

SIEMENS

Сети

SIMATIC NET

PROFIBUS-DP

Руководство

Оглавление

Основы PROFIBUS

Типы приборов и обмен данными в PROFIBUS-DP

PROFIBUS-DP в системе SIMATIC S7

Программирование и проектирование PROFIBUS-DP с помощью STEP 7

Пользовательское программирование DP-

Пример пользовательского обмена данными с помощью PROFIBUS-DP

Функции диагностики для PROFIBUS-DP

1

2

3

4

5

6

7

Указания по технике безопасности

Данное руководство содержит указания, которые вы должны соблюдать для обеспечения собственной безопасности, а также защиты от повреждений продукта и связанного с ним оборудования. Эти замечания выделены предупреждающим треугольником и представлены, в соответствии с уровнем опасности следующим образом:



Опасность

указывает, что если не будут приняты надлежащие меры предосторожности, то это **приведет** к гибели людей, тяжким телесным повреждениям или существенному имущественному ущербу.



Предупреждение

указывает, что при отсутствии надлежащих мер предосторожности это **может привести** к гибели людей, тяжким телесным повреждениям или к существенному имущественному ущербу.



Осторожно

указывает, что возможны легкие телесные повреждения и нанесение небольшого имущественного ущерба при непринятии надлежащих мер предосторожности.

Осторожно

указывает, что возможно повреждение имущества, если не будут приняты надлежащие меры безопасности.

Замечание

привлекает ваше внимание к особо важной информации о продукте, обращении с ним или к соответствующей части документации.

Квалифицированный персонал

К монтажу и работе на этом оборудовании должен допускаться только **квалифицированный персонал**. Квалифицированный персонал – это люди, которые имеют право вводить в действие, заземлять и маркировать электрические цепи, оборудование и системы в соответствии со стандартами техники безопасности.

Надлежащее использование

Примите во внимание следующее:



Предупреждение

Это устройство и его компоненты могут использоваться только для целей, описанных в каталоге или технической документации, и в соединении только с теми устройствами или компонентами других производителей, которые были одобрены или рекомендованы фирмой Siemens.

Этот продукт может правильно и надежно функционировать только в том случае, если он правильно транспортируется, хранится, устанавливается и монтируется, а также эксплуатируется и обслуживается в соответствии с рекомендациями.

Товарные знаки

SIMATIC®, SIMATIC HMI® и SIMATIC NET® - это зарегистрированные товарные знаки SIEMENS AG.

Некоторые другие обозначения, использованные в этих документах, также являются зарегистрированными товарными знаками; права собственности могут быть нарушены, если они используются третьей стороной для своих собственных целей.

Copyright © Siemens AG 2001 Все права защищены
Воспроизведение, передача или использование этого документа или его содержания не разрешаются без специального письменного разрешения. Нарушители будут нести ответственность за нанесенный ущерб. Все права, включая права, вытекающие из патента или регистрации практической модели или конструкции, сохраняются.

Siemens AG
Департамент автоматизации и приводов
Промышленные системы автоматизации
Пя 4848, D- 90327, Нюрнберг

Siemens Aktiengesellschaft

Отказ от ответственности

Мы проверили содержание этого руководства на соответствие с описанным аппаратным и программным обеспечением. Так как отклонения не могут быть полностью исключены, то мы не можем гарантировать полного соответствия. Однако данные, приведенные в этом руководстве, регулярно пересматриваются, и все необходимые исправления вносятся в последующие издания. Мы будем благодарны за предложения по улучшению содержания.

©Siemens AG 2001
Technical data subject to change.



Содержание

Глава 1. Основы PROFIBUS

1.1 Модель ISO/OSI	1
1.2 Архитектура протоколов и профили	2
1.2.1 PROFIBUS-DP	2
1.2.2 PROFIBUS-FMS	2
1.2.3 PROFIBUS-PA	3
1.3 Уровни PROFIBUS	3
1.3.1 Физический уровень (Layer 1) для DP/FMS (RS485)	3
1.3.2 Физический уровень (Layer 1) для DP/FMS (световоды)	7
1.3.3 Физический уровень (Layer 1) для PA	9
1.3.4 Fieldbus Data Link (Layer 2)	11
1.3.5 Прикладной уровень (Layer 7)	13
1.4 Топология шины	14
1.4.1 RS485-техника	14
1.4.2 Техника IEC 1158-2 (PROFIBUS-PA)	19
1.5 Управление доступом к шине в PROFIBUS	19
1.5.1 Метод обмена маркером	20
1.5.2 Метод Master-Slave	21
1.6 Шинные параметры	23

Глава 2 Типы приборов и обмен данными в PROFIBUS-DP

2.1 Типы приборов	1
2.1.1 DP-Master (класс 1)	1
2.1.2 DP-Slave	3
2.1.3 DP-Master (класс 2)	4
2.1.4 Комбинированные приборы DP	4
2.2 Обмен данными между типами DP-приборов	4
2.2.1 DP-коммуникационные связи и DP-обмен данными	4
2.2.2 Фаза инициализации, перезапуск и движение пользовательских данных	5
2.3 Цикл PROFIBUS-DP	9
2.3.1 Структура циклов PROFIBUS	9
2.3.2 Структура постоянного по времени PROFIBUS-DP цикла	10
2.4 Обмен данными через перекрестную связь	11
2.4.1 Коммуникационная связь Master-Slave при перекрестной связи	12
2.4.2 Коммуникационная связь Slave - Slave при перекрестной связи	12
2.5 Функциональное расширение DPV1	13

Глава 3. PROFIBUS-DP в системе SIMATIC S7

3.1 DP-интерфейсы в системах SIMATIC S7	1
3.2 Расширенные коммуникационные возможности	5
3.2.1 S7-функции	5
3.2.2 FDL-служба (SEND/RECEIVE)	6
3.3 Системные свойства DP-интерфейса в SIMATIC S7	6
3.3.1 Свойства запуска интерфейса DP-Master в SIMATIC S7	6
3.3.2 Выход из строя станции DP-Slave	6
3.3.3 Сигнал (Alarm) при удалении/вставке модуля	7
3.3.4 Диагностические сигналы от станций DP-Slave	7

3.3.5	Сигналы от процесса у станций DP-Slave.....	7
3.3.6	Сигнал состояния (Statusalarm) от DP-Slave'a.....	8
3.3.7	Сигнал модернизации (Update-alarm) от DP-Slave'a.....	8
3.3.8	Специфический для производителя сигнал от DP-Slave'a.....	8
3.4	Варианты DP-Slave'ов в системе SIMATIC S7	8
3.4.1	Компактные DP-Slave'ы	9
3.4.2	Модульные DP-Slave'ы	9
3.4.3	Интеллектуальные DP-Slave'ы (I-Slave'ы)	9

Глава 4. Программирование и проектирование PROFIBUS-DP с помощью STEP 7

4.1	Основы STEP 7	2
4.1.1	Объекты STEP 7	2
4.1.2	Проект STEP 7	2
4.2	Пример проекта с PROFIBUS-DP	3
4.2.1	Создание проекта STEP 7	3
4.2.2	Вставка объекта в проект STEP 7	4
4.2.3	Установки сети PROFIBUS	4
4.2.4	Проектирование аппаратной конфигурации с помощью HW-Config	11
4.2.5	Проектирование DP-Slave'ов	12
4.3	Использование коммуникационных процессоров CP 443-5 Ext и CP 342-5	20

Глава 5. Пользовательское программирование DP-интерфейса

5.1	Основы пользовательского DP-интерфейса	1
5.1.1	Организационные блоки	1
5.1.2	Основные системные функции для PROFIBUS-DP	2
5.2	Организационные блоки	7
5.2.1	Циклическая обработка главной программы (OB1)	7
5.2.2	Сигналы от процесса (OB40 – OB47)	7
5.2.3	Диагностические сигналы (OB82)	7
5.2.4	Сигналы снятия и установки модуля (OB83)	8
5.2.5	Ошибки выполнения программы (OB85)	11
5.2.6	Выход из строя носителей модуля (OB86)	12
5.2.7	Ошибка доступа к периферии	15
5.3	Функции для обмена DP-пользовательскими данными и функции для сигналов от процесса	16
5.3.1	Обмен консистентными DP-данными с помощью SFC14 DPRD_DAT и SFC15 DPWR_DAT	16
5.3.2	Управляющие команды SYNC и FREEZE с помощью функции SFC11 DPSYC_FR	18
5.3.3	Запуск сигнала от процесса на DP-Master'е с помощью SFC7 DP_PRAL	22
5.4	Диагностика DP с помощью SFC	23
5.4.1	Чтение стандартных диагностических данных DP-Slave с помощью SFC13 DPNRM_DG	23
5.4.2	Важный для распределенной периферии список состояний системы (SZL – нем., SSL – англ.)	25
5.4.3	Структура подписка SZL	25
5.4.4	Чтение подписка SZL с помощью SFC51 RDSYSST	25
5.4.5	Имеющиеся в распоряжение SZL-подписки	28

5.4.6	Особенности SFC 51 RDSYSST	29
5.5	Запись и чтение записей данных/параметров	29
5.5.1	Запись динамических параметров с помощью SFC55 WR_PARM	29
5.5.2	Запись предварительно определенных записей данных/параметров Из SDB с помощью SFC56 WR_DPARM	32
5.5.3	Запись всех предопределенных записей данных/параметров из SDB с помощью SFC57 PARM_MOD	33
5.5.4	Запись наборов данных/параметров с помощью SFC58 WR-REC	34
5.5.5	Чтение записи данных с помощью SFC59 RD_REC	37

Глава 6. Пример пользовательского обмена данными с помощью PROFIBUS-DP

6.1	Обмен данными с помощью команд доступа к периферии	1
6.2	Обмен консистентными данными с помощью SFC14 DPRD_DAT и SFC15 DPWR_DAT	3
6.2.1	Пользовательская программа для I-Slave (S7-300 с CPU315-2DP)	4
6.2.2	Пользовательская программа для DP-Master'a (S7-400 с CPU416-2DP)	6
6.3	Сигнал от процесса с помощью S7-300 как I-Slave: создание и обработка	7
6.3.1	Генерирование сигнала от процесса	7
6.3.2	Обработка сигнала от процесса с помощью S7-400 как DP-Master'a	8
6.3.3	Тестирование обработки сигнала от процесса в DP-Master'e	9
6.4	Передача записей (наборов) данных и параметров	10
6.4.1	Структура наборов данных (DS1) для аналоговых модулей SIMATIC S7-300	11
6.4.2	Пример применения: параметрирование аналогового входного модуля с помощью SFC55 WR_PARM	13
6.4.3	Тестирование перепараметрирования аналогового модуля с помощью SFC55 WR_PARM	15
6.4.4	Перепараметрирование аналогового входного модуля с помощью SFC56 WR_DPARM	15
6.4.5	Тестирование перепараметрирования аналогового модуля с помощью SFC56 WR_DPARM	16
6.5	Вызов (запуск) управляющих функций SYNC/FREEZE	16
6.5.1	Пример применения команд SYNC/FREEZE с DP-Master'ом IM467	18
6.5.2	Создание пользовательской программы для функций SYNC-/FREEZE	22
6.6	Обмен данными с помощью перекрестной связи	25
6.7	Обмен данными с помощью коммуникационного процессора CP342-5	33
6.7.1	CP342-5 как DP-Master	33
6.7.2	CP342-5 как DP-Slave	35
6.7.3	CP342-5 как активный DP-Slave	37

Глава 7. Функции диагностики для PROFIBUS-DP

7.1	Диагностика с помощью индикаторов SIMATIC S7 CPU, интерфейсов DP-Master и DP-Slave	1
7.1.1	Индикаторы S7-300	2
7.1.2	Индикаторы CPU S7-400 с DP-интерфейсом	4
7.2	Диагностика с помощью online-функций STEP 7	7
7.2.1	Функция Accessible Nodes в SIMATIC Manager	7
7.2.2	ONLINE-функции в SIMATIC Manager	10

7.2.3	Диагностика с помощью функции Module Information	11
7.2.4	Диагностика с помощью функции Diagnosing Hardware	17
7.3	Диагностика через программу пользователя	19
7.3.1	Диагностика DP-Slave с помощью SFC13 DPNRM_DG	19
7.3.2	Диагностика с помощью SFC51 RDSYSST в OB82	22
7.4	Диагностика с помощью диагностического блока FB125 в	
	S7-пользовательской программе	24
7.4.1	Блок диагностики FB125	24
7.4.2	Области применения FB125	25
7.4.3	Вызов блока диагностики DP FB125 в S7-пользовательской программе ..	25
7.4.4	Параметры блока диагностики DP FB125 и их значение	26
7.4.5	Описание экземпляра блока данных	30
7.4.6	Технические данные FB125	31
7.4.7	Применение FB125	31
7.5	Диагностика с помощью NCM	32

1. Основы PROFIBUS

(PROcess FieId BUS)

Введение

По сравнению с обычной реализацией структур автоматизации, уже на первый взгляд видны преимущества применения последовательной полевой шины. Экономия средств здесь образуется благодаря меньшей стоимости кабеля (кабель имеет меньшую длину) и применению полевых приборов. Это возможно, однако, только при стандартизированной и открытой полевой шине.

В 1987 году для немецкой промышленности был разработан и принят стандарт DIN E 19245 PROFIBUS. В 1996 году этот стандарт стал международной нормой EN 50170.

1.1 Модель ISO/OSI

Архитектура протоколов PROFIBUS ориентирована на уже установленные национальные и международные нормы. Так, архитектура протоколов базируется на модели OSI (Open System Interconnection).

На рис.1.1 изображена модель ISO/OSI для коммуникационных стандартов, состоящая из 7 уровней, подразделяющихся на два класса:

- ориентированных на пользователя с уровня 5 по уровень 7;
- ориентированных на сеть (уровни 1-4).

Уровни с 1 по 4 описывают пересылку передаваемых данных из одного пункта в другой, в то время как уровни с 5 по 7 предоставляют в распоряжение пользователя доступ к сети в соответствующей форме.

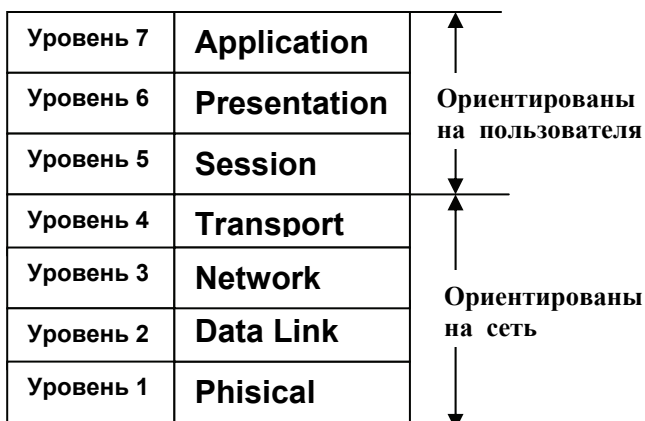


Рис. 1.1 Модель ISO/OSI для стандартов

1.2 Архитектура протоколов и профили

Из рис.1.2, представляющего архитектуру протоколов PROFIBUS, можно видеть, что в ней реализованы уровни 1,2 и 7. Для уровней 1 и 2 принят стандарт США EIA (Electronic Industries Association) RS485, международные нормы IEC 870-5-1 (Telecontrol Equipment and System) и EN 60870-5-1. Метод доступа к сети, службы передачи и управления данными ориентируются на DIN 19241, части 1-3 и нормы IEC 955 Process Data Highway/Тип С. Функции управления (FMA7) ориентированы на ISO DIS 7498-4 (Management Framework). С точки зрения пользователя PROFIBUS подразделяется на 3 профиля протокола: DP, FMS и PA.

	PNO-профиль для DP-устройств	PNO-профиль для FMS-устройств	PNO-профиль для DP-устройств
	Основные функции Расширенные функции		Основные функции Расширенные функции
	DP User Interface Direct Data Link Mapper (DDLМ)	Application Layer Interface (ALI)	DP User Interface Direct Data Link Mapper (DDLМ)
Layer 7 (Application)	<div style="text-align: center;">↑</div>	Application-Layer Fieldbus Message Specification (FMS)	<div style="text-align: center;">↑</div>
Layer 3-6		Н Е И С П О Л Ъ З У Ю Т С Я	
Layer 2 (Link)	Data Link Layer Fieldbus Data Link (FDL)	Data Link Layer Fieldbus Data Link (FDL)	IEC-Interface
Layer 1 (Phisik)	Phisical-Layer (RS485/LWL)	Phisical-Layer (RS485/LWL)	IEC 1158-2

Рис. 1.2 Архитектура протоколов PROFIBUS

1.2.1 PPOFIBUS-DP

PROFIBUS-DP применяет уровни 1 и 2, а также пользовательский интерфейс. Уровни с 3 по 7 не используются. Благодаря такой архитектуре достигается быстрая передача данных. Direct Data Link Mapper (DDLМ) организует доступ к уровню 2. В основу пользовательского интерфейса положены необходимые пользовательские функции, а также системные и аппаратно-зависимые функции различных типов PROFIBUS-DP-приборов.

Этот профиль протокола PROFIBUS оптимизирован для быстрого обмена данными специально для коммуникаций между системами автоматизации и децентрализованной периферией на полевом уровне.

1.2.2 PROFIBUS-FMS

В PROFIBUS-FMS применяются уровни 1,2 и 7. Пользовательский уровень состоит из FMS (Fieldbus Message Specification) и LLI (Lower Layer Interface).

FMS содержит пользовательский протокол и предоставляет в распоряжение коммуникационные службы.

LLI реализует различные коммуникационные связи и создает для FMS аппаратно-независимый доступ к уровню 2.

FMS применяется для обмена данными на уровне ячеек (PLC и PC). Мощные FMS-сервисы открывают широкие области использования и большую гибкость при передаче больших объемов данных.

PROFIBUS-DP и PROFIBUS-FMS применяют одинаковую технику передачи и единый протокол доступа к шине и поэтому могут работать через общий кабель.

1.2.3 PROFIBUS-PA

PROFIBUS-PA применяет расширенный PROFIBUS-DP-протокол передачи данных. Техника передачи согласно IEC 1158-2 обеспечивает надежность и питание полевых приборов через шину. Приборы PROFIBUS-PA могут благодаря применению специальных устройств (PROFIBUS-PA-Links) в простейшем случае интегрироваться в PROFIBUS-DP-сеть.

PROFIBUS-PA – специальная концепция, позволяющая подключать к общей шине датчики и приводы, находящиеся во взрывоопасной зоне.

1.3 Уровни PROFIBUS

1.3.1 Физический уровень (Layer 1) для DP/FMS (RS485)

В основной версии для экранированной витой пары уровню 1 PROFIBUS соответствует симметричная передача данных по стандарту EIA RS485 (также обозначается H2). Проводники шинных сегментов замкнуты с обеих сторон, скручены и экранированы (см. рис. 1.3)

Способ передачи

Для PROFIBUS назначен способ передачи RS485, базирующийся на полудуплексной, асинхронной синхронизации. Данные передаются внутри 11-разрядного кадра (рис. 1.4) в NRZ-коде (Non Return to Zero). Значения сигнала (биты) не изменяются во время передачи сигнала.

В то время, как передача бинарного значения “1” соответствует положительному значению на проводнике RxD/TxD-P (Receive/Transmit-Data-P), напротив, на проводнике RxD/TxD-N (Receive/Transmit-Data-N) присутствует “0”. Состоянию

покоя между отдельными телеграммами соответствует двоичный сигнал “1” (рис.1.5).

