

Société Française KEB
Z.I. de la Croix St. Nicolas • 14, rue Gustave Eiffel
F - 94510 LA QUEUE EN BRIE
Tél.: 0033 / 1 / 49620101 • FAX: 0033 / 1 / 45767495
E-mail: sfkeb.4@wanadoo.fr

KEB (UK) Ltd.
6 Chieftain Buisiness Park, Morris Close
Park Farm, Wellingborough, **GB** - Northants, NN8 6 XF
Tel.: 0044 / 1933 / 402220 • FAX: 0044 / 1933 / 400724
Internet: www.keb-uk.co.uk • E-mail: info@keb-uk.co.uk

KEB Italia S.r.l.
Via Newton, 2 • **I** - 20019 Settimo Milanese (Milano)
Tel.: 0039 / 02 / 33500782 • FAX: 0039 / 02 / 33500790
Internet: www.keb.it • E-mail: kebitalia@keb.it

KEB - YAMAKYU Ltd.
15 – 16, 2 – Chome, Takanawa Minato-ku
J – Tokyo 108 -0074
Tel.: 0081 / 33 / 445-8515 • FAX: 0081 / 33 / 445-8215
E-mail: kebjt001@d4.dion.ne.jp

KEB Portugal
Lugar de Salgueiros – Pavilhao A, Mouquim
P - 4760 V. N. de Famalicao
Tel.: 00351 / 252 / 371 318 • FAX: 00351 / 252 / 371 320
E-mail: keb.portugal@netc.pt

KEB Taiwan Ltd.
1F, No.19-5, Shi Chou Rd., Tounan Town
R.O.C. - Yin-Lin Hsian / Taiwan
Tel.: 00886 / 5 / 5964242 • FAX: 00886 / 5 / 5964240
E-mail: keb_taiwan@mail.apol.com.tw

KEBCO Inc.
1335 Mendota Heights Road
USA - Mendota Heights, MN 55120
Tel.: 001 / 651 / 4546162 • FAX: 001 / 651 / 4546198
Internet: www.kebco.com • E-mail: info@kebco.com

© KEB 0S.58.00A-K000 10/2002