



*Руководство
пользователя*

SM- Applications

Дополнительный модуль
для Unidrive SP

Номер по каталогу: 0471-0007-04
Редакция: 4

Общая информация

Изготовитель не несет никакой ответственности за любые последствия, возникшие из-за несоответствующей, небрежной или неправильной установки или регулировки дополнительных рабочих параметров оборудования или из-за несоответствия привода переменной скорости и двигателя.

Считается, что содержание этого руководства является правильным в момент его опубликования. В интересах выполнения политики непрерывного развития и совершенствования изготовитель оставляет за собой право без предварительного оповещения вносить изменения в технические условия или в рабочие характеристики или в содержание этого руководства.

Все права защищены. Никакую часть этого руководства нельзя воспроизводить или пересылать любыми средствами, электронными или механическими, путем фотокопирования, магнитной записи или в системах хранения и вызова информации без предварительного получения разрешения в письменной форме от издателя.

Версия программного обеспечения привода

Это изделие поставляется с последней версией программного обеспечения интерфейса пользователя и управления аппаратом. Если это изделие используется в новой или имеющейся системе с другими приводами, то возможны некоторые отличия между их программным обеспечением и программным обеспечением изделия. Из-за таких различий режим работы изделия может измениться. Это утверждение верно и для приводов, возвращенных из сервисного центра компании Control Techniques.

В случае возникновения вопросов обращайтесь в центр приводов Control Techniques Drive Centre.

Авторское право © Control Techniques Drives Ltd, апрель 2004 года

Редакция: 4

Аппаратура: SM-Applications выпуск 03.00

Программное обеспечение: V01.03.02

Содержание

1	Техника безопасности	6
1.1	Подразделы Предупреждение, Внимание и Примечание	6
1.2	Электрическая безопасность - общее предупреждение	6
1.3	Проектирование системы и безопасность персонала	6
1.4	Пределы воздействия на экологию	7
1.5	Соответствие нормам и правилам	7
1.6	Электродвигатель	7
1.7	Регулировка параметров	7
2	Введение	8
2.1	Введение	8
2.2	Модуль SM-Applications для Unidrive SP	8
2.3	Технические характеристики	9
2.4	Система разработки программ	9
2.5	Знания пользователя	9
3	Установка	10
3.1	Механическая установка	10
3.2	Электрические подключения	10
3.3	Подключение к CTNet	11
3.4	Кабель CTNet	11
3.5	Оконечная нагрузка сети CTNet	12
3.6	Подключение к EIA-RS485	13
3.7	Разъемы цифровых входов-выходов	15
3.8	Изоляция порта	15
3.9	Установка SmartCard	15
4	Приступаем к работе	17
4.1	Использование SYPT Workbench	18
4.2	Подключение компьютера к модулю SM-Applications	18
4.3	Выбор гнезда SM Applications	18
4.4	Конфигурирование передачи данных в SYPT	19
4.5	Создание узла Unidrive SP в SYPT	19
4.6	Перенос программ UD70 в модуль SM-Applications	19
4.7	Основы программирования на DPL	19
4.8	Пример программы	20
4.9	Загрузка программ	22
4.10	Предупреждения по работе с SYPT Workbench	22

5	Параметры	24
5.1	Обзор	24
5.2	Сохранение параметров	24
5.3	Параметры конфигурации	25
5.4	Меню 70-75 - регистры ПЛК	36
5.5	Меню 85 - параметры функции таймера	37
5.6	Меню 86 - параметры цифровых входов-выходов	39
5.7	Меню 88 - параметры состояния	40
5.8	Меню 90 - общие параметры	41
5.9	Меню 91 - параметры быстрого доступа	50
5.10	Меню 18, 19 - параметры приложения	55
5.11	Меню 20 - меню приложения	55
6	Передача данных	57
6.1	Порт последовательной передачи данных RS485	57
6.2	CTNet	60
6.3	Параметры отображения SM-Applications	60
7	Программирование на DPL	63
7.1	Заголовок программы	63
7.2	Задачи	64
7.3	Переменные	66
7.4	Параметры	69
7.5	Операторы	70
7.6	Основные команды DPL	71
7.7	Определяемые пользователем функциональные блоки	76
8	Фиксация и маркер в SM-Applications	79
8.1	Вход фиксации Freeze	79
8.2	Импульс маркера	80
9	CTSync	82
9.1	Обзор	82
9.2	Подключения	82
9.3	Ограничения	82
9.4	Функциональные блоки CTSync	82
9.5	Подвижный агрегат	84
9.6	Пример виртуального ведущего	86
10	Синхронизация дополнительных модулей	88
10.1	Обзор	88
10.2	Пример 1 синхронизации дополнительных модулей	88
10.3	Пример 2 синхронизации дополнительных модулей	92
10.4	Задачи управления положением	96

11	Диагностика	97
11.1	Ошибки времени выполнения	97
11.2	Просмотр кодов отключения Unidrive SP	97
11.3	Коды ошибок времени выполнения SM-Applications	98
11.4	Обработка ошибок времени выполнения в задаче ERROR	101
11.5	Техническая поддержка	101
12	Руководство по переходу	103
12.1	Различия в параметрах привода	103
12.2	Параметры UD70	103
12.3	Общие функции	105
12.4	Утилита SM-Applications Porting Aid	107
13	Краткий справочник	110

1 Техника безопасности

1.1 Подразделы Предупреждение, Внимание и Примечание

ПРИМЕЧ. В **Примечании** содержится информация, помогающая обеспечить правильную работу изделия.



Предупреждение содержит информацию, важную для устранения опасностей при работе.



Внимание содержит информацию, важную для исключения риска повреждения изделия или другого оборудования.

1.2 Электрическая безопасность - общее предупреждение

В приводе используются напряжения, которые могут вызвать сильное поражение электрическим током и/или ожоги и могут быть смертельными. При работе с приводом или вблизи него следует соблюдать предельную осторожность. Конкретные предупреждения приведены в нужных местах этого руководства.

1.3 Проектирование системы и безопасность персонала

Привод предназначен для профессионального встраивания в полную систему. В случае неправильной установки привод может создавать угрозу для безопасности.

В приводе используются высокие напряжения и сильные токи, в нем хранится большой запас электрической энергии, и он управляет оборудованием, которое может привести к травмам.

Необходимо тщательно выполнять установку системы, чтобы не допустить появления опасностей при нормальной работе и при поломке оборудования. Проектирование, монтаж, сдача в эксплуатацию и техническое обслуживание системы должно выполняться только соответственно обученным опытным персоналом. Такой персонал должен внимательно прочесть эту информацию по технике безопасности и все руководство пользователя.

Функции привода **ОСТАНОВ** и **ЗАЩИТНОЕ ОТКЛЮЧЕНИЕ** не отключают опасные напряжения с выхода привода и с любого внешнего блока. Перед выполнением работ на электрических соединителях необходимо отключить электрическое питание с помощью проверенного устройства электрического отключения.

За исключением единственной функции ЗАЩИТНОЕ ОТКЛЮЧЕНИЕ ни одну из функций привода нельзя использовать для обеспечения безопасности персонала, то есть их нельзя использовать для охраны условий труда.

Необходимо внимательно продумать все функции привода, которые могут создать опасность, как при обычной эксплуатации, так и в режиме неверной работы из-за поломки. Для любого применения, в котором поломка привода или его системы управления может привести к ущербу или способствовать его появлению, необходимо провести анализ степени риска и при необходимости принять специальные меры для снижения риска - например, установить устройства защиты от превышения скорости для случая выхода из строя системы управления скоростью или надежный механический тормоз для случая отказа системы торможения двигателем.

Функция ЗАЩИТНОЕ ОТКЛЮЧЕНИЕ была аттестована¹ как соответствующая стандарту EN954-1 категории 3 для предотвращения случайного запуска двигателя. Ее можно использовать для обеспечения безопасности. **Проектировщик системы несет полную ответственность за безопасность всей системы и ее соответствие требованиям стандартов обеспечения безопасности.**

¹Независимая аттестация от VIA была получена для габаритов от 1 до 3.

1.4 Пределы воздействия на экологию

Необходимо строго соблюдать все указания этого *Руководства пользователя Unidrive SP* относительно транспортировки, хранения, монтажа и эксплуатации привода, включая указанные пределы воздействия на экологию. К приводам нельзя прилагать чрезмерных механических усилий и нагрузок.

1.5 Соответствие нормам и правилам

Монтажник отвечает за соответствие требованиям всех действующих правил и норм, например, национальным правилам устройства электроустановок, нормам предотвращения аварий и правилам электромагнитной совместимости (ЭМС). Особое внимание следует уделить поперечному сечению проводов, выбору предохранителей и других средств защиты и подключению защитного заземления. В *Руководстве пользователя Unidrive SP* содержатся указания по достижению соответствия с конкретными стандартами ЭМС.

Внутри Европейского союза все механизмы, в которых может использоваться данный привод, должны соответствовать следующим директивам:

98/37/ЕС: Безопасность механизмов.

89/336/ЕЕС: Электромагнитная совместимость.

1.6 Электродвигатель

Проверьте, что электродвигатель установлен согласно рекомендациям изготовителя. Проверьте, что вал двигателя не поврежден.

Стандартные асинхронные электродвигатели с короткозамкнутым ротором предназначены для работы на одной скорости. Если предполагается использовать возможности привода для управления двигателем на скоростях выше проектной максимальной скорости, то настоятельно рекомендуется, прежде всего, проконсультироваться с изготовителем двигателя.

Низкая скорость работы может привести к перегреву двигателя из-за падения эффективности работы вентилятора охлаждения. Двигатель необходимо оснастить защитным термистором. При необходимости установите электровентилятор для принудительного охлаждения.

На степень защиты двигателя влияют настроенные в приводе значения параметров двигателя. Не следует полагаться на значения этих параметров по умолчанию.

Очень важно, чтобы в параметр Pr **0.46** “Номинальный ток двигателя” было введено правильное значение. Это влияет на тепловую защиту двигателя.

1.7 Регулировка параметров

Некоторые параметры сильно влияют на работу привода. Их нельзя изменять без подробного изучения влияния на управляемую систему. Следует предпринять специальные меры для защиты от нежелательных изменений из-за ошибки или небрежности.

2 Введение

ПРИМЕЧ. В этом руководстве параметры Unidrive SP и SM-Applications обозначаются как “#ММ.РР”, где ММ обозначает номер меню, а РР обозначает номер параметра внутри этого меню. Пожалуйста, смотрите полное описание параметров привода в Руководстве пользователя Unidrive SP.

Также смотрите раздел 5.2 *Сохранение параметров*, где приведена информация о сохранении параметров.

2.1 Введение

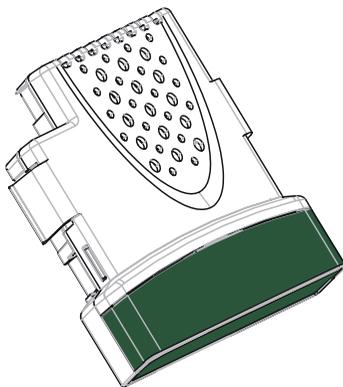
Современные приводы переменной скорости, например, Unidrive SP, имеют множество встроенных функций - управление рампой, контура ПИД, простое управление положением и т.п. Однако эти функции ограничены. Привод может делать только то, чем он оснащен, и если требуется управлять более сложным приложением, то пользователи часто должны использовать внешнее оборудование, например, ПЛК, для системного управления приводом.

Однако гибкость Unidrive SP можно существенно повысить при установке модуля SM-Applications. Этот модуль является дополнительным процессором для привода и позволяет пользователю использовать имеющиеся или писать собственные прикладные программы. Он также обладает мощными сетевыми возможностями, так что можно соединить вместе много приводов (и другого оборудования) для обмена технологической информацией, что позволяет получать полные прикладные решения.

2.2 Модуль SM-Applications для Unidrive SP

Модуль SM-Applications для привода Unidrive SP - это дополнительный модуль, который устанавливается в гнездо расширения на приводе Unidrive SP.

Рис. 2-1 Дополнительный модуль Unidrive SP



Модуль SM-Applications питается от внутреннего блока питания Unidrive SP.

ПРИМЕЧ. При работе с системой SYPT Workbench для программирования модуля SM-Applications вы должны указать SYPT, в каком гнезде установлен модуль. По умолчанию SYPT будет использовать 1, но это можно изменить с помощью *Slot Select*. Ее можно вызвать по пути : меню Пуск->Программы->SYPT Workbench->Slot Selector.

2.3 Технические характеристики

- Высокоскоростной специализированный микропроцессор
- Флэш-память 384 кбайт для программ пользователя
- Оперативная память 80 кбайт для программ пользователя
- Порт EIA-RS485 с поддержкой протоколов ANSI, Modbus-RTU ведомый и ведущий и Modbus-ASCII ведомый и ведущий
- Интерфейс к высокоскоростной сети CTNet со скоростью до 5 Мбит/с.
- Два цифровых входа 24 В
- Два цифровых выхода 24 В
- Двухпортовый интерфейс ОЗУ для обмена данными с Unidrive SP и другими дополнительными модулями
- Система программирования на базе задач, позволяющая в режиме реального времени управлять приводом и процессом.

2.4 Система разработки программ

Пользователь может разработать прикладные программы для модуля SM-Applications с помощью пакета программ SYPT Workbench. Пользователи, знакомые с изделием UD70 для Unidrive 1, могли уже работать с этой программой.

Программа SYPT Workbench содержит разные утилиты для разработки программ:

- Редактор конфигурации для конфигурирования приводов и подключения к сети CTNet.
- Программирование функциональных блоков и ступенчатой логики IEC61131-3
- Программирование на “родном” языке DPL
- Окно просмотра для контроля параметров любого привода или модуля, а также переменных программы пользователя
- Возможности отладки программы по шагам и с точкой останова

В программе SYPT Workbench вы можете связаться с SM-Applications либо по:

- Прямому подключению к порту RS485 на передней панели Unidrive SP
- Подключением к одному или нескольким модулям в сети CTNet (необходима интерфейсная плата CTNet для компьютера)

Программа SYPT Workbench работает под Microsoft Windows™ 98/ME/NT4/2000/XP.

2.5 Знания пользователя

При разработке прикладных программ важно иметь представление о программировании задач реального времени, управляемых событиями. Знание основ языка программирования BASIC также полезно, но не существенно. Средства многоступенчатых схем (LD) и функциональных блок-схем (FBD) в SYPT Workbench упрощают программирование для людей, знакомых с ПЛК.

В этом Руководстве считается, что пользователь имеет хотя бы начальные знания о Microsoft Windows™.

3 Установка

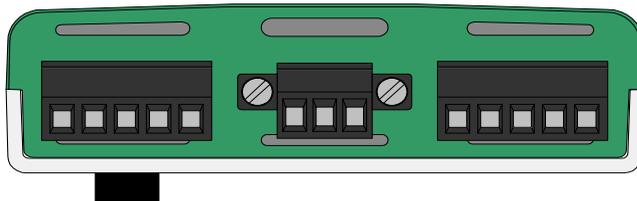
3.1 Механическая установка

Процедура установки модуля SM-Applications в привод Unidrive SP описана в монтажном листке, входящем в комплект модуля.

3.2 Электрические подключения

Модуль SM-Applications имеет две 5-контактные и одну 3-контактную клеммные колодки "под винт".

Рис. 3-1 Модуль SM-Applications - вид спереди



Клеммы нумеруются последовательно от клеммы 1 с левой стороны до клеммы 13 с правой стороны. Функции клемм указаны в таблице ниже:

Таблица 3.1 Разъемы модуля SM-Applications

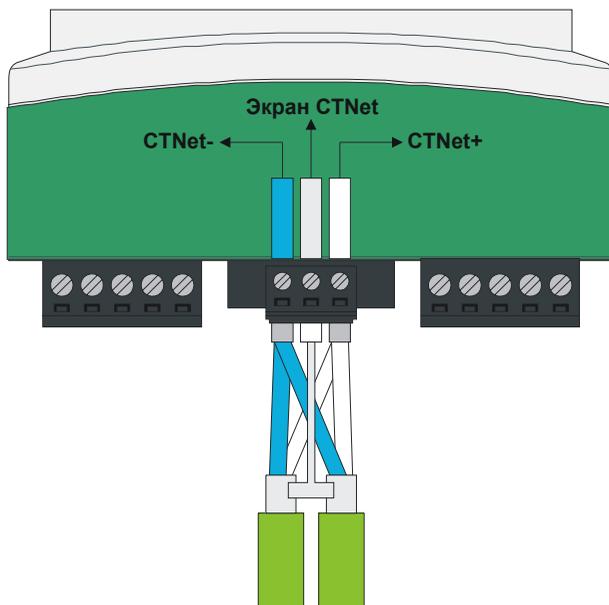
Клемма	Функция	Описание
1	0V SC	Подключение 0 В для порта RS485
2	/RX	Линия приема EIA-RS485 (отрицательная). Входящая.
3	RX	Линия приема EIA-RS485 (положительная). Входящая.
4	/TX	Линия передачи EIA-RS485 (отрицательная). Исходящая.
5	TX	Линия передачи EIA-RS485 (положительная). Исходящая.
6	CTNet A	Линия данных CTNet
7	CTNet Shield	Подключение экрана для CTNet
8	CTNet B	Линия данных CTNet
9	0V	Подключение 0 В для цифровых входов-выходов
10	DIO	Цифровой вход 0
11	DI1	Цифровой вход 1
12	DO0	Цифровой выход 0
13	DO1	Цифровой выход 1

3.3 Подключение к CTNet

ПРИМЕЧ. В этом Руководстве описаны только основы подключения к сети CTNet. Смотрите Руководство пользователя CTNet, где приведена полная информация.

Для подключения модуля SM-Applications к сети CTNet выполните соединения, показанные на рисунке ниже.

Рис. 3-2 Подключение к сети CTNet



Экраны обеих кабелей должны быть скручены вместе и подсоединены к центральной клемме 3-контактной колодки. Длину такой “косички” следует делать как можно короче. Такая конструкция обеспечивает непрерывность экрана.

3.4 Кабель CTNet

Кабель CTNet содержит одну витую пару и общий экран. Для подключения используется один набор клемм данных. Преимущество такого подхода заключается в том, что если клеммная колодка не подключена, то все равно сеть CTNet не разрывается (остается непрерывной).

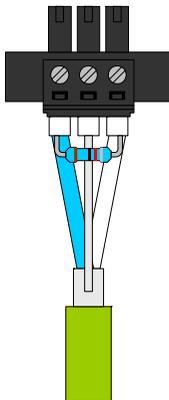
Сети CTNet работают при высоких скоростях передачи данных и поэтому в них надо использовать кабели, специально предназначенные для передачи высокочастотных сигналов. Кабели низкого качества могут ослабить сигналы, так что сигналы станут нечитаемыми на других узлах сети. Рекомендуется использовать для сети CTNet только тот кабель, который поставляется компанией Control Techniques.

3.5 Оконечная нагрузка сети CTNet

При использовании сетей высокоскоростной передачи данных очень важно, чтобы кабели данных сети с обоих концов были нагружены на номинальные нагрузочные резисторы. Это устраняет отражение сигналов от концов кабеля, которые могут вызывать помехи при работе.

Сопrotивление нагрузочного резистора должно точно совпадать с импедансом (волновым сопротивлением) кабеля. Для рекомендуемого зеленого кабеля Control Techniques CTNet резистор 82 Ома мощностью 0,25 Вт следует установить между линиями данных CTNet+ и CTNet- с ОБОИХ концов кабеля.

Рис. 3-3 Оконечная нагрузка сети CTNet



Неправильное подключение нагрузки кабеля может серьезно ухудшить работу сети. Если не установлены нагрузочные резисторы с нужным сопротивлением, то помехоустойчивость сети сильно снижается.

Если в сети CTNet установлено слишком много нагрузочных резисторов, то сеть будет перегружена и уровни сигналов снизятся. При этом узлы могут пропускать некоторые биты информации, что приведет к ошибкам передачи данных. При сильной перегрузке сети уровни сигналов могут быть такими низкими, что узлы вообще не могут обнаружить сетевой активности.

3.5.1 Подключение экрана кабеля CTNet

Подключение к сети CTNet следует выполнить, изолировав экраны кабелей от земли на каждом приводе Unidrive SP. Экраны кабелей следует скрутить вместе в той точке, где они выходят из кабеля, и сделать из них короткую “косичку” для подключения экрана кабеля сети CTNet к клеммной колодке, как показано выше. Для обеспечения безопасности экран кабеля CTNet нужно подключать к земле в одной точке. Такое подключение к земле предотвращает появление на кабеле напряжения в случае поломки другого устройства в сети CTNet. Для защиты от помех землю нужно подключать только в одной точке.

3.5.2 Максимальная длина сети и число узлов

Максимальное число узлов в одной сети CTNet равно 255, однако при этом сеть следует разбить на сегменты, разделенные повторителями. Максимальная длина сетевого кабеля в сети CTNet зависит от скорости в Бодах.

Повторители можно приобрести у Control Techniques. В таблице ниже показаны номер по каталогу для заказа различных повторителей.

Номер по каталогу	Описание
4500-0033	AI3-485X - Rev A
4500-0083	AI3-485X-CT - Rev D (Порт 1), Rev A (Порты 2 и 3)
4500-0082	AI3-CT - Rev D
4500-0032	AI2-485X/FOG-ST - Rev A (волоконная оптика)
4500-0081	AI2-CT/FOG-ST - Rev D (волоконная оптика)

Более подробные сведения приведены в Руководстве пользователя CTNet.

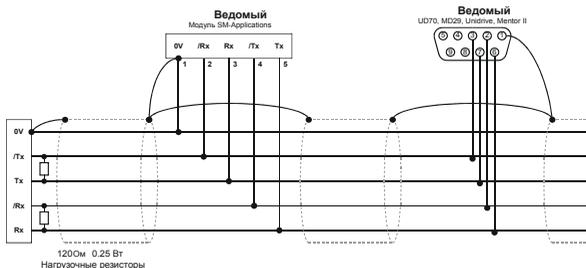
3.6 Подключение к EIA-RS485

Порт EIA-RS485 предназначен для низкоскоростной передачи данных (до 115200 бит/с). Этот порт стандартно поддерживает протоколы CT-ANSI ведомый, Modbus-RTU ведущий и ведомый и Modbus-ASCII ведущий и ведомый. Возможны подключения как по 2, так и по 4 проводам.

Более подробные сведения по работе с портом RS485 приведены в главе 6 *Передача данных*.

3.6.1 4-проводная сеть RS485

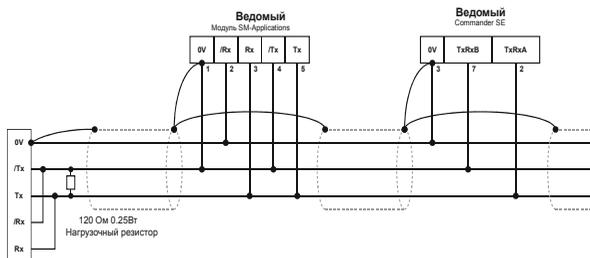
На схеме ниже показаны соединения, необходимые для 4-проводной сети RS485 с использованием ведущего контроллера с портом RS485. Модули SM-Applications можно настроить на работу в режиме ведущего контроллера, то при этом для управления сетью нужна программа DPL.



Чтобы обычный последовательный порт компьютера смог обмениваться данными с 4-проводной сетью RS485, потребуется переходник (преобразователь) типа RS232-RS485.

3.6.2 2-проводная сеть RS485

На схеме ниже показаны соединения, необходимые для 2-проводной сети RS485 с использованием ведущего контроллера с портом RS485. Модули SM-Applications можно настроить на работу в режиме ведущего контроллера, но при этом для управления сетью нужна программа DPL.



Чтобы обычный последовательный порт компьютера смог обмениваться данными с 2-проводной сетью RS485, потребуется переходник (преобразователь) типа RS232-RS485 с “разумным переключением трансивера” (известный также как “волшебный” преобразователь RS485). В качестве примера “волшебного” преобразователя можно указать преобразователь MA485F от Amplicon.

ПРИМЕЧ.

“Волшебный” преобразователь не нужен, если у ведущего контроллера имеется выход управления RTS. Этот выход включается при выполнении передачи с ведущего и отключается, когда ведущий не передает. Пакеты программ от Control Techniques (UniSoft, MentorSoft и SystemWise) HE переключают линию RTS.

3.6.3 Заземление

Для обеспечения безопасности рекомендуется экран кабеля передачи данных соединить проводником с низкой индуктивностью с точкой “чистой” земли. Такое заземление должно быть выполнено только в одной точке.

3.6.4 Прокладывание кабеля

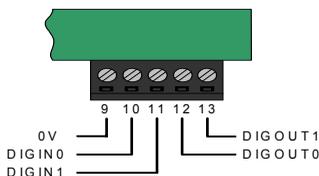
Кабель передачи данных не должен идти параллельно ни одному силовому кабелю, особенно кабелю, соединяющему привод и двигатель. Если параллельное расположение кабеля передачи данных и силового кабеля неизбежно, то расстояние между ними должно быть не менее 30 см.

Если кабели пересекаются под прямым углом, то это не должно создать никаких проблем. Максимальная длина кабеля для канала RS485 равна 1200 метров. Это длина только для низкой скорости передачи. Чем больше скорость передачи, тем короче должен быть кабель.

3.6.5 Нагрузочные сопротивления

Если используется система RS485 большой дальности с многими отводами, то на передающей и приемной парах должны быть установлены нагрузочные резисторы сопротивлением 120 Ом для снижения отражений сигналов. Однако при низких скоростях передачи это не обязательно.

3.7 Разъемы цифровых входов-выходов



Модуль SM-Applications оснащен 2 цифровыми входами, которые называются DIGIN0 и DIGIN1, и 2 цифровыми выходами с названиями DIGOUT0 и DIGOUT1. Этими входами и выходами можно управлять с помощью программы пользователя, загруженной в модуль SM-Applications.

Цифровые выходы используют положительную логику, так что они имеют напряжение +24 В в активном состоянии и могут выдавать ток до 20 мА. В пассивном состоянии выходы являются плавающими (отключенными). Цифровые выходы защищены от короткого замыкания и перегрузки. Порог отключения составляет 20 мА и при его превышении отключаются оба выхода.

Управление цифровыми входами-выходами проводится из меню 86 - смотрите раздел 5.6 *Меню 86 - параметры цифровых входов-выходов*.

3.8 Изоляция порта

Порты цифровых входов-выходов подключены к цепям управления главного привода.



Цепи входов-выходов изолированы от силовых цепей только основной (единственной) изоляцией. Монтажник должен обеспечить защиту внешних цепей управления от контакта с телом человека по крайней мере одним слоем изоляции (дополнительная изоляция), рассчитанной на переменное напряжение питания.

Порты CTNet и EIA/RS485 имеют дополнительную изоляцию от портов входов-выходов, что в целом обеспечивает двойную изоляцию от силовых цепей. Они имеют простое разделение (функциональная изоляция) друг от друга.



Для обеспечения состояния двойной изоляции портов данных:

- Все цепи, к которым подключен любой порт, должны иметь защитное разделение (то есть двойную изоляцию или одинарную изоляцию с заземлением)
- Все цепи, к которым подключены цепи управления приводом, должны иметь хотя бы основную изоляцию от силового напряжения.