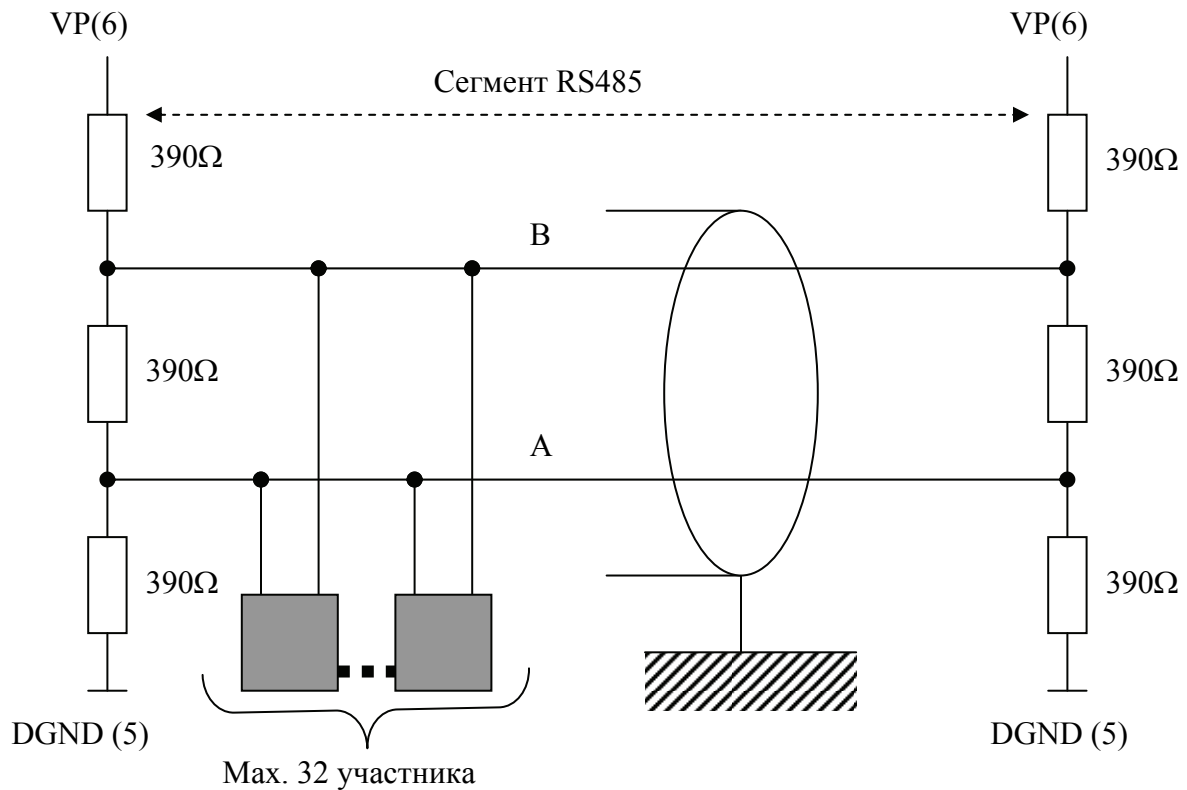
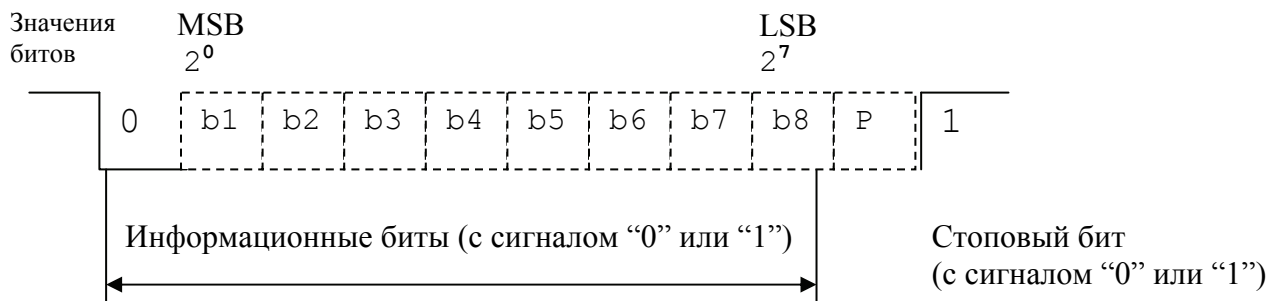


Рис. 1.3 Структура шинного сегмента RS485



LSB Least Signification Bit (младший бит)

MSB Most Signification Bit (старший бит)

Рис. 1.4 PROFIBUS UART-кадр

В литературе часто также оба проводника PROFIBUS обозначают как А-проводник и В-проводник. При этом А-проводник соответствует RxD/TxD-N, а В-проводник - RxD/TxD-P.

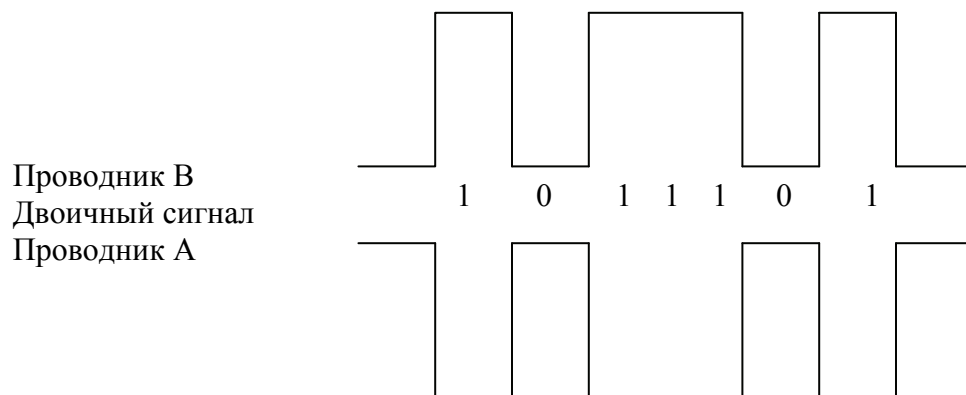


Рис. 1.5 Структура сигнала при передаче NRZ- кодом

Шина

В табл. 1.1 приведена максимально допустимая длина провода (длина сегмента) системы PROFIBUS. Эта длина зависит от скорости передачи. Внутри сегмента может быть до 32 участников.

Табл. 1.1. Максимальная длина сегмента в зависимости от скорости

Скорость передачи (kBit/s)	9,6-187,5	500	1500	12000
Длина сегмента (m)	1000	400	200	100

Данные о максимальной длине сегмента в табл.1.1 взяты из норм PROFIBUS. В табл.1.2 приведены параметры кабеля типа А:

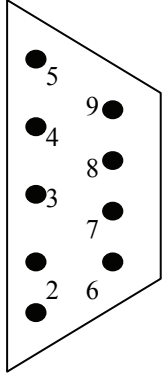
Табл.1.2. Спецификации PROFIBUS-кабеля типа А

Волновое сопротивление	От 135 до 165 Ом при частотах измерения от 3 до 20 MHz
Погонная емкость	< 30pF/m
Площадь сечения	> 0,34 mm ² , соотв. AWG22
Тип кабеля	Витая пара, 1x2 или 2x2 или 1x4 провода
Погонное сопротивление	< 110 Ом/km
Затухание сигнала	Max 9dB на всей длине отрезка провода
Экранирование	Медная оплетка или оплетка и экран из фольги

Подключение шины

В качестве стандарта для подключения участников к шине в нормах PROFIBUS EN 50170 рекомендуется 9-и штырьковый штекер, который изображен в табл.1.3. У каждого участника есть такой разъем с бухтовыми контактами, шинный кабель имеет разъем со штырьковыми контактами

Табл.1.3 Расположение контактов

Вид	Pin-№	Название сигнала	Обозначение
	1	SHIELD	Экран, напр., земля
	2	M24	-24v
	3	RxD/TxD-P	Прием/передача данных, плюс, провод В
	4	CNTR-P	Сигнал для управления направлением передачи, плюс.
	5	DGND	Данные
	6	VP	Напряжение питания, плюс
	7	P24	+24v
	8	RxD/TxD-N	Прием/передача данных, минус, провод А
	9	CNTR-N	Сигнал для управления направлением передачи, минус.

Окончание шины

Шинные провода данных с обеих сторон замкнуты на согласованные нагрузки (см. рис.1.3). Благодаря этим сопротивлениям устанавливается безопасный потенциал покоя на проводах шины, когда участники не обмениваются сообщениями (потенциал покоя между телеграммами). Шинные нагрузки имеются почти во всех стандартных разъемах PROFIBUS и могут быть активизированы с помощью переключателей.

Если используется шина со скоростью передачи более 1500 kBit/s, то нужно на основании потребляемой мощности подключенных участников и отраженной мощности использовать шинный штекер с дополнительной индуктивностью.

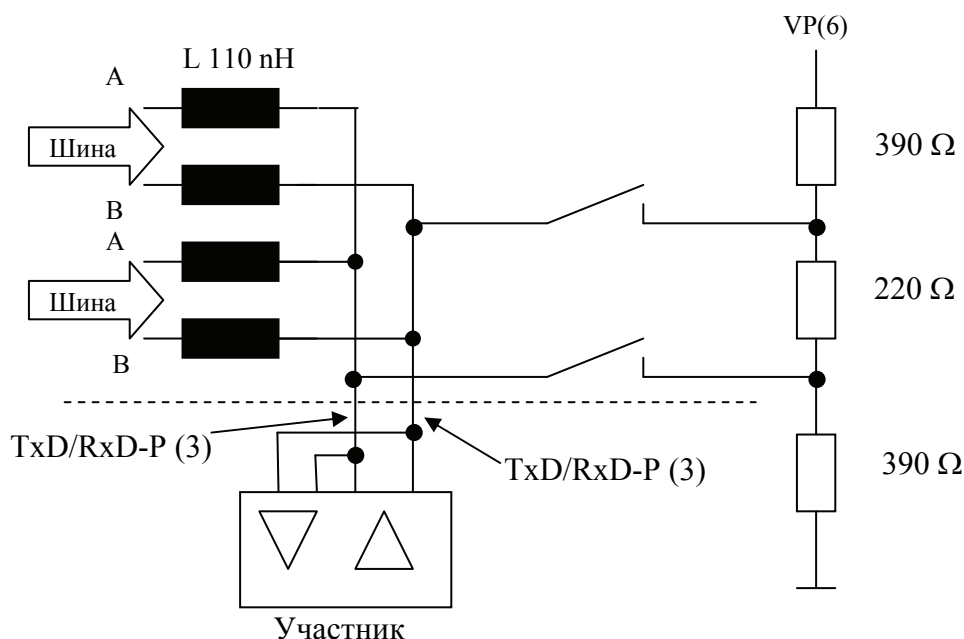


Рис. 1.6 Схема шинного штекера для скорости передачи более 1500 кБит/с

1.3.2 Физический уровень (Layer 1) для DP/FMS (световоды)

Дальнейшее использование уровня 1 PROFIBUS по норме PNO (Profibus Nutzer Organisation –нем.) “Техника оптической передачи для PROFIBUS”, версия 1.1 от 07.1993г. – это передача данных с помощью световодов. Благодаря оптоволокну внутри установки PROFIBUS между участниками может быть достигнуто расстояние до 15 km. Световодная техника устойчива к электромагнитным помехам и устанавливает безопасную разность потенциалов между участниками.

Благодаря простой технике подключения световодов, специальным пластиковым световодам, эта техника пришла на полевой уровень.

Среда передачи

В качестве среды передачи используются световоды со стеклянными или пластиковыми волокнами. В зависимости от используемого типа проводника длина связи может быть до 15 km при стеклянных световодах и до 80 m при пластиковых.

Подключение шины

Для подключения участников к световоду имеется различная техника (различные модули)

- *Модули OLM (Optical Link Module)*. Похож на репитер RS-485. Имеет два функционально разделенных электрических канала и выходы для одного или двух оптических каналов. Модули OLM соединяются с отдельными участниками или сегментами шины через интерфейс RS-485 (см. рис.1.7).

- *Модули OLP (Optic Link Plug).* С помощью модулей OLP можно соединять друг с другом оптическим волокном пассивных участников (Slave). Модули OLP подключаются прямо на 9-и штырьковый штекер участника. OLP получает энергию от участника и поэтому не нуждается в напряжении питания. Как видно из рис.1.8, для подключения активных участников шины (Master) к OLP-кольцу всегда используется OLM.
- *Интегрированное LWL-подключение (Licht Wellen Leiter – нем.).* Прямое подключение участников PROFIBUS к световоду. Возможно у приборов со встроенным LWL-вводом.

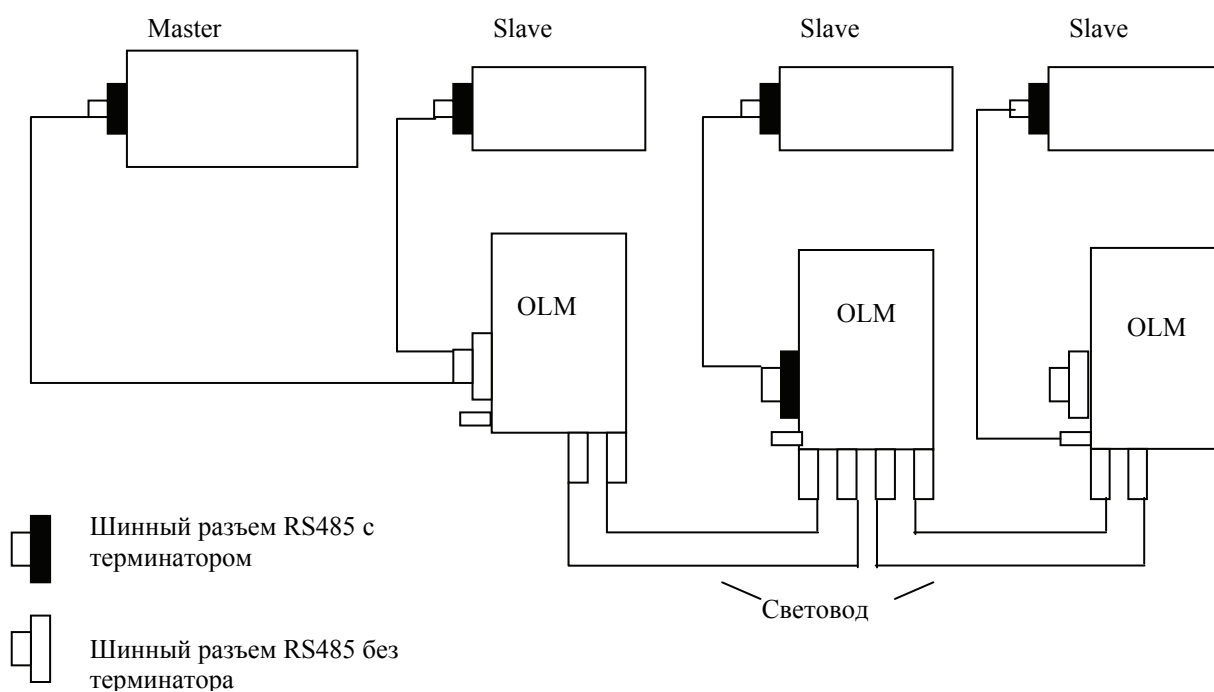


Рис 1.7 Пример шинной конфигурации с OLM-техникой

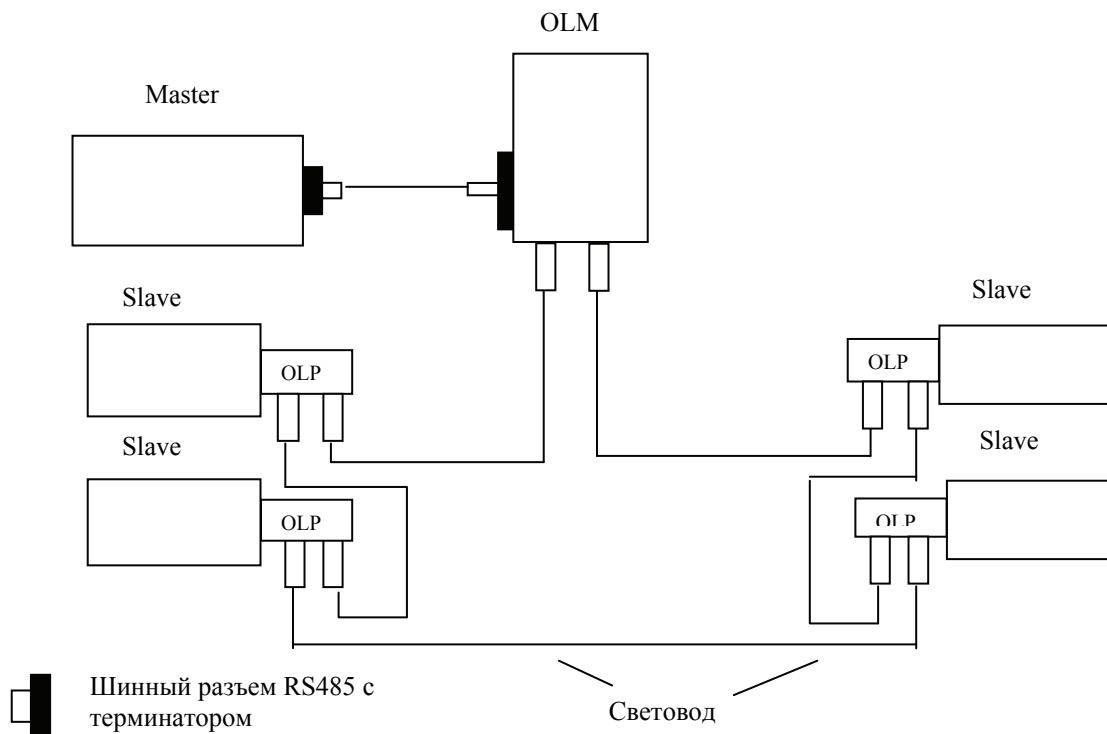


Рис 1.8 Оптическое однопроводное кольцо с OLM-технологией

1.3.3 Физический уровень (Layer 1) для PA

В PROFIBUS-PA используется передающая техника по IEC 1158-2. Эта техника позволяет достигнуть электробезопасности и питания полевых приборов прямо через шину. Для передачи данных используется бит-синхронизированный, с манчестерским кодом протокол передачи без постоянной составляющей (обозначается также как H1). При передаче данных с помощью манчестерского кода бинарный "0" передается как смена фронта с 0 на 1, а бинарная "1" – как смена фронта с 1 на 0. Данные передаются с помощью модуляции $\pm 9\text{mA}$ основного тока шинной системы I_B (рис.1.9).

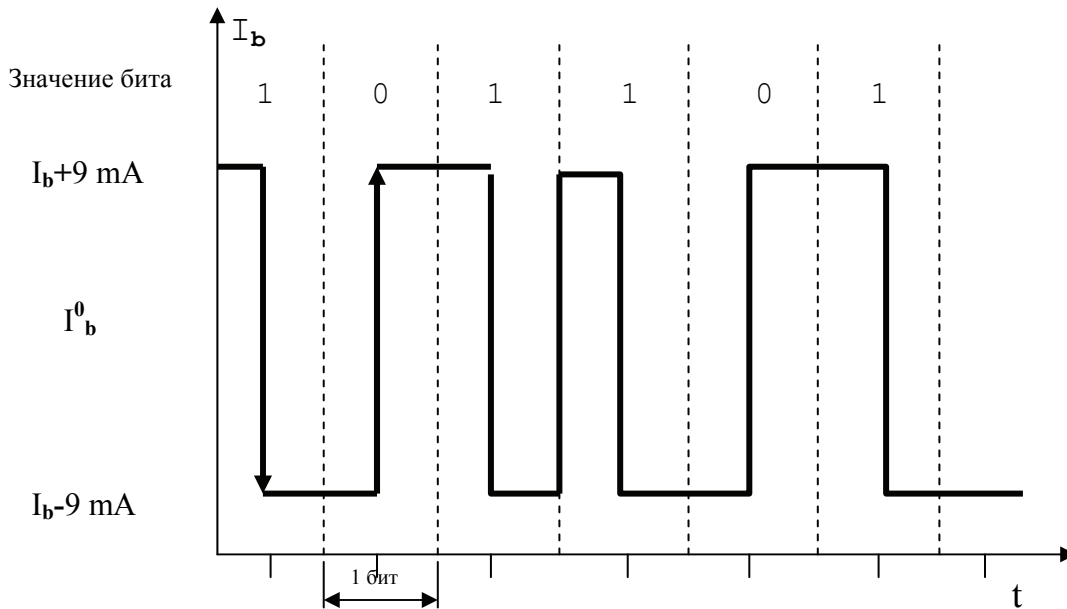


Рис. 1.9 Передача данных в PROFIBUS-PA с помощью модуляции тока (Манчестерский код II)

Скорость передачи составляет 31,25 кбит/с. В качестве среды передачи используется витой экранированный или неэкранированный провод. Шина, как это видно из рис. 1.10, состоит из сегментов, к которым подключены участники, сегменты замкнуты на RC-цепочки. К сегменту шины PA может быть подключено максимум 32 участника. Максимальная длина сегмента сильно зависит от применяемого источника питания, типа провода и потребления тока подключенными участниками.

Шинный провод

В качестве среды передачи для PROFIBUS-PA применяется 2-жильный кабель, технические данные которого не установлены/не нормированы. Свойства типов кабелей определяют максимальную длину шины, число подключаемых участников и чувствительность к электромагнитным шумам. На основании этого установлены для стандартных типов кабелей электрические и механические свойства.

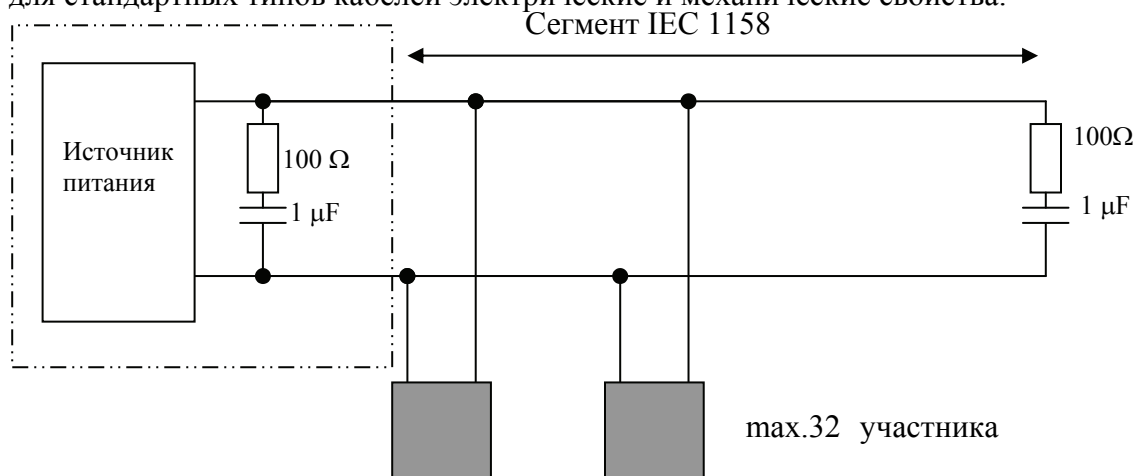


Рис. 1.10 Структура шинного сегмента PA

В DIN 61158-2 предложены для применения стандартные кабели для PROFIBUS-PA, называемые типами A...D.

Табл. 1.4 Предлагаемые типы кабеля для PROFIBUS-PA

	Тип А (основной)	Тип В	Тип С	Тип D
Структура кабеля	Витой, двухжильный, экранированный	Одна или несколько витых пар, экранир.	Несколько витых пар, неэкранир.	Несколько невитых пар, неэкранир.
Площадь сечения (номинальная)	0,8 мм ² (AWG18)	0,32 мм ² (AWG22)	0,13 мм ² (AWG26)	1,26 мм ² (AWG16)
Погонное сопротивление (пост. ток)	44 Ω/km	112 Ω/km	264 Ω/km	40 Ω/km
Волновое сопротивление при 31,25 kHz	100Ω±20%	100Ω±30%	**	**
Затухание при 39 kHz	3dB/km	5dB/km	8dB/km	8dB/km
Емкостное рассогласование	2nF/km	2nF/km	**	**
Групповое время запаздывания (7,9...39 kHz)	1,7μs/km	**	**	**
Степень экранирования	90%	**	-	-
Рекомендуемая длина сети, включая ¼-волновые согласующие шлейфы	1900 m	1200 m	400 m	200 m

1.3.4 Fieldbus Data Link (Layer 2)

Согласно модели OSI на втором уровне реализуются функции управления доступом к шине (раздел 1.2), обеспечение безопасности данных, а также выполнение протокола передачи и формирование телеграмм. Уровень 2 обозначается в PROFIBUS как FDL-уровень (Fieldbus Data Link).

Формат телеграммы уровня 2 (рис.1.11) способствует большей безопасности передачи. Вызывающая телеграмма имеет расстояние Хемминга HD (Hamming Distance) = 4. При HD = 4 может быть распознано до 3-х одновременных ошибок в фальсифицированных битах телеграммы данных. Это достигается благодаря применению особых стартового и завершающего знаков телеграммы, постоянно скользящей синхронизации, биту четности и контрольному байту.

При этом могут быть распознаны следующие ошибки:

- Ошибка символьного формата (четность, переполнение, ошибка фрейма)
- Ошибки протокола
- Ошибки разделителей начала и окончания
- Ошибки байта проверки фрейма
- Ошибки длины телеграммы

Телеграмма, у которой распознана ошибка, повторяется по крайней мере, один раз. Имеется возможность повторять телеграммы, проходящие по уровню 2 до 8 раз (шинный параметр “Retry”). Уровень 2 может осуществлять наряду с передачей данных “точка к точке”, также коммуникации во многие точки – Broadcast и Multicast.

При коммуникациях Broadcast активный участник посылает сообщение всем остальным участникам (Master’ам и Slave’ам). Прием данных не квитируется.

При коммуникациях Multicast активный участник посылает сообщение группе участников (Master’ам и Slave’ам). Прием данных не квитируется.

Службы, предлагаемые уровнем 2, приведены в табл. 1.5.

Табл. 1.5 Службы передачи PROFIBUS

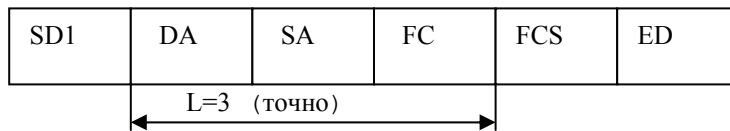
Служба	Функции	DP	PA	FMS
SDA	Посылка данных с квитированием			x
SRD	Данные посылаются и принимаются с квитированием	x	x	x
SDN	Данные посылаются без квитирования	x	x	x
CSRD	Циклическая посылка и прием с квитированием			x

В PROFIBUS-DP и –PA применяется, соответственно, подмножество служб уровня 2. Так, например, PROFIBUS-DP использует только службы SRD и SDN.

Службы вызываются через точки доступа к службе, SAP (Service Access Point), уровня 2 из выше стоящего уровня. В PROFIBUS-FMS используются эти точки доступа для адресации логических коммуникационных связей. В PROFIBUS-DP и –PA применяемые точки доступа строго упорядочены. У всех активных и пассивных участников можно использовать параллельно несколько точек доступа.

Различаются точки доступа источника SSAP (Source Service Access Point) и точки доступа цели DSAP (Destination Service Access Point).

Формат с постоянной длиной информационного поля



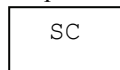
Формат с постоянной длиной информационного блока с данными



Формат с переменной длиной информационного блока



Короткое квитирование



Телеграмма-токен (маркер)



L	длина информационного поля
SC (Single Character)	отдельный символ, используется только для квитирования (SC=E5h)
SD1-SD4 (Start Delimiter)	стартовый байт для отличия различных форматов телеграмм (SD1=10h, SD2=68h, SD3=A2h, SD4=DCh)
LE / LEr (LEngth)	байт длины, указывает длину информационных полей у телеграмм с переменной длиной
DA (Destination Adress)	байт адреса цели, содержит информацию о приемнике
SA (Source Adress)	байт адреса источника, содержит информацию о передатчике
FC (Frame Control)	контрольный байт содержит информацию о службе для данного сообщения и приоритет сообщения
Data Unit	поле данных, может также содержать возможные расширения адреса телеграммы пользовательских данных
FCS (Frame Check Sequence)	проверочный байт, содержит контрольную сумму телеграммы, которая образуется операцией “И” без бита переполнения
ED (End Delimiter)	оконечный байт, указывает на конец телеграммы (ED=16h)

Рис. 1.11 Формат PROFIBUS-телеграмм

1.3.5 Прикладной уровень (Layer 7)

Уровень 7 модели ISO/OSI представляет в распоряжение пользователя полезные коммуникационные службы. Этот пользовательский уровень состоит в PROFIBUS из FMS (Fieldbus Message Spesification) и LLI (Lower Layer Interface) слоев.

Профиль FMS

Коммуникационные службы FMS соответствуют функциям прибора, т.е. в FMS-профилях PNO определен необходимый объем функций для конкретных требований. Эти FMS-профили устанавливают, что приборы различных производителей имеют одни и те же коммуникационные функции. Для FMS определены следующие профили:

Коммуникации между контроллерами (3.002)

Этот коммуникационный профиль устанавливает, какие FMS-службы применяются для коммуникаций между PLC. При помощи точно определенных классов контроллеров, установлены службы, параметры и типы данных, которые каждый PLC должен поддерживать.

Профиль для автоматизации зданий (3.011)

Этот профиль – отраслевой (специализированный) профиль и основа для многих открытых стандартов в автоматизации зданий. Описывает, как осуществляется обмен, управление, регулирование, обслуживание, обработка и архивирование сигналов (Alarm) в системах автоматизации зданий через FMS.

Коммутационные низковольтные приборы (3.032)

Этот профиль – отраслевой пользовательский FMS-профиль. Он определяет пользовательский образ действий низковольтных коммутационных приборов при коммуникациях через FMS.

Пользовательский интерфейс DP и DP-профили

PROFIBUS-DP применяет уровни 1 и 2. Через пользовательский интерфейс становятся доступными необходимые пользовательские функции, а также системные и аппаратные действия различных типов устройств PROFIBUS-DP.

Протокол PROFIBUS-DP открыто определяет, как передаются между участниками пользовательские данные по шине. Оценка передаваемых по протоколу пользовательских данных не происходит. Благодаря точно установленным параметрам профиля, приборы разных производителей могут совершать обмен. В настоящее время установлены следующие профили PROFIBUS-DP:

Профиль для NC/RC (3.052)

Профиль описывает, как происходит управление и обслуживание роботов через PROFIBUS-DP. На основании конкретной блок-схемы программы описывается движение и программное управление роботом.

Профиль для Encoder'а (преобразователя угол-код) (3.062)

Профиль описывает присоединение различных Encoder'ов к PROFIBUS-DP. Определены два аппаратных класса основных и дополнительных функций, как, например, масштабирование сигналов и расширенная диагностика.

Профиль для приводов с изменяемым числом оборотов (3.072)

Ведущие производители техники приводов разработали общий PROFIDRIVE-профиль. Профиль устанавливает, как приводы параметризуются и передают заданные и истинные значения. Благодаря этому становится возможным обмен данными приводов различных производителей.

Профиль содержит необходимые установки для вида работы регуляторов числа оборотов и позиционирования. Профиль устанавливает основные функции приводов и дает достаточное свободное пространство для специфических пользовательских расширений. Профиль содержит описание пользовательских функций DP или альтернативных функций FMS.

Профиль для управления и наблюдения, HMI (Human Machine Interface) (3.082)

Профиль устанавливает для приборов обслуживания и наблюдения (HMI) правила подключения этих приборов через PROFIBUS-DP к компонентам автоматизации. Профиль использует для коммуникаций расширенные функции PROFIBUS-DP.

Профиль для защищенной от ошибок передачи данных через PROFIBUS-DP (3.092)

В этом профиле устанавливаются дополнительные механизмы защиты данных для коммуникаций с защищенными от ошибок компонентами, как например, Not-AUS.

1.4 Топология шины

1.4.1 RS485-техника

Технологически система PROFIBUS состоит из нагруженной с двух сторон активной линии – шинной структуры, которая обозначается также, как сегмент шины RS-485. К шинному сегменту можно по стандарту RS-485 подключить до 32 RS-485 –участников. Каждый подключенный к шине участник, Master или Slave, представляет собой токовую нагрузку.

Повторитель (Repeater)

Если Вы должны подключить к системе PROFIBUS больше, чем 32 участника, то нужно использовать несколько шинных сегментов. Эти отдельные шинные сегменты, каждый максимум с 32-я участниками, должны быть соединены друг с другом через повторитель (усилитель мощности). Повторитель усиливает уровень передаваемого сигнала. Согласно EN 50170 не предусмотрена временная регенерация фазы бита во время передачи сигнала через повторитель. Из-за временных задержек и искажений двоичный сигнал может согласно EN 50170 проходить максимум три повторителя, которые работают как усилители мощности и включены последовательно. На практике, однако, повторитель-соединитель реализуется как восстановитель сигнала. Число повторителей, которые можно включить последовательно, таким образом зависит от его конструкции и изготовителя. Так, например, можно последовательно включить до 9 повторителей типа 6ES7 972-0AA00-0XA0 фирмы Siemens.

Максимальное удаление между двумя участниками шины зависит от скорости передачи. В табл. 1.6 даны значения для повторителя типа 6ES7 972-0AA00-0XA0.

Табл. 1.6 Максимально возможное расширение конфигурации PROFIBUS при включенных в ряд 9 повторителей в зависимости от скорости передачи

Скорость передачи (кБит/с)	9,6-187,5	500	1500	12000
Общая длина всех сегментов в метрах	10000	4000	2000	1000

Принципиальная схема, изображенная на рис. 1.12, поясняет свойства RS485-повторителя.

- Шинный сегмент 1, гнездо PG/PC и шинный сегмент 2 разделены друг от друга по потенциалам.
- Сигнал между шинным сегментом 1, гнездом PG/PC и шинным сегментом 2 усиливается.
- Повторитель имеет для шинных сегментов 1 и 2 подключаемое сопротивление.
- Благодаря разделителю – мосту М/РЕ повторитель может работать без заземления.

Только благодаря применению повторителя может быть достигнуто максимально возможное число участников в конфигурации PROFIBUS. Повторитель можно применять также для построения шинных структур типа “дерево” или ”звезда”. Также можно создать свободные от заземления структуры (разделение шинных сегментов друг от друга) с помощью повторителя и источника питания 24V без заземления (рис. 1.13).

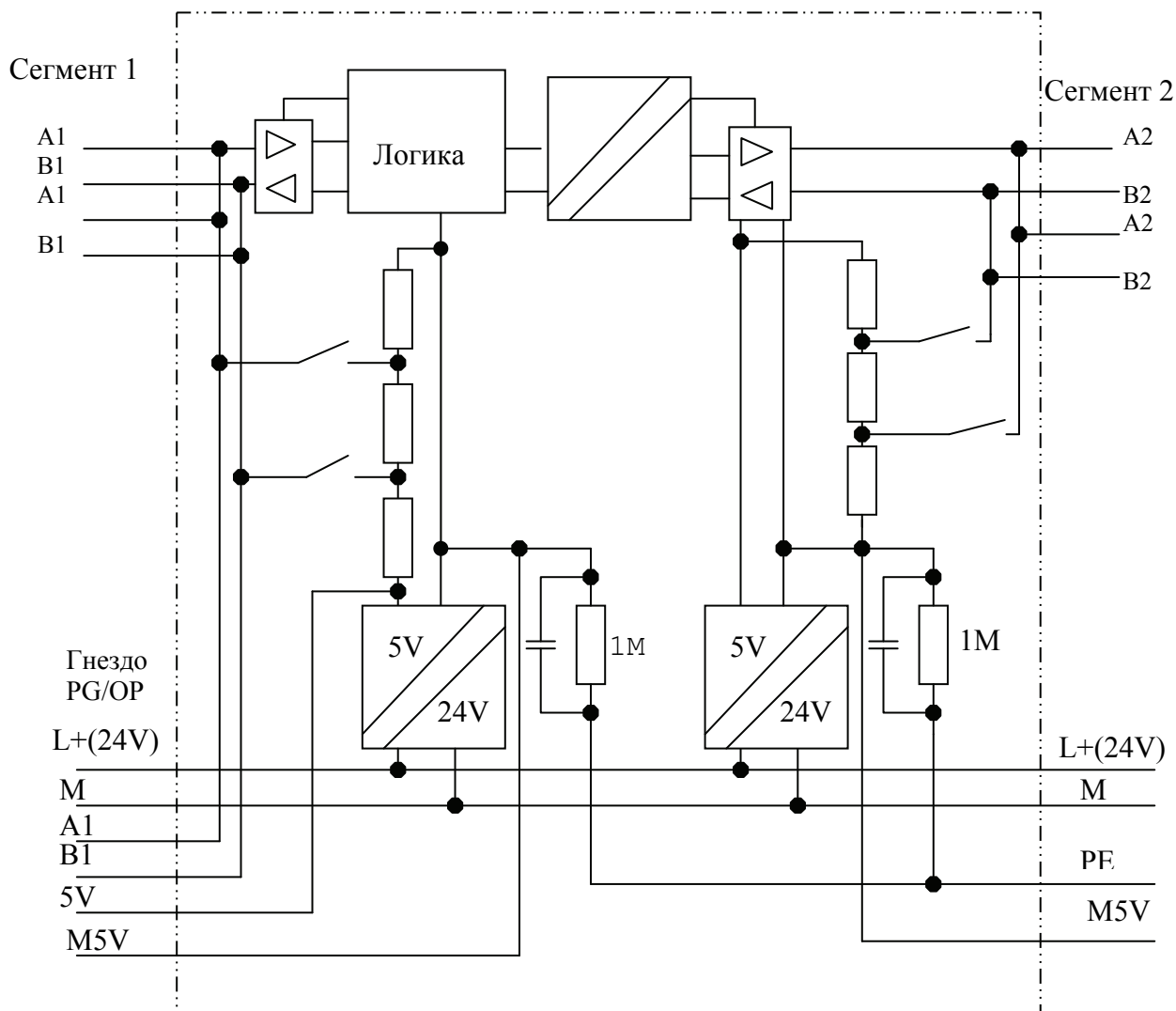


Рис. 1.12 Принципиальная схема RS485-репитера типа 6ES7 972-0AA00-0XA0

Повторитель также представляет нагрузку для соединения RS485. Подключенный RS485-повторитель уменьшает максимальное число участников на сегменте на 1. Это значит, что если на шинном сегменте находится повторитель, то можно на этот сегмент подключить максимум 31 участника. Число повторителей в общей шинной конфигурации не влияет на максимальное число участников (повторитель не занимает логического шинного адреса).