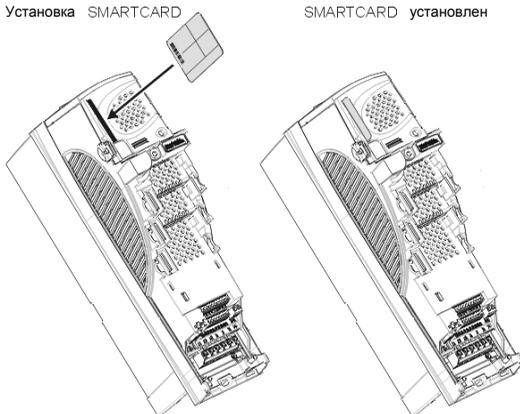
3.9 Установка SmartCard

Привод Unidrive SP можно оснастить SmartCard (смотрите Рис. 3-4 *Установка SmartCard* ниже), что позволяет пользователю читать и записывать данные на карту. Это полезно для хранения наборов параметров и других данных. Более подробная информация об использовании карты SmartCard с панели управления приведена в Руководстве пользователя Unidrive SP. Смотрите раздел 7.6.1 *Новые команды для модуля SM-Applications*, где приведена информация по использованию команд DPL для работы с картой SmartCard.

Рис. 3-4 Установка SmartCard

Установка SMARTCARD

SMARTCARD установлен



4 Приступаем к работе

В этой главе описаны основы структуры программы пользователя для модуля SM-Applications и некоторые аспекты использования программы SYPT Workbench. В приводе Unidrive SP есть 3 гнезда для дополнительных модулей, и каждое гнездо имеет собственное меню для конфигурирования параметров.

Рис. 4-1 Гнезда в приводе Unidrive SP

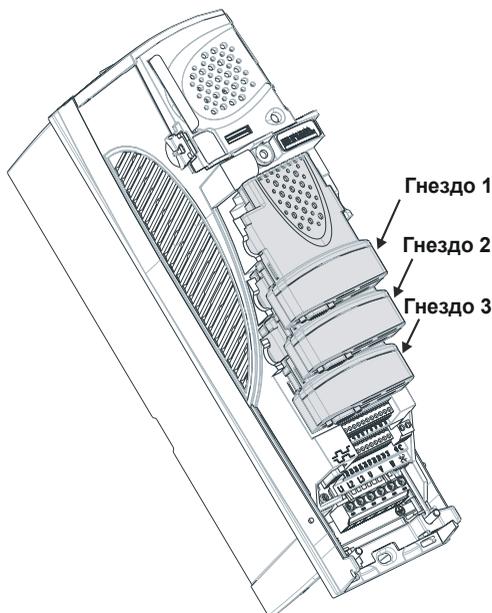


Таблица 4.1 Меню конфигурации гнезда

Гнездо	Меню
1	#15.PP
2	#16.PP
3	#17.PP

Внутри модуля SM-Applications меню текущего гнезда имеет псевдоним как меню 81. Поэтому при подключении к модулю из канала передачи данных или из программы пользователя проще всего обращаться к параметрам конфигурации как к меню 81.

На протяжении остальной части этого Руководства пользователя для обращения к конкретному параметру в любом гнезде будет использоваться формат #81.PP, например, параметр *Автостарт (Autorun)* будет обозначаться как #81.13.

Это также упрощает проблемы переносимости, поскольку модуль SM-Applications с программой, использующей меню 81, можно вставить в любое гнездо и программа будет работать без ошибок.

Если в приводе установлен модуль SM-Applications, то параметр идентификации модуля #81.01 покажет значение 301. Комбинация параметров #81.02 и #81.51 указывает версию микропрограммного обеспечения модуля. Это Руководство пользователя написано для микропрограммы версии V01.03.02.

4.1 Использование SYPT Workbench

Программа SYPT Workbench, используемая в изделии UD70 для Unidrive 1, является единственной платформой разработки программ для SM-Applications. Имеются определенные предупреждения по использованию программы SYPT Workbench с модулем SM-Applications, которые будут приведены дальше.

4.2 Подключение компьютера к модулю SM-Applications

Имеются два метода подключения программирующего компьютера к модулю SM-Applications, которые описаны ниже:

4.2.1 CTNet

С помощью сети CTNet вы можете подключить компьютер к сети приводов, что позволяет вам программировать привод и управлять им прямо с компьютера. Однако для этого в компьютере должна быть установлена интерфейсная плата CTNet. Компания Control Techniques поставляет платы конструктива PCI и PCMCIA для настольных и портативных компьютеров.

Сеть CTNet предоставляет очень высокие скорости загрузки и обновления для отслеживания работы в реальном масштабе времени.

Смотрите раздел 3.3 *Подключение к CTNet*, где описано подключение сети CTNet к модулю SM-Applications.

4.2.2 Последовательный порт RS485

С помощью специального преобразователя RS232-RS485 вы можете подключить компьютер к последовательному порту RJ45 на передней панели привода. Для этой цели компания Control Techniques выпускает специальный кабель - этот же кабель используется с другими изделиями Control Techniques, в которых используется разъем RJ45 RS485, например, Commander SE.

Разъем RJ45 расположен под небольшой шторкой на передней панели Unidrive SP как раз ниже гнезда для панели управления. Разводка этого разъема описана в Руководстве пользователя Unidrive SP.

При таком типе подключения вы можете управлять только тем приводом и модулем SM-Applications, к которому подключен компьютер.

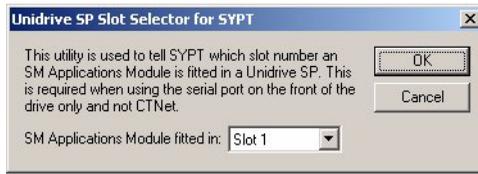
Рис. 4-2 Кабель для передачи данных



4.3 Выбор гнезда SM Applications

При работе с SYPT Workbench с подключением по RS232/485 вы должны указать SYPT, в каком гнезде установлен модуль SM-Applications. Для этого используется программа *Slot Selector* (вызов по Пуск->Программы->Меню SYPT Workbench).

Рис. 4-3 Программа выбора гнезда



4.4 Конфигурирование передачи данных в SYPT

Перед попыткой установить связь с модулем SM-Applications вы должны настроить SYPT на применение нужного протокола передачи данных:

1. В SYPT Configuration Editor выберите **Communications** в меню Options
2. При подключении через CTNet выберите CTNet в качестве протокола и проверьте правильную скорость передачи (при нажатии кнопки Help будут показаны параметры других настроек).
3. При подключении через RS232/485 к передней панели привода выберите протокол **CT-RTU** и выберите соответствующий порт RS232 COM. Также проверьте, что параметры привода #11.24 и #11.25 настроены в **rtU** и **19200** соответственно (это значения по умолчанию). В настоящее время SYPT может передавать данные по этому протоколу только со скоростью 19200 Бод.
4. Нажмите ОК.

4.5 Создание узла Unidrive SP в SYPT

Узел Unidrive SP создается точно так же, как другие типы узла (привода):

1. Вставьте новый узел, выбрав пункт **Node** в меню Insert.
2. Дважды щелкните по имени нового узла (или выберите пункт меню Edit->Node properties).
3. В раскрывшемся списке Target Type выберите **SMAPPL** (тип привода автоматически изменится на Unidrive SP).
4. Нажмите ОК.



1 : Node_1

Новым узлом теперь будет Unidrive SP.

4.6 Перенос программ UD70 в модуль SM-Applications

Если вы собираетесь преобразовать программы с платформы UD70 для Unidrive 1 в программы для Unidrive SP, то прочтите важную информацию в главе 12 *Руководство по переносу программ*.

4.7 Основы программирования на DPL

Модуль SM-Applications можно программировать с помощью смеси из многоступенчатых схем (LD), функциональных блок-схем (FBD) и кода DPL (язык программирования привода). Вместе все это называется *Программа DPL*.

На самом верхнем уровне программа состоит из:

- Заголовок программы - указывает название программы, автора, версию и т.п. Он настраивается с помощью окна свойств узла в SYPT.
- Тело программы - состоит из разделов *задач*, содержащих разделы LD, FBD и DPL. Оно создается в редакторе DPL Editor внутри SYPT.

Разделы задач охватывают блоки инструкций, которые должны быть выполнены микропроцессором в конкретное время, например, каждые 8 мсек или при первом включении модуля. Каждая задача имеет свое конкретное имя, назначение и приоритет. Более подробно это описано в разделе 7.2 *Задачи*.

4.7.1 Библиотека функциональных блоков

Программа SYPT Workbench поставляется с обширной библиотекой готовых функциональных блоков. Они выполняют задачи от простых (счетчик) до более сложных, например, контуры ПИД и генераторы профиля s-рампы. Эти готовые блоки вместе называются Библиотекой функциональных блоков (FBL).

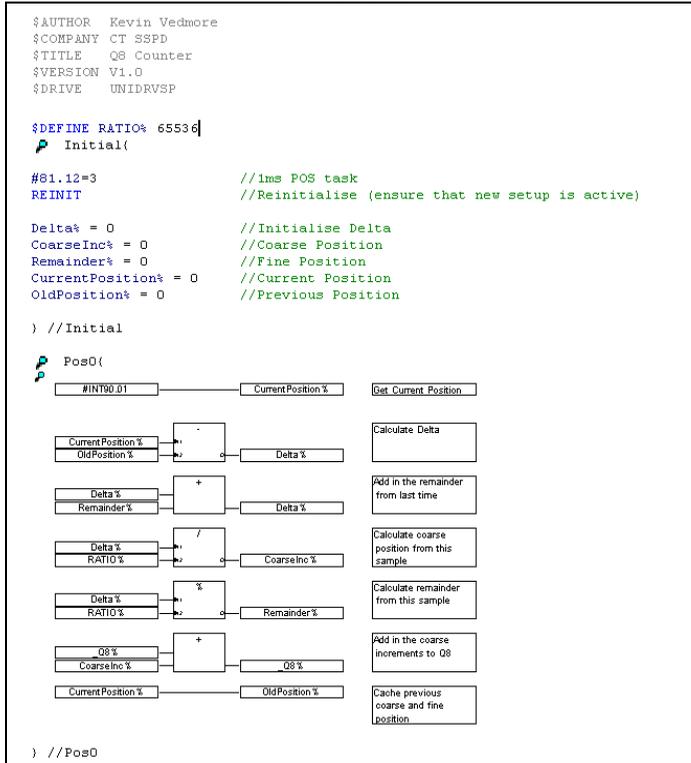
Все функции внутри FBL описаны в оперативной справочной системе.

Вы можете также создать в программе свои функциональные блоки. Если вы создали генератор нового удобного профиля, то вы можете оформить его как функциональный блок пользователя (UDFB) и встроить его в свою главную программу DPL. Более подробно это описано в разделе 7.7 *Определяемые пользователем функциональные блоки* и в оперативной справочной системе.

4.8 Пример программы

Ниже показан пример программы DPL:

Рис. 4-4 Пример программы DPL



Эта программа считывает с привода информацию обратной связи о положении (которая масштабируется до 2^{32} /оборот), вычисляет изменение (которое будет пропорционально скорости) и преобразует его в импульсы энкодера (на основе стандартного импульсного энкодера) и добавляет это в аккумулятор (интегратор). Этот пример показывает основные концепции доступа к параметрам и применения математических функций. Этот пример может быть полезен для людей, переходящих с платформы UD70 для Unidrive 1, поскольку он показывает, как воссоздать накапливающее положение энкодера `_Q8%`, которое было доступно на этом изделии.

При просмотре программы вы обнаружите четыре основных раздела:

- Раздел заголовка
- Начальная задача **Initial**
- Задача **Pos0**
- Функциональная блок-схема

4.8.1 Раздел заголовка

Этот раздел автоматически создается программой SYPT по параметрам из диалогового окна свойств узла. Он содержит такие сведения, как название программы, ее автора и версию.

4.8.2 Начальная задача Initial

Как объясняется далее в разделе 7.2 *Задачи*, это *задача*, которая выполняется при первом включении питания модуля SM-Applications и при его сбросе при условии, что установлен параметр автозапуска Autorun (смотрите раздел о сохранении параметров). В этой задаче имеются операторы DPL, которые инициализируют в нуль некоторые целые переменные (обозначенные “хвостовым” символом %).

4.8.3 Задача Pos0 и функциональная блок-схема

Поскольку эта программа работает с информацией обратной связи по положению, то основная часть работы выполняется задачей POS0. Все операции, относящиеся к скорости, положению и моменту, обычно выполняются в задаче POS0 и POS1 или CLOCK, которая теперь синхронизована с приводом. В этом случае здесь будет единственная функциональная блок-схема, которая выполняет все вычисления, нужные для определения положения инкрементного энкодера.

Задача выполняет следующие основные действия:

1. Считывает текущее значение обратной связи с энкодера
2. Вычитает ранее считанное значение обратной связи по положению для нахождения изменения (дельты).
3. Масштабирует значение в фактические импульсы энкодера, предполагая использование стандартного инкрементного энкодера (а не энкодера SinCos).
4. Добавляет это приращение в аккумулятор (интегратор).
5. Запоминает текущее положение энкодера для следующего цикла расчетов.

В этом примере программы используется переменная `_Q8%`. Это 32-битное значение подобно любой другой переменной, но оно является частью специального набора *регистров*, называемого набором регистров ПЛК. Эти регистры ПЛК удобны за счет того, что их можно сохранять в энергонезависимой памяти и также вызывать через параметры в меню с 70 по 76. Более подробно это описано в разделе 5.4 *Меню 70-75 - регистры ПЛК*.

ПРИМЕЧ. Если вы хотите сами создать и попробовать эту программу, и ранее вы не использовали программу SYPT Workbench, то рекомендуется сначала внимательно прочитать всю эту главу, а затем прочесть главу “Приступаем к работе” в Руководстве пользователя SYPT Workbench, в которой объясняется, как создавать такую программу.

Чтобы обеспечить выполнение задачи POS0, параметр #81.12 необходимо настроить в ненулевое значение в начальной задаче Initial. После настройки этого параметра нужно подать команду REINIT (смотрите ниже).

```
#81.12 = 3 //Период активации задачи Pos 1 мсек
REINIT //Повторная инициализация
```

4.9 Загрузка программ

По умолчанию программы можно загружать в модуль SM-Applications только если сигнал включения привода не активен (#6.15=0). Такое поведение можно отменить, настроив параметр #81.37 в 0.

4.10 Предупреждения по работе с SYPT Workbench

Ниже перечислены известные проблемы, возникающие при использовании программы SYPT Workbench с модулем SM-Applications:

- **Выбор гнезда**
Если вы подключаетесь к приводу Unidrive SP с помощью последовательного разъема RJ45 на передней панели, то SYPT по умолчанию ожидает, что модуль SM-Applications вставлен в ГНЕЗДО 1 привода. Однако вы можете изменить это на любое другое гнездо с помощью прилагаемой утилиты SLOT SELECT (вызов проводится по цепочке меню Пуск->Программы->SYPT Workbench->Slot Selecter).
- **Точки останова в задачах POS0/1**
Если вы настроите точку останова в любой из этих задач, то точка останова сработает, но флаг не будет показан и сбросить точку останова можно будет только за счет сброса дополнительного модуля.
- **Задачи EVENTx**
Нельзя проводить пошаговую отладку и размещать точки останова в любой из задач EVENT.
- **Определяемые пользователем функциональные блоки (UDFB)**
Модуль SM Applications имеет очень большую гибкость относительно входных и выходных аргументов UDFB (имеется большое количество входов и выходов и нет никаких ограничений на выравнивание). Однако программа SYPT все же применяет старые ограничения UD70, так что в настоящее время нельзя воспользоваться преимуществами новой гибкости.

5 Параметры

5.1 Обзор

Модуль SM-Applications содержит две базы данных параметров:

- База данных Unidrive SP
Эта база содержит полный набор параметров привода. Модуль SM-Applications кэширует эту базу данных в своей собственной энергонезависимой *флэш* памяти. При включении питания модуль проверяет, соответствует ли кэш содержимому привода. Если нет, то база данных загружается с привода, при этом на дисплее привода на несколько секунд появляется слово “Loading”. Такая загрузка не повторится до тех пор, пока модуль SM-Applications не будет переставлен в другой привод с другой версией микропрограммы или не будет обновлена микропрограмма привода.
- База данных модуля SM-Applications
Эта база данных содержит все параметры, хранящиеся локально на модуле, например, регистры ПЛК, а также другие “короткие” параметры (меню 90, 91, и т.д.)

5.2 Сохранение параметров

В зависимости от типа сохраняемого параметра имеются различные способы сохранения параметров. Они подробно описаны в следующих разделах:

5.2.1 Сохранение параметров SM-Applications

При выполнении показанных ниже действий в модуле SM-Applications сохраняются следующие параметры:

- Меню 70, 71, 74 и 75 (эквивалентно наборам регистров P, Q, T и U)
- Меню 20

Для сохранения параметров по запросу:

1. Настройте #81.19 в 1
2. Нажмите кнопку Сброс
Параметр #81.19 будет автоматически сброшен в нуль и произойдет сброс привода.

Для сохранения параметров при падении напряжения (UU):

- Настройте #81.20 в 1 (не забывайте, что изменение этого параметра вступает в силу только после сброса модуля)

Обратите внимание, что простое выполнение описанных выше операций не сохранит меню 20. Для сохранения меню 20 вам надо выполнить описанные выше операции, но перед этим обязательно установить параметр #81.21 в 1. Для вступления в силу изменения значения этого параметра не нужно выполнять сброс модуля.

5.2.2 Восстановление параметров меню 20

Для восстановления параметров меню 20 при включении питания параметр #81.21 должен быть равен 1 при включении питания, поэтому нужно сохранить параметры привода. Смотрите раздел 5.2.3 *Сохранение параметров привода*.

5.2.3 Сохранение параметров привода

При выполнении показанных ниже действий в приводе сохраняются следующие параметры:

- Меню с 1 по 14, 18, 19 и 21.
- Меню 15, 16 и 17, если модуль установлен в соответствующем гнезде.

Для сохранения параметров привода:

1. Настройте #0.00=1000 (параметр ноль в любом меню при работе с панели привода)
2. Настройте #10.38=100 (имитирует нажатие кнопки Сброс на панели привода)

5.3 Параметры конфигурации

Параметры основной конфигурации (или настройки) хранятся в соответствующем меню для гнезда, в которое вставлен модуль SM-Applications.

Гнездо	Меню
1	15
2	16
3	17

Кроме этих меню, доступен псевдоним на соответствующее меню в виде местного меню 81 внутри модуля SM-Applications. Это меню можно вызвать из пользовательской программы DPL или по каналу связи (CTNet/CT-RTU/RS485), оно обеспечивает удобный способ чтения и изменения параметров настройки, при этом не надо знать, в каком именно гнезде установлен модуль SM-Applications.



Если не указано иное, то эти параметры считываются только при первом включении питания SM-Applications, при сбросе и по команде *REINIT* DPL. Изменение любого из этих параметров “на ходу” не дает немедленного эффекта.

Для сброса модуля SM-Applications с панели привода введите значение 1070 в параметр ноль и нажмите кнопку Сброс.

ПРИМЕЧ. В этом Руководстве пользователя параметры конфигурации будут обозначаться как #81.xx. При настройке параметров непосредственно с панели привода используйте вместо этого соответствующее меню 15, 16 или 17.

ПРИМЕЧ. Указанная для любого параметра скорость обновления означает скорость, с которой параметр обновляется для чтения или при записи - когда новое значение вступит в силу. “Инициализация” означает, что параметр считывается только при сбросе модуля или по команде *REINIT* DPL.



Изменение режима работы привода сбрасывает все параметры конфигурации и приложения назад в их значения по умолчанию, так же как и параметры привода. Этого можно избежать, если использовать в параметре ноль код **1255** вместо обычного кода **1253**. При этом в значения по умолчанию будут сброшены только параметры привода, а меню от 15 до 20 не изменятся.

5.3.1 Описания параметров

#81.01	Код модуля		
Доступ	RO	Диапазон	0 до 499
По умолчанию	Отсутствует	Скорость обновления	Отсутствует

Код модуля указывает тип модуля, установленного в соответствующее гнездо. Для модуля SM-Applications этот код равен **301**.

#81.02	Версия микропрограммы - Старшая		
Доступ	RO	Диапазон	00.00 до 99.99
По умолчанию	Отсутствует	Скорость обновления	Отсутствует

Указывает старший номер версии операционной системы в модуле SM-Applications. Совместно с параметром #81.51 образует полный номер версии.

#81.03	Состояние программы DPL		
Доступ	RO	Диапазон	0 до 3
По умолчанию	0	Скорость обновления	изменение через 1 мсек

Сообщает состояние выполнения программы DPL пользователя в модуле SM-Applications. Определены следующие значения состояния:

На дисплее	Значение	Описание
noPE	0	Нет программы DPL
StoP	1	Программа DPL остановлена
run	2	Программа DPL работает
triP	3	Ошибка времени выполнения. Работает задача ERROR или программа DPL остановлена

#81.04	Доступный ресурс системы		
Доступ	RO	Диапазон	0 до 100
По умолчанию	Отсутствует	Скорость обновления	200 мсек

Показывает ресурс свободного процессора в процентах к текущему времени фонового исполнения, вычисляемым за 200 мсек.

#81.05	Адрес RS485		
Доступ	RW	Диапазон	0 до 255
По умолчанию	11	Скорость обновления	Инициализация

Определяет адрес этого узла для протоколов передачи данных ANSI и Modbus. Для протокола ANSI диапазон адреса от 11 до 99, где первая цифра - это адрес группы, а вторая цифра - номер блока. Обе цифры должны быть в диапазоне 1-9. Нуль не допускается, так как он используется ведущим для адресации группы узлов.

Этот параметр не действует, если режим RS485 равен 25 (CTSunc) или 26 (CTSunc).

#81.06	Режим RS485 Mode		
Доступ	RW	Диапазон	0 до 255
По умолчанию	1	Скорость обновления	Инициализация

Определяет режим работы (или протокол) для расположенного на плате порта RS485. Описание этих режимов приведено в Таблице 6.1 *Режимы последовательной связи - параметр #81.06*.

#81.07	Скорость в бодах RS485		
Доступ	RW	Диапазон	0-9 (300-115200 bps)
По умолчанию	4 (4800)	Скорость обновления	Инициализация

Определяет скорость в Бодах (или в битах в секунд, bps) для расположенного на плате порта RS485. Поддерживаются следующие скорости:

На дисплее	Значение	Описание
300	0	300 bps
600	1	600 bps
1200	2	1200 bps
2400	3	2400 bps
4800	4	4800 bps
9600	5	9600 bps
19200	6	19200 bps
38400	7	38400 bps
57600	8	57600 bps
115200	9	115200 bps

Обратите внимание, что этот параметр игнорируется, когда режим порта RS485 настроен в 25 (ведущий CTSunc) или 26 (ведомый CTSunc).

#81.08	Задержка переключения RS485		
Доступ	RW	Диапазон	0 до 255 мсек
По умолчанию	2 мсек	Скорость обновления	Инициализация

Определяет фиксированную задержку между приемом сообщения портом RS485 и передачей ответа. Это может быть полезным в 2-проводных сетях, когда ведущему необходимо некоторое время для переключения из режима передачи в режим приема. Всегда имеется задержка не менее 1 мсек, а этот параметр позволяет удлинить ее.

#81.09	Задержка включения RS485 Tx		
Доступ	RW	Диапазон	0 до 1 мсек
По умолчанию	0 мсек	Скорость обновления	Инициализация

Этот параметр позволяет внести задержку 1 мсек между тем, когда SM-Applications включает передатчик RS485, и фактическим выполнением передачи. Он может потребоваться только если окажется, что получатель принимает искаженное начало передаваемых данных.

#81.10 Маршрутизация печати DPL			
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Инициализация

Управляет тем, куда посылается вывод по команде DPL PRINT. Если настроен в нуль (Откл), то выход посылается клиенту программирования (SYPT Workbench), а если настроен в 1 (Вкл), то он посылается на порт RS485.

#81.11 Такт распределения задачи (мсек)			
Доступ	RW	Диапазон	0 до 200 мсек
По умолчанию	10 мсек	Скорость обновления	Инициализация

Определяет период распределения (время такта) для задачи DPL CLOCK в миллисекундах. Нулевое значение отключает задачу CLOCK.

ПРИМЕЧ. До версии 01.05.00 Unidrive SP значение по умолчанию для этого параметра было 10 мсек.

#81.12 Период распределения задачи POS			
Доступ	RW	Диапазон	0 до 6
По умолчанию	0	Скорость обновления	Инициализация

Определяет период распределения (выполнения) задач POS для достижения нужной производительности приложения с учетом ресурсов, доступных для выполнения программы DPL пользователя. Имеются следующие значения:

На дисплее	Значение	Описание
diSAbled	0	Отключено
0.25	1	250 мсек
0.5	2	500 мсек
1	3	1 мсек
2	4	2 мсек
4	5	4 мсек
8	6	8 мсек

Настройте этот параметр, чтобы программа DPL пользователя автоматически запускалась при включении питания/сбросе. Если значение изменено и новая настройка должна использоваться при включении питания, то проверьте, что выполнено сохранение параметров **привода**.

ПРИМЕЧ. В Unidrive SP с версией программы 01.03.00 и меньше дисплей показывал не фактический период, а соответствующее ему число, например, 250 мсек показывалось на дисплее как 1.

#81.13	Разрешение автозапуска		
Доступ	RW	Диапазон	0/1
По умолчанию	1	Скорость обновления	Инициализация

#81.14	Разрешение глобального отключения во время выполнения		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Инициализация

Настройка этого параметра в 1 заставляет Unidrive SP выполнить отключение при появлении определенных ошибок времени выполнения в модуле SM-Applications / программе DPL пользователя.

Дополнительная информация приведена в разделе 11.1 *Ошибки времени выполнения*.

#81.15	Запрет сброса при очистке отключения		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Инициализация

Если этот параметр равен 0, то модуль будет сброшен при очистке отключения привода. Если параметр настроен в 1, то модуль не реагирует на сброс отключения привода (то есть продолжает работать).

#81.16	Скорость обновления данных энкодера		
Доступ	RW	Диапазон	0-3
По умолчанию	0	Скорость обновления	Инициализация

Если этот параметр равен 0, то данные APC и параметры энкодера в меню 90 обновляются каждые 250 мксек.

Если этот параметр равен 1, то данные APC и параметры энкодера в меню 90 обновляются непосредственно перед выполнением каждой задачи POS.

Если этот параметр равен 2, то данные APC и параметры энкодера в меню 90 обновляются непосредственно перед выполнением каждой задачи CLOCK.

Если параметр равен 3, то данные APC и параметры энкодера в меню 90 никогда не обновляются. Поскольку они не обновляются, то освобождается больше ресурсов процессора.

#81.17	Разрешить отключения по превышению диапазона параметров		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Инициализация

Определяет действие, выполняемое когда программа DPL пользователя пытается записать в параметр недопустимое значение. Если настроен в 1, то происходит отключение по ошибке времени выполнения (номер 44); если настроен в 0, то значение будет автоматически ограничено предельным значением параметра

#81.18	Разрешение сторожевого таймера		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Инициализация

При установке включает в программе DPL функцию сторожевого таймера. Тогда команда DPL WDOG должна выполняться каждые 200 мсек. Это можно использовать для защиты программы от сбоя. Если команда не выполнена в течение 200 мсек, то привод выполнит отключение **SLx.tO**. Обратите внимание, что команда WDOG должна также быть выполнена для включения сторожевого таймера. Обычно она выполняется в конце задачи инициализации.