Четвертьволновые отрезки линии передачи

Благодаря прямому подключению участников, например, через 9-штырьковый штекер, в линейной структуре возникают Четвертьволновые отрезки линии

передачи. По EN50170 при скорости передачи 1500 kBit/s допустима длина отрезка менее 6,6 м. Как правило эти отрезки должны быть еще уменьшены.

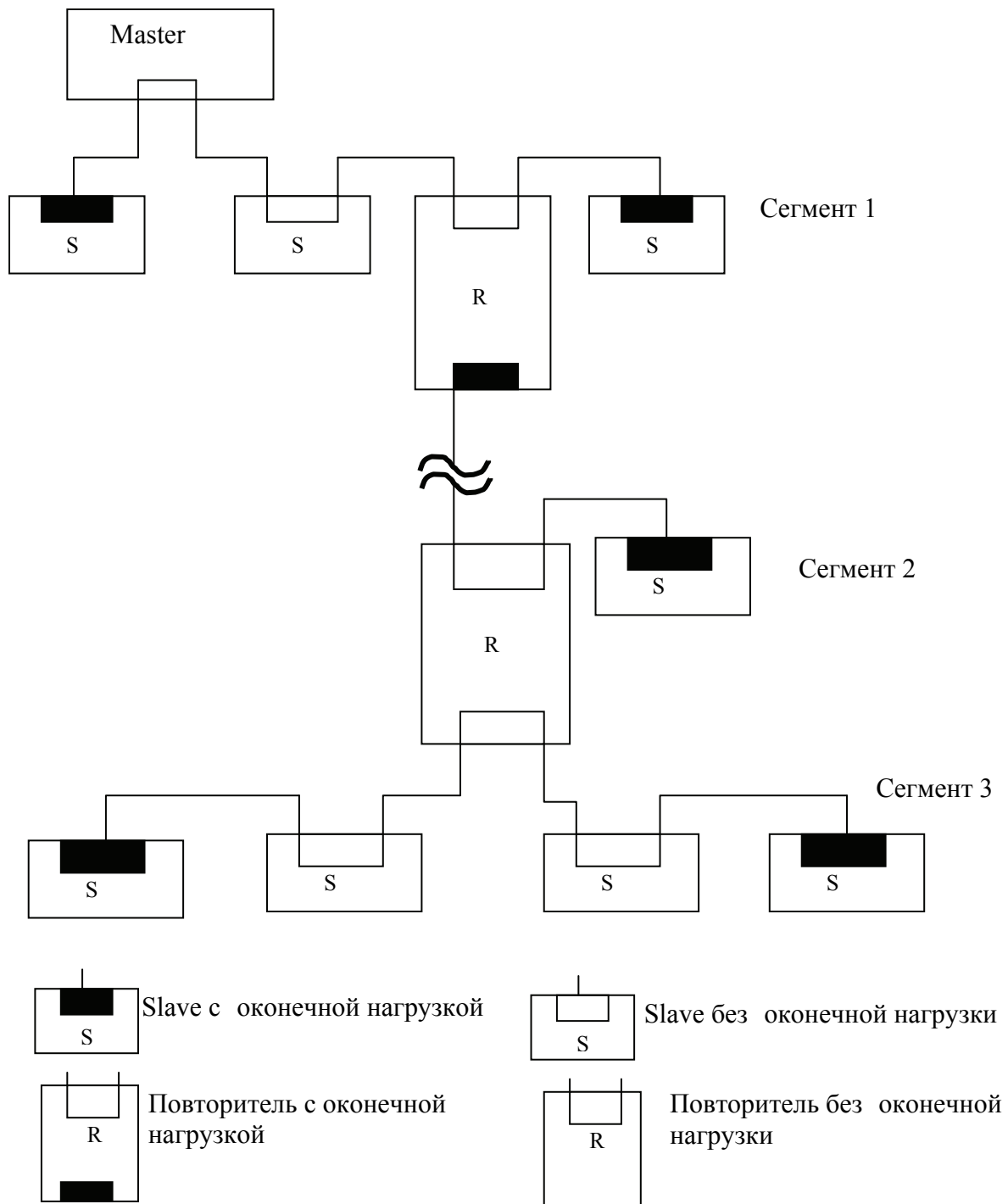


Рис.1.13 Шинная конфигурация с повторителем

Исключение здесь представляет временное подключение устройств программирования и диагностики.

Четвертьволновые отрезки линии передачи могут в зависимости от числа и длины быть причиной переотражений и приводить к искажениям телеграмм. При скорости передачи более 1500 kBit/s Четвертьволновые отрезки линии передачи недопустимы. Программаторы и приборы диагностики могут быть в этом случае подключены через “активные” шинные соединители.

Оптоволоконная техника

При использовании оптоволоконной техники имеются уже готовые шинные структуры, такие, как линия, дерево, звезда, различные варианты кольцевой структуры. С помощью OLM (Optic Link Module) можно реализовать как однопроводное (однофазное) кольцо, так и резервированное двухпроводное (двухфазное) кольцо (рис.1.14).

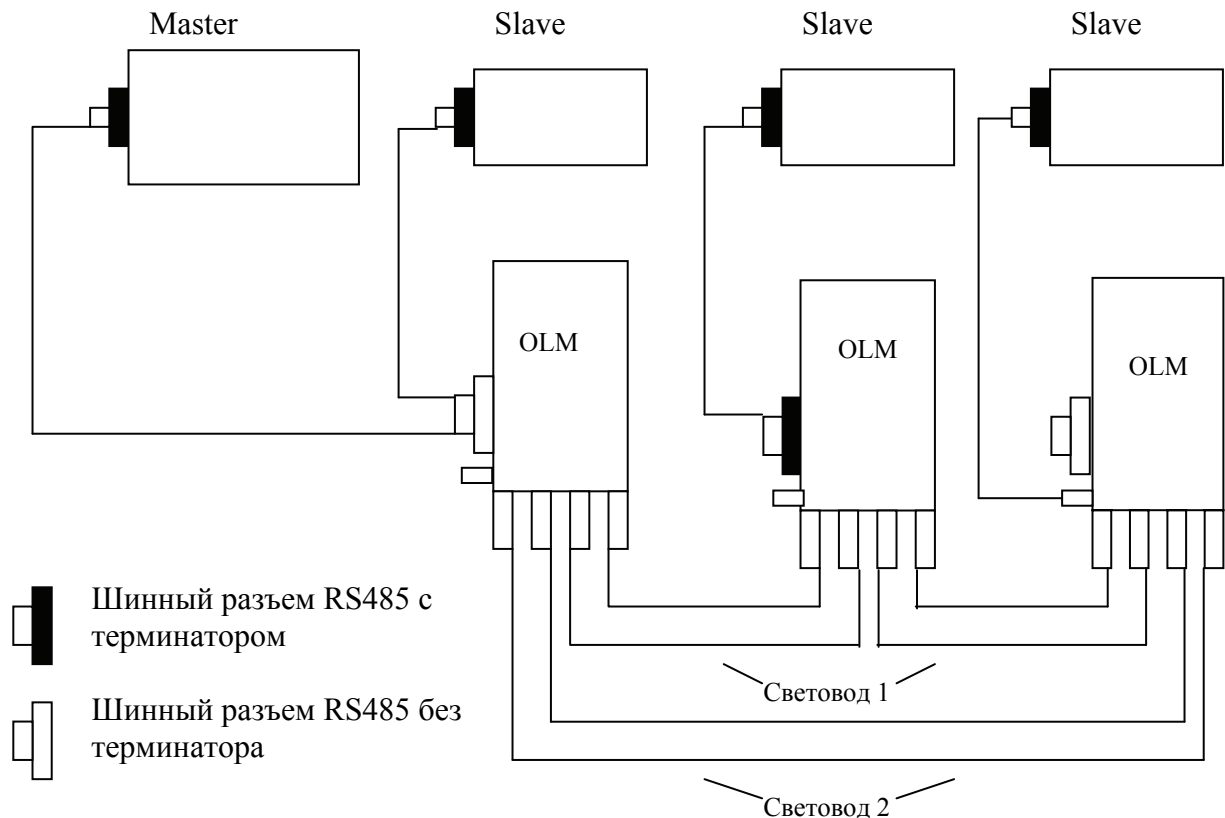


Рис 1.14 Резервированное двухпроводное (двухфазное) кольцо

При однопроводной структуре OLM соединяются друг с другом одинарным оптическим кабелем. Если появляется неисправность, как например, разрыв оптоволоконна или выход из строя OLM, кольцо разрывается. При резервированном оптоволоконном кольце OLM соединяются друг с другом двумя дуплексными оптическими кабелями. В этом случае система может сама распознать

неисправность и сконфигурировать шинную систему в линейную структуру. Сообщения о повреждениях сообщаются через соответствующие контакты и могут быть обработаны. Как только разрыв оптоволокну будет устранен, шинная система конфигурируется опять в резервированное кольцо.

1.4.2 Техника по IEC 1158-2 (PROFIBUS-PA)

С помощью PROFIBUS-PA могут быть реализованы отдельные структуры: линейные, древовидные, звездообразные, а также их комбинации.

Количество шинных сегментов, занятых участниками шины зависит от установленных источников питания, тока, потребляемого участниками, типа кабеля и экрана шинной системы. На шинную систему можно подключить до 32 участников. Чтобы повысить надежность системы, можно сделать сегмент резервированным. Подключение шинных сегментов PA к сегменту PROFIBUS-DP осуществляется с помощью сегментных разветвителей - DP/PA-Copler (рис.1.15) или сегментных соединителей - DP/PA-Link.

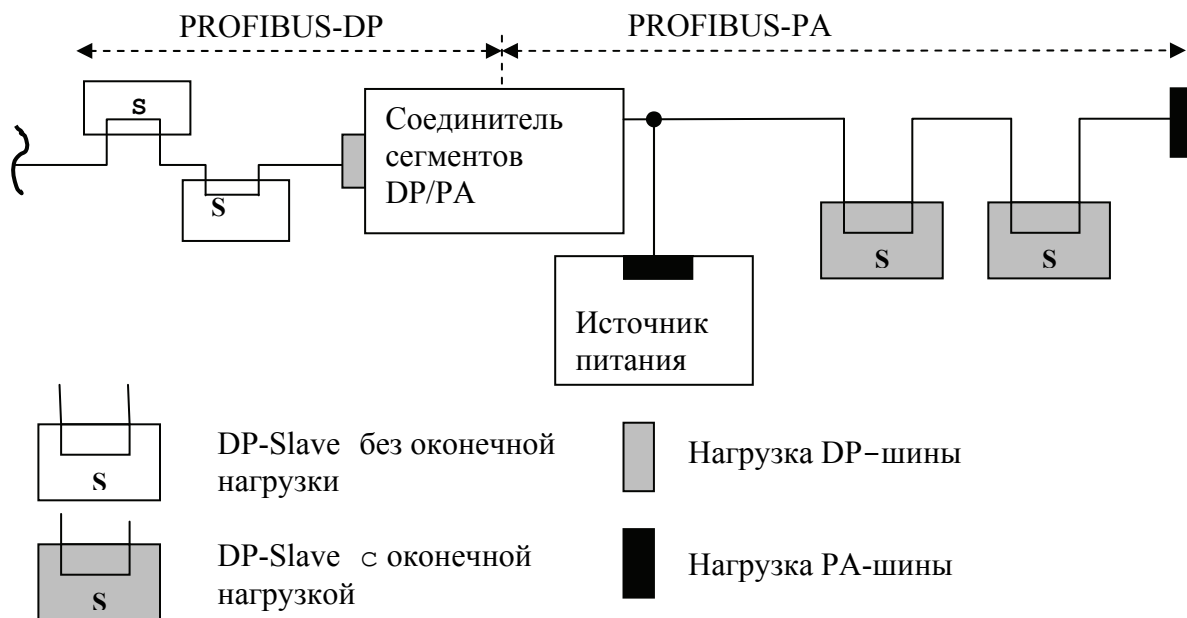


Рис. 1.15 Конфигурация шины с соединителем сегментов DP/PA

1.5 Управление доступом к шине в PROFIBUS

К управлению доступом к шине PROFIBUS предъявляются два существенных требования.

С одной стороны для надежных коммуникаций между равноправными приборами автоматизации или PC необходимо, чтобы каждый участник в течение

определенного временного окна получал доступ к шине для решения своих коммуникационных задач.

С другой стороны для обмена данными между сложными приборами автоматизации или РС и простой децентрализованной периферией требуется быстрый обмен данными с возможно малыми издержками протокола.

Это достигается благодаря гибридно построенному управлению доступом к шине, состоящим из децентрализованного обмена маркером (токеном) между активными участниками (Master'ами) и централизованного обмена Master-Slave для обмена данными между активными и пассивными участниками шины PROFIBUS.

Активный участник, который владеет маркером, берет на себя в данное время функции мастера на шине, чтобы проводить коммуникации с пассивными и активными участниками.

Обмен сообщениями по шине происходит при этом через адресацию участников. Каждому PROFIBUS-участнику назначается однозначный адрес. Адрес назначается из области от 0 до 126. При этом максимальное число участников, находящихся на шине, не превышает 127.

С этим управлением доступом к шине могут быть реализованы следующие конфигурации системы:

- “Чистая” система Master-Master (обмен маркером)
- “Чистая” система Master-Slave (Master-Slave)
- Комбинация обоих методов

Метод доступа к PROFIBUS не зависит от используемой среды передачи, например, медь или оптоволокно, и соответствует EN 50170, том 2.

1.5.1 Метод обмена маркером

Активные участники, подключенные к PROFIBUS, упорядочены по возрастанию их адреса в логическое маркерное кольцо (Token-Ring) (рис.1.16).

Под маркерным кольцом (Token Ring) здесь понимается организационное кольцо из активных участников, в котором маркер (Token) всегда передается от одного участника к следующему. Маркер, а с ним и право на доступ к среде передачи, передается при этом через специальную маркер-телеграмму между активными участниками. Исключение представляет активный участник с наивысшим на шине адресом HSA (Highest Station Address). Он передает маркер исключительно активному участнику с наименьшим шинным адресом, чтобы замкнуть маркерное кольцо.

Время одного обращения маркера через всех активных участников называется *временем обращения маркера*. С помощью устанавливаемого заданного времени обращения маркера Ttr (Time Target Rotation) определяется максимально разрешенное время обращения маркера.

Управление доступом к шине активных участников (MAC – Medium Access Control) осуществляется как на фазе инициализации, так и на фазе функционирования маркерного кольца.

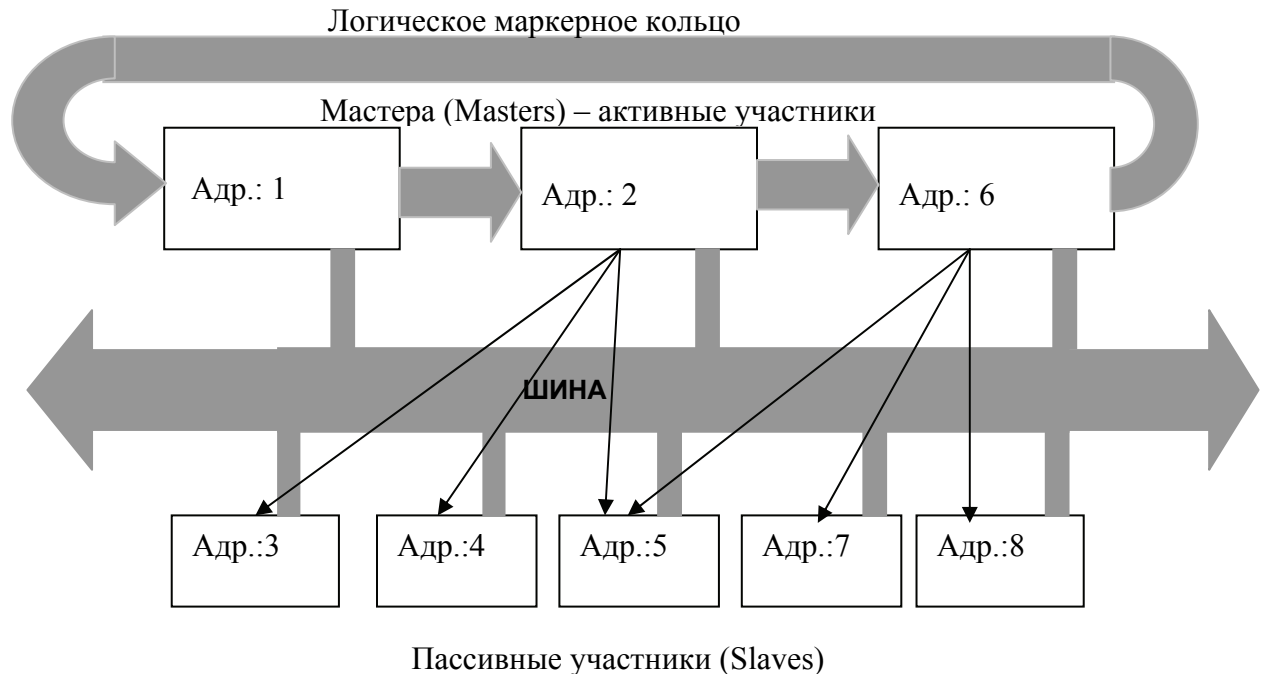


Рис. 1.16 Метод обмена маркером (токеном)

При этом устанавливаются адреса всех имеющихся на шине активных участников и заносятся в LAS (List of Active Station – список активных станций). Для управления маркером при этом особенно важны адреса предыдущей станции PS (Previous Station), от которой маркер получается, и следующей станции NS (Next Station), которой маркер предназначается. Кроме того, LAS также нужна, чтобы при текущей работе исключать из кольца вышедших из строя или дефектных активных участников и, соответственно, принимать вновь появившихся участников без помех текущему обмену данными по шине.

1.5.2 Метод Master-Slave

Если логическое маркерное кольцо состоит только из одного активного и нескольких пассивных участников, то это соответствует “чистой” системе Master-Slave (рис.1.17).

Метод Master-Slave делает возможным мастеру (активному участнику), который имеет право прямой передачи, опрашивать назначенных ему Slaves (пассивных участников). Мастер при этом имеет возможность принимать сообщения от Slave, и соответственно, передавать.

Типичная стандартная шинная конфигурация PROFIBUS-DP базируется на этом методе управления шиной. Активная станция (DP-Master) обменивается в циклической последовательности данными с пассивными станциями (DP-Slaves).