#81.19	Запрос сохранения		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	100 мсек

Настройка этого параметра в 1 вызывает немедленное сохранение всех энергонезависимых данных модуля SM-Applications. Они состоят из наборов регистров P/Q/T/U PLC и опционно из меню 20 (зависит от настройки #81.21).

ПРИМЕЧ. Обратите внимание, что это также вызывает сброс модуля и этот параметр автоматически возвращается в 0. Он также сбрасывается при выполнении отключения Unidrive SP. Меню 81 не сохраняется.

#81.20	Разрешить сохранение по отключению "UU"		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Немедленно

Настройка этого параметра в 1 означает, что все энергонезависимые данные модуля SM-Applications будут автоматически сохранены перед выполнением отключения питания привода.

ПРИМЕЧ. Обратите внимание, что при выполнении сохранения 'UU' модуль SM-Applications будет сброшен.

#81.21	Разрешение сохранения и восстановления меню 20		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Немедленно

Если настроен в 1, то меню 20 будет сохраняться/восстанавливаться вместе с другими энергонезависимыми параметрами при запросе сохранения (#x.19=1) и при сохранении по отключению питания (#x.20=1). Если меню 20 должно быть восстановлено при включении питания, то пользователь должен обеспечить сохранение этого параметра в приводе перед отключением питания.

Поскольку меню 20 - это глобальное меню привода, то для сохранения и восстановления меню 20 следует использовать только один модуль, установленный в Unidrive SP. Поэтому если в приводе установлены несколько модулей SM-Applications, то этот параметр должен быть установлен **только в одном** из них, иначе меню 20 не будет правильно восстановлено при включении питания.

ПРИМЕЧ. В отличие от других параметров настройки параметры #81.20 и #82.21 **не кэшируются**, что означает, что все изменения параметра немедленно вступают в силу.

#81.22	ID маркерного кольца CTNet		
Доступ	RW	Диапазон	0 до 255
По умолчанию	0	Скорость обновления	Инициализация

Этот параметр позволяет пользователю указать идентификатор кольцевой сети CTNet, к которой подключен модуль SM-Applications. Если в системе имеется только одна кольцевая сеть, то этот параметр можно оставить в его значении по умолчанию. Если в системе используются несколько кольцевых сетей, то каждой сети надо назначить свой идентификатор. Комбинация идентификатора CTNet Token Ring ID и адреса узла CTNet должна быть уникальной

#81.23	Адрес узла CTNet		
Доступ	RW	Диапазон	0 до 255
По умолчанию	0	Скорость обновления	Инициализация

Определяет адрес узла для CTNet. Каждый узел в CTNet должен иметь уникальный адрес, настройка его в нуль отключает сеть CTNet на этом узле.

#81.24	Скорость CTNet в Бодах		
Доступ	RW	Диапазон	0 до 3
По умолчанию	1(2.5)	Скорость обновления	Инициализация

Указывает скорость передачи данных в CTNet. Все узлы сети должны быть настроены на одинаковую скорость. Скорости определяются так:

На дисплее	Значение	Описание
5.000	0	5 Мбит/с
2.500	1	2.5 Мбит/с
1.250	2	1.25 Мбит/с
0.625	3	625 кбит/с

#81.25	Настройка синхронизации CTNet		
Доступ	RW	Диапазон	0 до 9999
По умолчанию	0	Скорость обновления	Инициализация

Указывает скорость создания сообщений синхронизации в CTNet. Это сообщение указывает всем узлам, когда передавать циклические данные. Этот параметр должен быть настроен только в одном узле сети CTNet.

Этот параметр обновления имеет формат SSFF, где FF определяет скорость обновления канала быстрых циклических данных, а SS определяет скорость медленных циклических данных как кратное от FF. Поэтому если параметр имеет значение 1510, то быстрые циклические данные устанавливаются каждые 10 мсек, а медленные - каждые 150 мсек. При использовании простого режима (смотрите ниже) нужно настроить только FF (скорость быстрого цикла).

#81.26, #81.28, #81.30	Параметры настройки простого режима CTNet		
Доступ	RW	Диапазон	0 до 25503
По умолчанию	0	Скорость обновления	Инициализация

#81.27, #81.29, #81.31- #81.34	Параметры настройки простого режима CTNet		
Доступ	RW	Диапазон	0 до 9999
По умолчанию	0	Скорость обновления	Инициализация

Эти параметры определяют источник и назначения для циклических данных простого режима CTNet.

Параметр	Формат	Канал	Описание
#81.26	NNSS	1	Определяет номер и гнездо узла назначения NNN = Номер узла(1-255) SS = Номер гнезда (1-3) например, значение 201 означает узел ID 2, гнездо 1.
#81.27	MMPP	1	Определяет параметр-источник привода, который должен передаваться MM = Номер меню PP = Номер параметра например, значение 302 означает #3.02 (скорость)
#81.28	NNSS	2	Номер и гнездо узла назначения для канала 2
#81.29	MMPP	2	Параметр-источник привода для канала 2
#81.30	NNSS	3	Номер и гнездо узла назначения для канала 3
#81.31	MMPP	3	Параметр-источник привода для канала 3
#81.32	MMPP	1	Параметр назначения гнезда 1 для входящих данных
#81.33	MMPP	2	Параметр назначения гнезда 2 для входящих данных
#81.34	MMPP	3	Параметр назначения гнезда 3 для входящих данных

Более подробная информация о данных простого режима сети CTNet приведена в Руководстве пользователя CTNet.

#81.35	ID задачи события синхро CTNet		
Доступ	RW	Диапазон	0 до 4
По умолчанию	0	Скорость обновления	Инициализация

Указывает, какая из задач EVENT будет запущена при приеме или создании сообщения синхронизации CTNet. Это синхро сообщение вырабатывается ведущим узлом (который может быть этим узлом) в сети CTNet через постоянные интервалы времени. Определены следующие состояния:

На дисплее	Значение	Описание
Disabled	0	Не запланировано никаких задач событий
Event	1	Запланирована задача EVENT
Event1	2	Запланирована задача EVENT1
Event2	3	Запланирована задача EVENT2
Event3	4	Запланирована задача EVENT3

#81.36	Диагностика CTNet		
Доступ	RO	Диапазон	-3 до 32767
По умолчанию	Отсутствует	Скорость обновления	1 секунда

Состояние сети CTNet отображается в параметре диагностики CTNet. Если модуль SM-Applications успешно обменивается данными с сетью CTNet, то в параметре отображается число сообщений за одну секунду.

Таблица 5.1 Диагностика CTNet

#MM.36	Состояние	Описание
>0	Сеть исправна	Указывает количество сообщений, которые обрабатываются в каждую секунду.
0	Сеть исправна, передачи данных нет	Была установлена кольцевая сеть нижнего уровня и она активна, но узел не принимает никаких сообщений данных CTNet.
-1	RECON	Было обнаружено изменение конфигурации сети.
-2	Ошибка инициализации	Модуль SM-Applications не может сконфигурировать интерфейс CTNet. Проверьте, что адрес узла и скорость передачи настроены правильно.
-3	MYRECON	Модуль SM-Applications вызвал повторное конфигурирование сети CTNet

#81.37	Запрет загрузки, если привод разрешен		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Инициализация

Если этот параметр установлен и пользователь пытается загрузить в этот модуль новую программу пользователя DPL или операционную систему и привод разрешен, то загрузка будет запрещена и произойдет отключение по ошибке времени выполнения 70, если установлен параметр глобального отключения по ошибке времени выполнения (#81.14).

Поскольку загрузка останавливает нормальную работу модуля, можно считать опасным выполнение загрузки при работе системы привода. Таким образом, настройка этого параметра предотвращает опасную загрузку в таких условиях.

#81.38	Отключение во время выполнения APC		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Инициализация

Если этот параметр равен 0, то при возникновении невосстановимой ошибки APC (например, использование неинициализированной функции SAM) привод выполнит отключение с ошибкой времени выполнения 81. Если этот параметр равен 1, то привод не отключится при возникновении невосстановимой ошибки APC.

#81.39	Состояние синхронизации между SM-Applications и приводом		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	NA

Этот параметр показывает состояние синхронизации текущего модуля. Он указывает, является или нет модуль ведущим устройством синхронизации.

Состояние синхронизации	Состояние
0	Запрос ведущего синхронизации равен 0 и ведущим устройством синхронизации является другой модуль.
1	Дополнительный модуль является ведущим синхронизации.
3	Дополнительный модуль является ведущим синхронизации, но частота синхронизации отсутствует или не в диапазоне.

#81.42	Фиксация положения энкодера обратной связи		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Инициализация

Если в модуле SM-Applications возникает фиксация, то положение энкодера обратной связи и счетчика оборотов можно запомнить, если этот параметр равен 1. Они запоминаются в параметрах #90.19 и #90.20 (смотрите раздел 5.8 *Меню 90 - общие параметры*). Более подробные сведения о входе фиксации Freeze приведены в главе 8 *Фиксация и маркер в SM-Applications*.

#81.43	Инверсия входа фиксации		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Инициализация

Если этот параметр равен 0, то фиксация происходит по нарастающему фронту на входе DIGIN0 модуля SM-Applications. Если он равен 1, то фиксация происходит по спадающему фронту на входе DIGIN0. Более подробные сведения о входе фиксации Freeze приведены в главе 8 *Фиксация и маркер в SM-Applications*.

#81.44	Уровень приоритета задачи		
Доступ	RW	Диапазон	0 до 255
По умолчанию	0	Скорость обновления	Инициализация

Этот параметр может изменять уровни приоритета разных задач. Доступ к параметру проводится побитно (поразрядно):

Бит	Значение
0	Значение 0 дает задачам такой же уровень приоритета, как в UD70. Настройка значения в 1 снижает приоритет CTNet ниже приоритета задач POS0 и POS1. При этом снижаются ошибки синхронизации задач POS, но может возникнуть нехватка ресурсов для задачи CTNet.
1	Значение 0 означает, что задача связи между модулями имеет приоритет выше, чем у задач POS. Настройка этого значения в 1 делает приоритет задач связи между модулями ниже, чем у задач POS.

#81.48	Номер строки с ошибкой		
Доступ	RO	Диапазон	32 бита
По умолчанию	0	Скорость обновления	По ошибке

Указывает номер строки программы DPL, которая вызвала ошибку времени выполнения. Это справедливо только если:

- Программа пользователя была скомпилирована с опцией *отладки*
- Ошибка такая, которую может вызвать код программы пользователя, например, деление на 0 (50) или параметр не существует (41).

Если не выполняется одно из этих условий, то параметр номера строки с ошибкой показывает нуль (0).

#81.49	ID программы пользователя		
Доступ	RO/RW	Диапазон	Знаковый 16-битный
По умолчанию	0	Скорость обновления	Смотрите Примечание

Этот параметр позволяет пользователю ввести код ID в свою программу. Это может, например, быть номер версии программы. Для записи этого параметра используйте функциональный блок SETUSERID().

#81.50	Код ошибки времени выполнения		
Доступ	RO	Диапазон	0 до 255
По умолчанию	0	Скорость обновления	По ошибке

Если происходит ошибка времени выполнения, то ее номер помещается в этот параметр.

Более подробные сведения приведены в главе 11.1 *Ошибки времени выполнения*.

#81.51	Микропрограмма - младший номер версии		
Доступ	RO	Диапазон	0 до 99
По умолчанию	Отсутствует	Скорость обновления	Отсутствует

Указывает младший номер версии операционной системы в модуле SM-Applications. Используется вместе с #81.02 для образования полного номера версии микропрограммы.

5.4 Меню 70-75 - регистры ПЛК

Эти меню предоставляют доступ к регистрам ПЛК. Регистры ПЛК - это знаковые 32-битные целые переменные, доступные для программ пользователя и для передачи данных по CTNet.

Регистры ПЛК разделены в 6 наборов по 100 параметров, пронумерованных от 00 до 99. К регистрам можно также получить доступ из программы DPL пользователя с помощью специального имени переменной или имени массива.

Номер меню	Доступ	Переменная DPL (x=номер регистра)	Число регистров	Описание
70	RW	_Px%, _P%[x]	100	Общего назначения. Сохраняемые.
71	RW	_Qx%, _Q%[x]	100	Общего назначения. Сохраняемые.
72	RW	_Rx%, _R%[x]	100	Используются для исходящих каналов циклических данных CTNet. Несохраниваемые.
73	RW	_Sx%, _S%[x]	100	Используются для входящих каналов циклических данных CTNet. Несохраниваемые
74	RW	_Tx%, _T%[x]	100	Общего назначения. Сохраняемые.
75	RW	_Ux%, _U%[x]	100	Общего назначения. Сохраняемые.

В таблице выше вы видите, что каждый параметр внутри меню 70 до 75 имеет эквивалентную переменную DPL. Это значит, что вы можете использовать любой формат для доступа к параметрам внутри этих меню.

например, **#72.01=1** эквивалентно **_R01%=1**, **#75.65=66** будет эквивалентно **_U65%=66** и т.д.

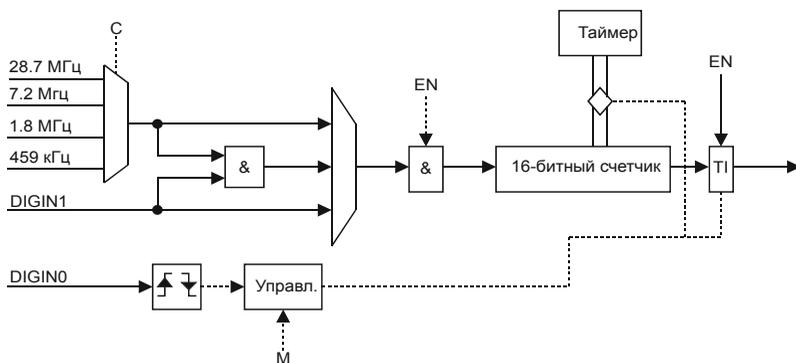
Все меню 70, 71, 74 и 75 можно сохранять в энергонезависимой флэш-памяти по запросу или автоматически, если привод отключается по падению напряжения (Смотрите раздел 5.2 *Сохранение параметров*, где это описано подробнее).

Меню 72 и 73 используются для передачи циклических данных CTNet, но если эта функция не используется, то эти регистры можно использовать для любых других целей. Однако этого следует избегать, если в дальнейшем предполагается использовать циклические данные.

5.5 Меню 85 - параметры функции таймера

В модуль SM-Applications встроен аппаратный счетчик, который имеет следующие функции:

- 16-разрядный инкрементный счетчик.
- Скорость счета задается внутренним синхрогенератором. Делитель синхроимпульсов настраивается как частота/1, частота/4, частота/16, частота/64.
- Скорость счета задается от внешних синхроимпульсов с цифрового входа DIGIN1. Максимальная скорость счета составляет 600 кГц.
- Таймер можно использовать для планирования одной из 4 задач DPL Event при переполнении счетчика или при сигнале фиксации на входе DIGIN1.
- Пользователь может выбрать значение переполнения счетчика во всем полном 16-разрядном диапазоне.
- Таймер можно настроить на кэширование счета по нарастающему или спадающему фронту на входе DIGIN0.



#85.01	Слово управления		
Доступ	RW	Диапазон	13 бит
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Таблица 5.2 Слово управления - параметр #85.01

Бит	Символ	Функция
b0-b2	TE	Планирование задачи таймера EVENT, если флаг TI установлен: 0=Нет планирования задач Event 1=Запланирована задача Event 2=Запланирована задача Even1 3=Запланирована задача Even2 4=Запланирована задача Even3
b3	EN	Включение таймера: 0=Таймер отключен 1=Таймер включен

Таблица 5.2 Слово управления - параметр #85.01

Бит	Символ	Функция
b4	CS	Источник синхроимпульсов: 0=внутренний синхрогенератор 1=внешние синхроимпульсы со входа DIGIN1.
b5-b6	C	Выбор делителя частоты внутренних синхроимпульсов (игнорируется при выборе внешних синхроимпульсов): 0=частота /1 (28.7 МГц) 1=частота /4 2=частота /16 3=частота /64
b7-b8	M	Режим таймера: <u>Режим 0 (00): Режим свободного счета</u> Счетчик считает выбранные синхроимпульсы. Флаг TI устанавливается при "прокручивании" счетчика. <u>Режим 1 (01): Режим захвата 1</u> Счетчик считает выбранные синхроимпульсы. Нарастающий фронт на входе DIGIN0 зашелкивает текущее содержимое счетчика в параметре КЭШ ЗАХВАТА ТАЙМЕРА и устанавливается флаг TI. Затем счетчик продолжает увеличиваться (TI не устанавливается при "прокрутке" счетчика). <u>Режим 2 (10): Режим захвата 2</u> Счетчик считает выбранные синхроимпульсы. Спадающий фронт на входе DIGIN0 зашелкивает текущее содержимое счетчика в параметре КЭШ ЗАХВАТА ТАЙМЕРА и устанавливается флаг TI. Затем счетчик продолжает увеличиваться (TI не устанавливается при "прокрутке" счетчика).

#85.02	Слово состояния		
Доступ	RO	Диапазон	0 до 3
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Таблица 5.3 Слово состояния - параметр #85.02

Бит	Символ	Функция
b0	TI	Флаг события таймера: 0=Не было никаких событий 1=Возникло событие (смотрите описание параметра #85.01) Примечание: Бит TI автоматически сбрасывается операционной системой, если флаг TE в #85.01 имеет ненулевое значение. В противном случае он сбрасывается при чтении слова состояния.
b1	OV	Флаг прокручивания счетчика при переполнении 0=Прокрутки счетчика не происходит 1=Происходит прокрутка счетчика (возврат в 0) Примечание: Этот флаг верен для ВСЕХ режимов счетчика и автоматически очищается при чтении регистра состояния.

#85.03	16-битовый счетчик таймера		
Доступ	RW	Диапазон	16 бит
По умолчанию	Отсутствует	Скорость обновления	Немедленно

С помощью того параметра можно в любой момент времени прочитать и записать текущее значение счетчика.

#85.04	Предел прокрутки		
Доступ	RW	Диапазон	16 бит
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Этот параметр указывает значение, при котором выполняется прокрутка параметра #85.03.



Если вы настроите малое значение предела прокрутки при высокой частоте счета, то модуль SM-Applications может “зависать”. Это вызывается тем, что непрерывно обслуживаются прерывания прокрутки и не остается ресурсов процессора для обслуживания других функций. Если это произойдет, то пользователь может сбросить модуль SM-Applications за счет выполнения сброса привода 1070. Если программа DPL настраивает параметры таймера, то перед сбросом следует запретить автозапуск программы (#17.13=0). После сброса пользователь может инициализировать таймер правильным значением предела прокрутки.

#85.05	Кэш захвата таймера		
Доступ	RW	Диапазон	16 бит
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Этот параметр хранит захваченное значение при использовании таймера в режимах 1 или 2 (режимы захвата).

5.6 Меню 86 - параметры цифровых входов-выходов

Модуль SM-Applications оснащен двумя цифровыми входами и двумя цифровыми выходами. Эти входы и выходы управляются из этого меню.

#86.01	Цифровой вход 0		
Доступ	RO	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	Немедленно

#86.02	Цифровой вход 1		
Доступ	RO	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Эти два параметра позволяют считать состояние цифровых входов 0 и 1.

Неактивный вход (низкий уровень) дает значение 0, а активный вход (высокий уровень) дает значение 1.

#86.03	Цифровой выход 0		
Доступ	RW	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	Немедленно

#86.04	Цифровой выход 1		
Доступ	RW	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Эти два параметра управляют цифровыми выходами 0 и 1. Настройка в 0 переводит выход в низкий уровень, а настройка в 1 - в высокий (+24 В).

#86.05	Цифровые выходы 0 и 1		
Доступ	RW	Диапазон	0 до 3
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Этот параметр обеспечивает управление обоими цифровыми выходами и является альтернативой отдельного управления каждым выходом с помощью #86.03/4. Бит 0 управляет цифровым выходом 0 (#86.03), а бит 1 управляет цифровым выходом 1 (#86.04).

5.7 Меню 88 - параметры состояния

#88.01	Код ошибки / Сброс		
Доступ	RW	Диапазон	0 до 9999
По умолчанию	Отсутствует	Скорость обновления	По ошибке

Этот параметр имеет две функции - при чтении он возвращает код ошибки времени выполнения, аналогичный #81.50 (обратите внимание - он не возвращает коды отключения привода). Этот параметр сбрасывается в нуль при сбросе и при запуске выполнения программы пользователя.

Если в этот параметр записать значение 1070, то модуль SM-Applications запустит “теплый” перезапуск привода и других модулей. Это можно использовать для перезапуска программы пользователя (при условии, что автозапуск #81.13=1) и для очистки любого отключения привода. Такую операцию сброса можно выполнить в любое время, а не только после ошибки времени выполнения или в задаче ERROR.



Запись значения 1070 в параметр #88.01 приводит к автоматической очистке любого отключения привода, а также к сбросу всех дополнительных модулей, установленных в Unidrive SP. Это поведение отличается от поведения изделия UD70 на Unidrive 1, когда сброс привода не выполнялся.

#88.02	Ошибка в задаче		
Доступ	RO	Диапазон	0 до 50
По умолчанию	Отсутствует	Скорость обновления	При ошибке

Параметр Ошибка в задаче можно использовать для того, чтобы определить, в какой задаче возникла ошибка. Этот параметр имеет верное значение, только если он читается из задачи ERROR после отключения по ошибке времени выполнения. Величины этого параметра означают следующее:

Величина	Задача
50	Системная System
1	Начальная Initia
2	Фоновая Background
3	Clock
4	Error
5	Pos0
6	Pos1

Величина	Задача
7	Event
8	Event1
9	Event2
10	Event3

Если отсутствует условие ошибки, то возвращается значение 0.

Более подробная информация об этих параметрах приведена в главе 11 *Диагностика* на стр. 97.

5.8 Меню 90 - общие параметры

Это меню содержит заданные значения и значения обратной связи от привода, а также другую информацию о состоянии.

ПРИМЕЧ. При переносе программ UD70 в модуль SM-Applications нужно соблюдать осторожность, так как эти параметры отличаются от аналогичных в UD70.

#90.01	Положение энкодера обратной связи (2^{32} /оборот)		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Содержит положение энкодера обратной связи.

Старшие 16 бит масштабируются в 65536 отсчетов на оборот независимо от типа датчика обратной связи или настройки масштабирования в приводе. Младшие 16 бит дают точное положение, выдаваемое датчиком обратной связи, масштабированное на 65536. Для стандартных энкодеров это значение обычно равно нулю, но высокоточные датчики, например, энкодеры SinCos, могут обеспечить такую дополнительную точность.

Импульсы маркера и т.п. не влияют на этот параметр.



Более подробные сведения по использованию этих параметров обратной связи приведены в системе оперативной справки в SYPT Workbench.

#90.02	Счетчик оборотов энкодера обратной связи		
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Содержит счетчик оборотов энкодера обратной связи.

#90.03	Положение опорного энкодера (2^{32} /оборот)		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Содержит заданное положение энкодера.

Старшие 16 бит масштабируются в 65536 отсчетов на оборот независимо от типа датчика обратной связи или настройки масштабирования в приводе. Младшие 16 бит дают точное положение, выдаваемое датчиком обратной связи, масштабированное на 65536. Для стандартных энкодеров это значение обычно равно нулю, то высокоточные датчики, например, энкодеры SinCos, могут обеспечить такую дополнительную точность.

Импульсы маркера и т.п. не влияют на этот параметр.

#90.04 Счетчик оборотов опорного энкодера			
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Содержит счетчик оборотов опорного энкодера.

#90.10 Режим привода			
Доступ	RO	Диапазон	Знаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Предоставляет однозначный метод определения режима работы привода Unidrive SP. Рекомендуется использовать этот параметр вместо #11.31 или #0.48, поскольку эти параметры указывают запрошенный, а не фактический режим.

Определены следующие значения режима:

Величина	Режим
26	Разомкнутый контур
27	Векторный в замкнутом контуре
28	Серво
29	Рекуперация
20	Commander SE

Для программирования изменения режима работы привода используйте функциональные блоки MODEXFER или CMODEXFER.

#90.11 Состояние привода и слово управления			
Доступ	RW	Диапазон	Знаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	Немедленно

При записи в этот параметр обновляется слово состояния. При чтении этого параметра считывается значение слова состояния (аналогично параметру #10.40).

Таблица 5.4 Слово управления

Бит	Описание
b15	Если установлен, то значение #1.46 копируется из b6
b14	Если установлен, то значение #1.45 копируется из b5
b13	Устанавливает значение #18.33 (Меню приложения 1, бит 3)
b12	Если установлен, то значение #6.32 копируется из b3
b11	Если установлен, то значение #6.31 копируется из b2

Таблица 5.4 Слово управления

Бит	Описание
b10	Если установлен, то значение #6.30 копируется из b1
b9	Если установлен, то значение #6.15 копируется из b0
b8	Устанавливает значение #18.32 (Меню приложения 1, бит 2)
b7	Устанавливает значение #18.31 (Меню приложения 1, бит 1)
b6	Устанавливает значение #1.46 (бит выбора предустановки 1)
b5	Устанавливает значение #1.45 (бит выбора предустановки 0)
b4	Отключение пользователя. Если установлен, то сразу отключает привод.
b3	Устанавливает значение #6.32 (Бит последовательности 2: Назад)
b2	Устанавливает значение #6.31 (Бит последовательности 1: Толчки)
b1	Устанавливает значение #6.30 (Бит последовательности 0: Вперед)
b0	Устанавливает значение #6.15 (Разрешение привода)

Таблица 5.5 Слово состояния

Бит	Описание
b15	Не используется
b14	#10.15 (Отказ силового питания)
b13	#10.14 (Работа по направлению)
b12	#10.13 (Дана команда направления)
b11	#10.12 (Тревога тормозного резистора)
b10	#10.11 (Активен тормозной IGBT)
b9	#10.10 (Рекуперация)
b8	#10.09 (Выход привода на пределе тока)
b7	#10.08 (Достигнута нагрузка)
b6	#10.07 (Выше заданной скорости)
b5	#10.06 (На скорости)
b4	#10.05 (Ниже заданной скорости)
b3	#10.04 (Работа на или ниже минимальной скорости)
b2	#10.03 (Нулевая скорость)
b1	#10.02 (Привод работает)
b0	#10.01 (Привод исправен)

#90.12 Причина планировки задачи Event			
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	По Event

#90.13 Причина планировки задачи Event1			
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	По Event1

#90.14 Причина планировки задачи Event2			
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	По Event2

#90.15 Причина планировки задачи Event3			
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	По Event3

Эти четыре параметра указывают причину, по которой была запущена (запланирована) данная конкретная задача EVENT. Значение имеет смысл только при запущенной конкретной задаче EVENT.

Отдельные биты величины параметра имеют следующие значения:

Биты	Описание
0-1	Запустившее задачу гнездо 0 = Локальное гнездо 1 = Гнездо 1 2 = Гнездо 2 3 = Гнездо 3
2-7	Причина запуска. 0-31 = Инициирован другой дополнит. модуль 32 = Синхронизация CTNet 33 = Блок таймера 34-63 = Определенная пользователем причина по команде DPL SCHEDULEEVENT.

#90.18 Флаг фиксации энкодера обратной связи			
Доступ	RW	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Для захвата фиксируемого положения этот параметр надо сбросить в 0. После фиксации этот параметр становится = 1. Для повторной активации просто сбросьте его в 0.

#90.19	Фиксация положения энкодера обратной связи		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

#90.20	Фиксация оборотов энкодера обратной связи		
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Эти 2 параметра хранят положение и обороты энкодера обратной связи в момент активации входа фиксации.

#90.21	Отключить проверку положения энкодера привода		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Немедленно

Привод регулярно проверяет положение, получаемое по синусным и косинусным волнам с энкодера SINCOS по каналу последовательной связи. Настройте этот параметр в 1 для отключения этой проверки.

#90.22	Регистр передачи порта энкодера привода		
Доступ	RW	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Если параметр Проверка положения энкодера привода отключен (#90.21=1), то этот параметр можно использовать для связи с энкодером, подключенном к приводу через последовательный порт.

#90.23	Регистр приема порта энкодера привода		
Доступ	RW	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Если параметр Проверка положения энкодера привода отключен (#90.21=1), то этот параметр можно использовать для связи с энкодером, подключенном к приводу через последовательный порт.

#90.24	Номер гнезда SM-Applications		
Доступ	RO	Диапазон	Беззнаковый 8-битный
По умолчанию	Отсутствует	Скорость обновления	Инициализация

Этот параметр сообщает номер гнезда, в котором установлен модуль SM-Applications.

#90.25	Положение маркера энкодера обратной связи (2^{32}/оборот)		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Старшие 16 бит масштабируются в 65536 отсчетов на оборот независимо от типа датчика обратной связи или настройки масштабирования в приводе.

#90.26	Обороты маркера энкодера обратной связи (2^{16}/оборот)		
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Этот параметр выдает счетчик оборотов маркера энкодера обратной связи.

#90.27	Номер версии базы данных SM-Applications		
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	Инициализация

Номер версии базы данных считывается из базы данных после включения питания.

#90.28	Флаг фиксации опорного энкодера		
Доступ	RW	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Для захвата фиксируемого положения этот параметр надо сбросить в 0. После фиксации этот параметр становится = 1. Для повторной активации просто сбросьте его в 0.

#90.29	Фиксация положения опорного энкодера		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

#90.30	Фиксация оборотов опорного энкодера		
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Эти 2 параметра хранят соответственно положение и обороты опорного энкодера в момент активации входа фиксации.

#90.31	Обороты и грубое положение энкодера обратной связи		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

#90.32	Обороты и грубое положение опорного энкодера		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Эти два параметра хранят 16-битовые обороты в старшем слове и 16-битовое положение в младшем слове для энкодера обратной связи (#90.31) и опорного энкодера (#90.32).

#90.33	Фиксация оборотов и грубого положения энкодера обратной связи		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

#90.34	Фиксация оборотов и грубого положения опорного энкодера		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Эти два параметра хранят 16-битные обороты в старшем слове и 16-битное положение в младшем слове в момент активации входа фиксации.

#90.35	Положение маркера опорного энкодера (2^{32}/оборот)		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Этот параметр хранит значение положения опорного энкодера в момент активации импульса маркера.

#90.36	Обороты маркера опорного энкодера (2 ¹⁶ /оборот)		
Доступ	RO	Диапазон	Беззнаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Этот параметр хранит значение счетчика оборотов опорного энкодера в момент активации импульса маркера.

#90.37	Обороты и грубое положение маркера энкодера обратной связи		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

#90.38	Обороты и грубое положение маркера опорного энкодера		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Эти два параметра хранят 16-битовые обороты в старшем слове и 16-битовое положение в младшем слове для энкодера обратной связи (#90.37) и опорного энкодера (#90.38) в момент активации импульса маркера.

#90.39	Состояние кнопки панели привода		
Доступ	RO	Диапазон	Знаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	> 40 мсек

#90.40	Запуск задачи Event		
Доступ	WO	Диапазон	Беззнаковый 16-битный
По умолчанию	0	Скорость обновления	Немедленно

После настройки этого параметра в значение он запустит выполнение одной из задач Event модуля SM-Applications.

Величина	Действие
0	Не запускать задачу Event
1	Запуск задачи Event
2	Запуск задачи Event1
3	Запуск задачи Event2
4	Запуск задачи Event3

#90.41	Флаг маркера опорного энкодера		
Доступ	RW	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

#90.42	Флаг маркера энкодера обратной связи		
Доступ	RW	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	Смотрите #81.16

Эти два параметра устанавливаются в 1 при активации импульса маркера соответствующего энкодера, но только в том случае, если были настроены параметры разрешения флага маркера (параметры #90.45 и #90.46). Для повторного срабатывания по маркеру пользователь должен сбросить эти параметры в нуль. Пользователь не может установить эти параметры в 1.

#90.43	Источник опорного энкодера		
Доступ	RW	Диапазон	Беззнаковый 8-битный
По умолчанию	Отсутствует	Скорость обновления	Немедленно

#90.44	Источник энкодера обратной связи		
Доступ	RW	Диапазон	Беззнаковый 8-битный
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Эти два параметра определяют источник для опорных (заданных) данных и данных обратной связи. Допустимые источники указаны в таблице ниже.

Величина	Описание
0	Энкодер привода
1	Гнездо 1
2	Гнездо 2
3	Гнездо 3
4	Программа пользователя
5	Не настроен

#90.45	Разрешение флага опорного маркера		
Доступ	RW	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	Немедленно

#90.46	Разрешение флага маркера обратной связи		
Доступ	RW	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Эти два параметра необходимо установить в 1, чтобы разрешить установку флагов маркеров (параметры #90.41 и #90.42) при активации импульса маркера.

#90.47	Разрешение фиксации опорного		
Доступ	RW	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	Немедленно

#90.48	Разрешение фиксации обратной связи		
Доступ	RW	Диапазон	0/1
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Эти два параметра необходимо установить в 1, чтобы разрешить установку флагов фиксации (параметры #90.18 и #90.28) при активации входа фиксации.

5.9 Меню 91 - параметры быстрого доступа

Параметры в этом меню являются виртуальными параметрами модуля SM-Applications, которые обеспечивают более быструю скорость обновления или улучшенное разрешение по сравнению с параметрами привода.

#91.01	Разрешение ссылки		
Доступ	RW	Диапазон	Беззнаковый 8-битный
По умолчанию	0	Скорость обновления	Немедленно

Этот параметр разрешает использование ссылочных параметров, описанных ниже в этом разделе. Вы должны установить в этом параметре соответствующий бит. Смотрите следующую таблицу.

Бит	Функция	Соответствующий параметр
0	Разрешение ссылки задания скорости	#91.02
1	Разрешение ссылки задания жесткой скорости	#91.03
2	Разрешение ссылки задания момента	#91.04
3	Аналоговый выход 1	-
4	Аналоговый выход 2	-
5	Зарезервировано	-
6	Зарезервировано	-
7	Зарезервировано	-

#91.02	Задание скорости (#1.21)		
Доступ	RW	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Настраивает задание скорости в единицах **0.001 об/мин**. Это значение зеркально повторяется в параметре привода #1.21 (предустановленное задание скорости 1), поэтому для управления приводом с помощью этого параметра надо убедиться, что в приводе выбрана предустановленная скорость 1 (#1.14=3, #1.15=1).

При использовании этого параметра проверьте, что в #91.01 установлен бит 0 и что соответственно настроена скорость полной шкалы в #91.05. Обратите внимание, что этот параметр действует только в векторном режиме замкнутого контура и в режиме Серво.

#91.03	Задание жесткой скорости (#3.22)		
Доступ	RW	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Управляет заданием жесткой скорости привода в единицах **0.001 об/мин**.

При использовании этого параметра проверьте, что в #91.01 установлен бит 1 и что соответственно настроена скорость полной шкалы в #91.05. Обратите внимание, что этот параметр действует только в векторном режиме замкнутого контура и в режиме Серво.

#91.04	Заданное значение момента (#4.08)		
Доступ	RW	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Указывает уставку момента (параметр привода #4.08) в единицах 0.01%.

Для использования этого параметра проверьте, что в #91.01 установлен бит 2.

#91.05	Скорость полной шкалы (об/мин)		
Доступ	RW	Диапазон	Знаковый 32-битный
По умолчанию	1500	Скорость обновления	Отсутствует

Настройте его в максимальную (абсолютную) скорость, которая когда-либо будет записана в #91.02 или #91.03. Задается в единицах 1 об/мин.

Это задает ограничение на значения скорости, посылаемые в привод. При попытке записать в #91.02 или #91.03 значения скорости, большие чем указанное в #91.05 значение в об/мин произойдет ограничение скорости или ошибка времени выполнения по превышению диапазона.

#91.06	Обратная связь по скорости		
Доступ	RO	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Этот параметр возвращает значение обратной связи по скорости в приводе Unidrive SP в единицах 0,01 об/мин в режимах замкнутого контура. Однако при использовании энкодера низкого разрешения может быть некоторая нестабильность на низкой скорости. Например, при скорости 10 об/мин с энкодером 1024 имп/об этот параметр может прыгать между 0 и 14.65 об/мин. Он аналогичен параметру Unidrive SP #3.02.

#91.07		Обратная связь по току (#4.02)	
Доступ	RO	Диапазон	Знаковый 16-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Это значение берется из параметра #4.02 и выражается в единицах 0,01 А (то есть 150=1,5 Ампер).

#91.08		Значение аналогового входа 1 привода	
Доступ	RO	Диапазон	±4000
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Это значение берется из аналогового входа 1 привода и масштабируется до ±4000 для представления +/- полной шкалы входного сигнала. Сведения о частоте опроса аналоговых входов приведены в Руководстве пользователя привода.

#91.09		Значение аналогового входа 2 привода	
Доступ	RO	Диапазон	±1000
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Это значение берется из аналогового входа 2 привода и масштабируется до ±1000 для представления +/- полной шкалы входного сигнала. Сведения о частоте опроса аналоговых входов приведены в Руководстве пользователя привода.

#91.10		Значение аналогового входа 3 привода	
Доступ	RO	Диапазон	±1000
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Это значение берется из аналогового входа 3 привода и масштабируется до ±1000 для представления +/- полной шкалы входного сигнала. Сведения о частоте опроса аналоговых входов приведены в Руководстве пользователя привода.

#91.16		Цифровые входы привода	
Доступ	RO	Диапазон	Беззнаковый 8-битный
По умолчанию	Отсутствует	Скорость обновления	250 мксек

Этот параметр аналогичен параметру привода #8.20, он предоставляет в одном параметре состояние 7 цифровых входов. При этом учитывается уровень логики и инвертирование сигналов.

Биты параметра распределены следующим образом:

Бит	Цифровой вход
0	F1
1	F2
2	F3
3	F4
4	F5

Бит	Цифровой вход
5	F6
6	Разрешение (Enable)
7	Зарезервировано - Читается как ноль

#91.17	Число принятых верных сообщений CTSync		
Доступ	RW	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	

Этот параметр увеличивается на 1 при каждом приеме хорошего сообщения CTSync с правильной контрольной суммой.

#91.18	Число принятых неверных сообщений CTSync		
Доступ	RW	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	

Этот параметр увеличивается на 1 при каждом приеме сообщения CTSync с неправильной контрольной суммой.

#91.19	Число пропущенных сообщений CTSync		
Доступ	RW	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	

Этот параметр увеличивается на 1 каждый раз, когда сообщение не было принято, хотя модуль ожидал сообщения.

#91.20	Слишком мала ширина сигнала синхронизации CTSync		
Доступ	RW	Диапазон	Знаковый 32-битный
По умолчанию	Отсутствует	Скорость обновления	Синхронизация

Этот параметр увеличивается на 1 каждый раз, когда сигнал синхронизации имеет неверную ширину. Этот параметр может увеличиться во время синхронизации, но после завершения синхронизации он не меняется. Если этот параметр увеличивается после синхронизации, то возможен слишком большой шум и помехи в сети RS485. Проверьте все разъемы и подключения.

#91.21	Управление синхронизацией между модулями		
Доступ	RW	Диапазон	0 до 2
По умолчанию	0	Скорость обновления	Немедленно

Этот параметр позволяет пользователю настроить модуль SM-Applications в схеме синхронизации между модулями. Более подробное описание этого вопроса приведено в разделе 10 *Синхронизация дополнительных модулей*.

Величина	Описание
0	Не требуется, чтобы модуль участвовал в схеме синхронизации между модулями.
1	Модуль SM-Applications должен участвовать в схеме синхронизации между модулями в качестве поставщика. Описание термина "поставщик" смотрите в разделе 10.1 <i>Обзор</i> .
2	Модуль SM-Applications должен участвовать в схеме синхронизации между модулями в качестве потребителя. Описание термина "потребитель" смотрите в разделе 10.1 <i>Обзор</i> .

#91.22	Состояние синхронизации между модулями		
Доступ	RO	Диапазон	Беззнаковый 8-битный
По умолчанию	Отсутствует	Скорость обновления	Немедленно

Этот параметр показывает состояние модуля SM-Applications в схеме синхронизации между модулями. Более подробное описание этого вопроса приведено в разделе 10 *Синхронизация дополнительных модулей*..

Бит	Значение	Описание
1 до 0	Запрошена роль синхронизации между модулями	Это аналогично битам 1:0 описанного выше параметра управления синхронизацией между модулями.
2	Достигнута роль синхронизации между модулями	Этот бит указывает, что модуль, пытавшийся стать поставщиком синхронизации между модулями, добился этой роли. Если на данном приводе несколько модулей пытались стать поставщиками синхронизации, то хотя бы в одном из этих модулей этот бит будет сброшен. Для модуля, пытающегося стать потребителем синхронизации между модулями, он указывает, что он не запросил быть поставщиком и что поставщик был обнаружен в другом гнезде и используется как источник данных синхронизации. (Если поставщик был найден в другом гнезде, но скорость данных синхронизации не совместима с потребителем, то этот бит будет сброшен, так как поставщик, хотя и найден, не может использоваться как источник данных синхронизации.)
3	Выход поставщика синхронизации соответствует техническим условиям	Этот бит имеет смысл только для модулей, которые были назначены поставщиками синхронизации между модулями и ведомыми CTSync. Этот бит указывает, что сигнал, выдаваемый поставщиком синхронизации между модулями, соответствует допускам привода и что привод сейчас синхронизован с частотой CTSync ведомого. Обратите внимание, что этот бит может быть установлен, только если биты 2-0 этого параметра имеют значения 1, 0, 1, указывая, что модуль - поставщик (0, 1) и что достигнута роль поставщика (1).
7 до 4	<i>Зарезервировано</i>	<i>Читаются как нули</i>

5.10 Меню 18, 19 - параметры приложения

Эти два меню называются параметрами приложения, поскольку пользователь может свободно использовать их по своему усмотрению.

Компоновка двух этих меню аналогична. Все параметры доступны для чтения/записи с модуля SM-Applications (и через порт связи), но могут только читаться на панели управления приводом.

#1x.01	Целый, чтение-запись, сохраняется при отключении питания		
Доступ	RW	Диапазон	Знаковый 16-битный
По умолчанию	0	Скорость обновления	Отсутствует

Этот параметр автоматически сохраняется приводом при отключении питания.

#1x.02- #1x.10	Целый, только чтение		
Доступ	RW (привод RO)	Диапазон	Знаковый 16-битный
По умолчанию	0	Скорость обновления	Отсутствует

#1x.11- #1x.30	Целый, чтение-запись		
Доступ	RW	Диапазон	Знаковый 16-битный
По умолчанию	0	Скорость обновления	Отсутствует

#1x.31- #1x.50	Битовый, чтение-запись		
Доступ	RO	Диапазон	0/1
По умолчанию	0	Скорость обновления	Отсутствует

Параметры #1x.11-#1x.50 сохраняются в энергонезависимой памяти привода.

5.11 Меню 20 - меню приложения

Это меню, как и меню 18 и 19, содержит параметры, не влияющие на работу привода, которые можно использовать для самых разных целей.

ПРИМЕЧ. Это меню НЕ сохраняется в энергонезависимой памяти привода. Однако по запросу его можно сохранить во флэш памяти модуля SM-Applications. Если установлено несколько модулей SM-Applications, то по очевидным причинам только один из них можно настроить на сохранение и восстановление этого меню.

Если параметр #81.21 установлен, то это меню будет сохраняться и восстанавливаться модулем SM-Applications.

#20.01- #20.20	Целый, чтение-запись		
Доступ	RW	Диапазон	Знаковый 16-битный
По умолчанию	0	Скорость обновления	Отсутствует

Эти параметры являются обычными целыми числами в диапазоне от -32768 до 32767.

#20.21- #20.40	Длинное целое, чтение-запись		
Доступ	RW	Диапазон	Знаковый 32-битный
По умолчанию	0	Скорость обновления	Отсутствует

Эти параметры являются длинными целыми, то есть имеют диапазон от -2^{31} до $2^{31}-1$. Если значение параметра превышает максимальную величину, которую можно показать на дисплее панели привода (9,999,999), то будет показано "-----". Нельзя ввести значения, превышающего максимальное отображаемое на дисплее значение.

6 Передача данных

6.1 Порт последовательной передачи данных RS485

В модуле SM-Applications имеется порт последовательной передачи данных RS485. Информация по аппаратному подключению и разводке кабеля приведена в разделе 3 *Установка*.

Этот порт поддерживает ряд встроенных протоколов: CT-ANSI ведомый, Modbus RTU в режимах ведущего и ведомого, Modbus ASCII в режимах ведущего и ведомого и 3 режима пользователя. Возможны подключения по 2-проводной и по 4-проводной схемам.

Таблица 6.1 Режимы последовательной связи - параметр #81.06

Режим	Описание
1	4-проводной CT-ANSI Порт настроен на 1 стартовый бит, 7 битов данных, четную четность и 1 стоповый бит.
2	Зарезервировано
3	Зарезервировано
4	Зарезервировано
5	2-проводной CT-ANSI Порт настроен на 1 стартовый бит, 7 битов данных, четную четность и 1 стоповый бит.
6	Режим пользователя. 1 стартовый бит, 7 битов данных, ЧЕТНАЯ четность, 1 стоповый бит (всего 10 битов)
7	Режим пользователя. 1 стартовый бит, 8 битов данных, ЧЕТНАЯ четность, 1 стоповый бит (всего 11 битов)
8	Режим пользователя. 1 стартовый бит, 8 битов данных, НЕТ четности, 1 стоповый бит (всего 10 битов)
9	Зарезервировано
10	Зарезервировано
11	Зарезервировано
12	Зарезервировано
13, 43, 73	4-проводной Modbus RTU ведомый Порт RS485 настроен на: Режим 13: 1 стартовый бит, 8 битов данных, НЕТ четности, 2 стоповых бита. Режим 43: 1 стартовый бит, 8 битов данных, ЧЕТНАЯ четность, 1 стоповый бит. Режим 73: 1 стартовый бит, 8 битов данных, НЕЧЕТНАЯ четность, 1 стоповый бит.
14, 44, 74	4-проводной Modbus ASCII ведомый Порт RS485 настроен на: Режим 14: 1 стартовый бит, 7 битов данных, НЕТ четности, 2 стоповых бита. Режим 44: 1 стартовый бит, 7 битов данных, ЧЕТНАЯ четность, 1 стоповый бит Режим 74: 1 стартовый бит, 7 битов данных, НЕЧЕТНАЯ четность, 1 стоповый бит
15, 45, 75	2-проводной Modbus RTU ведомый Порт RS485 настроен на: Режим 15: 1 стартовый бит, 8 битов данных, НЕТ четности, 2 стоповых бита. Режим 45: 1 стартовый бит, 8 битов данных, ЧЕТНАЯ четность, 1 стоповый бит. Режим 75: 1 стартовый бит, 8 битов данных, НЕЧЕТНАЯ четность, 1 стоповый бит.
16, 46, 76	2-проводной Modbus ASCII ведомый Порт RS485 настроен на: Режим 16: 1 стартовый бит, 7 битов данных, НЕТ четности, 2 стоповых бита Режим 46: 1 стартовый бит, 7 битов данных, ЧЕТНАЯ четность, 1 стоповый бит Режим 76: 1 стартовый бит, 7 битов данных, НЕЧЕТНАЯ четность, 1 стоповый бит

Таблица 6.1 Режимы последовательной связи - параметр #81.06

Режим	Описание
17, 47, 77	4-проводной Modbus RTU master Порт RS485 настроен на: Режим 17: 1 стартовый бит, 8 битов данных, НЕТ четности, 2 стоповых бита. Режим 47: 1 стартовый бит, 8 битов данных, ЧЕТНАЯ четность, 1 стоповый бит. Режим 77: 1 стартовый бит, 8 битов данных, НЕЧЕТНАЯ четность, 1 стоповый бит.
18, 48, 78	4-проводной Modbus ASCII master Порт RS485 настроен на: Режим 18: 1 стартовый бит, 7 битов данных, НЕТ четности, 2 стоповых бита Режим 48: 1 стартовый бит, 7 битов данных, ЧЕТНАЯ четность, 1 стоповый бит Режим 78: 1 стартовый бит, 7 битов данных, НЕЧЕТНАЯ четность, 1 стоповый бит
19, 49, 79	2-проводной Modbus RTU master Порт RS485 настроен на: Режим 19: 1 стартовый бит, 8 битов данных, НЕТ четности, 2 стоповых бита. Режим 49: 1 стартовый бит, 8 битов данных, ЧЕТНАЯ четность, 1 стоповый бит. Режим 79: 1 стартовый бит, 8 битов данных, НЕЧЕТНАЯ четность, 1 стоповый бит.
20, 50, 80	2-проводной Modbus ASCII master Порт RS485 настроен на: Режим 18: 1 стартовый бит, 7 битов данных, НЕТ четности, 2 стоповых бита Режим 48: 1 стартовый бит, 7 битов данных, ЧЕТНАЯ четность, 1 стоповый бит Режим 78: 1 стартовый бит, 7 битов данных, НЕЧЕТНАЯ четность, 1 стоповый бит
25	Ведущий (Master) SM-Applications CT-Sync Скорость передачи фиксирована на уровне 896875 bps
26	SM-Applications CT-Sync ведомый. Скорость передачи фиксирована на уровне 896875 bps

Если будет выбран неверный или неподдерживаемый режим, то режим по умолчанию вернется назад в 1 (4-проводной CT-ANSI) и может произойти ошибка времени выполнения 49.

Скорость передачи в Бодах указана в параметре **#81.07**.

Адрес этого блока указан в параметре **#81.05**.

6.1.1 CT-ANSI

Протокол CT-ANSI - точно такой же, что реализован в приводе. Более подробные сведения приведены в документации на привод и в оперативной справочной системе.

ПРИМЕЧ. Протокол CT-ANSI поддерживает чтение и запись полных 32-разрядных значений. Изделие UD70 для Unidrive 1 не поддерживает таких больших значений.

Через порт RS485 можно получить доступ ко всем параметрам привода, а также к параметрам модуля SM-Applications.

6.1.2 Modbus-RTU

Поддерживаются режимы ведущего и ведомого протокола Modicon Modbus-RTU.

В ведомом режиме RTU поддерживаются коды следующих функций:

Функция	Описание
FC3	Предустановка одного регистра
FC16	Предустановка нескольких регистров
FC23	Чтение/Запись нескольких регистров

Максимальное число регистров, которые можно прочитать/записать за одну операцию, равно 16.

Параметры привода отображаются на регистры Modbus как **40000 + MenuX100 + Параметр**. Например, параметр #01.21 будет регистром номер 40121.

К данным параметров можно обращаться как в 16-битном, так и в 32-битном режимах. Режим выбирается с помощью 2 старших битов адреса регистра следующим образом:



Биты поля типа 15-14	Доступ
00	16-бит. Обратная совместимость
01	32-бит.
10	Зарезервировано
11	Зарезервировано

Таким образом, для доступа к параметру #70.01 в 32-битном режиме номер регистра $40000 + (0x4000 + 70X100 + 01) = 63385$.

Если 32-битный параметр читается в режиме 16-битного доступа, то будут возвращены младшие 16 битов регистра.

Обратите внимание, что фактически передаваемый в протоколе номер регистра на 1 меньше запрошенного и не содержит смещения 40000. Большинство ведущих блоков Modbus автоматически обращают эту разницу на -1, но некоторые этого не делают.

В режиме ведущего в программе DPL пользователя используются следующие команды:

- RtuReadHoldingRegs
- RtuReadHoldingParas
- RtuReadInputRegs
- RtuPresetMultipleRegs
- RtuPresetMutlipleParas
- RtuMasterReply
- RtuMasterStatus

6.1.3 Modbus ASCII

Поддерживаются режимы ведущего и ведомого протокола Modicon Modbus ASCII.

В ведомом режиме Modbus ASCII поддерживаются коды следующих функций:

Функция	Описание
FC3	Предустановка одного регистра
FC16	Предустановка нескольких регистров
FC23	Чтение/Запись нескольких регистров



Дополнительные сведения приведены в оперативной справке.

Полная информация по протоколу Modbus приведена на сайте Modicon по адресу www.modicon.com. Обратите внимание, что 32-битовый доступ разработан только в Control Techniques.

6.1.4 Режимы пользователя

Эти режимы выключают все внутренние протоколы и позволяют пользователю напрямую обращаться к порту RS485 непосредственно из программы DPL. Их можно использовать совместно с DPL ANSI - ANSIREAD, ANSIWRITE и т.д. Определенные пользователем протоколы можно также реализовать с помощью команд PUTCHAR и GETCHAR языка DPL.

6.2 CTNet

Полное описание CTNet не может быть приведено в этом Руководстве для пользователя, оно приведено в отдельном Руководстве пользователя CTNet.

6.3 Параметры отображения SM-Applications

Модуль SM-Applications имеет внутренние параметры, которые могут записываться и считываться модулями fieldbus, которые также установлены в приводе Unidrive SP. Это может оказаться удобным способом связи между двумя шинами fieldbus. Эти параметры показаны в таблице ниже.

Таблица 6.2 Внутренние параметры модуля SM-Applications

Параметры SM-Applications	Ссылка на параметр	Ссылка на гнездо 1	Ссылка на гнездо 2	Ссылка на гнездо3
Регистры ПЛК _Pxx%	#70.xx	#100.xx	#130.xx	#160.xx
Регистры ПЛК _Qxx%	#71.xx	#101.xx	#131.xx	#161.xx
Регистры ПЛК _Rxx%	#72.xx	#102.xx	#132.xx	#162.xx
Регистры ПЛК _Sxx%	#73.xx	#103.xx	#133.xx	#163.xx
Регистры ПЛК _Txx%	#74.xx	#104.xx	#134.xx	#164.xx
Регистры ПЛК _Uxx%	#75.xx	#105.xx	#135.xx	#165.xx
Параметры локальной конфигурации	#81.xx	#111.xx	#141.xx	#171.xx
Параметры функции таймера	#85.xx	#115.xx	#145.xx	#175.xx
Параметры цифрового I/O	#86.xx	#116.xx	#146.xx	#176.xx
Параметры состояния	#88.xx	#118.xx	#148.xx	#178.xx
Общие параметры	#90.xx	#120.xx	#150.xx	#180.xx
Параметры быстрого доступа	#91.xx	#121.xx	#151.xx	#181.xx

Модуль интерфейса fieldbus считывает и записывает данные непосредственно из или в внутренние регистры модуля SM-Applications. Модуль интерфейса fieldbus может считывать данные и записывать данные в модуль SM-Applications, установленный в любое гнездо привода Unidrive SP, для этого надо просто указать целевые параметры, показанные в Таблице 6.2.