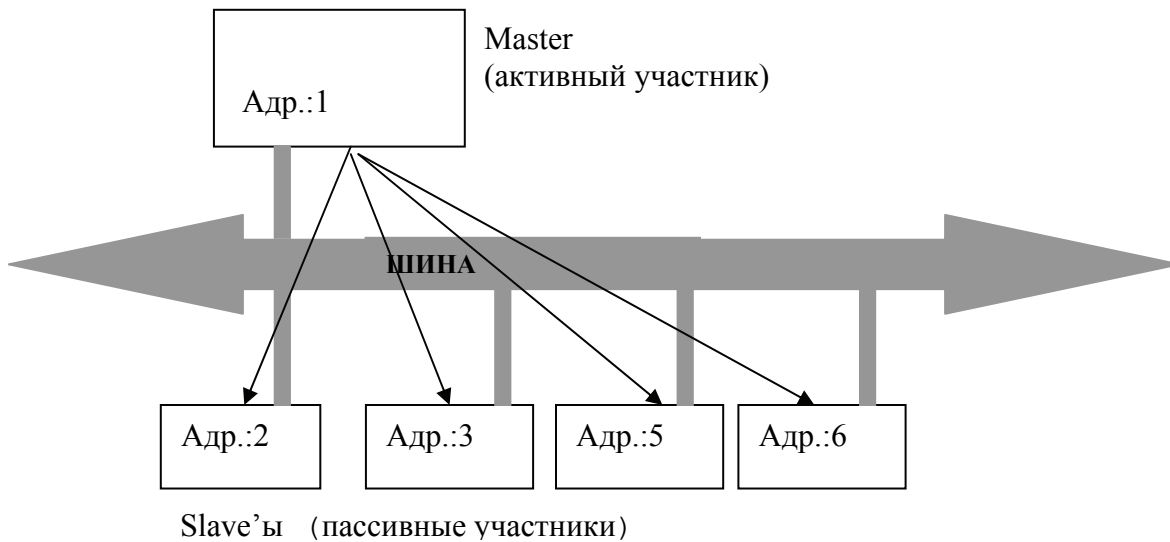


Рис. 1.17 Метод доступа Master-Slave

1.6 Шинные параметры

Безупречное функционирование сети PROFIBUS достигается только тогда, когда установленные шинные параметры соответствуют друг другу. Шинные параметры, заданные для одного участника, должны устанавливаться для каждого другого участника сети, так что они идентичны во всей сети. В целом шинные параметры зависят от выбранной скорости передачи и задаются соответствующим инструментом проектирования. Изменения этих наборов параметров должно проводиться только опытным персоналом. Важнейшие шинные параметры и их значения:

- **Tslot_Init** – предварительно установленное значение для Tslot; параметр Tslot изменяется в зависимости от конфигурации сети.
- **Tslot** – время ожидания приема. Время ожидания приема (slot-time) определяет максимальное время ожидания передатчика, то есть максимальное время, которое передатчик ожидает ответ от партнера.
- **Max. Tsdr** (Maximum Station Delay Responder) – максимальная задержка ответа станции. Определяет максимальный отрезок времени, требуемый отвечающим узлом, чтобы редактировать протокол.
- **Min. Tsdr** (Minimum Station Delay Responder) – минимальная задержка ответа станции. Определяет минимальный отрезок времени, требуемый отвечающим узлом, чтобы редактировать протокол.

- **Tset** (Setup Time) – время установки – время, которое может пройти между приемом “телеграммы данных” и реакцией на нее (ответа на телеграмму)
- **Tqui** (Quier-Time for Modulator) – время переключения модулятора. Время, требуемое, чтобы переключиться из режима передачи на режим приема.
- **GAP-Factor** – определяет, через сколько оборотов маркера новый участник, подключенный к маркерному кольцу, будет включен в это кольцо. (GAP – пропуск, пробел в адресной области от собственного адреса участника до адреса следующего активного участника шины)
- **Retry Limit** – максимальное количество повторений вызова. Параметр устанавливает, сколько попыток предпринимается максимально, чтобы достичь участника.
- **Tid2** (idle-time 2) – время покоя 2. Определяет время задержки после отправки запроса без ответа.
- **Trdy** (ready-time) – время для квитирования или ответа
- **Tid1** (idle-time 1) – время покоя 1. Устанавливает время ожидания (задержки) после приема ответа.
- **Ttr** (Target-Rotation-Time) – заданное время обращения маркера – максимальное время, имеющееся в распоряжении для обращения маркера. В этот промежуток времени все активные участники (Master’а) один раз получают маркер. Разность между заданным временем обращения маркера и фактическим временем обращения маркера определяет, сколько времени остается у Master’ов для отправки телеграмм (данных) Slave’ам.
- **Ttr typicalli** (Tipical Ttr) – типичное время цикла – среднее время реакции на шине, если все спроектированные Slave’ы обмениваются с DP-Master’ами данными, ни один Slave не сообщает диагностических данных и нет никакой дополнительной передачи по шине для PG.
- **Response Monitoring** – время, в течение которого Master реагирует на выход из строя Slave.

Все шинные параметры описывают таким образом времена, которые должны точно соответствовать друг другу. Единицей для измерения этих параметров является tBIT (time_Bit). Один tBIT – это время передачи по шине одного бита и называется также временем передачи бита. Это время зависит от скорости передачи и вычисляется следующим образом:

$$tBIT = 1/\text{скорость передачи (бит/с)}$$

Например, для скорости передачи 12 Мбит/с время передачи бита – 83 ns, а для скорости передачи 1,5 Мбит/с – 667 ns.

2. Типы приборов и обмен данными в PROFIBUS-DP

Введение

PROFIBUS-DP выполняет высокие временные требования для обмена данными в области децентрализованной периферии и полевых устройств.

Типичной DP-конфигурацией обладает одномастерная структура (Mono-Master-Struktur – нем.) (рис.2.1)

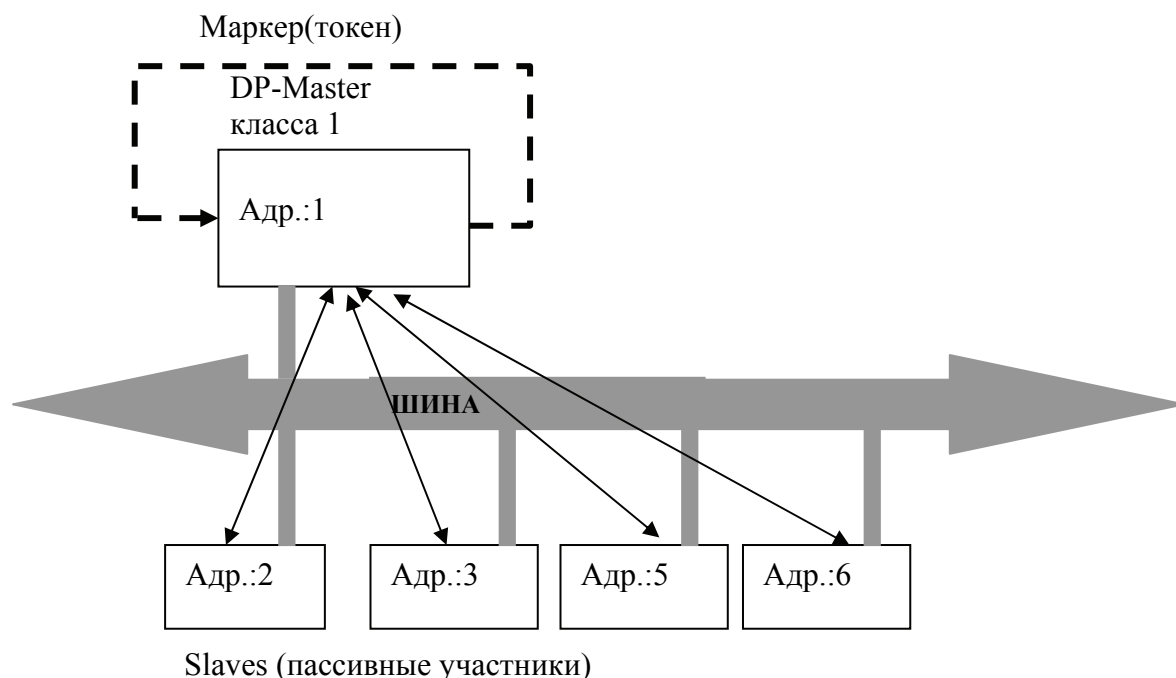


Рис. 2.1 Структура с одним мастером

Коммуникации между DP-Master'ом и DP-Slave'ом осуществляется по принципу Master-Slave. Это означает, что DP-Slaves только по требованию DP-Master'а становятся активными на шине. DP-Slaves для этого располагаются в списке вызовов (Polling-Liste) DP-Master'а друг за другом.

Обмен пользовательскими данными между DP-Master'ом и DP-Slave'ом происходит циклически, без учета содержания пользовательских данных.

Рис.2.2 показывает принципиальную обработку списка вызовов (Polling-Liste) в DP-Master'е.

Цикл сообщений между DP-Master'ом и одним DP-Slave'ом состоит из *кадра запроса (Request Frame)* DP-Master'а и принадлежащего DP-Slave'у подтверждения или *кадра подтверждения (Response Frame)* (квитирование).

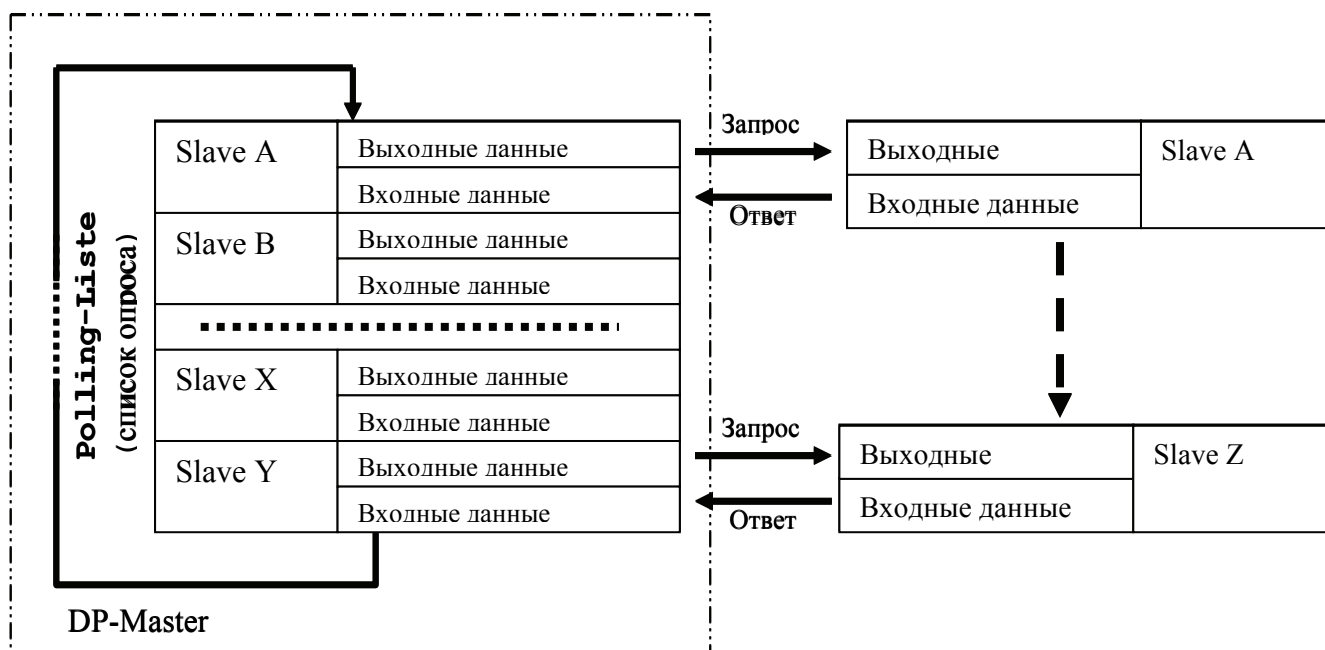


Рис.2.2 Обработка списка опроса DP-Мастером

Определенные в EN 50170 свойства участников PROFIBUS на уровнях 1 и 2, могут иметь DP-системы с несколькими мастерами (Multi-Master-Struktur – нем.)

На практике это означает, что к шине подключено несколько станций DP-Master. Master'а между собой могут обмениваться пользовательскими данными, например, с помощью FMS-коммуникаций (рис.2.2).

2.1 Типы приборов

2.1.1 DP-Master (класс 1)

Этот DP-Master обменивается пользовательскими данными с DP-Slaves циклически. В частности он имеет задачи, которые выполняются следующими функциями протокола:

- *Set_Prm* и *Chk_Cfg*

Благодаря параметрированию DP-Slaves на старте, рестарте и фазе обмена данными, DP-Slave'ам передаются специфические для них параметры.

При конфигурировании определяется число входных и выходных байтов для каждого DP-Slave.

- *Data_Exchange*

Циклический обмен входными и выходными данными с DP-Slave'ами, назначенными данному DP-Master'у.

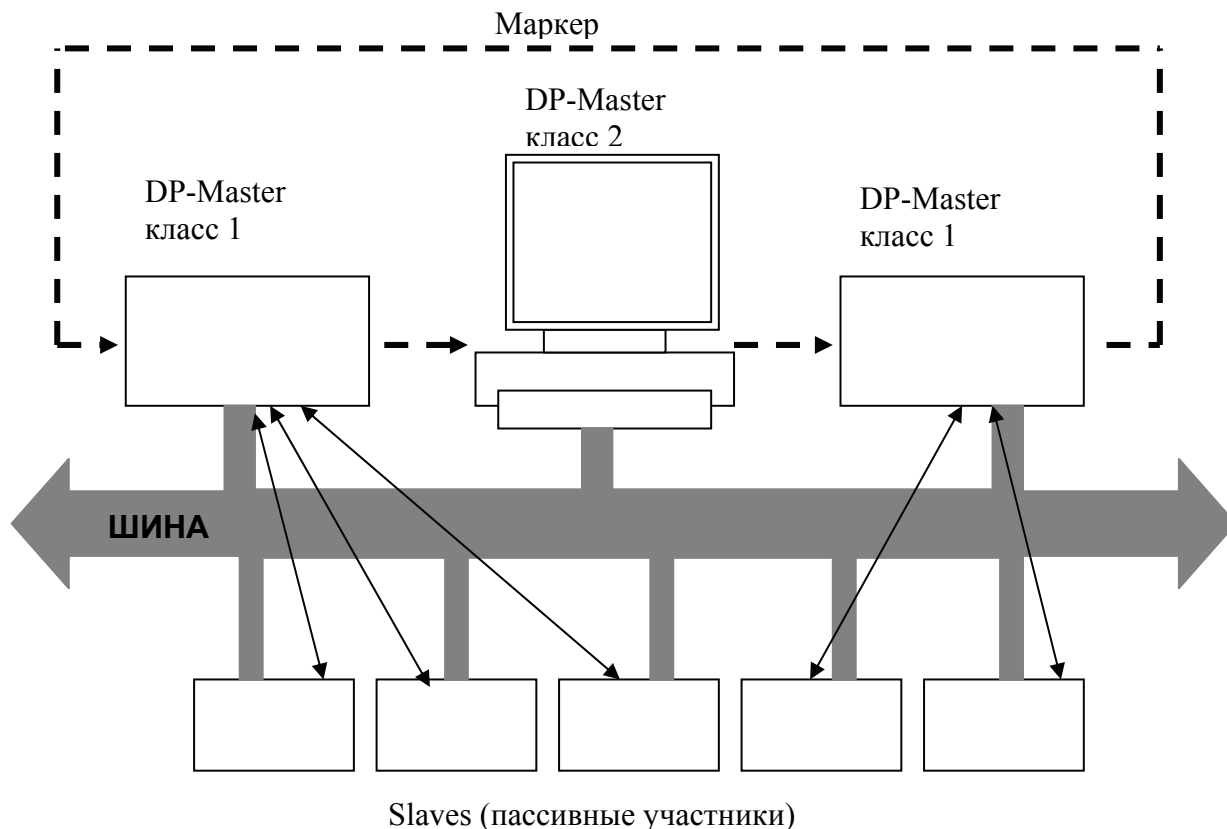


Рис. 2.3 Структура PROFIBUS с несколькими мастерами

- *Slave_Diag*

На фазе запуска или во время циклического обмена пользовательскими данными читается диагностическая информация из DP-Slave.

- *Global_Control*

Благодаря командам управления сообщает DP-Slave DP-Master'у свое рабочее состояние. Далее могут посылаться управляющие команды отдельным DP-Slave'ами или определенным группам DP-Slave'ов для синхронизации входных и выходных данных (команды Sync и Freeze).

2.1.2 DP-Slave

DP-Slave обмениваются пользовательскими данными только с DP-Master'ом, который его предварительно параметрировал и конфигурировал. DP-Slave в состоянии сообщить DP-Master'у свою диагностику или выдать сигнал о событии процесса.

2.1.3 DP-Master (класс 2)

DP-Master (класс 2) – это программаторы, приборы для диагностики и управления шиной, которые поддерживают кроме уже названных функций DP-Master’а (класс 1), еще другие специальные функции. Это:

- *RD_Inp* и *RD_Outp*

Читаются входные и выходные данные от DP-Slave параллельно с обменом данными с DP-Master’ом.

- *Get_Cfg*

С помощью этой функции считываются актуальные конфигурационные данные DP-Slave.

- *Set_Slave_Add*

С помощью этой функции DP-Slave’ам и DP-Master’ам можно назначать новый шинный адрес, на сколько они это поддерживают.

Возможен еще ряд коммуникационных функций DP-Master’а (класс 2) для работы с DP-Master’ом (класс 1).

2.1.4 Комбинированные приборы DP

Возможна комбинация между названными типами DP-приборов внутри одного модуля. Например, на практике типичны комбинации

- DP-Master (класс 1) с DP-Master’ом (класс 2)
- DP-Slave с DP-Master’ом (класс 1)

2.2 Обмен данными между типами DP-приборов

2.2.1 DP-коммуникационные связи и DP-обмен данными

Инициатор коммуникационного задания обозначается в PROFIBUS-DP как *Requestor* (“запросчик”), а соответствующий партнер по коммуникациям – как *Responder* (“ответчик”). Все телеграммы-запросы DP-Master’а (класс 1) передаются на уровень 2 как служебные телеграммы “high-prio” (высокого приоритета). Соответствующие телеграммы-ответы DP-Slave’ов передаются на уровень 2 как служебные телеграммы “low-prio” (низкого приоритета).

DP-Slave имеет возможность при однократном обмене изменить приоритет “low-prio” на “high-prio” для ответной телеграммы. Это необходимо при передаче диагностических сообщений или сообщения о событии.

Передача данных происходит без установления соединения через соединения one-to-one или one-to-many (один с одним или один со многими) (только команды управления и перекрестная связь).

В табл.2.1 показаны коммуникационные возможности отдельно по функциям запроса и ответа для DP-Master и DP-Slave.

Табл. 2.1 Коммуникационные связи между DP-приборами

Функция/служба (по EN 50170)	DP-Slave		DP-Master (класс 1)		DP-Master (класс 2)		Через SAP №	Через службу уровня 2
	Requ	Resp	Requ	Resp	Requ	Resp		
Data Exchange		M	M	O			Default-SAP	SRD
RD_Inp		M		O			56	SRD
RD_Outp		M		O			57	SRD
Slave_Diag		M	M	O			60	SRD
Set_Prm		M	M	O			61	SRD
Chk_Cfg		M	M	O			62	SRD
Get_Cfg		M		O			59	SRD
Global Control		M	M	O			58	SRD
Set Slave Add		O		O			55	SRD
M-M- Kommunikation			O	O	O		54	SRD/SDN
DP-V1-служба		O	O	O			51/50	SRD

Requ = Requester – “запросчик”;

Resp = Responder – “ответчик”;

M = Mandatory Function – обязательная функция;

O = Optional Function – необязательная (по выбору) функция.

2.2.2 Фаза инициализации, перезапуск и движение пользовательских данных

Как можно сделать вывод из рис. 2.4, DP-Master должен параметризовать и конфигурировать DP-Slave перед тем, как он сможет обмениваться пользовательскими данными с DP-Slave. Если это имеет место, то готовность к работе DP-Slave’a проверяется DP-Master’ом с помощью диагностических данных. Если DP-Slave сообщает готовность для параметрирования, DP-Master посылает сразу же после этого данные параметрирования и конфигурирования. После повторной проверки готовности к работе DP-Slave’a с помощью диагностических данных, DP-Master начинает циклически обмениваться с DP-Slave’ом пользовательскими данными.

Данные параметрирования (Set_Prm)

С помощью данных параметрирования сообщаются DP-Slave’у необходимые локальные и глобальные параметры, свойства и функции. Содержание данных параметрирования устанавливается при проектировании DP-Master’a. Это происходит при создании проекта во время проектирования DP-Slave’ов, часто в прямом диалоге или косвенно через доступ к имеющимся параметрам и специфическим для DP-Slave’a GSD-файлам (Geräte Stamm Daten – данные о происхождении прибора -нем.).

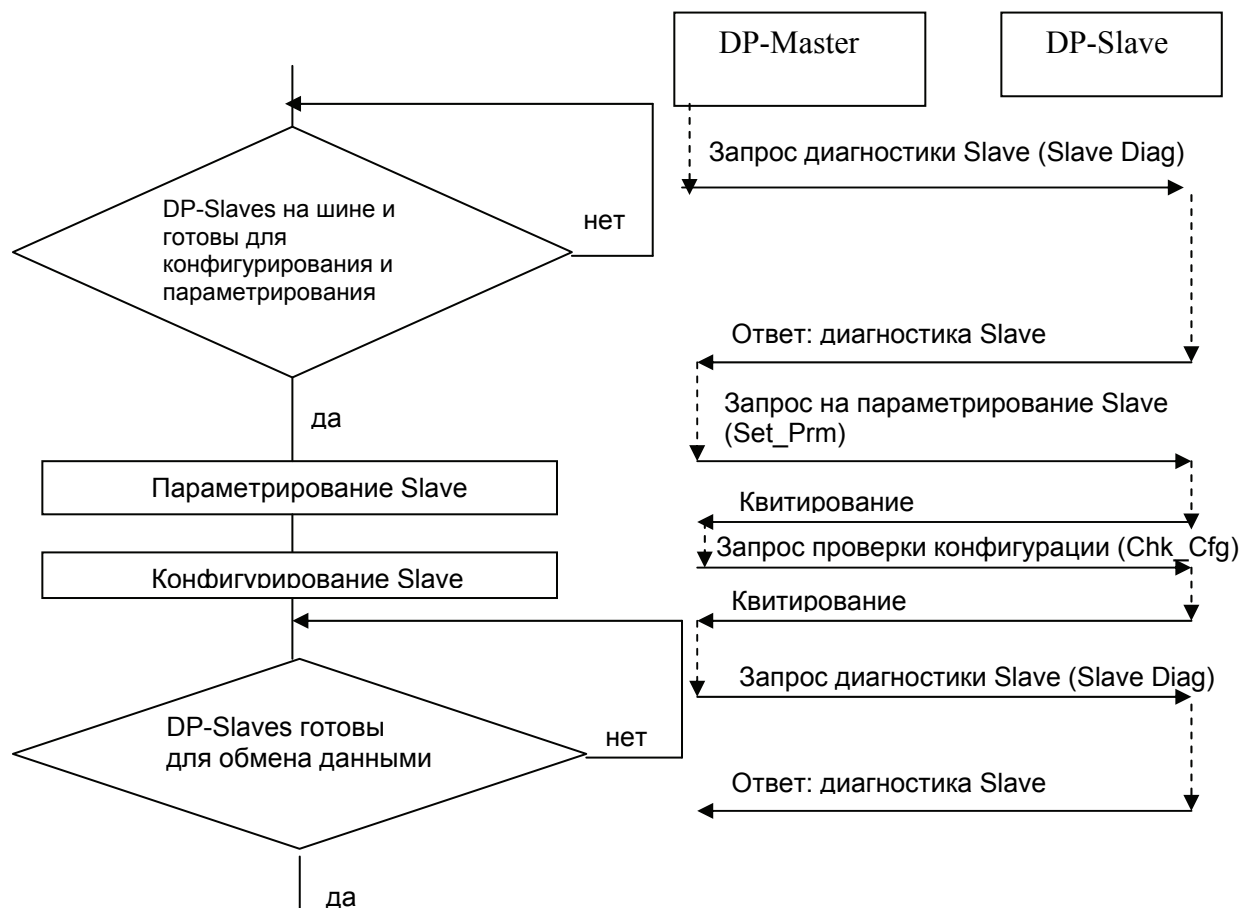


Рис. 2.4 Фаза инициализации DP-Slave

Структура телеграммы для параметрирования состоит из определенных в EN 50170 частей и, если это нужно, специфической для пользователя и DP-Slave'a части.

Важнейшие составляющие телеграммы параметрирования:

- *Station-Status*

Состояние станции содержит специфические для Slave'a функции и настройки. Так, например, здесь определяется, должен ли быть активирован контроль срабатывания. Определяется, открыт или закрыт доступ к DP-Slave'у и, если это предусмотрено при проектировании, должен ли DP-Slave работать с управляющими командами Sync и Freeze.

- *Watchdog*

Watchdog – контроль времени срабатывания – должен определять выход из строя DP-Master'a. Если времени контроль срабатывания активирован и определил выход из строя DP-Master'a, стираются локальные выходные данные (необходимо использовать заменяющие значения). DP-Slave может эксплуатироваться на шине с или без контроля времени срабатывания. Среда проектирования предлагает руководствуясь шинной конфигурацией и установленной скоростью передачи время контроля срабатывания, которое может быть получено при проектировании (см. также шинные параметры).

- *Ident-Number*

Идентификационный номер (ID) DP-Slave'a присваивается PNO (Profibus Nutzer Organisation (нем.) – организация пользователей PROFIBUS) при сертификации. ID DP-Slave'a хранится в GSD-файле. DP-Slave принимает телеграмму параметризации, если только полученный ID совпадает с собственным. Благодаря этому предотвращается ошибочное параметрирование.

- *Group-Ident*

Благодаря групповому идентификатору можно объединить DP-Slave'ы в группы для управляющих команд “Sync” и “Freeze”. Возможно максимум 8 групп.

- *User-Prm-Data*

Благодаря параметрируемым данным DP-Slave'ов (User-Prm-Data) устанавливаются для DP-Slave'ов специфические пользовательские данные. Это могут быть, например, начальные установки или регулировочные параметры.

Данные конфигурирования (Chk_Cfg)

С помощью конфигурационной телеграммы DP-Master сообщает DP-Slave'у через формат опознавания объем и структуру участвующих в обмене входных/выходных данных. Эта область, называемая также модулем, согласуется по форме между DP-Master'ом и DP-Slave'ом: байтовая структура или структура из слов (формат опознавания). Через этот формат опознавания можно устанавливать на модуль входные/выходные области размером максимум в 16 байт/слов. Внутри формата конфигурационной телеграммы различаются следующие, зависящие от DP-Slave, установки:

- DP-Slave имеет статически определенные области входов/выходов. Например, периферийный модуль ET200B.
- DP-Slave имеет в зависимости от конфигурации/выполнения динамические области входов/выходов. Например, периферийные модули ET200M или привод.
- Области входов/выходов DP-Slave устанавливаются через специальный формат опознавания, зависящий от DP-Slave и изготовителя. Например, S7-DP-Slave аналоговый модуль ET200B, DP/AS I-Link и ET200M).

Области входных/выходных данных, которые содержат связанную информацию и которые не могут передаваться структурами байтов или слов, должны обрабатываться как *консистентные данные*.

К ним относятся, например, области параметров для регуляторов или наборы параметров для привода. С помощью специальных форматов опознавания (зависящих от DP-Slave и изготовителя) можно устанавливать области входов и выходов (модули) с максимальной длиной 64 байт/слов.

Пользовательские области входов и выходов (модули) DP-Slave'ов хранятся в GSD-файлах и при проектировании предлагаются соответствующим инструментом проектирования.

Диагностические данные (Slave_Diag)

Посредством запроса диагностических данных DP-Master проверяет на фазе запуска, имеется ли DP-Slave и готов ли он для параметрирования. Сообщаемые DP-Slave'ом диагностические данные состоят из определенной в EN 50170 диагностической части и необязательной, специфической для DP-Slave диагностической информации. Через диагностические данные DP-Slave сообщает DP-Master'у свое рабочее состояние и в случае диагностики – причину диагностического сообщения. DP-Slave имеет возможность сообщать с помощью ответных телеграмм службы Data_Exchange уровня 2 с “high-prio” на уровень 2 DP-Master'а локальные диагностические данные, которые требует DP-Master для оценки.

Если нет актуального диагностического события, то ответная телеграмма службы Data_Exchange имеет идентификатор “low-prio”.

Диагностические данные DP-Slave'ов могут быть, однако, без специального сообщения от диагностического события затребованы DP-Master'ом в любое время.

Пользовательские данные (Data_Exchange)

DP-Slave проверяет принятые от DP-Master'а данные параметрирования и конфигурирования.

Если нет ошибок и DP-Master'ом разрешены желаемые установки, DP-Slave сообщает DP-Master'у, что он готов для обмена данными. С этого момента DP-Master обменивается с DP-Slave'ом запроектированными пользовательскими данными (рис.2.5).

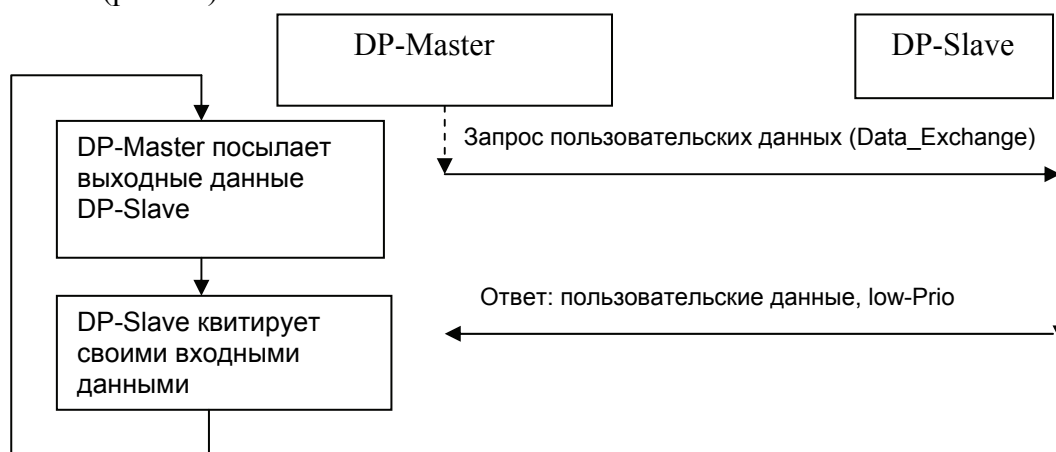


Рис. 2.5 Циклический обмен пользовательскими данными DP-Slave с DP-Master'ом

При обмене пользовательскими данными DP-Slave реагирует на телеграммы-запросы Data_Exchange DP-Master'а (класс 1), который его параметрировал и конфигурировал. Другие телеграммы пользовательских данных DP-Slave отбрасывает. Внутри пользовательских данных нет дополнительных управляющих или структурных знаков для описания передаваемых данных, то есть передаются чистые пользовательские данные. Как изображено на рис.2.6, DP-Slave может во время ответа через изменение класса приоритета служебных телеграмм с “low-prio” на “high-prio” сообщить DP-Master'у, что имеется

актуальное диагностическое событие или диагностическая информация о состоянии. Настоящая диагностическая информация или информация о состоянии сообщается DP-Slave'ом один раз в ответ на это DP-Master'у диагностической телеграммой по его требованию. После получения диагностических данных продолжается обмен с DP-Slave'ом данными, как запроецировано пользователем. С помощью телеграмм запрос-ответ можно обмениваться пользовательскими данными между DP-Master'ом и DP-Slave'ом в обоих направлениях объемом до 244 байт.

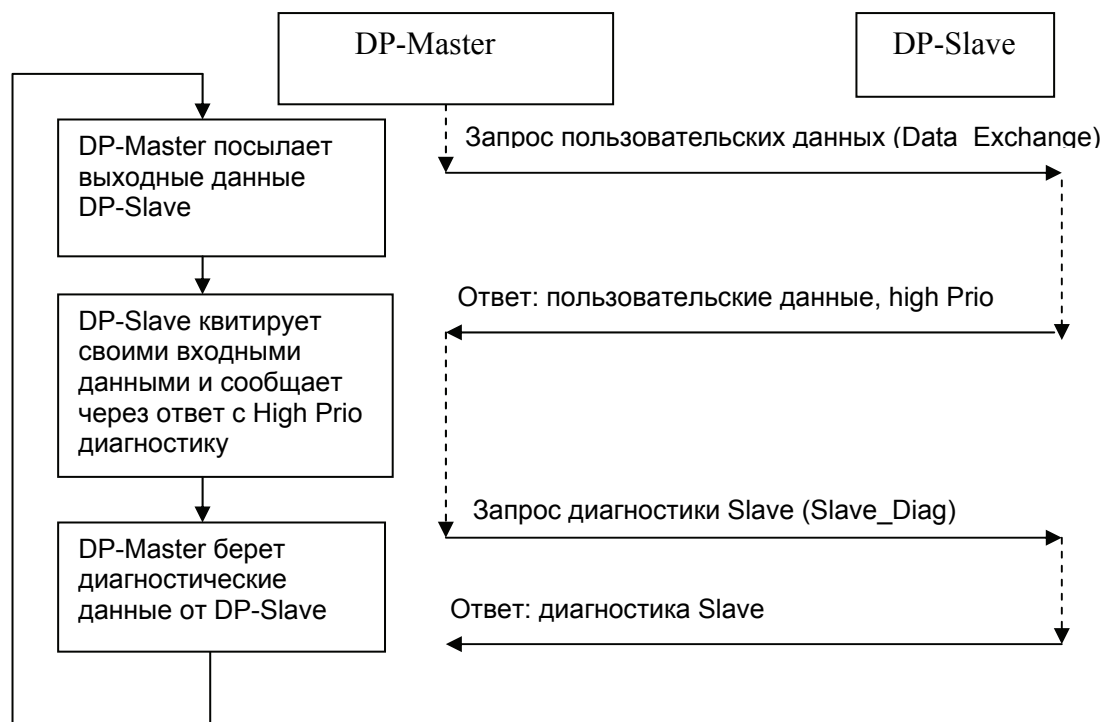


Рис. 2.6 DP-Slave сообщает о актуальном диагностическом событии

2.3 Цикл PROFIBUS-DP

2.3.1 Структура циклов PROFIBUS

Рис. 2.7 показывает структуру DP-цикла в шинной системе DP с одним мастером.

Постоянная компонента DP-циклов составляет при этом часть цикловой телеграммы, состоящей из управления доступом к шине (управление маркером и состоянием участников) и обменом данными (Data_Exchange) с DP-Slave'ами. Наряду с этим циклическим движением данных есть однако внутри DP-цикла также ряд зависящих от событий, *ациклических телеграмм*.

К этим ациклическим телеграммам относятся:

- Обмен данными во время фазы инициализации DP-Slave
- Диагностические функции DP-Slave'a

- Коммуникации с DP-Master'ом 2-го класса
- Коммуникации с другими мастерами
- Обусловленные уровнем 2 повторения телеграмм при помехах
- Ациклическое движение данных по DP-V1
- On-line функции с PG
- HMI - функции

В зависимости от этой ациклической телеграммы конкретный DP-цикл может удлиняться.

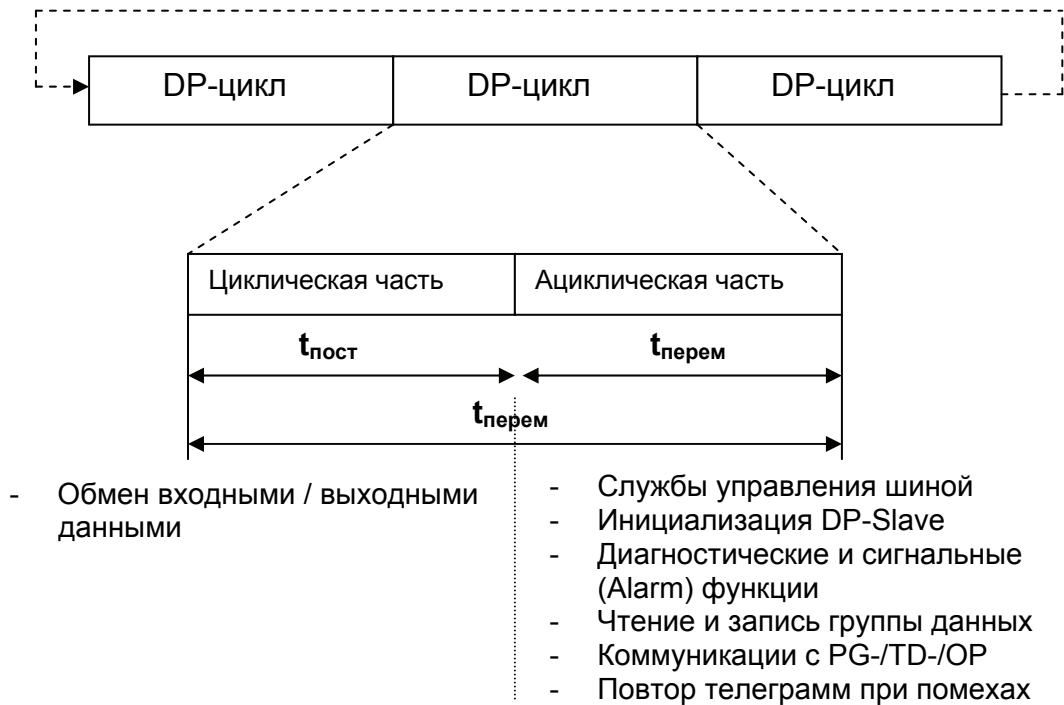


Рис.2.7 Структура цикла PROFIBUS-DP

Таким образом, шинный цикл всегда состоит из постоянной по времени циклической части и существующей не всегда, зависимой от событий, переменной ациклической части телеграммы.

2.3.2 Структура постоянного по времени PROFIBUS-DP цикла

В некоторых случаях при автоматизации выгодно, когда шинный цикл DP по времени остается одинаковым и, таким образом, обмен данными может происходить строго периодически. Это находит применение, например, в области техники приводов для самосинхронизации нескольких приводов.

В отличие от нормального цикла DP, DP-Master'ом предусматривается (резервируется) при постоянном по времени цикле DP для ациклической части коммуникаций определенная часть времени.

Как представлено на рис.2.8, DP-Master для этого обеспечивает, чтобы эта зарезервированная временная часть была не превышена благодаря тому, что он допускает только определенное число ациклических событий.

Если зарезервированное время не нужно, то DP-Master заполняет паузу “пустым временем”. Благодаря этому гарантируется, что заданное постоянное время будет выдержано с точностью до микросекунды.