ПРИМЕЧ. Если в приводе Unidrive SP установлен единственный модуль SM-Applications, то можно без проблем обращаться к обычным параметрам SM-Applications, поскольку SM-PROFIBUS-DP автоматически переправит их в модуль SM-Applications.

6.3.1 Пример конфигурации 1

Рассмотрим привод Unidrive SP в следующей конфигурации:

- Гнездо 1 - свободное
- Гнездо 2 - модуль SM-Applications
- Гнездо 3 - модуль SM-PROFIBUS-DP

Если из сети PROFIBUS-DP поступил запрос на чтение параметра #71.08, то он будет переправлен в модуль SM-Applications с наименьшим номером гнезда, то есть в гнездо 2. Будет возвращено значение _Q08% из гнезда 2.

Если из сети PROFIBUS-DP поступил запрос на чтение параметра #131.08, то он будет отослан прямо в модуль SM-Applications в гнезде 2. Будет по-прежнему возвращено значение _Q08% из гнезда 2.

Если из сети PROFIBUS-DP поступил запрос на чтение параметра #101.08, то он будет отослан прямо в модуль SM-Applications в гнезде 1. Поскольку в гнезде 1 нет никакого модуля SM-Applications, будет возвращено сообщение об ошибке, указывающее, что запрошенный параметр не существует.

Если бы в гнезде 1 был установлен модуль SM-DeviceNet, то для чтения или записи данных вы могли бы использовать прямую ссылку на параметр гнезда, что предоставляет вам простой шлюз для передачи данных между DeviceNet и Profibus-DP.

6.3.2 Пример конфигурации 2

Рассмотрим привод Unidrive SP в следующей конфигурации:

- Гнездо 1 - модуль SM-Applications
- Гнездо 2 - модуль SM-Applications
- Гнездо 3 - модуль SM-PROFIBUS-DP

Если из сети PROFIBUS-DP поступил запрос на чтение параметра #71.08, то он будет переправлен в модуль SM-Applications с наименьшим номером гнезда, то есть в гнездо 1. Будет возвращено значение _Q08% из гнезда 1.

Если из сети PROFIBUS-DP поступил запрос на чтение параметра #131.08, то он будет отослан прямо в модуль SM-Applications в гнезде 2. Будет возвращено значение _Q08% из гнезда 2.

Если из сети PROFIBUS-DP поступил запрос на чтение параметра #101.08, то он будет отослан прямо в модуль SM-Applications в гнезде 1. Будет возвращено значение _Q08% из гнезда 1.

ПРИМЕЧ. Если в приводе Unidrive SP установлены два модуля SM-Applications, то к параметрам модуля SM-Applications лучше всего обращаться с помощью прямой ссылки на параметры гнезда. Если используются обычные параметры модуля SM-Applications, и модуль SM-Applications вынут из гнезда 1, то эти ссылки на параметры будут переправлены в гнездо 2.

6.3.3 Пример конфигурации 3

Рассмотрим привод Unidrive SP в следующей конфигурации:

- Гнездо 1 - модуль SM-Applications
- Гнездо 2 - модуль SM-Applications
- Гнездо 3 - модуль SM-Applications

Модуль SM-Applications не позволяет считывать или записывать данные по прямым ссылкам на параметры гнезда, поэтому передачу данных следует выполнять другим методом.

Если модуль SM-Applications из гнезда 1 желает считать регистр #71.08 из модуля SM-Applications в гнезде 3, то это можно сделать через один из 32-битных параметров приложения (#20.21-#20.40). Модуль SM-Applications в гнезде 1 должен записать данные регистра в параметр приложения. Модуль SM-Applications из гнезда 3 затем может считать этот параметр.

Код для гнезда 1 :

#20.21 = #71.08

Код для гнезда 3 :

#71.08 = #20.21

Эти данные могут быть также считаны модулем SM-Applications, установленным в гнезде 2.

ПРИМЕЧ. Модуль SM-Applications не может обращаться к внутреннему меню #1xx.xx. Он должен обращаться к ним по номеру ссылки на параметр, например, если вам нужно считать параметр #104.35, то вы должны использовать #74.35. Приведенный выше пример кода не синхронизирует обновление параметра #71.08 в двух модулях. Более подробные сведения приведены в разделе 10 *Синхронизация дополнительных модулей*. Если модули SM-Applications подключены к одной сети CTNet, то это можно сделать с помощью циклических данных. Более подробная информация по CTNet приведена в Руководстве пользователя CTNet.

7 Программирование на DPL

В этой главе рассмотрены вопросы:

- Базовая структура и синтаксис программы DPL
- Основные команды DPL
- Новые функции, предоставляемые модулем SM-Applications



Этот раздел никак не следует рассматривать как полное руководство по языку DPL. Полные справочные данные по командам и блокам функций DPL приведены в оперативных справочных руководствах.

7.1 Заголовок программы

Каждая программа DPL начинается с раздела заголовка. Программа SYPT Workbench сама создает этот раздел для пользователя. Обычно в нем содержится следующая информация:

- Название программы
- Автор программы
- Номер версии программы

7.1.1 Псевдонимы

Сразу же после заголовка пользователь может разместить раздел *псевдонимов*. Псевдонимы используются в тексте программы 'вместо' различных выражений или констант, то есть могут заменять:

- численное константное выражение
- адрес регистра или параметра
- выражение или оператор DPL

Псевдонимы определяются с помощью оператора (директивы) \$DEFINE.

```
$DEFINE имя значение
```

Например, хорошим тоном считается использование псевдонимов для именования всех используемых в программе параметров привода.

```
$DEFINE PRESET_REF_1 #1.21  
$DEFINE PRESET_REF_2 #1.22  
$DEFINE SPEED_FB #3.02
```

Рекомендуется также использовать в именах всех псевдонимов только ЗАГЛАВНЫЕ буквы, чтобы их легко можно было отличить от обычных переменных.

ПРИМЕЧ. Рекомендуется также, чтобы имена представляющих целые значения псевдонимов заканчивались на символ '%'. В графических утилитах программиста (QLD/FBD), SYPT будет рассматривать все псевдонимы без символа % как величины с плавающей запятой. Поэтому их нельзя будет использовать на входе LD и на входе только с целыми значениями.

Директива \$DEFINE НЕ создает никакого машинного кода и никак не ускоряет выполнение вашей программы - она просто позволяет вам использовать более удобные имена для обозначения некоторых элементов в программе.

7.2 Задачи

Программа DPL состоит из отдельных разделов, называемых задачами. Внутри задач пользователь записывает инструкции программы, которые должны выполняться микропроцессором при определенных условиях или через определенные интервалы времени. Каждая задача имеет свое конкретное имя, назначение и приоритет и только одна из каждой задачи может присутствовать в программе DPL. Ниже кратко описаны общие задачи:

Таблица 7.1 Общие задачи

Имя задачи	Приоритет	Назначение
INITIAL	3	Самая первая задача, которая выполняется после включения питания или после сброса. Эта задача обычно используется для инициализации параметров привода и переменных программы. Ни одна другая задача не может выполняться, пока эта задача не будет завершена.
BACKGROUND	1	Задача низкого приоритета, используемая для не критичных по времени функций. Способ выполнения этой задачи близко аналогичен циклу сканирования в ПЛК. Обычно эта задача организована в виде одного большого цикла, и в конце задачи стоит инструкция перехода назад на начало. Если эта задача будет завершена, то она больше не будет выполняться.
CLOCK	2	Задача, выполняемая через фиксированный интервал времени (от 1 до 200 мсек) и используемая в некоторых связанных с временем приложениях, например, для создания профиля ramпы. Эта задача теперь синхронизована с циклом управления привода уровня 2 и ее можно использовать вместо старой задачи Encoder.
POS0 POS1	4	Две задачи реального времени, которые выполняются через кратное число циклов управления приводом (от 250 мсек до 8 мсек). Обычно эти задачи используются для управления контуром скорости или тока привода в таких приложениях, как позиционирование. Сначала выполняется задача POS0, и сразу после нее выполняется задача POS1.
EVENT	5	Задачи событий выполняются только при возникновении определенного события. События могут собираться с разных источников, например, из CTNet, других дополнительных модулей в приводе Unidrive SP или из программы пользователя. Задачи EVENT имеют высший приоритет, поэтому они обычно содержат очень малое число инструкций. Их можно связывать в цепочки для прерывания сервисных подпрограмм.
EVENT1	5	Смотрите описание выше.
EVENT2	5	Смотрите описание выше.
EVENT3	5	Смотрите описание выше.
ERROR	1	Эта задача выполняется только при возникновении в программе DPL пользователя ошибки времени выполнения (например, деления на ноль). Ее можно использовать для безопасного завершения ненормального выполнения программы. Перед запуском задачи ERROR останавливается выполнение всех других задач.

ПРИМЕЧ. При использовании задач CLOCK, POS0 и POS1 рекомендуется не применять в них такого кода, как циклы FOR и DO WHILE. Они могут вызвать в DPL ошибку переполнения (tr54).

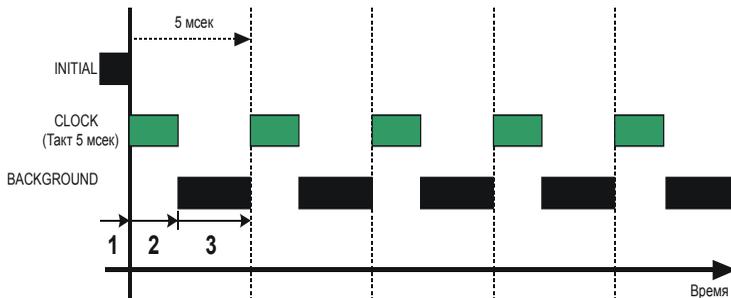
ПРИМЕЧ. Можно по-прежнему использовать задачи UD70 ENCODER и SPEED. Теперь они являются псевдонимами для задач POS0 и POS1 соответственно (то есть если в программе есть задача ENCODER, то это все равно, что в ней есть задача POS0). Интервал времени для обеих задач не фиксирован, как в UD70, а указывается пользователем. Задачу CLOCK в модуле SM-Applications можно использовать вместо задачи ENCODER в UD70, это дает интервал времени, лучше соответствующий интервалу задачи UD70 ENCODER, чем интервал времени задач POS0 и POS1.

Все инструкции программы **должны** располагаться внутри задачи. Для периодически выполняемых задач, например, POS0, POS1 и CLOCK, все инструкции задачи должны быть выполнены в течение определенного времени, поэтому внутри этих задач можно выполнять только критичные по времени функции.

Задачи позиционирования состоят из POS0, APC и POS1 и выполняются именно в этом порядке, если они имеются (то есть если они настроены на выполнение).

Задачи имеют разные уровни приоритета, поэтому одна задача может прерывать другую задачу. Чем выше номер приоритета в показанной выше таблице, тем выше приоритет этой задачи. Поэтому задача POS0 может прерывать задачу CLOCK, которая может прерывать фоновую задачу BACKGROUND.

Концепция взаимного прерывания задач показана на следующей схеме:

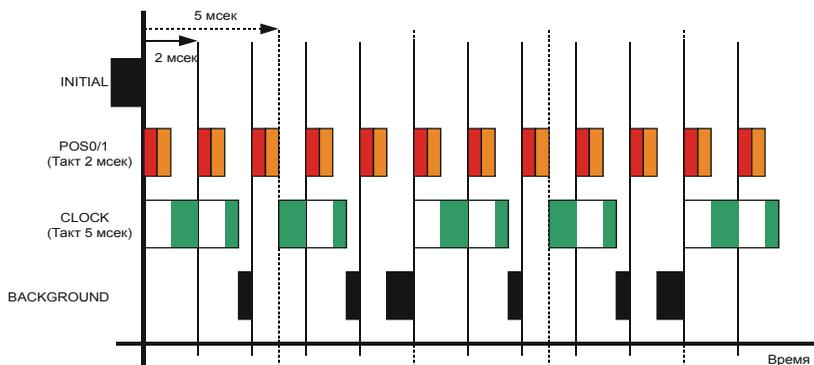


Обозначения:

1. Задача INITIAL имеет исключительное управление ресурсами. Не может выполняться никакая другая задача.
2. Приоритет задачи CLOCK выше, чем у фоновой задачи BACKGROUND.
3. Задача CLOCK завершена и теперь может выполняться задача BACKGROUND - до следующего такта системных часов.

Обратите особое внимание на то, что задача CLOCK запускается периодически через фиксированные интервалы времени (на схеме выше через 5 мсек). Это означает, что все инструкции внутри задачи CLOCK **ДОЛЖНЫ** быть выполнены менее чем за 5 мсек, иначе задача BACKGROUND не получит промежутка для своего выполнения, либо может произойти отключение по перегрузке процессора.

На следующей схеме показано, что происходит при использовании еще задач POS:



Здесь видно, что задачи POS0 и POS1 прерывают задачу CLOCK, которая в свою очередь прерывает фоновую задачу BACKGROUND. Как видно, это сильно загруженная программа, так как для выполнения фоновой задачи остается совсем немного времени. Для определения того, насколько велика загрузка модуля SM-Applications можно использовать параметр свободного ресурса процессора #81.04.

7.2.1 Задачи EVENT

Предусмотрены четыре задачи событий. Задача события может быть запущена по:

- Приему кадра синхронизации CTNet SYNC (настройка через #81.35)
- Запуск от программы пользователя
Новая команда DPL SCHEDULEEVENT. Смотрите оперативную справочную систему.

7.3 Переменные

7.3.1 Типы

Имеются три основных типа переменных:

1. Целая переменная
2. Переменная с плавающей запятой двойной точности
3. Переменная с плавающей запятой обычной точности

Целая переменная указывается символом % в конце имени переменной. Если в имени переменной нет символа %, то это переменная формата плавающей запятой.

Таблица 7.2 Типы переменных

Тип	Представление	Диапазон
Целый	32-битовая со знаком.	-2147483648 до 2147483647
Одинарный плавающий	32-битовая, 1 бит знака, 8 - экспоненты и 23 - мантиссы.	±3.40282e+038
Двойной плавающий	64 бита: 1 бит знака, 52 бита мантиссы, 11 битов экспоненты	±1.79769e+308

Примеры переменных:

```
speed% = 1234 // целая переменная
value = 55.6 // переменная с плав. запятой
```

Для указания того, какой тип переменных с плавающей запятой - одинарной или двойной точности - используется во всей программе, в начале программы ставится специальный оператор. По умолчанию используются переменные двойной точности. Для использования переменных одинарной точности сразу после секции заголовка программы (с \$TITLE и т.п.) надо вставить следующую строку:

```
$flt single
```

7.3.2 Имена переменных

Первым символом имени переменной должна быть буква. Последующие символы могут быть буквами, цифрами, и символом подчеркивания (_).

- ПРИМЕЧ.**
- Имена переменных чувствительные к регистру (например, имена speed%, SPEED% и Speed% относятся к трем разным переменным).
 - Редакторы SYPT QuickLD и FBD позволяют использовать только имена переменных длиной не более 16 символов, включая символ %.

7.3.3 Инициализация переменных

Перед использованием любой переменной в программе ей нужно присвоить начальное значение. Обычно это делает задача INITIAL. Например,

```
Initial {
speed_sp% = 0
ramp% = 0
}
```

7.3.4 Область и время жизни переменной

Переменные могут быть глобальными или локальными. Все переменные, определенные в программах DPL, являются глобальными, то есть к ним имеет доступ и их может изменять любая задача. Исключением являются переменные в определенном пользователем функциональном блоке, которые являются локальными (то есть к ним нельзя обратиться снаружи определенного пользователем функционального блока).

Ни одна переменная DPL не переживает сброса модуля SM-Applications. Не забывайте, что восстановление привода из состояния отключения также вызывает сброс модуля.

7.3.5 Массивы неизменного размера

Программа DPL может содержать массивы целых или плавающих переменных. Разрешаются массивы только фиксированного размера (одномерные).

Массив необходимо сначала объявить с помощью оператора DIM (обычно в задаче Initial), при этом число элементов массива указывается в квадратных скобках после имени массива, например:

```
DIM myarray%[20] // Целый массив из 20 элементов
DIM array2[30] // Массив из 30 плавающих элементов
```

Все элементы массива нумеруются от 0 до число_элементов - 1. В приведенном выше примере первый элемент массива myarray%[] - это:

```
myarray%[0]
```

а последний элемент - это:

```
myarray%[19]
```

Предусмотрены две функции, с помощью которых во время выполнения программы можно определить верхнюю и нижнюю границу индекса массива. Это функции UPPER и LOWER. Для массива myarray%[], UPPER вернет значение 19, а LOWER вернет 0.

7.3.6 Константные массивы

Константные массивы, как следует из их названия, содержат неизменные предопределенные значения. Значения константного массива определяются в программе DPL с помощью специального раздела (смотрите термин CONST в оперативной справочной системе). Можно определять только целые значения. Преимущество константного массива заключается в том, что размер массива ограничивается только размером памяти программы, а не ОЗУ переменных. Размер памяти программы составляет 384 кбайт, он используется для хранения скомпилированного файла DPL, данных константных массивов и опционно самого файла DPL.

7.3.7 Память для хранения - число переменных

Все переменные, динамические массивы и регистры ПЛК располагаются в области памяти размером 80 кбайт. Каждая целая переменная и переменная с плавающей запятой одинарной точности занимает 4 байта (32 бита), а переменная с плавающей запятой двойной точности занимает 8 байт (64 бита). Кроме того, в памяти размещаются другие элементы, например, индексы для параметров.

Компилятор DPL оповестит вас, если вы достигнете предела имеющейся памяти.

7.3.8 Адресация битов в переменных

Все целые переменные и массивы можно адресовать побитно. Это означает, что внутри переменной можно отдельно читать или записывать каждый отдельный бит. Для доступа к битам после имени переменной поставьте точку (.) и затем номер бита от 0 до 31.

Пример 1 (простая переменная):

```
flags% = 0 // инициализируем все 32 бита в 0
flags%.0 = 1 // настроим бит 0 в 1

// теперь проверим, настроены ли бит 0 и бит 1 в значение 1.
IF flags%.0 & flags%.1 = 1 THEN
  PRINT "Условие выполнено."
ENDIF
```

Example 2 (массивы):

```
DIM myarray%[10]
...
IF myarray%.1[4] = 1 THEN;проверим бит 1 элемента #4.
  PRINT "Условие выполнено."
ENDIF
```

Примечание: Номер бита должен быть константой - переменные не допускаются.

7.3.9 Регистры ПЛК

Область “ПЛК” - это специальный набор predetermined 32-битных регистров. Регистры ПЛК делятся на 6 наборов из 100 параметров, пронумерованных от 00 до 99. К регистрам можно также обратиться из программы DPL пользователя с помощью специального имени переменной или имени массива. Четыре из наборов регистров можно сохранять во флэш-памяти модуля SM-Applications. Дополнительная информация о регистрах ПЛК приведена в разделе 5.4 *Меню 70-75 - регистры ПЛК*.

7.3.10 Файлы ОЗУ

Файлы ОЗУ позволяют пользователю сохранить “файлы” в ОЗУ пользователя в модуле SM-Applications. Их можно загрузить и выгрузить с помощью команд DPL. Их преимущество заключается в том, что за одну операцию вы можете вызвать или записать массив чисел, а не отдельный элемент массива.



Дополнительные сведения о файлах ОЗУ и примеры программ приведены в оперативной справочной системе.

7.4 Параметры

Параметры можно разделить на две категории:

- Параметры привода
- Параметры SM-Applications

Параметры привода размещаются в ведущем приводе. Большинство из них влияет на работу привода, однако некоторые зарезервированы как “прикладные программы”. Это параметры меню 18, 19 и 20.

Параметры SM-Applications являются локальными и доступны только внутри модуля SM-Applications. Эти параметры предоставляют доступ к дополнительным функциям модуля SM-Applications и обеспечивают более быстрый доступ к некоторым параметрам привода.

ПРИМЕЧ. Модуль SM-Applications гарантирует, что используемая им база данных параметров привода всегда совпадает с этой базой в ведущем приводе Unidrive SP. Когда модуль SM-Applications только установлен в Unidrive SP и в первый раз включается питание, на дисплее привода может на несколько секунд появиться слово “Loading”. Это указывает процесс синхронизации баз данных в модуле SM-Applications. Такая синхронизация выполняется только при первой установке модуля в привод. При последующих включениях питания слово “Loading” появляется только кратковременно.

7.4.1 Чтение и запись параметров

Чтение и запись параметров выполняется с помощью команды #. Доступ к параметрам ведется в формате #MM.PP точно так же, как с клавиатуры привода. Например, для чтения параметра обратной связи по скорости (параметр 03.02), используйте:

```
speed% = #3.02
```

для записи параметра задания скорости (например, 01.22), используйте:

```
#01.22 = 1500
```

Обратите внимание, что передний нуль в поле меню/параметра можно опускать. Например, все конструкции #3.02, #03.02, #03.2 и #3.2 обращаются к одному параметру.

7.4.2 Целые параметры с фиксированной запятой

Работа с целыми параметрами с фиксированной запятой выполняется медленнее, чем с целыми параметрами. Для ускорения при записи и чтении параметров можно использовать специальную команду #INT. При использовании этой команды для целых с фиксированной запятой десятичные позиции будут автоматически удаляться.

Например, параметр #1.19 имеет диапазон значений 0.000 - 0.099. При чтении этого параметра с помощью:

```
speed_fine% = #INT1.19
```

будут возвращены целые значения от 0 до 99. При записи команда:

```
#INT1.19 = 45
```

настроит параметр в значение 0.045 (как команда #1.19=0.045). Преимуществом этого является то, что программа DPL может использовать целые переменные (%) вместо плавающих, что ускоряет работу.

7.5 Операторы

DPL предоставляет все стандартные операторы:

Таблица 7.3 Стандартные операторы в порядке очередности выполнения

Оператор	Значение
-	Арифметическое изменение знака
!	Логическое отрицание (унарная)
!(..., nbit)	Отрицание битов <i>nbit</i>
*	Умножение
/	Деление
%	Модуль (остаток)
+	Сложение
-	Вычитание
&	Побитовое И
	Побитовое ИЛИ
^	Побитовое исключающее ИЛИ (XOR)

Таблица 7.4 Условные операторы в порядке очередности выполнения

Оператор	Значение
=	Равенство
<	Меньше чем
>	Больше чем
<=	Меньше или равно
>=	Больше или равно
<>	Неравенство
AND	Логическое И
OR	Логическое ИЛИ
NOT	Логическое НЕТ

7.6 Основные команды DPL

Версия языка DPL, реализованная для модуля SM-Applications, обратно совместима с продуктом UD70 для Unidrive 1, но в нее добавлено несколько новых выражений.



Полные сведения о языке DPL и библиотеке функциональных блоков приведена в системе оперативной справки.

7.6.1 Новые команды для модуля SM-Applications

Эти новые команды были введены в язык DPL для модуля SM-Applications и отсутствовали в UD70 для Unidrive 1.

Цикл FOR

Это новый оператор для модуля SM-Applications.

```
FOR переменная = целое_выражение to целое_выражение [STEP константа]
  операторы
LOOP
```

CASE

Это альтернативный вариант для конструкции выбора IF-ELSEIF-ENDIF.

```
SELECT целое_выражение
  CASE целая_константа
    операторы
  [CASE целая_константа, целая_константа ...
    [операторы]]
  [ELSE
    [операторы]]
ENDSELECT
```

Эта конструкция позволяет удобно проверить несколько константных значений. Можно включать любое число операторов CASE.

ПРИМЕЧ. В двух показанных выше примерах некоторые участки текста заключены в квадратные скобки ([и]). Участок кода внутри квадратных скобок является опциональным.

ПРИМЕЧ. Операторы CASE выполняются аналогично тому, как в программах типа Visual Basic, в которых управление программы НЕ передается к следующему CASE, как это выполняется в языке программирования C.

MAX_INT, MIN_INT, MIN_FLOAT, MAX_FLOAT

Это специальные предопределенные ключевые слова, которые распознаются компилятором DPL и заменяются соответствующим численным значением.

Таблица 7.5 Min/Max

Ключевое слово	Величина
MIN_INT	-2147483648
MAX_INT	2147483647
MIN_FLOAT	-3.40282e+038 (модель одинарной точности) -1.79769e+308 (модель двойной точности)
MAX_FLOAT	3.40282e+038 (модель одинарной точности) 1.79769e+308 (модель двойной точности)

UPPER/LOWER

Эти функции принимают массив в качестве параметра и возвращают верхний и нижний индекс массива соответственно. Например:

```
// Создадим массив из 1000 элементов
DIM array%(1000)

// now,

l% = LOWER(array%) // вернет значение 0
u% = UPPER(array%) // вернет значение 999.

// найдем сумму всех величин в массиве array%
total%=0
FOR i% = LOWER(array%) TO UPPER(array%)
    total% = total% + array%(i%) //добавим элемент массива к total
LOOP
```

TRUNC

Эта функция преобразует значение с плавающей запятой в целое, используя обрезание (отрасывание дробной части) вместо округления.

Например:

```
// инициализируем переменную с плавающей запятой
floatval = 1.56

int1% = floatval // автопреобразование округляет до 2.
int2% = INT(floatval) // явное преобразов. по INT округляет до 2
int3% = TRUNC(floatval) // явное преобразов. по TRUNC дает 1
```

SCHEDULEEVENT

Этот функц. блок используется для планирования задачи EVENT. Его аргументы:

- Номер гнезда
Указывает, какое гнездо запускает задачу события. Сейчас разрешен только 0, что означает локальное гнездо.
- ID задачи
Диапазон 0-3 для указания запускаемой задачи EVENT
- Причина
Определяемая пользователем причина. Должна быть не менее 34. Эту величину можно проверить в EVENT, посмотрев параметр #90.12-#90.15.

```
BACKGROUND {
... некоторый код
// Спланировать локальную задачу event1 с кодом причины 45.
a% = SCHEDULEEVENT(0, 1, 45)
... еще некоторый код
}

EVENT1 {
IF #90.13 = 45 THEN
    // задача запущена из DPL
ENDIF
}
```

CTNETDIAGNOSTICS

Возвращает диагностическую информацию по CTNet. Смотрите оперативную справочную систему. Эта функция заменяет специальные переменные, например, NOFMESSAGES, которые использовались в изделии UD70.

В этой команде отсутствуют аргументы, она возвращает 10 значений. В таблице ниже описаны выходные значения и эквивалентные переменные UD70.

Таблица 7.6 Выходные значения CTNetDiagnostics

Выход	Величина	Переменная UD70
1	Полное число сообщений, обработанных этим узлом	NOFMESSAGES
2	Число переполнений циклических данных	NOFOVERRUNS
3	Потерянные сообщения RX	NOFLOSTMESSAGES
4	Число повторных попыток	NOFRETRIES
5	Число повторных подключений	NOFRECONS
6	Число избыточных NAK	NOFEXENAKS
7	Повторяющиеся супс	NOFDUPSYNCS
8	Число локально созданных повторн. подключений	NOFMYRECONS
9	Число нециклических сообщений	NOFNONCYCLICMESSAGES
10	Число потерянных сообщений маршрутизации	Отсутствует

GETPARATTR

Эта функция используется для получения атрибутов параметров, например, максимальных и минимальных значений, флагов только чтения и т.п.

```
(max%, min%, flags%) = GETPARATTR(menu%, par%)
```

CModeXfer

Эта функция позволяет пользователю изменить режим привода, при этом ни один из установленных дополнительных модулей не выполнит аппаратного сброса. Она позволяет плавно изменять режим привода. Во время смены режима привода шины fieldbus не смогут записывать в параметры и это обрабатывается на системном уровне. Во время этого периода они НЕ получат сообщения "ошибка записи параметра".

ПРИМЕЧ. При использовании этой команды меню с 15 по 20 не сбрасываются в значения по умолчанию.

RtuReadRegs

RtuReadParas

RtuReadInputRegs

RtuPresetRegs

RtuPresetParas

RtuMasterReply

RtuMasterStatus

Эти команды позволяют пользователю воспользоваться функциями Ведущего для SM-Applications Modbus RTU. Смотрите оперативную справочную систему.

PFIXREAD6/PFIXWRITE6

Эти блоки обеспечивают чтение и запись параметров привода с фиксированной точностью в 6 десятичных знаков.

SETUSERID

Эта команда используется для настройки идентификатора пользователя - параметра #81.49.

```
SETUSERID(101) // настроить #81.49 в 101.
```

ANSIREPLY

Эта команда использовалась в UD70, однако для модуля SM-Applications ее синтаксис был изменен.

```
(status%, reply%) = ANSIREPLY()
```

Это функция порта RS485, которая используется совместно с функциями ANSIREAD и ANSIWRITE.

Примеры ANSIREPLY:

```
result% = ANSIREADN(12, 1811) //выполнить чтение ansi
//если сообщение было успешно отправлено
IF result% = 0 THEN
    //сообщение не успешно отправлено
    goto top:
ENDIF

timeout% = 0 //сообщение успешно отправлено, поэтому инициал. таймаут
DO
    (status%, reply%) = ANSIREPLY()// получить состояние и значение
    DELAY(1) // задержка 100 мсек
LOOP WHILE status% = -65536 AND timeout% < 50 //таймаут=50хтакты таймера
```

```

Background{
top:
value% = #18.11
result% = ANSIWRITEN(12, 1811, value%, 1);записать величину в удаленный
привод
IF result% = 0 THEN
    //сообщение не послано успешно
    goto top:
ENDIF

CALL get_reply: //получим ответ

GOTO top:
} //Background

get_reply:{
timeout% = 0
DO
    (status%, reply%) = ANSIREPLY()
LOOP WHILE status% = -65536 AND timeout% < 50
} //get_reply:

```

Первый выходной аргумент возвращает состояние команды ANSIREPLY и может принимать одно из следующих значений:

- 65536 = Ответ еще не принят
- 65537 = Ответ принят, но с плохой контрольной суммой
- 65538 = Принят EOT (то есть параметр не существует)
- 65539 = Принят NAK
- 65540 = ACK received

AssRAM

UnassRAM

RamLength

SetRamLength

Эти команды позволяют программисту использовать в модуле SM-Applications файлы ОЗУ. Файлы ОЗУ предоставляют средство для доступа к массивам программы пользователя, например, с помощью файловых служб SMP. Более подробные сведения по этим командам и файлам ОЗУ приведены в оперативной справочной системе.

OpenReadSmartCard

OpenWriteSmartCard

CloseSmartCard

ReadSmartCardByte

WriteSmartCardByte

GetNextSmartCardFile

ReadReadOnlyBit

WriteReadOnlyBit

Эти команды позволяют пользователю использовать карты SmartCard привода Unidrive SP. За подробной информацией обращайтесь в ваш локальный центр поддержки приводов.

7.6.2 Команды и функциональные блоки DPL

Имеется обширный список команд и функций, которые можно использовать в программе DPL. Этот список слишком большой для этого Руководства, так что посмотрите его в оперативной справочной системе.

7.7 Определяемые пользователем функциональные блоки

7.7.1 Обзор

SYPT стандартно поставляется с предопределенной библиотекой функциональных блоков, которые можно использовать как с утилитами графического программирования (LD и FBD), так и в простом DPL.

Система определяемых пользователем функциональных блоков UDFB позволяет пользователю создавать свои собственные функциональные блоки, которые будут автоматически доступны в утилитах графического программирования (схемы функциональных блоков Function Block Diagrams и схемы QuickLD) в дополнение к функциям стандартной библиотеки.

Сам UDFB подобен автономной секции программы DPL и поэтому может состоять из смеси простых команд DPL, блоков FBD и схем QLD, а также из других UDFB. Однако обратите внимание, что в UDFB вы не можете создать секции стандартной задачи (например, POS0).

7.7.2 Область действия UDFB

Каждый блок UDFB является локальным по отношению к узлу программы DPL, в которой он определен. Чтобы сделать UDFB доступным для программ в другом узле, надо просто скопировать и вставить секцию UDFB в программу другого узла.

Блок UDFB отображается внутри редактора DPL или SYPT подобно задаче - то есть в "сжимаемой" секции - и рекомендуется располагать все блоки UDFB в начале программы, поскольку перед использованием блока UDFB он должен быть определен.

7.7.3 Инкапсуляция и хранение данных

В отличие от любой задачи в программе DPL блоки UDFB являются самодостаточными единицами программы (то есть они инкапсулированы). Это означает, что каждый блок UDFB имеет свой собственный уникальный набор переменных (локальные переменные).

Блок UDFB взаимодействует с программой DPL в узле только через свои входные и выходные аргументы. Блок UDFB не может получить доступа к глобальным переменным DPL внутри программы DPL, а также к переменным внутри других блоков UDFB.

Конечно, блок UDFB может обращаться к параметрам привода и к регистрам модуля SM-Applications, которые считаются глобальными, однако этого не рекомендуется делать, особенно в блоках, которые могут повторно использоваться другими программами или приложениями. Те ситуации, когда блоку необходим прямой доступ к параметрам или регистрам, возможны в особых случаях использования изделия.

При каждом использовании блока UDFB в программе DPL создается его отдельный экземпляр, который является копией UDFB с уникальными локальными переменными.

Примечание: Локальные переменные UDFB нельзя просматривать в окне просмотра программы SYPT

7.7.4 Правила именования UDFB

Чтобы обеспечить уникальность имен UDFB и исключить возможность конфликтов с именами блоков стандартной библиотеки функций, имя блока UDFB должно начинаться на символ подчеркивания (`_`). Длина этого имени ограничена 16 символами, однако рекомендуется использовать короткие имена, чтобы они правильно отображались в редакторах SYPT FBD и QuickLD, например `_myFunc`, `_PID1` and `_my_Func`

Вот примеры недопустимых имен:

`myFunc`, `UDFB1`

7.7.5 Входные и выходные аргументы

Блок UDFB может передавать данные через свои аргументы следующих типов:

- Целые переменные
- Переменные с плавающей запятой
- Целые массивы
- Массивы с плавающей запятой

Входные и выходные аргументы являются стандартными переменными DPL - то есть они чувствительны к регистру и их имена должны начинаться на букву, а не на цифру. Длина имен входных аргументов ничем не ограничена, однако редакторы FBD и QuickLD внутри SYPT показывают только первые 5 символов имени аргумента.

Количество входных и выходных аргументов ограничивается только доступной памятью в отличие от изделия UD70, в котором могло быть не более 10 целых входных и 10 целых выходных аргументов.

ПРИМЕЧ. Поскольку программа SYPT Workbench была разработана для UD70, утилиты, используемые для создания блоков UDFB, по-прежнему налагают ограничения на входные и выходные аргументы и на порядок аргументов с плавающей запятой и целых. Поэтому в системе программирования SYPT Workbench V1 нельзя реализовать полной гибкости модуля SM-Applications V1. Более подробные сведения по ограничениям приведены в оперативной справочной системе.

7.7.6 Разделы кода UDFB

Код внутри блока UDFB делится на два раздела (секции):

- Раздел начального кода (инициализация)
- Раздел основного кода

Раздел начального кода используется для описания и инициализации всех локальных переменных, которые будут использоваться в UDFB. Начальный раздел выполняется для каждого экземпляра UDFB при запуске или сбросе (это происходит до выполнения задачи DPL Initial).

ПРИМЕЧ. Входные и выходные аргументы блока UDFB нельзя использовать в начальном разделе UDFB.

В основном разделе находится фактический код функционального блока, который выполняет основную работу данной функции. Входные и выходные аргументы имеют смысл только внутри основного раздела кода блока.

Два этих раздела (секции) разделяются ключевым словом FBbody. Начальный код размещается до этого ключевого слова, основной код - после.

Не забывайте, что фактический код блока может состоять из смеси команд DPL, схем FBD и схем QuickLD.

Ниже приведен пример простого блока UDFB, который складывает два числа и масштабирует на predetermined величину (0.5):

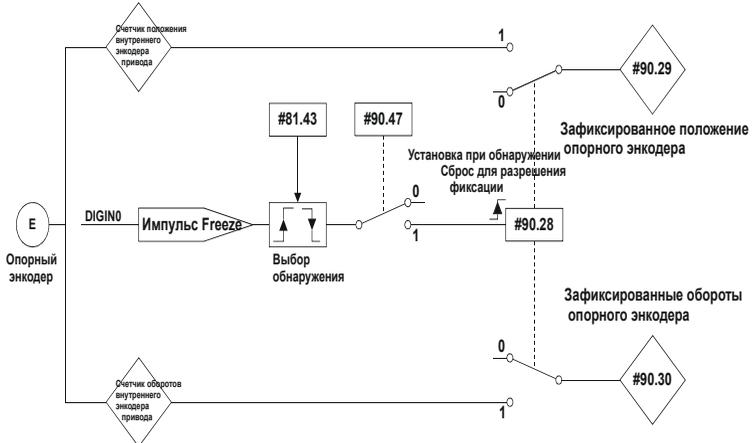
```
(output%) = _simplefb(input1%, input2%) {  
  // начальный код инициализации:  
  
  scale% = 500    // инициализируем локальную переменную  
  
  FBbody  
  // основной код блока:  
  
  output% = input1% + input2% * scale% / 1000  
  
}
```

8 Фиксация и маркер в SM-Applications

8.1 Вход фиксации Freeze

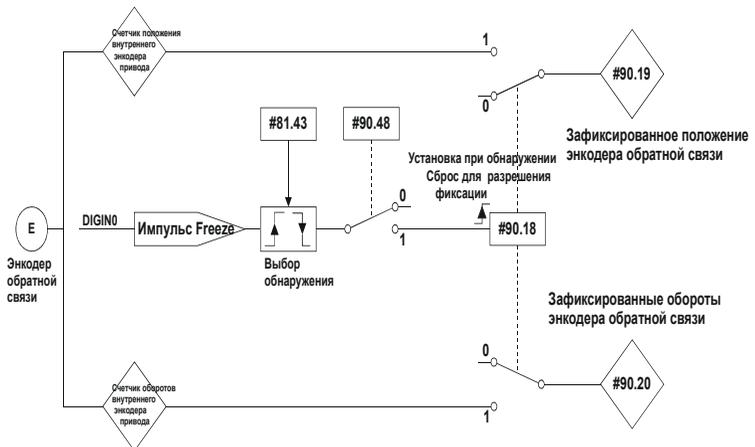
Цифровой вход 0 модуля SM-Applications (DIGIN0) можно использовать для фиксации (или “замораживания”) счетчиков опорного энкодера и энкодера обратной связи.

Рис. 8-1 Вход фиксации задания SM-Applications



Счетчик оборотов энкодера кэшируется в параметр #90.29, а положение энкодера кэшируется в параметр #90.30.

Рис. 8-2 Вход фиксации обратной связи SM-Applications



Счетчик оборотов энкодера кэшируется в параметр #90.19, а положение энкодера кэшируется в параметр #90.20.

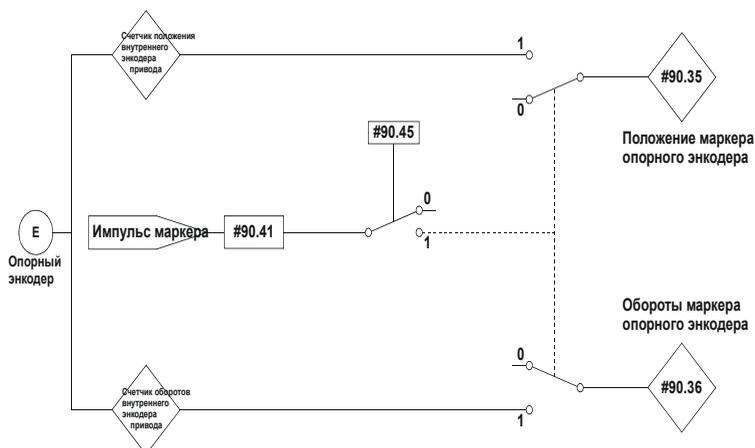
Положение фиксации обоих энкодеров (задания и обратной связи) можно захватывать на переднем или на заднем фронте импульса фиксации. Это определяется настройкой параметра #81.43 в 0 (передний фронт) или 1 (задний фронт). Параметр #81.42 позволяет записывать положение в параметры #90.19 и #90.29, а счетчик оборотов в параметры #90.20 и #90.30.

При поступлении импульса фиксации параметры #90.18 и #90.28 автоматически устанавливаются в 1, так что положение можно записать в #90.19 и #90.29, а счетчик оборотов можно записать в #90.20 и #90.30. Параметр #90.18 и #90.28 необходимо сбросить в 0, если пользователь желает обновить данные при следующем импульсе фиксации.

8.2 Импульс маркера

Модуль SM-Applications может кэшировать данные положения и счетчика оборотов в тот момент, когда импульс маркера поступает на энкодер задания (опорный) или на энкодер обратной связи.

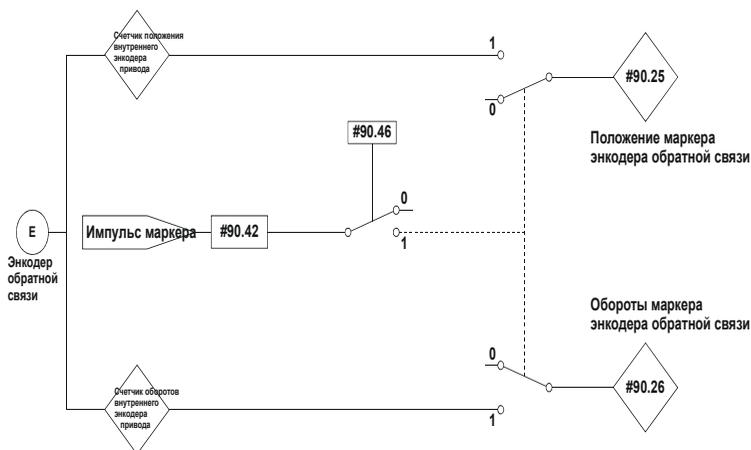
Рис. 8-3 Опорный маркер в SM-Applications



Положение маркера кэшируется в параметр #90.35, а счетчик оборотов маркера кэшируется в параметр #90.36.

Пользователь настраивает #90.41 в нуль, а привод настраивает #90.41 в 1 при КАЖДОМ обнаружении маркера. Пользователь должен обеспечить считывание данных маркера перед поступлением следующего импульса маркера.

Рис. 8-4 Маркер обратной связи в SM-Applications



Положение маркера кэшируется в параметр #90.25, а счетчик оборотов маркера кэшируется в параметр #90.26.

Пользователь настраивает #90.42 в нуль, а привод настраивает #90.42 в 1 при КАЖДОМ обнаружении маркера. Пользователь должен обеспечить считывание данных маркера перед поступлением следующего импульса маркера.

9.1 Обзор

Модуль SM-Applications можно использовать для синхронизации двух или нескольких приводов Unidrive SP. Этим обеспечивается, что приводы выполняют свои внутренние функции на точно одинаковой частоте и в одной шкале времени, то есть все операции выполняются одновременно.

Также из одного модуля (ведущий Master) в другие (ведомые Slave) по сети CTSync можно передать 3 значения данных. Они состоят из 2 x 32-битовых целых со знаком и 1 x 8-битового целого без знака.

Один (и только один) модуль SM-Applications должен быть настроен в качестве ведущего (Master), а все остальные модули должны быть настроены ведомыми (Slave), если им надо участвовать в работе схемы CTSync. Ведущий вырабатывает опорные данные, которые передаются всем ведомым по сети. Ведущего также можно настроить на работу в качестве ведомого, например, если нужно синхронизовать два привода. В этом случае ведущий будет вырабатывать опорные данные, а также следовать этим опорным данным. Ведомый также будет следовать этим опорным данным.

9.2 Подключения

Система SM-Applications CTSync работает при подключении портов RS485 модулей SM-Applications в сеть как в 2-проводном, так и в 4-проводном вариантах. Смотрите раздел 3.6 *Подключение к EIA-RS485*, где приведены сведения по подключению портов RS-485 модулей SM-Applications.

Для упрощения разводки линии передачи ведомого и линии приема ведущего можно не подключать в 4-проводном варианте сети. Так можно делать, поскольку ведущий не принимает ответа от ведомого.

9.3 Ограничения

- В сети может быть только один ведущий CTSync
- Все задачи Pos ведущего и ведомых CTSync должны быть настроены на одинаковое время обновления (параметр #81.12)
- Без применения повторителей линии можно соединить не более 8 узлов в 2-проводном и 16 узлов в 4-проводном вариантах сети.
- Максимальная длина кабеля сети RS485 составляет 200 метров.

9.4 Функциональные блоки CTSync

Для использования функциональности CTSync в модуле SM-Applications имеются шесть функциональных блоков. Они перечислены ниже.

9.4.1 CTSYNCSetMasterReferences

```
CTSYNCSetMasterReferences(Reference1%, Reference2%, AuxiliaryReference%)
```

Входные аргументы	
Имя аргумента	Диапазон
Reference1	32-битовый со знаком
Reference2	32-битовый со знаком
AuxiliaryReference	Беззнаковый 8-битный

Этот функциональный блок позволяет ведущему CTSync записывать опорные данные во все ведомые узлы сети CTSync. Эта команда не оказывает никакого действия при использовании на ведомом устройстве CTSync.

9.4.2 CTSYNCGetSlaveReferences

```
(Reference1%, Reference2%, AuxiliaryReference%, Status%) =  
CTSYNCGetSlaveReferences()
```

Выходные аргументы	
Имя аргумента	Диапазон
Reference1%	32-битовый со знаком
Reference2%	32-битовый со знаком
AuxiliaryReference%	Беззнаковый 8-битный

Этот функциональный блок позволяет CTSync считывать опорные данные, созданные ведущим CTSync. Его можно использовать как на ведущем, так и на ведомых.

Блок возвращает последние принятые от ведущего опорные значения.

Выходные аргументы:

- Status%: 1: ОК
- 0: Принято 0 или слишком мало данных.
- 1: Принято больше байтов, чем ожидалось.
- 2: Ошибка контрольной суммы в принятых данных.

Если Status не равен ОК, то значения Reference1%, Reference2% и AuxiliaryReference% не изменяются.

9.4.3 CTSYNCSetupOutputChannel

```
Result% = CTSYNCSetupOutputChannel(Channel%, Menu%, Parameter%)
```

Этот функциональный блок конфигурирует один из 3 каналов на маршрутизацию всех проходящих данных в указанный параметр привода.

Входные аргументы:

- Channel%: 1, 2 или 3 для 3 доступных "каналов".
- Menu%: Номер меню привода для записи.
- Parameter%: Номер параметра привода для записи.

Выходные аргументы:

- Result%: 1: Операция завершена успешно.
- 1: Указан неверный канал.
- 3: Указанный канал используется другой задачей.
- 4: Параметр не существует или только для чтения

9.4.4 CTSYNCEnableOutputChannel

```
Result% = CTSYNCEnableOutputChannel(Channel%)
```

Этот функциональный блок позволяет указанному каналу записывать свои данные в привод Unidrive SP в начале каждой выборки положения подвижного агрегата.

Входные аргументы:

Channel%: 1, 2 или 3 для 3 доступных "каналов".

Выходные аргументы:

Result%: 1: Операция завершена успешно.
0: Канал настроен неправильно.
-1: Указан неверный канал.
-3: Указанный канал используется другой задачей.

9.4.5 CTSYNCDisableOutputChannel

```
Result% = CTSYNCDisableOutputChannel(Channel%)
```

Этот функциональный блок заставляет указанный канал прекратить запись своих данных в привод Unidrive SP.

Входные аргументы:

Channel%: 1, 2 или 3 для 3 доступных "каналов".

Выходные аргументы:

Result%: 1: Операция завершена успешно.
-1: Указан неверный канал.
-3: Указанный канал используется другой задачей.

9.4.6 CTSYNCWriteOutputChannel

```
Result% = CTSYNCWriteOutputChannel(Channel%, value%)
```

Этот функциональный блок записывает значение в указанный канал.

Входные аргументы:

Channel%: 1, 2 или 3 для 3 доступных "каналов".
Value%: Записываемое значение.

Выходные аргументы:

Result%: 1: Операция завершена успешно.
0: Канал настроен неправильно.
-1: Указан неверный канал.
-3: Указанный канал используется другой задачей.

Если записываемое значение превышает возможный диапазон параметра, то в приводе выполняется отключение (tr44), если #81.14=1 и #81.17=1, или значение обрезается до диапазона, если любой из этих параметров настроен в 0.

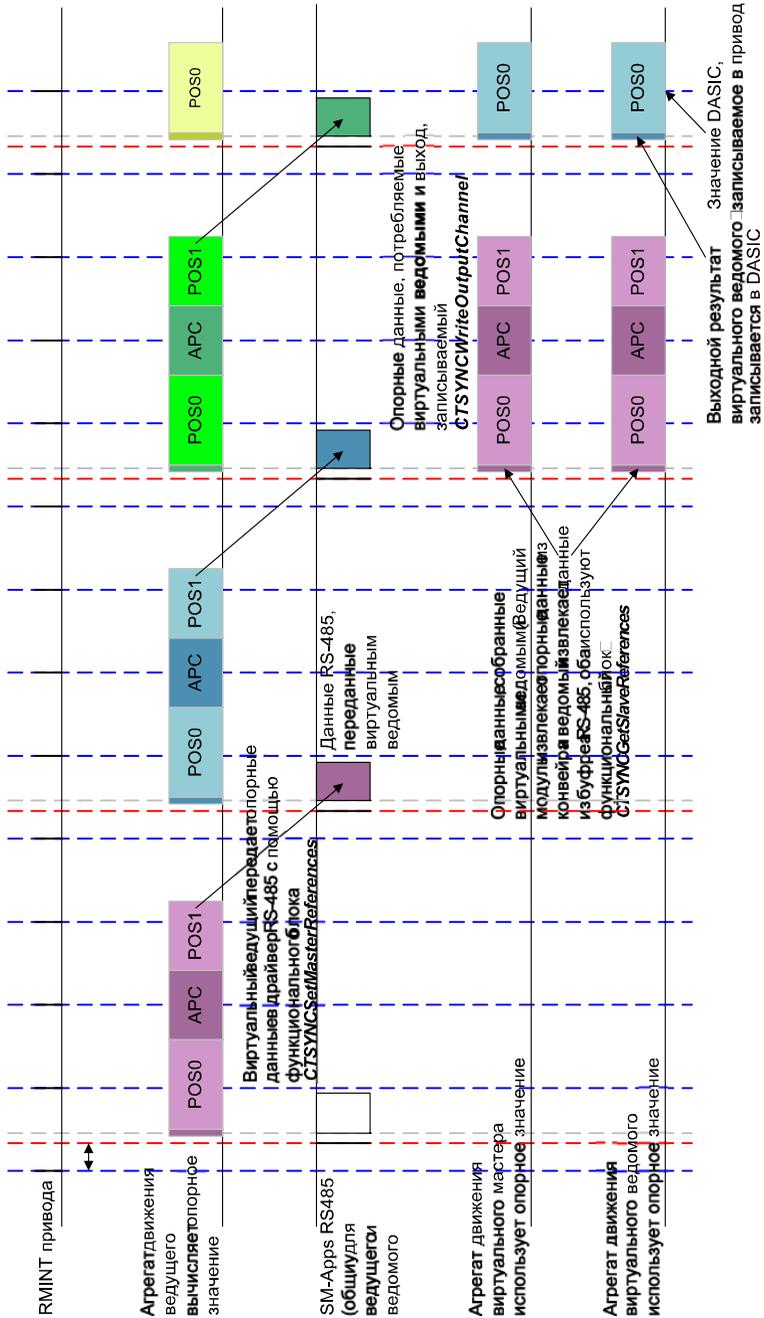
9.5 Подвижный агрегат

Ниже показаны задачи подвижного агрегата для каждой выборки подвижного агрегата:

1. Агрегат движения ведущего вычисляет опорное значение.
2. Ведущий с помощью функционального блока **CTSYNCSetMasterReferences** передает опорные данные в драйвер RS-485.
3. Данные RS-485 передаются ведомому (ведомым).
4. Ведомый с помощью функционального блока **CTSYNCGetSlaveReferences** извлекает данные.
5. С помощью функционального блока **CTSYNCWriteOutputChannel** извлеченные данные записываются в указанные каналы (если это нужно).
6. Выходные значения ведомого записываются в параметры привода с помощью ASIC привода.

Информация о синхронизации приведена на Рис. 9-1 *Синхронизация подвижного агрегата*

Рис. 9-1 Синхронизация подвижного агрегата



9.6 Пример виртуального ведущего

9.6.1 Пример кода ведущего

Следующий код показывает, как следует настроить ведущего для выработки опорных данных и передачи данных ведомым. Также показано, как ведущий работает в качестве ведомого, используя выработанные им опорные данные. Это выполняется с помощью функционального блока **CTSYNCGetSlaveReferences**.

В этом примере вырабатываются опорные данные ведущего для треугольной рампы, а также реализован виртуальный ведомый, который выводит данные в привод.

```
Initial
{
    #81.06=25 //Ведущий CTSync Master
    #81.12=2 //Задача Pos через 500 мксек
    REINIT

    //Настроим опорное значение рампы виртуального ведущего.
    Ramp% = 0

    //Сконфигурируем выходной канал ведомого 1 в меню 18, параметр 11.
    CTSYNCSetupOutputChannel(1,18,11)

    //Включим сконфигурированный выходной канал 1.
    CTSYNCEnableOutputChannel(1)
}

POS0
{
    //Сначала выполним функцию ведомого, так что
    //у ведомого и ведущего будет идентичная синхронизация:
    (Slaveref1%,Slaveref2%,Auxref%, Status%) = CTSYNCGetSlaveReferences()

    //Проверим Status%, чтобы убедиться в правильности приема данных.
    if Status% = USR_VALUE_OK then

        //Вывод в привод в начале следующего периода движения.
        CTSYNCwriteOutputChannel(1,Slaveref1%)

    endif

    //Выполним функцию ведущего, в данном случае рампу:
    Ramp% = Ramp% + 1

    //В случае необходимости сбросим профиль рампы
    if Ramp% > 999 then

        //Вернем рампу назад в 0
        Ramp% = 0
    endif

    //Перешлем опорные данные ведомым. Используется только reference1.
    CTSYNCSetMasterReferences(Ramp%,0,0)
}
```

9.6.2 Пример кода ведомого

Следующий код показывает, как ведомый может читать опорные данные, выработанные виртуальным ведущим.

```
Initial
{
    #81.06=26 //ведомый CTSync Slave
    #81.12=2 //задача Pos через 500us
    REINIT

    //настроим опорное значение ramпы виртуального мастера.
    Ramp% = 0

    //Сконфигурируем выходной канал ведомого в меню 18, параметр 11.
    CTSYNCSetupOutputChannel(1,18,11)

    //включим сконфигурированный выходной .
    CTSYNCEnableOutputChannel(1)
}

POS0
{
    //Сначала выполним функцию ведомого, так что
    //у ведомого и ведущего будет идентичная синхронизация:
    (Slaveref1%,Slaveref2%,Auxref%, Status%) = CTSYNCGetSlaveReferences()

    // Проверим. что данные были приняты правильно.
    if Status% = USR_VALUE_OK then

        //Вывод в привод в начале следующего периода движения.
        CTSYNCWriteOutputChannel(1,Slaveref1%)
    endif
}
```

10 Синхронизация дополнительных модулей

10.1 Обзор

Схема синхронизации между дополнительными модулями создает механизм, позволяющий задачам управления положением в одном или нескольких модулях синхронизоваться с аналогичными задачами в другом модуле того же привода без использования внешних соединений.