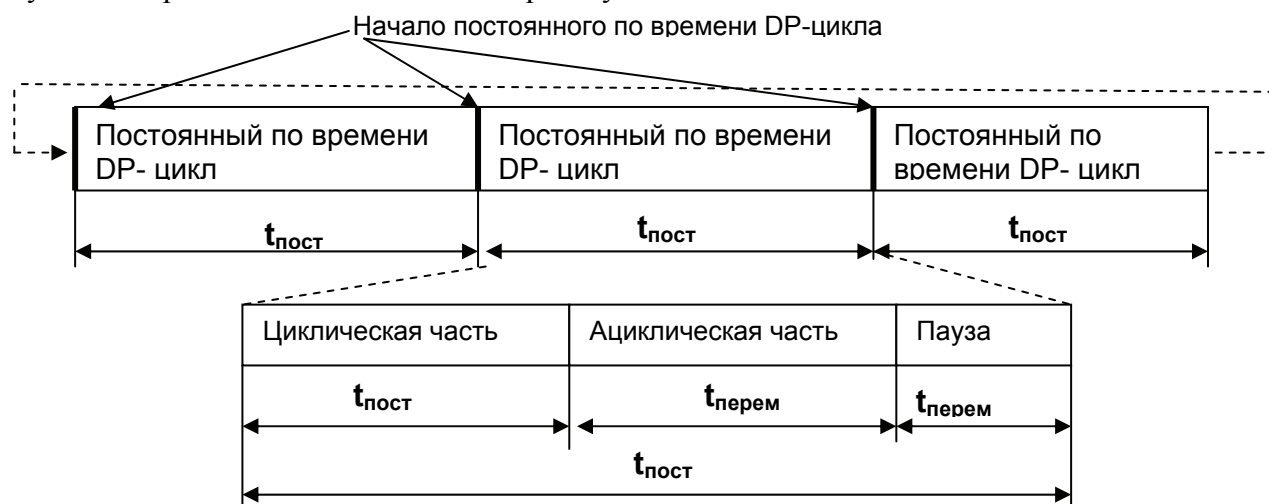


Рис.2.8 Структура постоянного по времени цикла PROFIBUS-DP

Задание времени для постоянного шинного цикла DP осуществляется при создании проекта в STEP 7. Предлагаемое в STEP 7 значение времени определяется, руководствуясь спроектированной конфигурацией установки и учитывая определенные, типичные части ациклических служб. При проектировании постоянного по времени цикла есть возможность изменить предлагаемое STEP 7 значение его длительности.

В настоящее время постоянный по времени DP-цикл может быть установлен только в системе с одним мастером.

2.4 Обмен данными через перекрестную связь

Дальнейшая возможность обмена данными в PROFIBUS-DP при применении SIMATIC S7 заключается в использовании перекрестной связи. При спроектированной перекрестной связи DP-Slave отвечает DP-Master'у не телеграммой one-to-one (Slave -> Master), а специальной телеграммой one-to-many (Slave -> nn). Таким образом входные данные Slave'а, содержащиеся в ответной телеграмме, предоставляются не только соответствующему мастеру, но и всем участникам шины.

Возможные при перекрестной связи коммуникации “Master-Slave” и ”Slave-Slave” поддерживаются не всеми SIMATIC S7 DP-Master'ами и не всеми вариантами DP-Slave'ов. Если такие коммуникации поддерживаются, то они проектируются с помощью STEP 7. Возможна смесь из обоих вариантов коммуникационных связей.

2.4.1 Коммуникационная связь Master-Slave при перекрестной связи

На рис. 2.9 представлены возможные коммуникационные связи на основе конфигурации, состоящей из 3-х S7 DP-Master'ов и 4-х DP-Slave'ов. Все представленные на этом рисунке DP-Slave'ы посылают свои входные данные, как телеграммы one-to-many.

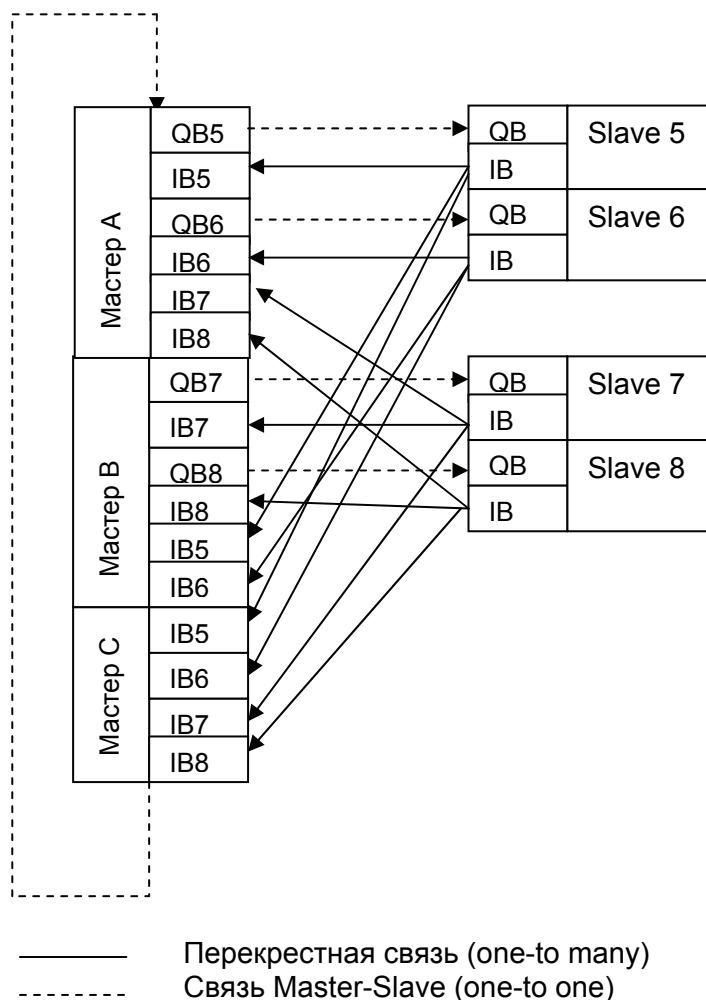


Рис. 2.9 Коммуникационные связи Master-Slave при перекрестной связи

DP-Master A, которому приписаны Slave'ы 5 и 6, принимает также входные данные Slave'ов 7 и 8. Также DP-Master B, которому приписаны Slave'ы 7 и 8, принимает входные данные Slave'ов 5 и 6. DP-Master'у C Slave'ов не приписано вовсе. Однако этот DP-Master принимает входные данные всех Slave'ов, эксплуатирующихся в шинной системе.

2.4.2 Коммуникационная связь Slave-Slave при перекрестной связи

Следующий вариант обмена данными при перекрестной связи представляет коммуникационную связь Slave-Slave, изображенную на рис.2.10, при использовании в качестве Slave'ов I- Slave'ов (интеллектуальных Slave'ов, см. раздел 3.4.3), как, например, CPU 315-2DP.

При этом I- Slave также в состоянии принимать входные данные от других DP-Slave'ов.

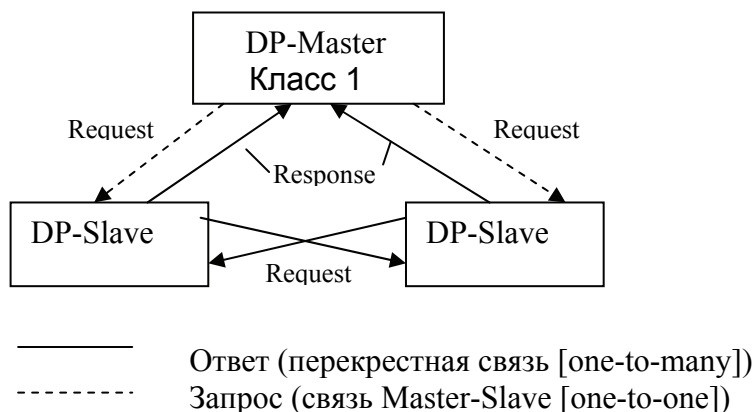


Рис.2.10 Коммуникационные связи Slave-Slave при перекрестных связях

2.5 Функциональное расширение DPV1

Постоянно усложняющиеся требования к DP-Slave'ам требуют расширенной коммуникационной функциональности PROFIBUS-DP. Это относится к ациклическому обмену данными, а также к способности посылки Alarm'ов.

Чтобы выполнить эти требования, международный стандарт EN 50170 том 2 был расширен. Описанные в стандарте расширения касаются как DP-Slave'ов, так и DP-Master'ов. Эти функциональные расширения, называемые также DPV1-расширениями, являются опциональными (необязательными) для стандартного протокола. Благодаря этому гарантируется, что прежние PROFIBUS-DP-полевые приборы и приборы с расширением DPV1 могут совместно эксплуатироваться и этим достигается внутренняя совместимость.

При этом справедливы следующие правила:

- DP-Slave с DPV1-расширением может работать с DP-Master'ом без DPV1-функциональности. DPV1-функциональность DP-Slave может не использоваться.
- DP-Slave без DPV1-расширения может без ограничений эксплуатироваться с DPV1-Master'ом.

Для краткости DP-Master с DPV1-расширением будем называть также DPV1-Master'ом. Это справедливо также для DP-Slave'ов, которые поддерживают расширение EN 50170. Их будем называть DPV1-Slave'ами.

Благодаря расширению стандарта таким образом открыт путь для создания нового поколения полевых приборов. При проектировании чаще всего ставится однако вопрос, какие различия существуют между различными вариантами DP-Slave'ов.

- Стандартные DP-Slave'ы (DP-Norm-Slaves) обладают только основной функциональностью, которая описана в стандарте EN 50170, без DPV1-расширения. Таким образом у DP-Slave'а не может быть ациклического обмена данными и в модели Alarm'ов этих Slave'ов предусмотрены только диагностические сигналы. Стандартные DP-Slave'ы проектируются с помощью GSD-файлов (GerTMte-StamDaten-файл) в соответствующем инструменте проектирования.
- DPS7-Slave'ы – дальнейшее развитие стандартных DP-Slave'ов фирмы SIEMENS. Это расширение может однако использоваться только с модулями-мастерами SIMATIC S7. У DPS7-Slave'ов возможен ациклический обмен данными. В них также встроена расширенная модель Alarm'ов. Если DPS7-Slave'ы проектируются через GSD-файлы и подключаются к DP-Master'у другого производителя, то они ведут себя как стандартные DP-Slave без DPV1-расширения по EN 50170, том 2. Полная функциональность DPS7-Slave'ов достигается только при проектировании с помощью SIMATIC STEP7 и при работе с DP-Master'ами SIMATIC S7.
- DPV1-Slave'ы – это Slave'ы с DPV1-расширением по EN 50170 том 2. Это расширение относится к модели Alarm'ов и стандартизации ациклического обмена данными. DPV1-Slave'ы могут работать с любым DPV1-Master'ом с полной функциональностью. Эти Slave'ы проектируются с помощью GSD-файлов версии 3.

Таблица 2.2 дает обзор, какие диагностические и аварийные сигналы могут быть в каких DP-Slave'ах. При этом предполагается, что Slave'ы работают с соответствующим Master'ом.

Табл. 2.2 Имеющиеся Alarm'ы и ациклический обмен данными у DP-Slave'ов

	Стандартные DP-Slave'ы	DPS7-Slave'ы	DPV1-Slave'ы
Диагностические сигналы	X	X	X
Сигнал от процесса	-	X	X
Сигнал при удалении блока	-	X	X
Сигнал при вставке блока	-	X	X
Сигнал состояния	-	-	X
Сигнал модернизации	-	-	X
Сигнал, специфический для изготовителя	-	-	X
Ациклический обмен данными	нет	Да, с модулем S7-DP-Master	Да, с модулем DPV1-Master

3. PROFIBUS-DP в системе SIMATIC S7

Введение

PROFIBUS – составная часть системы SIMATIC S7. Децентрализованная периферия (DP) обрабатывается благодаря связи через проект STEP 7, как центральная периферия. Поведение системы при выходе из строя, диагностике и alarm'ах от SIMATIC S7 DP-Slave'ов также соответствует поведению центральной периферии. Через встроенный или устанавливаемый интерфейс PROFIBUS-DP можно также подключать к системе автоматизации полевые приборы со сложными техническими функциями. Свойства PROFIBUS, определенные на уровнях 1 и 2 и прозрачность внутренних системных коммуникационных возможностей (S7-функции) позволяют эксплуатировать в системах SIMATIC S7 PROFIBUS-DP программаторы (PG), PC, а также приборы обслуживания и наблюдения.

3.1 DP-интерфейсы в системах SIMATIC S7

В системах SIMATIC S7-300 и SIMATIC S7-400 различают два варианта интерфейсов PROFIBUS-DP:

- *Встроенные в CPU интерфейсы* (CPU 315-2, CPU 316-2, CPU 318-2, CPU 412-1, CPU 412-2, CPU 413-2, CPU 414-2, CPU 414-3, CPU 416-2, CPU 416-3, CPU 417-4)
- *Интерфейс, образованный с помощью подключения в PLC IM (интерфейсного модуля) или CP (коммуникационного процессора) (IM467, IM467-FO, CP443-5 (Extended), CP342-5)*

В зависимости от рабочих характеристик CPU различают также рабочие характеристики DP-интерфейса. В таблицах с 3.1 по 3.4 представлены важнейшие технические данные интерфейсов PROFIBUS-DP для SIMATIC S7-300 и SIMATIC S7-400, которые как встроены в CPU, так и подключаемы к ним. За исключением CP342-5 децентрализованная периферия обрабатывается через DP-интерфейс так же, как центральная. DP-интерфейс CP342-5 работает отдельно от CPU. Обмен пользовательскими данными происходит через вызов специальных функций (FC) из программы пользователя.

Интерфейсы S7-300-DP CPU 315-2, CPU 318-2 и CP342-5 могут использоваться в DP-системах и как DP-Master и как DP-Slave. При эксплуатации DP-интерфейса как DP-Slave'а предстоит выбор вида управления доступом к шине: “DP-Slave как активный участник” или ” DP-Slave как пассивный участник”. DP-Slave, который применяется как активный, ведет себя во время обмена данными с DP-Master'ом через DP-протокол как (пассивный) DP-Slave. Как только этот “активный DP-Slave” обладает маркером, можно обмениваться данными с другими участниками через другие коммуникационные службы, такие, как FDL или S7-функции. Нужно заметить, что подобный выбор доступен не для каждого DP-интерфейса. Например, DP-Slave CP342-5 можно объявить как активным, так и пассивным участником. Подобная возможность позволяет одновременно с DP-функциями, эксплуатировать PG, OP и обмен данными S7-CPU друг с другом через DP-интерфейс SIMATIC S7.

Табл.3.1 Технические данные интерфейса PROFIBUS-DP, подключаемого к системе S7-300.

Модуль	CP342-5		CP342-5	
Заказной номер	6GK7 342-5DA00-0XA0 6GK7 342-5DA01-0XA0		6GK7 342-5DA02-0XA0	
Вид работы	DP-Master	DP-Slave	DP-Master	DP-Slave
Скорость передачи, кБит/с	9,6...1500	9,6...1500	9,6...12000	9,6...12000
Мак. число DP-Slave	64	-	64	-
Мак. число модулей	-	32	-	32
Входн. байтов на Slave	Max 240	-	Max 240	-
Выходн. байтов на Slave	Max 240	-	Max 240	-
Входн. байтов как Slave	-	Max 86	-	Max 86
Выходн. байтов как Slave	-	Max 86	-	Max 86
Консистентный блок данных	Max 240 байт	Max 86 байт	Max 240 байт	Max 128 байт
Пользоват. область входов	Max 240 байт	-	Max 240 байт	-
Пользоват. область выходов	Max 240 байт	-	Max 240 байт	-
Мак. параметр. данных на Slave	242 байта	-	242 байта	-
Мак. конфигур. данных на Slave	242 байта	-	242 байта	-
Мак. диагност. данных на Slave	240 байта	-	240 байта	-
Поддержка перекрестной связи	нет	нет	нет	нет
Постоянный по времени цикл	нет	нет	нет	нет
SYNC/FREEZE	да	нет	да	нет
Режим DPV1	нет	нет	нет	нет

Табл. 3.2 Технические данные интерфейсов PROFIBUS-DP, встроенных в систему S7-300

Модуль	CPU3152DP		CPU3152DP		CPU3162DP		CPU318-2DP		
Заказной номер	6ES7315-2AF01 и 2AF02		6ES7315-2AF03-0AB0		6ES7316-2AG00-0AB0		6ES7318-2AF00-0AB0		
Число интерфейсов	2 (1-ый – только MPI)		2 (1-ый – только MPI)		2 (1-ый – только MPI)		2		
							1-ый	2-ой	1-ый и 2-ой
Вид работы	M ¹⁾	S ²⁾	M ¹⁾	S ²⁾	M ¹⁾	S ²⁾	MPI/M ³⁾	M/MPI ⁴⁾	S ²⁾
Скорость передачи, кбит/с	9,6-12000	9,6-12000	9,6-12000	9,6-12000	9,6-12000	9,6-12000	9,6-12000	9,6-12000	9,6-12000
Мак. число DP-Slave'ов	64	-	64	-	64	-	32	125	-
Мак. число модулей	512	32	512	32	512	32	512	1024	32
Входных байтов на Slave ⁵⁾	122	-	244	-	244	-	244	244	-
Выходных байтов на Slave ⁵⁾	122	-	244	-	244	-	244	244	-
Входных байтов как Slave ⁵⁾	-	122	-	244	-	244	-	-	244
Выходных байтов как Slave ⁵⁾	-	122	-	244	-	244	-	-	244
Консистентный блок данных ⁵⁾	32	32	32	32	32	32	128	128	32
Пользоват. область входов	1K	-	1K	-	2K	-	2K	8K	-
Пользоват. область выходов	1K	-	1K	-	2K	-	2K	8K	-
Мак. параметрич. данных на Slave, байт	244		244		244		244	244	
Мак. конфигурац. данных на Slave, байт	244		244		244		244	244	
Мак. диагностич. данных на Slave, байт	240		240		240		240	240	
Поддержка перекрестной связи	нет	нет	да	да	да	да	да	да	да
Постоянный по времени цикл	нет	-	да	-	да	-	да	да	-
SYNC/FREEZE	нет	нет	да	нет	да	нет	да	да	нет
Режим DPV1	нет	нет	нет	нет	нет	нет	с FW3.0	с FW3.0	с FW3.0

¹⁾ DP-Master; ²⁾ DP-Slave; ³⁾ MPI/DP-Master; ⁴⁾ DP-Master/MPI; ⁵⁾ max. значения.

Табл. 3.3 Технические данные интерфейса PROFIBUS-DP, встроенного в системы S7-400.