Синхронизация выполняется за счет того, что один модуль вырабатывает сигнал запуска, один или несколько модулей принимают его. Такие модули называются Поставщиком и Потребителями соответственно. Сигнал запуска используется для изменения планирования агрегата POS.

Если модуль является Поставщиком, то запуск может проводиться с периодом 250 мксек, 500 мксек, 1 мсек, 2 мсек, 4 мсек или 8 мсек, то есть модули Поставщиков вырабатывают сигнал запуска в каждый период POS, как указано в параметре #81.12.

Эта схема является дополнением к схеме CTSync (смотрите раздел 9 CTSync, где приведена дополнительная информация), хотя ее можно использовать совместно с CTSync для синхронизации модулей в нескольких приводах Unidrive SP (смотрите раздел 10.3 *Пример 2 синхронизации дополнительных модулей*).

Если модуль SM-Applications является Потребителем, то он может запускать свои задачи управления положением с такой же периодичностью или быстрее, чем сигнал синхронизации (запуска) от Поставщика.

Если модуль SM-Applications является Потребителем и его задача управления положением настроена на работу медленнее, чем период поступления сигнала синхронизации (запуска) от Поставщика, то тогда модуль не сможет определить фазу Поставщика и поэтому он не может выполнять попыток синхронизации с сигналом Поставщика (смотрите раздел 10.3 *Пример 2 синхронизации дополнительных модулей*).

10.2 Пример 1 синхронизации дополнительных модулей

В этом примере показана ситуация, когда нужно синхронизовать задачи POS0 в модулях в приводе Unidrive SP.

В следующей таблице показаны настройки для каждого из трех модулей:

Таблица 10.1 Настройки параметров

	Параметр	Величина	Описание
Гнездо 1	#81.06	1 (по умолчанию)	Нет в схеме CTSync
	#81.12	2	Период задачи управления положением 500 мксек
	#91.21	2	Потребитель синхронизации между модулями
	#91.22	6	<i>Достигнуто состояние Потребителя синхронизации между модулями</i>
Гнездо 2	#81.06	1 (по умолчанию)	Нет в схеме CTSync
	#81.12	1	Период задачи управления положением 250 мксек
	#91.21	2	Потребитель синхронизации между модулями
	#91.22	6	<i>Достигнуто состояние Потребителя синхронизации между модулями</i>

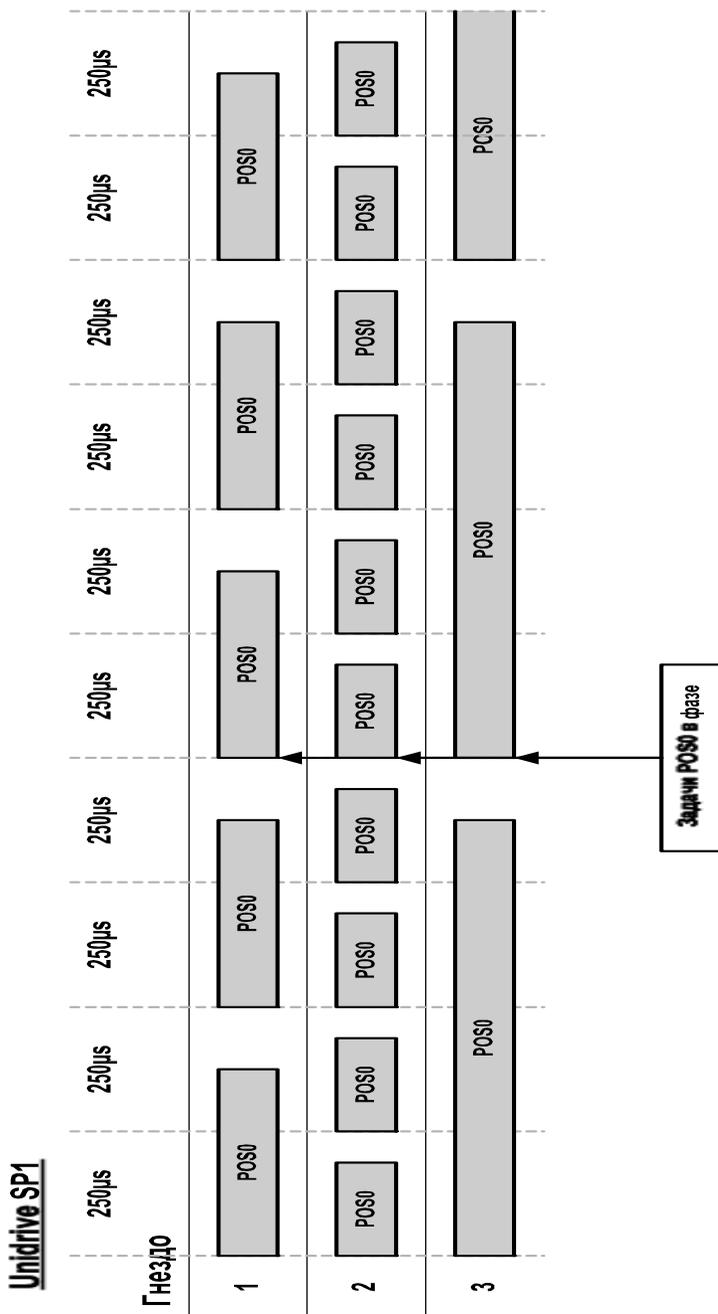
Таблица 10.1 Настройки параметров

	Параметр	Величина	Описание
Гнездо 3	#81.06	1 (по умолчанию)	Нет в схеме CTSync
	#81.12	3	Период задачи управления положением 1 мсек
	#91.21	1	Поставщик синхронизации между модулями
	#91.22	5	<i>Достигнуто состояние Поставщика синхронизации между модулями</i>

Выделенный курсивом параметр #91.22 - это параметр состояния, его не нужно настраивать. Он указывает, достиг ли модуль функциональности синхронизации между модулями, запрошенной в параметре #91.21. Более подробные сведения приведены в описании параметра #91.22.

Перед показанной выше настройкой параметров задачи POS могут выполняться с разбросом фазировки по отношению друг к другу, как показано на Рис. 10-1 *Выполнение задачи POS ДО синхронизации между модулями*. После настройки параметров в показанные выше значения задачи POS будут выполняться в фазе друг с другом, как показано на Рис. 10-2 *Выполнение задачи POS ПОСЛЕ синхронизации между модулями*.

Рис. 10-2 Выполнение задачи POS ПОСЛЕ синхронизации между модулями



10.3 Пример 2 синхронизации дополнительных модулей

В этом примере показана ситуация, когда нужно синхронизовать задачи POS0 в модулях в двух привода Unidrive SP.

Для этого работу модуля в каждом приводе нужно сконфигурировать так, чтобы он участвовал в схеме CTSync (смотрите раздел 9 CTSync, где приведена дополнительная информация). Один модуль должен быть сконфигурирован как ведущий CTSync Master, а один модуль должен быть сконфигурирован как ведомый CTSync Slave.

В следующих таблицах показаны настройки для каждого из 6 модулей:

Таблица 10.2 Настройки параметров для Unidrive SP 1

	Параметр	Величина	Описание
Гнездо 1	#81.06	1 (по умолчанию)	Нет в схеме CTSync
	#81.12	2	Период задачи управления положением 500 мксек
	#91.21	2	Потребитель синхронизации между модулями
	#91.22	6	<i>Достигнуто состояние Потребителя синхронизации между модулями</i>
Гнездо 2	#81.06	1 (по умолчанию)	Нет в схеме CTSync
	#81.12	1	Период задачи управления положением 250 мксек
	#91.21	2	Потребитель синхронизации между модулями
	#91.22	6	<i>Достигнуто состояние Потребителя синхронизации между модулями</i>
Гнездо 3	#81.06	25 (по умолчанию)	Ведущий CTSync Master
	#81.12	3	Период задачи управления положением 1 мсек
	#91.21	1	Поставщик синхронизации между модулями
	#91.22	5	<i>Достигнуто состояние Поставщика синхронизации между модулями</i>

Таблица 10.3 Настройки параметров для Unidrive SP 2

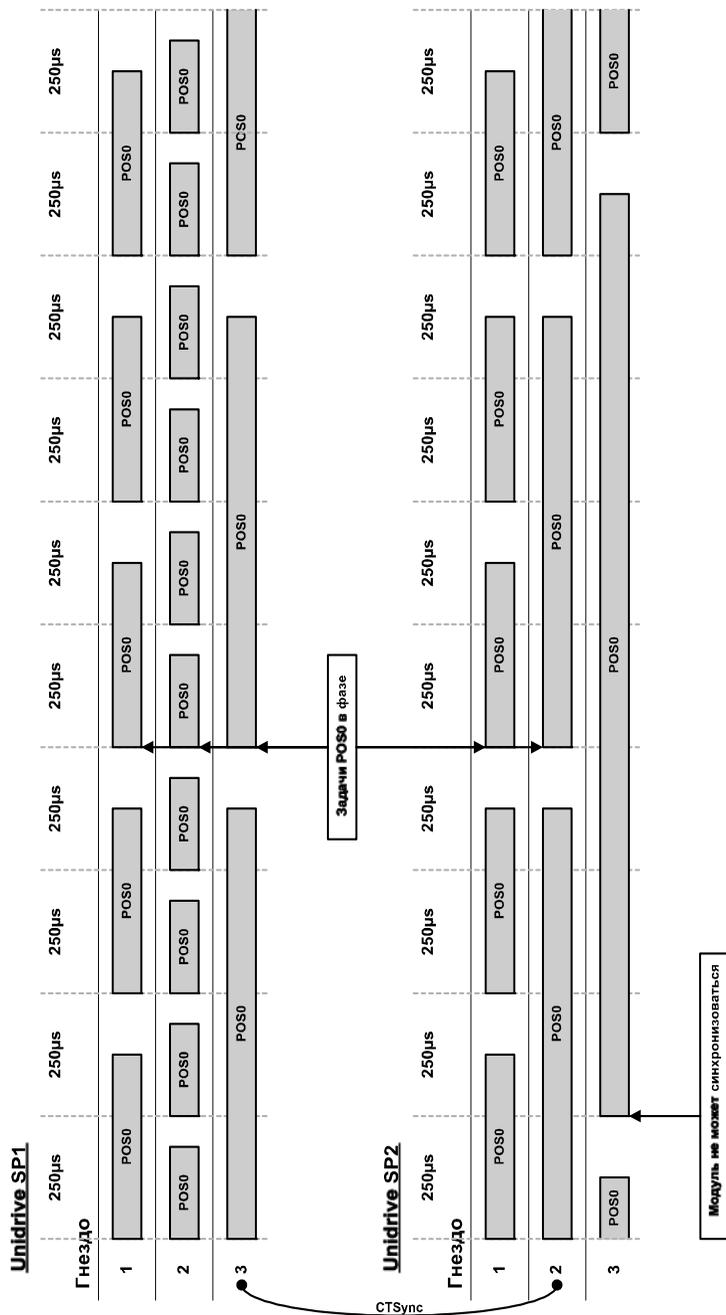
	Параметр	Величина	Описание
Гнездо 1	#81.06	1 (по умолчанию)	Нет в схеме CTSync
	#81.12	2	Период задачи управления положением 500 мксек
	#91.21	2	Потребитель синхронизации между модулями
	#91.22	6	<i>Достигнуто состояние Потребителя синхронизации между модулями</i>
Гнездо 2	#81.06	26	Ведомый CTSync Slave
	#81.12	3	Период задачи управления положением 1 мсек
	#91.21	1	Поставщик синхронизации между модулями
	#91.22	13	<i>Достигнуто состояние Поставщика синхронизации между модулями по сигналу DINT</i>
Гнездо 3	#81.06	1 (по умолчанию)	Нет в схеме CTSync
	#81.12	4	Период задачи управления положением 2 мсек
	#91.21	2	Потребитель синхронизации между модулями
	#91.22	2	<i>Состояние Потребителя синхронизации между модулями не достигнуто (слишком медленная задача POS)</i>

Выделенный курсивом параметр #91.22 - это параметр состояния, его не нужно настраивать. Он указывает, достиг ли модуль функциональности синхронизации между модулями, запрошенной в параметре #91.21. Более подробные сведения приведены в описании параметра #91.22..

ПРИМЕЧ. Если модуль является ведомым CTSync Slave, то он должен быть Поставщиком синхронизации между модулями на приводе, в котором он установлен, поэтому параметр #91.21 автоматически настраивается в значение 1.

Перед показанной выше настройкой параметров задачи POS могут выполняться с разбросом фазировки по отношению друг к другу, как показано на Рис. 10-3 *Выполнение задачи POS ДО синхронизации между модулями по CTSync*. После настройки параметров в показанные выше значения задачи POS будут выполняться в фазе друг с другом, как показано на Рис. 10-4 *Выполнение задачи POS ПОСЛЕ синхронизации между модулями по CTSync*.

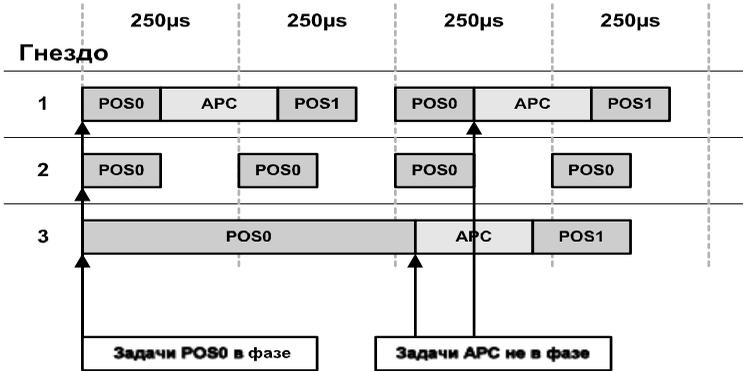
Рис. 10-4 Выполнение задачи POS ПОСЛЕ синхронизации между модулями по CTSync



10.4 Задачи управления положением

Задачи управления положением - это задачи POS0, APC и POS1. Эти задачи выполняются в указанном порядке (сначала выполняется POS0, затем выполняется APC и затем POS1). Поскольку задачи POS0 и APC могут не иметь одинакового времени для выполнения на разных модулях, то задачи APC и POS1 нельзя сфазировать. Поэтому при использовании схемы синхронизации между модулями пользователь должен понимать, что все задачи управления положением в целом в каждом модуле запускаются сфазировано (одновременно), но это не относится к отдельным задачам. На Рис. 10-5 *Задачи управления положением* показан пример такой ситуации.

Рис. 10-5 Задачи управления положением



11 Диагностика

В этой главе описаны следующие вопросы:

- Ошибки времени выполнения и коды отключения
- Обработка ошибок времени выполнения
- Состояние сети CTNet
- Техническая поддержка

11.1 Ошибки времени выполнения

Ошибка времени выполнения - это ошибка, которая возникает при выполнении в модуле SM-Applications конкретной операции. Она может возникнуть как результат ошибки выполнения программы DPL пользователя (например, попытка записи в отсутствующий параметр, или попытка деления значения на ноль), как результат неправильной настройки (например, неверная настройка CTNet) или как результат системной ошибки, например, перегрузка процессора или таймаут *сторожевого таймера*.

При появлении ошибки может быть выполнено одно или несколько из следующих действий:

- Программа пользователя может быть остановлена или может быть запрещен ее запуск
- Привод может отключиться с кодом **SLx.Er** (где x - это номер гнезда), причем код ошибки времени выполнения помещается в параметр #MM.50
- Привод может отключиться с другим кодом **SLx.****
- Может быть выполнена задача DPL **ERROR** (если она существует).

Что из этого будет выполняться, зависит от типа ошибки и от настройки глобального параметра разрешения отключения по ошибке времени выполнения #MM.14. Это подробно описано в разделе 11.3 ниже.

11.2 Просмотр кодов отключения Unidrive SP

В таблице ниже показаны возможные коды отключения, которые будут отображаться на панели привода Unidrive SP, если в модуле SM-Applications обнаружена ошибка, вызывающая отключение привода. Не забывайте, что не все ошибки времени выполнения приводят к отключению привода.

Таблица 11.1 Коды просмотра отключения Unidrive SP

Гнездо, вызвавшее отключение			Ошибка	Описание
Гнездо 1	Гнездо 2	Гнездо 3		
SL1.HF	SL2.HF	SL3.HF	Аппаратная ошибка	Unidrive SP обнаружил наличие дополнительного модуля, но не может связаться с ним.
SL1.tO	SL2.tO	SL3.tO	Таймаут сторожев. таймера	Указывает, что программа пользователя, использующая функцию <i>сторожевого таймера</i> , не смогла подать команду WDOG в течение 200 мсек.

Таблица 11.1 Коды просмотра отключения Unidrive SP (Продолжение)

Гнездо, вызвавшее отключение			Ошибка	Описание
Гнездо 1	Гнездо 2	Гнездо 3		
SL1.Er	SL2.Er	SL3.Er	Ошибка	Отключение времени выполнения, вызванное модулем SM-Applications либо из-за ошибки в программе DPL пользователя, либо по другой причине. Фактический код ошибки помещается в параметр #MM.50.
SL1.nF	SL2.nF	SL3.nF	Не установлен	Модуль SM-Applications был отсоединен во время работы или модуль SM-Applications вышел из строя. Это отключение может также возникнуть, если гнездо Unidrive SP настроено для модуля SM-Applications, но модуль не установлен в это гнездо.
SL1.dF	SL2.dF	SL3.dF	Установлен другой	Это отключение возникает, если модуль SM-Applications установлен в гнездо, где раньше стоял другой модуль, или если он установлен в ранее не использовавшееся гнездо.

11.3 Коды ошибок времени выполнения SM-Applications

Если модуль SM-Applications обнаруживает ошибку во время работы, то код ошибки помещается в следующий параметр:

#81.50	Код ошибки SM-Applications		
Доступ	RO	Диапазон	0 до 255
По умолчанию	Отсутствует	Скорость обновления	По ошибке

Для определенных ошибок пользователь может выбрать, что должен отключиться и привод Unidrive SP. Это настраивается в параметре разрешения отключения по глобальной ошибке времени выполнения:

#81.14	Разрешение отключения по глобальной ошибке времени выполнения		
Доступ	RW	Диапазон	0/1
По умолчанию	0	Скорость обновления	Отсутствует

Если этот параметр настроен в 1 (On), то привод будет отключаться по ВСЕМ ошибкам времени выполнения.

В таблице ниже указаны коды ошибок и их значения, а также то, будет ли отключаться привод, остановится ли программа пользователя, и будет ли запущена задача DPL ERROR.

Примечания:

Таблица 11.2 Коды ошибок модуля SM-Applications (Продолжение)

Код ошибки	Причина	Отключ. привод?	Задача ERROR?	Останов прог?
58	Испорчено содержимое флэш-памяти ПЛК. Ошибка возникает при запуске и означает, что набор регистров ПЛК (P/Q/T/U) и меню 20 не были восстановлены. Если эта неисправность повторяется, то это может быть вызвано поломкой аппаратуры, так что обращайтесь к поставщику.	Да	Да	Не раб.
59	Привод не принял модуль в качестве ведущего синхронизации	Да	Да	Да
60	Отказ аппаратуры CTNet. Обратитесь к вашему поставщику.	Может	Нет	Нет
61	Неверная конфигурация CTNet. Проверьте все параметры конфигурации	Может	Нет	Нет
62	Неверная скорость CTNet. Проверьте #MM.24 и сетевые подключения.	Может	Нет	Нет
63	Неверный идентификатор узла CTNet. Проверьте #MM.23	Может	Нет	Нет
64	Перегрузка цифрового выхода. При этом оба цифровых выхода переводятся в пассивный режим и остаются пассивными, пока не будет устранено условие перегрузки. Порог отключения составляет 20 мА.	Да	Да	Да
65	Неверные параметры функционального блока. Вы вызвали функц. блок в программе DPL, но один или несколько входных аргументов указаны неверно.	Да	Да	Да
66	Куча пользователя слишком велика. Программа скомпилирована для целевого модуля, в котором больше памяти ОЗУ, чем в данном модуле. Происходит при запуске.	Да	Нет	Не раб.
67	Файл ОЗУ не существует или был указан id не для файла ОЗУ.	Да	Да	Да
68	Указанный файл ОЗУ не ассоциирован с массивом.	Да	Да	Да
69	Ошибка обновления кэша базы данных параметров привода в памяти флэш.	Да	Нет	Не раб.
70	Программа пользователя загружена при включенном (разрешенном) приводе. Произойдет, если #MM.37 = 1 и загружена программа.	Может	Нет	Да
71	Отказ изменения режима привода	Да	Нет	Да
72	Неверная операция буфера CTNet	Да	Да	Да
73	Отказ быстрой инициализации параметра	Да	Нет	Нет
74	Превышение температуры	Да	Да	Да
75	Не доступна аппаратура. Программа пользователя пытается получить доступ к отсутствующей аппаратуре, например, к цифровому I/O, порту RS485 или CTNet в модуле SM-Applications Lite.	Да	Да	Да
76	Нельзя определить тип модуля. Модуль не распознан.	Да	Нет	Не раб.
77	Ошибка связи между модулями и модулем в гнезде 1.	Да	Да	Да
78	Ошибка связи между модулями и модулем в гнезде 2.	Да	Да	Да
79	Ошибка связи между модулями и модулем в гнезде 3.	Да	Да	Да
80	Ошибка связи между модулями и модулем в неизвестном гнезде.	Да	Да	Да
81	Внутренняя ошибка APC. Смотрите параметр #81.38 (например, использование неинициализированной функции CAM).	Может	Да	Да
82	Отказ связи с приводом.	Может	Да	Да

11.4 Обработка ошибок времени выполнения в задаче ERROR

Определенные ошибки времени исполнения могут запускать задачу DPL ERROR, если она имеется. Это предоставляет удобный метод безопасной обработки условий ошибки и выполнения необходимых действий, например, управляемого останова системы или аварийной сигнализации об ошибке.

Если запускается задача ERROR, то все другие задачи DPL будут остановлены. Поэтому задача ERROR имеет исключительные права выполнения. После завершения задачи ERROR программа DPL заканчивает работу и не работает ни одна другая задача DPL (хотя можно сбросить и перезапустить программу - более подробно это описано ниже).

ПРИМЕЧ. Отключения привода не запускают задачу ERROR. Ее могут запустить только некоторые ошибки программы DPL.

В задаче ERROR можно использовать все стандартные команды DPL, а также большинство функциональных блоков. Можно обращаться ко всем параметрам привода и модуля SM-Applications.

Код ошибки времени выполнения можно определить по следующему параметру:

#88.01	Состояние / Сброс ошибки		
Доступ	RW	Диапазон	0 до 9999
По умолчанию	Отсутствует	Скорость обновления	По ошибке

Этот параметр имеет две функции. При чтении он возвращает код ошибки времени выполнения, такой же, как в #81.50 (внимание - он не возвращает кодов отключения привода). Этот параметр сбрасывается в нуль при сбросе и при запуске программы пользователя.

Если в этот параметр записать значение 1070, то модуль SM-Applications начнет мягкий (теплый) перезапуск привода и всех других дополнительных модулей. Это можно использовать для перезапуска программы пользователя (при условии, что параметр автозапуска #81.13=1) и очистки состояния отключения привода. Такую операцию сброса можно выполнять в любой момент времени, не только после ошибки времени выполнения и не только в задаче ERROR.



При записи 1070 в параметр #88.01 автоматически очищается любое отключение привода, а также выполняется сброс всех дополнительных модулей в приводе Unidrive SP. Это поведение отличается от изделия UD70 на Unidrive 1, где сброс привода не выполнялся.

Задачу, вызвавшую ошибку времени выполнения, можно определить при чтении параметра #88.02, как описано выше.

Если пользователь желает отключить привод (если он еще не был отключен), то в параметр #10.38 надо записать соответствующий код отключения.

11.5 Техническая поддержка

При обращении к вашему поставщику за технической поддержкой надо всегда подготовить для него информацию, взяв ее из следующих параметров.

11.5.1 Микропрограмма модуля

#81.02	Микропрограмма - старшая версия		
Доступ	RO	Диапазон	00.00 до 99.99
По умолчанию	Отсутствует	Скорость обновления	Отсутствует

#81.51	Микропрограмма - младшая версия		
Доступ	RO	Диапазон	0 до 99
По умолчанию	Отсутствует	Скорость обновления	Отсутствует

Полную версию микропрограммы модуля SM-Applications можно прочитать из соответствующего гнезда. Это руководство написано для модулей SM-Applications с микропрограммой версии V01.03.02. В таблице ниже показано, как образовать полный номер версии микропрограммы по значениям двух этих параметров.

Таблица 11.3 Версия микропрограммы

Старшая версия	Младшая версия	Версия микропрограммы
1.01	5	V01.01.05

12 Руководство по переносу программ

В этой главе описаны некоторые основные отличия между изделием UD70 для Unidrive 1 и модулем SM-Applications для Unidrive SP, которые могут оказаться полезными при преобразовании (переносе) программ DPL пользователя.

12.1 Различия в параметрах привода

Набор параметров модуля Unidrive SP имеет небольшое число отличий от набора параметров Unidrive 1. Все параметры, к которым есть доступ в программе DPL, в идеальном случае должны быть сконфигурированы операторами `#define` в начале программы, что упрощает преобразование.

Поэтому при переносе программы вы должны внимательно просмотреть текст программы и найти все ссылки на параметры привода. Прежде всего следует искать символ `#`, а затем просмотреть всю программу в поисках функциональных блоков, которые могут иметь аргументами указатели на параметры (например, `PFixRead` или `WRNET`).

После нахождения всех параметров убедитесь, что они такие же, как в Unidrive SP и выражены в тех же единицах. При необходимости измените номер параметра или примените масштабирование.

12.2 Параметры UD70

12.2.1 Параметры настройки

Параметры настройки теперь располагаются в меню 15, 16 и 17, в то время как в UD70 они всегда размещались в меню 17. Настоятельно рекомендуется, чтобы новые программы DPL пользователя, которые изменяют параметры настройки, были переделаны на использование меню 81, что устраняет необходимость определять, какое именно из меню 15/16/17 используется.

В таблице ниже приведены некоторые основные отличия между двумя меню настройки:

Таблица 12.1 Изменения параметров настройки

Параметр	Описание	Изменения
#17.03	Номер строки с ошибкой	Теперь это #81.48.
#17.06	Режим RS485	Режимы 2,3,4,9,10,11,12 больше не поддерживаются. Режимы пользователя 6,7,8 не поддерживаются в текущей версии программного обеспечения.
#17.08	Указатель RS485	Не поддерживается. Теперь это параметр задержки переключения
#17.09	Указатель 2 RS485	Не поддерживается. Теперь это параметр задержки включения TX
#17.10	Коэф. масштаба RS485	Не поддерживается. Теперь это параметр маршрута печати DPL
#17.11	Время такта синхронизации	Для версий привода < 01.05.00 значение по умолчанию равно 0. Для версий привода >= 01.05.00 значение по умолчанию равно 10 мсек.
#17.12	Контроллер положения	Этот параметр теперь управляет периодом активизации новой задачи POS0/1, которая отличается от подобных задач скорости и энкодера UD70.
#17.15	Режим отключения RS485	Не поддерживается. Сейчас параметр не используется.

Таблица 12.1 Изменения параметров настройки (Продолжение)

Параметр	Описание	Изменения
#17.16	Источник синхронизации канала IO	Канал IO не поддерживается. Теперь параметр используется для выбора периода обновления энкодера.
#17.20	Сохранение во флэш при отключении питания	Этот параметр теперь действует сразу же.
#17.21	Отключить MD29MON	Этот параметр теперь управляет сохранением меню 20.
#17.22	RS232 привод-привод	Не поддерживается. Этот параметр теперь определяет идентификатор маркерного кольца CTNet.
#17.23	Управление запуском EVENT	Смотрите #81.35. Этот параметр теперь идентификатор CTNet.
#17.24	Переключение 2-проводного ANSI	Смотрите параметр #81.08. Теперь параметр используется для CTNet.
#17.25+	Не используются	Теперь имеется 51 параметр настройки.

12.2.2 Меню 18/19

Эти меню остались неизменными.

12.2.3 Меню 20

- Меню 20 теперь состоит из 40 параметров, а не из 50. Параметры с #20.21 по #20.40 теперь являются полными 32-битовыми параметрами со знаком.
- Ранее первые 20 параметров меню 20 считались зарезервированными для модулей передачи данных. Теперь это не так, поскольку параметры связи собраны в меню 15/16/17, следовательно, все меню 20 теперь свободно.
- Также изменено сохранение меню 20. Для сохранения меню надо сначала установить параметр #81.21 в 1 и затем установить #81.19 или #81.20. Причина такого изменения в том, что теперь в приводе может быть установлено два или три модуля SM-Applications и поскольку меню 20 сохраняется и восстанавливается из флэш памяти модуля, то только один модуль должен быть настроен на работу с меню, иначе оно будет неправильно восстановлено.

12.2.4 Регистры ПЛК

- Хорошие новости - теперь имеются дополнительно 200 **сохраняемых** регистров. Это банки регистров T и U (или меню 74 и 75). Регистры P/Q/R/S остаются такими же, как раньше, однако поскольку теперь нет встроенного контроллера положения, набор регистров Q свободен для использования.
- На UD70 с CTNet были зарезервированы некоторые регистры из наборов R и S. Это больше не так - все 100 регистров обоих наборов можно использовать для ссылок на циклические данные CTNet (внимание - 3 регистра в R и S все еще зарезервированы для ссылок на циклические данные простого режима).

12.2.5 Параметры меню 90

В частях этого меню имеются значительные изменения и программы НУЖНО изменить, чтобы учесть эти изменения. Параметры положения энкодера #90.01/ #90.03 теперь предоставляют полную информацию о положении (включая точное положение) промасштабированную так, что полное 32-битное значение соответствует одному обороту. Отдельные счетчики оборотов доступны в #90.02 и #90.04.

12.3 Общие функции

12.3.1 Аппаратура

- CTNet стандартно имеется на модуле.
- Порт программирования RS232 больше недоступен. Программирование теперь выполняется по протоколу CT-RTU через последовательный порт RJ45 на передней панели привода, или через CTNet. (Старая программа DPL Toolkit теперь не поддерживается.)
- Цифровой I/O использует теперь уровни 24 В положительной логики, а не ТТЛ. Теперь есть два цифровых выхода вместо только одного. Также теперь цифровой I/O более доступен для работы за счет использования винтовых клемм.

12.3.2 Язык DPL

DPL сохраняет обратную совместимость. Были внесены следующие улучшения:

- Добавлены новые конструкции языка, например, FOR...LOOP и SELECT...CASE.
- Вложенность: Размер стека теперь выделяется под задачи и он больше, чем у UD70, так что возможен больший уровень глубины вложений.
- Был добавлен новый тип данных - плавающая запятая с одинарной точностью, который имеет преимущества малого времени исполнения по сравнению с двойной точностью (этот тип используется по умолчанию). Одинарная или двойная точность указывается как глобальная опция программы с помощью новой директивы \$FLT SINGLE в заголовке программы.

ПРИМЕЧ. Тип данных с одинарной точностью не является прямой заменой для двойной точности и пользователи должны проверить, что одинарной точности вполне достаточно для их приложений.

12.3.3 Математические функции

- Сложность выражений теперь существенно повышена, в UD70 вы были ограничены тем, что вы сможете сделать в одном выражении. Теперь вы можете создавать более сложные выражения.
- Добавлен новый оператор преобразования типов TRUNC. Он преобразует плавающее в целое с обрезанием, а не с округлением дробной части.
- Сравнение значений с плавающей запятой: в UD70 для сравнения чисел с плавающей запятой использовался довольно неточный и непредсказуемый нечеткий алгоритм. Теперь в модуле SM-Applications применен стандартный метод сравнения, как в других языках программирования. Однако могут возникать неожиданные побочные эффекты. Например,

```
f = 1.2 * 3
IF f = 3.6 THEN
    // Результат будет не истина.
ENDIF
```

Для проверки на равенство или неравенство лучше использовать диапазон. Например:

```
f = 1.2 * 3
IF f > 3.59 or f < 3.61 THEN
    // Это будет истина.
ENDIF
```

12.3.4 Задачи

Были изменены задачи.

- Задачи ENCODER и SPEED больше не используются. Вместо них добавлены задачи POS0 и POS1. Также, задачу CLOCK можно использовать как замену задаче ENCODER. Задачи POS0, POS1 и CLOCK выполняются синхронно с приводом (как раньше задача SPEED), но теперь можно указывать кратные времена - от 250 мксек до 8 мсек для POS0 и POS1 и от 5 мсек до 200 мсек для задачи CLOCK. Можно по-прежнему использовать имена SPEED и ENCODER, поскольку это псевдонимы для POS0 и POS1 (это удобно при работе с SYPT Workbench V1, так как имеется проблема с точками прерываний в POS0/1), но не забывайте, что синхронизация **отличается** и что может потребоваться изменить код для учета этого. Также обратите внимание. что синхронизация не зависит от частоты ШИМ привода, как это было в UD70.
- Добавлены три новые задачи EVENT, но это не должно вызвать никаких проблем при переносе программ.
- Задачи EVENT теперь можно планировать из DPL с помощью нового функционального блока SCHEDULEEVENT.

12.3.5 Определяемые пользователем функциональные блоки

- В UD70 был предел на параметры - 10 целых входных и 10 целых выходных. Были также ограничения на выравнивание. В модуле SM-Applications этих ограничений нет. Число входных и выходных параметров ограничено только памятью. Не должно быть никаких проблем с переносом. (Примечание: утилита SYPT Workbench пока налагает все старые ограничения на программы, разрабатываемые для модуля SM-Applications).

12.3.6 Передача данных

CTNet

- Интерфейс CTNet теперь является стандартным (не опционным). Все параметры конфигурации были изменены. Синхроимпульсы вырабатываются немного по-другому в истинных миллисекундах, а не в ближайших кратных к 1.38 мсек, как в UD70. Также запуск задачи EVENT происходит как на ведущем, так и на ведомом узлах - UD70 не делал этого.
- Все параметры настройки переместились из меню 20 в меню опций (15/16/17/81).
- Специальные диагностические переменные, например, NOFMESSAGES, больше не поддерживаются. Вместо этого диагностическую информацию возвращает новый функциональный блок CTNETDIAGS. Старые программы с этими диагностическими переменными не компилируются и их надо изменить, используя новый функциональный блок.
- Запуск задачи EVENT по сообщению SYNC теперь также происходит и на узле, фактически вырабатывающем сообщение SYNC, а также на ведомых узлах.

Порт RS485

- Протокол CT-ANSI расширен до полной поддержки 32-битного чтения/записи.
- Режим “коробчатого” IO 10 больше не поддерживается.
- Режимы “привод с приводом” и каскадные 2, 3 и 4 не поддерживаются.
- Режим RS232 “привод с приводом” не поддерживается.
- Режим 9 не поддерживается из-за аппаратных ограничений.
- Режимы 11-12 не поддерживаются. Режим 11, который “обходил” программные буферы UD70, больше не нужен, поскольку программные буферы не используются в пользовательских режимах.
- Теперь доступны новые режимы ведущего Modbus-RTU и соответствующие команды.

12.4 Утилита SM-Applications Porting Aid

12.4.1 Обзор

Система SYPT Workbench v1.6.0+ теперь содержит утилиту Porting Aid, которая предупреждает пользователей о тех функциях, которые в модуле SM-Applications реализованы не так, как в модулях UD7X. Предупреждения будут показаны системой SYPT Workbench в окне ошибок. В этом окне ошибок будет указан номер строки, к которой относится информация об ошибке/переносе, и в случае двойного щелчка по номеру строки в окне ошибок SYPT Workbench покажет исходную строку DPL, содержащую измененный пункт. Такую утилиту переноса можно активировать, используя новую директиву \$PORTING сразу после строк главного заголовка программы:

```
$AUTHOR Kevin Vedmore
$COMPANY Control Techniques
$TITLE Test Program
$VERSION v1.0
$DRIVE UNIDRVSP
$PORTING
```

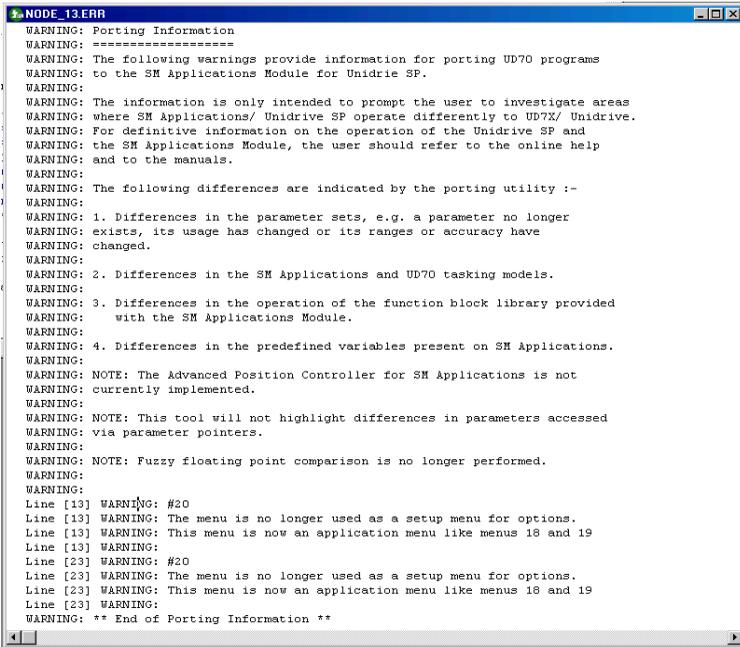
12.4.2 Выводимые различия

Система SYPT Workbench при использовании директивы \$PORTING может сообщить следующую информацию о различиях:

1. Задачи POS0 и POS1 будут рекомендованы вместо старых задач Speed, Torque и Encoder. Задача Clock будет рекомендована в качестве альтернативы использованию задачи Pos вместо старой задачи Encoder.
2. Имеются различия в наборах параметров Unidrive SP или SM-Applications, например, было изменено количество цифр после запятой, изменены имена или правила использования, а некоторые параметры больше не существуют или были перемещены.
Примечание: Конкретные предупреждения об изменении числа цифр после запятой будут показаны только в случае доступа к параметру с помощью директивы #INT. При доступе к параметру без использования #INT будет показано только общее предупреждение об изменении числа знаков после запятой.
3. Разная работа функциональных блоков RTL и OS.

Ниже показан пример сообщений, выводимых утилитой Porting Aid.

Рис. 12-1 Окно с информацией о переносе



```

NODE_13.ERR
WARNING: Porting Information
WARNING: =====
WARNING: The following warnings provide information for porting UD70 programs
WARNING: to the SM Applications Module for Unidrive SP.
WARNING:
WARNING: The information is only intended to prompt the user to investigate areas
WARNING: where SM Applications/ Unidrive SP operate differently to UD7X/ Unidrive.
WARNING: For definitive information on the operation of the Unidrive SP and
WARNING: the SM Applications Module, the user should refer to the online help
WARNING: and to the manuals.
WARNING:
WARNING: The following differences are indicated by the porting utility :-
WARNING:
WARNING: 1. Differences in the parameter sets, e.g. a parameter no longer
WARNING: exists, its usage has changed or its ranges or accuracy have
WARNING: changed.
WARNING:
WARNING: 2. Differences in the SM Applications and UD70 tasking models.
WARNING:
WARNING: 3. Differences in the operation of the function block library provided
WARNING: with the SM Applications Module.
WARNING:
WARNING: 4. Differences in the predefined variables present on SM Applications.
WARNING:
WARNING: NOTE: The Advanced Position Controller for SM Applications is not
WARNING: currently implemented.
WARNING:
WARNING: NOTE: This tool will not highlight differences in parameters accessed
WARNING: via parameter pointers.
WARNING:
WARNING: NOTE: Fuzzy floating point comparison is no longer performed.
WARNING:
WARNING:
Line [13] WARNING: #20
Line [13] WARNING: The menu is no longer used as a setup menu for options.
Line [13] WARNING: This menu is now an application menu like menus 18 and 19
Line [13] WARNING:
Line [23] WARNING: #20
Line [23] WARNING: The menu is no longer used as a setup menu for options.
Line [23] WARNING: This menu is now an application menu like menus 18 and 19
Line [23] WARNING:
WARNING: ** End of Porting Information **

```

Перевод текста сообщений в окне:

ПРЕДУПРЕЖДЕНИЕ:Информация о переносе
ПРЕДУПРЕЖДЕНИЕ:=====
ПРЕДУПРЕЖДЕНИЕ:Следующие предупреждения содержат сведения о переносе программ
ПРЕДУПРЕЖДЕНИЕ:UD70 в модуль SM Applications для Unidrive SP
ПРЕДУПРЕЖДЕНИЕ:
ПРЕДУПРЕЖДЕНИЕ:Эта информация имеет целью только предложить пользователю изучить
ПРЕДУПРЕЖДЕНИЕ:участки, где SM Applications/Unidrive SP работает не так, как UD7X/ Unidrive
ПРЕДУПРЕЖДЕНИЕ:Подробная информация о работе Unidrive SP и модуля SM Applications
ПРЕДУПРЕЖДЕНИЕ:приведена в системе оперативной справки и в руководствах.
ПРЕДУПРЕЖДЕНИЕ:
ПРЕДУПРЕЖДЕНИЕ:Утилита переноса указывает следующие различия:
ПРЕДУПРЕЖДЕНИЕ:
ПРЕДУПРЕЖДЕНИЕ:1. Различия в наборах параметров, например, параметр больше
ПРЕДУПРЕЖДЕНИЕ:не существует, его применение изменено или изменилась
ПРЕДУПРЕЖДЕНИЕ:его точность.
ПРЕДУПРЕЖДЕНИЕ:
ПРЕДУПРЕЖДЕНИЕ:2. Различия в моделях задач SM Applications и UD70.
ПРЕДУПРЕЖДЕНИЕ:
ПРЕДУПРЕЖДЕНИЕ:3. Различия в работе библиотеки функциональных блоков, поставляемых
ПРЕДУПРЕЖДЕНИЕ:вместе с модулем SM Applications.
ПРЕДУПРЕЖДЕНИЕ:
ПРЕДУПРЕЖДЕНИЕ:4. Различия в предопределенных переменных, имеющих в SM Applications.
ПРЕДУПРЕЖДЕНИЕ:
ПРЕДУПРЕЖДЕНИЕ:ПРИМЕЧАНИЕ: Улучшенный контроллер положения АРС для SM Applications
ПРЕДУПРЕЖДЕНИЕ:в настоящее время не реализован.
ПРЕДУПРЕЖДЕНИЕ:
ПРЕДУПРЕЖДЕНИЕ:ПРИМЕЧАНИЕ: Эта утилита не отмечает разницы в параметрах, доступ к
ПРЕДУПРЕЖДЕНИЕ:которым проводится через указатели на параметры.
ПРЕДУПРЕЖДЕНИЕ:
ПРЕДУПРЕЖДЕНИЕ:ПРИМЕЧАНИЕ: Больше не выполняется нечеткого сравнения чисел с
ПРЕДУПРЕЖДЕНИЕ:плавающей запятой
ПРЕДУПРЕЖДЕНИЕ:
Строка [13] ПРЕДУПРЕЖДЕНИЕ:#20

Строка [13] ПРЕДУПРЕЖДЕНИЕ: Это меню больше не используется как меню настройки опций.
Строка [13] ПРЕДУПРЕЖДЕНИЕ: Это меню сейчас находится в прикладном меню, как меню 18 и 19
Строка [13] ПРЕДУПРЕЖДЕНИЕ:
Строка [23] ПРЕДУПРЕЖДЕНИЕ:#20
Строка [23] ПРЕДУПРЕЖДЕНИЕ: Это меню больше не используется как меню настройки опций.
Строка [23] ПРЕДУПРЕЖДЕНИЕ: Это меню сейчас находится в прикладном меню, как меню 18 и 19
Строка [23] ПРЕДУПРЕЖДЕНИЕ:
ПРЕДУПРЕЖДЕНИЕ:** Конец информации по переносу **

13 Краткий справочник

Полное описание этих параметров приведено в разделе 5 *Параметры*.

Таблица 13.1 Параметры

Параметр	Описание	Диапазон	По умолчанию
#81.01	Код дополнительного модуля	0-499	Отсутств.
#81.02	Версия микропрограммы модуля	0-99.99	Отсутств.
#81.03	Состояние программы DPL	0-3	Отсутств.
#81.04	Доступные системные ресурсы %	0-100%	Отсутств.
#81.05	Адрес RS485	0-255	11
#81.06	Режим RS485	0-255	1
#81.07	Скорость в Бодах RS485	0-9	4
#81.08	Задержка переключения RS485	0-255 мс	2 мс
#81.09	Задержка включения Tx в RS485	0-1 мс	0 мс
#81.10	Маршрут печати DPL	0-1	0
#81.11	Длительность такта синхронизации (мс)	0-200 мс	0
#81.12	Период планирования задачи POS	0-6	0
#81.13	Включение автозапуска	0-1	1
#81.14	Включение глобального отключения по ошибке времени выполнения	0-1	0
#81.15	Зарезервировано	0	0
#81.16	Скорость обновления данных энкодера	0-3	0
#81.17	Разрешение отключений по превышению диапазона	0-1	0
#81.18	Включение сторожевого таймера	0-1	0
#81.19	Запрос сохранения	0-1	0
#81.20	Сохранение при отключении 'UU'	0-1	0
#81.21	Включить меню 20 в операции сохранения/восстановления	0-1	0
#81.22	Идентификатор маркерного кольца CTNet	0-255	0
#81.23	Адрес узла CTNet	0-255	0
#81.24	Скорость в Бодах CTNet	0-3	1
#81.25	Настройка синхронизации CTNet (SSFF)	0-9999	0
#81.26	Узел назначения простого режима 1 CTNet (NNSS)	0-25503	0
#81.27	Параметр источника простого режима 1 CTNet (MMPP)	0-9999	0
#81.28	Узел назначения простого режима 2 CTNet (NNSS)	0-25503	0
#81.29	Параметр источника простого режима 2 CTNet (MMPP)	0-9999	0
#81.30	Узел назначения простого режима 3 CTNet (NNSS)	0-25503	0
#81.31	Параметр источника простого режима 3 CTNet (MMPP)	0-9999	0
#81.32	Назначение гнезда 1 простого режима CTNet (MMPP)	0-9999	0
#81.33	Назначение гнезда 2 простого режима CTNet (MMPP)	0-9999	0
#81.34	Назначение гнезда 3 простого режима CTNet (MMPP)	0-9999	0
#81.35	Идентификатор задачи CTNet Sync EVENT	0-4	0
#81.36	Диагностический параметр CTNet	-3-32767	Отсутств.
#81.37	Запрет загрузки в случае разрешения привода	0-1	0
#81.38	Отключение во время выполнения APC	0-1	0
#81.39	Состояние синхронизации между модулями привода	0-3	0

Таблица 13.1 Параметры (Продолжение)

Параметр	Описание	Диапазон	По умолчан.
#81.42	Положение фиксации обратной связи энкодера	0-1	0
#81.43	Инверсия фиксации	0-1	0
#81.44	Уровень приоритета задачи	0-255	0
#81.48	Номер строки DPL с ошибкой	32 бита	0
#81.49	Используйте ID программы	16 бит	0
#81.50	Код ошибки времени выполнения	0-255	0
#81.51	Младшая версия программного обеспечения	0-99	Отсутств.

Таблица 13.2 Коды ошибок SM-Applications

Код ошибки	Причина	Отключ. привод?	Задача ERROR?	Останов прог?
39	Переполнение стека программы пользователя	Да	Нет	Да
40	Неизвестная ошибка - обращайтесь к поставщику	Да	Нет	Да
41	Параметр не существует. Программа DPL пользователя пыталась записать или прочитать несуществующий параметр.	Может	Да	Да
42	Попытка записи в параметр только для чтения.	Может	Да	Да
43	Попытка чтения из параметра только для записи.	Может	Да	Да
44	Значение параметра вне диапазона. (Программа DPL пользователя записала в параметр неверное значение.) Если параметр #MM.17=0, то записываемое значение автоматически обрезается и ошибки не возникает.	Может	Да	Да
45	Неверные режимы синхронизации	Да	Нет	Не раб.
46	Не используется	-	-	-
47	Потеряна синхронизация с ведущим CTSync Master	Да	Да	Да
48	RS485 не в режиме пользователя. Возникает, если подана команда DPL для порта RS485 в режиме пользователя, но порт RS485 не находится в режиме пользователя.	Да	Да	Да
49	Неверная конфигурация RS485. Например, неверный режим.	Да	Да	Да
50	Арифметическая ошибка - деление на ноль или переполнение.	Может	Да	Да
51	Индекс массива не в диапазоне. Например, агг%[20] где массив агг% согласно оператору DIM имеет только 19 элементов.	Может	Да	Да
52	Отключение по слову управления пользователя. Происходит при установке бита отключения в слове управления #90.11	Да	Нет	Нет
53	Программа DPL несовместима с целевым модулем. Например, загружена программа для UD70.	Да	N/A	N/A

Таблица 13.2 Коды ошибок SM-Applications (Продолжение)

Код ошибки	Причина	Отключ. привод?	Задача ERROR?	Останов прог?
54	Переполнение задачи DPL. Это происходит, если код DPL в задаче реального времени (например, POS0) нельзя выполнить вовремя. С помощью параметра #88.02 можно определить в какой задаче возникла такая ошибка. Проверьте правильность периода планирования задачи и что в задаче нет циклов. Может также возникнуть из-за внешнего воздействия, например, прием большого пакета данных по CTNet. Эту проблему можно преодолеть изменением приоритета CTNet, чтобы он был ниже, чем у задач POS. Однако при этом может не хватать ресурсов для обработки задачи CTNet. Смотрите описание параметра #81.44 на стр. 35, где приведена дополнительная информация.	Может	Да	Да
55	Не используется	-	-	-
56	Неправильная конфигурация блока таймера.	Да	Да	Да
57	Функциональный блок не существует.	Да	Да	Не раб.
58	Испорчено содержимое флэш-памяти ПЛК. Ошибка возникает при запуске и означает, что набор регистров ПЛК (P/Q/T/U) и меню 20 не были восстановлены. Если эта неисправность повторяется, то это может быть вызвано поломкой аппаратуры, так что обращайтесь к поставщику.	Да	Да	Не раб.
59	Привод не принял модуль в качестве ведущего синхронизации	Да	Да	Да
60	Отказ аппаратуры CTNet. Обратитесь к вашему поставщику.	Может	Нет	Нет
61	Неверная конфигурация CTNet. Проверьте все параметры конфигурации	Может	Нет	Нет
62	Неверная скорость CTNet. Проверьте #MM.24 и сетевые подключения.	Может	Нет	Нет
63	Неверный идентификатор узла CTNet. Проверьте #MM.23	Может	Нет	Нет
64	Перегрузка цифрового выхода. При этом оба цифровых выхода переводятся в пассивный режим и остаются пассивными, пока не будет устранено условие перегрузки. Порог отключения составляет 20 мА.	Да	Да	Да
65	Неверные параметры функционального блока. Вы вызвали функц. блок в программе DPL, но один или несколько входных аргументов указаны неверно.	Да	Да	Да
66	Куча пользователя слишком велика. Программа скомпилирована для целевого модуля, в котором больше памяти ОЗУ, чем в данном модуле. Происходит при запуске.	Да	Нет	Не раб.
67	Файл ОЗУ не существует или был указан id не для файла ОЗУ.	Да	Да	Да
68	Указанный файл ОЗУ не ассоциирован с массивом.	Да	Да	Да
69	Ошибка обновления кэша базы данных параметров привода в памяти флэш.	Да	Нет	Не раб.
70	Программа пользователя загружена при включенном (разрешенном) приводе. Произойдет, если #MM.37 = 1 и загружена программа.	Может	Нет	Да
71	Отказ изменения режима привода	Да	Нет	Да
72	Неверная операция буфера CTNet	Да	Да	Да
73	Отказ быстрой инициализации параметра	Да	Нет	Нет

Таблица 13.2 Коды ошибок SM-Applications (Продолжение)

Код ошибки	Причина	Отключ. привод?	Задача ERROR?	Останов прог?
74	Превышение температуры	Да	Да	Да
75	Не доступна аппаратура. Программа пользователя пытается получить доступ к отсутствующей аппаратуре, например, к цифровому I/O, порту RS485 или CTNet в модуле SM-Applications Lite.	Да	Да	Да
76	Нельзя определить тип модуля. Модуль не распознан.	Да	Нет	Не раб.
77	Ошибка связи между модулями и модулем в гнезде 1.	Да	Да	Да
78	Ошибка связи между модулями и модулем в гнезде 2.	Да	Да	Да
79	Ошибка связи между модулями и модулем в гнезде 3.	Да	Да	Да
80	Ошибка связи между модулями и модулем в неизвестном гнезде.	Да	Да	Да
81	Внутренняя ошибка APC. Смотрите параметр #81.38 (например, использование неинициализированной функции CAM).	Может	Да	Да
82	Отказ связи с приводом.	Может	Да	Да

Таблица 13.3 Клеммы

Клемма	Функция	Описание
1	0V SC	Подключение 0 В для порта RS485
2	/RX	Линия приема EIA-RS485 (отрицательная). Входящая.
3	RX	Линия приема EIA-RS485 (положительная). Входящая.
4	/TX	Линия передачи EIA-RS485 (отрицательная). Исходящая.
5	TX	Линия передачи EIA-RS485 (положительная). Исходящая.
6	CTNet A	Линия данных CTNet
7	CTNet Shield	Подключение экрана для CTNet
8	CTNet B	Линия данных CTNet
9	0V	Подключение 0 В для цифровых входов-выходов
10	DIGIN0	Цифровой вход 0
11	DIGIN1	Цифровой вход 1
12	DIGOUT0	Цифровой выход 0
13	DIGOUT1	Цифровой выход 1