Модуль	CPU 412-1	CPU 412-2		CPU 413-2	CPU 414-2
Заказной номер	6ES7 412-1XF03-0AB0	6ES7 412-2XG00-0AB0		6ES7 413-2XG0?-0AB0	6ES7 414-2X?00 или 2X?00 -0AB0
Число интерфейсов	1	2		2 (1-й только MPI)	2 (1-й только MPI)
Вид работы	MPI/DP-Master	MPI/DP-Master	DP-Master	DP-Master/MPI	DP-Master
Скорость передачи, кбит/с	9,6...12000	9,6...12000	9,6...12000	9,6...12000	9,6...12000
Max. число DP-Slave'ов	32	32	125	64	96
Входных байт на Slave	Max 244	Max 244	Max 244	Max 122	Max 122
Консистентный блок данных, байт	Max 128	Max 128	Max 128	Max 122	Max 122
Пользовательская область входов, кбайт	2	2	2	2	4
Выходных байт на Slave	Max 244	Max 244	Max 244	Max 122	Max 122
Пользовательская область выходов	2	2	2	2	4
Max. парам. данных на Slave, байт	244	244	244	244	244
Max. конфигурац. данных на Slave, байт	244	244	244	244	244
Max. диагност. данных на Slave, байт	240	240	240	240	240
Поддержка перекрестной связи	да	да	да	нет	нет
Постоянный по времени цикл	да	да	да	нет	нет
SYNC/FREEZE	да	да	да	Только через модули расширения CP/IM	Только через модули расширения CP/IM
Режим DPV1	с FW 3.0	с FW 3.0	с FW 3.0	нет	нет

Табл. 3.3 Продолжение

Модуль	CPU 414-2		CPU 413-3		CPU 416-2
Заказной номер	6ES7 414-2XG03-0AB0		6ES7 414-3XJ00-0AB0		6ES7 416-2X?00 или -2X?01- 0AB0
Число интерфейсов	2		3 (3-й интерфейс IF964-DP устанавливается как DP-Master)		2 (1-й только MPI)
Вид работы	MPI/DP-Master	DP-Master/MPI	MPI/DP-Master	DP-Master/MPI	DP-Master
Скорость передачи, кбит/с	9,6...12000	9,6...12000	9,6...12000	9,6...12000	9,6...12000
Max. число DP-Slave'ов	32	125	32	125	96
Входных байт на Slave	Max 244	Max 244	Max 244	Max 244	Max 122
Консистентный блок данных, байт	Max 128	Max 128	Max 128	Max 128	Max 122
Пользовательская область входов, кбайт	2	6	2	6	8
Выходных байт на Slave	Max 244	Max 244	Max 244	Max 244	Max 122
Пользовательская область выходов, кбайт	2	6	2	6	8
Max. парам. данных на Slave, байт	244	244	244	244	244
Max. конфигурац. данных на Slave, байт	244	244	244	244	244
Max. диагност. данных на Slave, байт	240	240	240	240	240
Поддержка перекрестной связи	да	да	да	да	нет
Постоянный по времени цикл	да	да	да	да	нет
SYNC/FREEZE	да	да	да	да	*)
Режим DPV1	с FW 3.0	с FW 3.0	с FW 3.0	с FW 3.0	с FW 3.0

*) Только через модули расширения CP/IM

Табл. 3.3 Продолжение

Модуль	CPU 416-2		CPU 416-3	
Заказной номер	6ES7 416-2XK02-0AB0		6ES7 416-2XL00-0AB0	
Число интерфейсов	2		3 (3-й интерфейс – подкл. IF964-DP как DP-Master)	
Вид работы	MPI/DP-Master	DP-Master/MPI	MPI/DP-Master	DP-Master/MPI
Скорость передачи, кбит/с	9,6...12000	9,6...12000	9,6...12000	9,6...12000
Макс. число DP-Slave'ов	32	125	32	125
Входных байт на Slave	Max 244	Max 244	Max 244	Max 244
Консистентный блок данных, байт	Max 128	Max 128	Max 128	Max 128
Пользовательская область входов, кбайт	8	2	2	8
Выходных байт на Slave	Max 244	Max 244	Max 244	Max 244
Пользовательская область выходов, кбайт	8	2	2	8
Макс. парам. данных на Slave, байт	244	244	244	244
Макс. конфигурац. данных на Slave, байт	244	244	244	244
Макс. диагност. данных на Slave, байт	240	240	240	240
Поддержка перекрестной связи	да	да	да	да
Постоянный по времени цикл	да	да	да	да
SYNC/FREEZE	да	да	да	да
Режим DPV1	с FW 3.0	с FW 3.0	с FW 3.0	с FW 3.0

Табл. 3.3 Продолжение

Модуль	CPU 417-4		IF964DP	IM467/ IM467FO
Заказной номер	6ES7 417-4XL00-0AB0		6ES7 964-2AA00-0AB0	6ES7 467-S?J00 или S?J01-0AB0
Число интерфейсов	4(3-й и 4-й интерфейсы – подкл. IF964-DP как DP-Master)		1	1
Вид работы	MPI/DP-Master	DP-Master/MPI	В S7-400 CPU – только DP-Master	DP-Master
Скорость передачи, кбит/с	9,6...12000	9,6...12000	9,6...12000	9,6...12000
Макс. число DP-Slave'ов	32	125	Max 125 ¹⁾	125
Входных байт на Slave	Max 244	Max 244	Max 244 ¹⁾	Max 244
Консистентный блок данных, байт	Max 128	Max 128	Max 128 ¹⁾	Max 128
Пользовательская область входов, кбайт	2	8	Зависит от CPU	4
Выходных байт на Slave	Max 244	Max 244	Max 244 ¹⁾	Max 244
Пользовательская область выходов, кбайт	2	8	Зависит от CPU	4
Макс. парам. данных на Slave, байт	Max 244	Max 244	Max 244 ¹⁾	Max 244
Макс. конфигурац. данных на Slave, байт	Max 244	Max 244	Max 244 ¹⁾	Max 244
Макс. диагност. данных на Slave, байт	Max 240	Max 240	Max 240 ¹⁾	Max 240
Поддержка перекрестной связи	да	да	Зависит от CPU	нет
Постоянный по времени цикл	да	да	Зависит от CPU	нет
SYNC/FREEZE	да	да	Зависит от CPU	да
Режим DPV1	с FW 3.0	с FW 3.0	Зависит от CPU	нет

¹⁾ - у S7-400 CPU

Табл. 3.3 Продолжение

Модуль	IM467	CP443-5 EXT	CP443-5 EXT
Заказной номер	6ES7 467-5GJ02-0AB0	6GK7 443-5DX00 или 5DX01-0XE0	6GK7 443-5DX02-0XE0
Число интерфейсов	1	1	1
Вид работы	DP-Master	DP-Master	DP-Master
Скорость передачи, кбит/с	9,6...12000	9,6...12000	9,6...12000
Max. число DP-Slave'ов	125	125	125
Входных байт на Slave	Max 244	Max 244	Max 244
Консистентный блок данных, байт	Max 128	Max 128	Max 128
Пользовательская область входов, кбайт	4	4	4
Выходных байт на Slave	Max 244	Max 244	Max 244
Пользовательская область выходов, кбайт	4	4	4
Max. парам. данных на Slave, байт	Max 244	Max 244	Max 244
Max. конфигурац. данных на Slave, байт	Max 244	Max 244	Max 244
Max. диагност. данных на Slave, байт	Max 240	Max 240	Max 240
Поддержка перекрестной связи	да	нет	да
Постоянный по времени цикл	да	нет	да
SYNC/FREEZE	да	да	да
Режим DPV1	нет	нет	с 6GK7 443-5DX03-0XE0

3.2 Расширенные коммуникационные возможности

Активный DP-интерфейс (DP-Master и активный DP-Slave) SIMATIC S7-300 и S7-400 поддерживает одновременно с DP-функциями следующие коммуникационные возможности:

- S7-функции через встроенный и подключаемый DP-интерфейсы и
- PROFIBUS-FDL – службу (SEND/RECEIVE) только через CP.

3.2.1 S7-функции

S7-функции представляют коммуникационные службы между S7-CPU (друг с другом) и SIMATIC-HMI – системами (Human Machine Interface). S7-функции – составная часть всех SIMATIC S7 – приборов. S7-функции реализуют в частности:

- Сложную on-line функциональность STEP 7 для программирования, тестов, ввода в эксплуатацию и диагностики SIMATIC S7-приборов автоматизации (S7-300/400)
- Передачу данных и областей данных между отдельными SIMATIC S7 – станциями максимум до 64 кбайт.
- Запись и чтение данных без дополнительных коммуникационных пользовательских программ у партнера по коммуникации между SIMATIC S7 – станциями.

- Запуск управляющих функций, таких как остановка, новый запуск, повторный запуск CPU-партнера по коммуникации.
- Функции наблюдения для партнера по коммуникациям, например, актуальное рабочее состояние CPU-партнера.

3.2.2 FDL-служба (SEND/RECEIVE)

Коммуникации через уровень 2 (Layer 2) делают возможными постоянные FDL-посылки и прием блоков данных длиной до 240 байт. Обмен данными, базирующийся на SDA-телеграммах (Send Data with Acknowledge), используется как при коммуникациях внутри систем автоматизации SIMATIC S7, так и при обмене данными между SIMATIC S7 и S5 – системами, а также с PC.

FDL-служба осуществляется в SIMATIC S7 через вызов функций (AG_SEND и AG_RECV) внутри пользовательской программы.

3.3 Системные свойства DP-интерфейса в SIMATIC S7

Интерфейс DP-Master в системах SIMATIC S7, за исключением CP342-5, используется так, как это описано в разделах 3.3.1 – 3.3.8.

3.3.1 Свойства запуска интерфейса DP-Master в SIMATIC S7

При децентрализованной структуре установки часто по техническим и топологическим причинам не возможно подключить все электрические машины или части установки. На практике это при известных обстоятельствах означает, что при запуске DP-Master'a еще не все запроецированные DP-Slave'ы имеются в наличии. Перед началом циклической работы после включения питания во время фазы запуска (Startup) DP-Master должен параметризовать и конфигурировать все назначенные ему Slave'ы. Для систем S7-300 и S7-400 можно установить максимальное время ожидания сообщения о готовности всех DP-Slave'ов с помощью параметра “*“Finished” Message by Means of Modules*”. Область устанавливаемых значений лежит между 1ms и 65000 ms. По умолчанию установлено значение 65000 ms. По истечению этого времени CPU переходит в состояние STOP или RUN в зависимости от установки параметра “*Startup at Present Configuration Not Equal to Actual Configuration*”(Запуск при несовпадении заданной и действительной конфигураций).

3.3.2 Выход из строя станции DP-Slave

Выход из строя DP-Slave'ов, например, из-за сбоя источника питания, обрыва шины или ее дефекта, сообщается операционной системой CPU через вызов организационного блока OB86 (выход из строя носителя модулей, DP-сети или DP-Slave). OB86 вызывается операционной системой, как при приходящем, так и при уходящем событии. Если OB86 не запрограммирован, то при выходе из строя DP-сети или DP-Slave'a CPU переходит в состояние STOP. Таким образом, при выходе из строя децентрализованной периферии система SIMATIC S7 ведет себя так же, как при выходе из строя центральной периферии.

3.3.3 Сигнал (Alarm) при удалении/вставке модуля

Модули, расположенные в центральной корзине и спроектированные в системе SIMATIC S7, периодически опрашиваются центральным процессором и при их удалении или вставке CPU получает соответствующий сигнал.

SIMATIC DPS7-Slave'ы и DPV1-Slave'ы также могут отслеживать эти события и при их наступлении сообщать о них DP-Master'у. Благодаря этому в CPU стартует блок OB83, при этом он будет при удалении стартовать как приходящее событие, а при вставке – как уходящее. При вставке модуля в спроектированный слот в состоянии работы (RUN) операционная система проверяет, соответствует ли тип вставленного модуля спроектированному типу. Затем стартует OB83 и при совпадении типов спроектированного и вставленного модулей происходит параметрирование (назначение параметров) вставленного модуля. Если OB83 отсутствует в CPU, то при появлении сигнала удаления/вставки центральный процессор переходит в состояние STOP.

3.3.4 Диагностические сигналы от станций DP-Slave

Модули с диагностическими способностями в области децентрализованной периферии в состоянии сообщать о событиях с помощью диагностических прерываний как, например, частичный выход из строя станции, обрыв провода у сигнального модуля, короткое замыкание/перегрузка периферийного канала или выход из строя источника питания. При приходящих или уходящих диагностических сигналах операционная система CPU вызывает организационный блок OB82 для обработки диагностических сигналов. Если OB82 не запрограммирован, CPU переходит в состояние STOP. Возможные диагностические события и структуры их сообщений в зависимости от сложности DP-Slave'ов частично описаны в EN 50170. Внутри DP-Slave SIMATIC S7 возможные диагностические события согласованы с системной диагностикой SIMATIC S7.

3.3.5 Сигналы от процесса у станций DP-Slave

DP-Slave SIMATIC S7, способные генерировать сигналы от процесса, могут сигнализировать через шину CPU DP-Master'у о событиях в процессе, как, например, выход за пределы верхней или нижней границы значения аналогового сигнала. Для обработки сигналов от процесса в системе SIMATIC S7 зарезервированы организационные блоки OB40 ... OB47, которые вызываются операционной системой в случае поступления сигнала. Таким образом, обработка возбуждаемых сигналов от процесса в системе SIMATIC S7 идентично, как у децентрализованной, так и у центральной периферии. Следует, однако, принять во внимание, что время реакции на сигналы от процесса, возбуждаемые децентрализованной периферией по сравнению с сигналами от процесса, возбуждаемыми центральной периферией, обусловлено временем прохождения телеграммы по шине и последующей обработкой сигнала в DP-Master'е и поэтому больше.

3.3.6 Сигнал состояния (Statusalarm) от DP-Slave'a

DPV1-Slave'ы могут вызывать сигнал состояния. Если модуль DPV1-Slave меняет свое рабочее состояние, например RUN на STOP, то об этой смене состояния может быть сообщено DP-Master'у с помощью сигнала состояния. Точное событие, которое вызывает сигнал состояния, устанавливается изготовителем и может быть взято из документации DPV1-Slave'a.

Благодаря сигналу состояния операционная система CPU вызывает организационный блок OB55. Если этот блок не запрограммирован, то CPU не смотря на это остается в состоянии RUN. OB55 имеется только у S7-CPU, поддерживающих DPV1.

3.3.7 Сигнал модернизации (Update-alarm) от DP-Slave'a

DPV1-Slave может сигнализировать DP-Master'у о изменении параметров модулей с помощью сигнала модернизации. Благодаря этому сигналу в CPU вызывается OB56. OB56 имеется только у S7-CPU, поддерживающих DPV1. Если этот блок не запрограммирован, то CPU при появлении сигнала модернизации не смотря на это остается в состоянии RUN. Какое событие в DPV1-Slave'e сообщает о себе сигналом модернизации, определяется изготовителем. Точная информация может быть взята из описания DPV1-Slave'a.

3.3.8 Специфический для производителя сигнал от DP-Slave'a

Специфический для производителя сигнал может поступать только от слота DPV1-Slave'a к DP-Master'у. Благодаря ему вызывается организационный блок OB57. OB57 имеется только у S7-CPU, поддерживающих DPV1. Если OB57 не запрограммирован, CPU не смотря на это, остается в состоянии RUN. Определение, когда DPV1-Slave вызывает специфический для производителя сигнал, зависит от Slave'a, соответственно, у интеллектуального Slave'a от его приложения (выполняемой программы пользователя) и в общем определяется производителем. Информацию о том, может ли DPV1-Slave и, если может, то когда, посылать специфический для производителя сигнал, можно взять из документации на Slave.

3.4 Варианты DP-Slave'ов в системе SIMATIC S7

DP-Slave'ы, в системе SIMATIC S7 по структуре и функциям подразделяются на 3 группы:

- *Компактные DP-Slave'ы*
- *Модульные DP-Slave'ы*
- *Интеллектуальные DP-Slave'ы (I-Slave'ы)*

3.4.1 Компактные DP-Slave'ы

Компактные DP-Slave'ы обладают не изменяемой структурой периферии в области входов и выходов. Такие DP-Slave'ы представлены в ряду цифровых станций ET 200 В (В – для блочной периферии). В зависимости от числа необходимых периферийных каналов и диапазонов напряжения, можно выбрать из спектра блоков ET 200 В подходящий модуль.

3.4.2 Модульные DP-Slave'ы

У DP-Slave'ов, построенных по модульному принципу, структура применяемых областей входов и выходов переменная и устанавливается при проектировании. Типичные представители этого типа DP-Slave'ов – станции ET 200 М.

К головному (интерфейсному) модулю ET 200 М (IM 153) можно подключать до 8-и периферийных модулей из спектра модулей S7-300.

3.4.3 Интеллектуальные DP-Slave'ы (I-Slave'ы)

Системы автоматизации S7-300 могут использоваться через CPU 315-2, CPU 316-2, CPU 318-2 или CP 342-5 как DP-Slave'ы. Такие полевые приборы используются для предварительной обработки сигналов и обозначаются в системе SIMATIC S7 как “Intelligente DP-Slaves (интеллектуальные DP-Slave'ы)”, коротко I-Slave'ы. Структура применяемых входных и выходных областей для S7-300 как DP-Slave'а определяется при проектировании с помощью утилиты HW-Config.

Отличительная черта I-Slave это то, что DP-Master'у предоставляются в распоряжение области входов/выходов не реально имеющейся периферии, а области входов/выходов, которые отображаются в CPU (области отображения входов и выходов, не занятые периферией).

4. Программирование и проектирование PROFIBUS-DP с помощью STEP 7

Введение

Программный пакет STEP 7 – базовое программное обеспечение для программирования и проектирования систем SIMATIC S7. Эта глава описывает основные средства базового пакета STEP 7, применяемые к PROFIBUS-DP.

Предпосылки: установленный на PG или PC программный пакет STEP 7 и соответствующие знания работы под Win95 или WinNT.

Базовый пакет STEP 7 состоит из различных приложений (рис. 4.1).

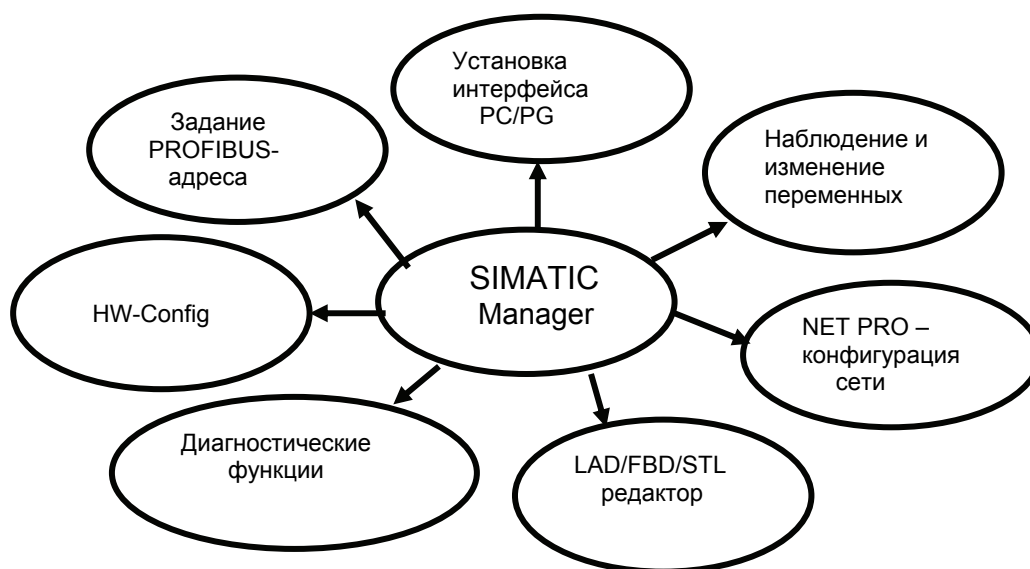


Рис.4.1 Важные для PROFIBUS-DP STEP7-приложения, вызываемые из SIMATIC Manager

Они применяются для:

- Конфигурирования и параметрирования аппаратуры
- Конфигурирования сетей и соединений
- Загрузки и тестирования пользовательской программы

Благодаря ряду дополнительных пакетов, например, таких, как языки программирования SCL, S7GRAPH или HiGraph, базовый пакет STEP 7 расширяется для соответствующих приложений. С помощью центрального инструмента SIMATIC Manager все необходимые приложения могут быть вызваны в графическом виде. Все данные и установки для системы автоматизации структурированы внутри проекта и представлены в виде объектов. Пакет STEP 7 снабжен обширной online-помощью вплоть до контекстной помощи для указанного контейнера, объекта и появляющегося сообщения об ошибке.

4.1 Основы STEP 7

4.1.1 Объекты STEP 7

Рис. 4.2 показывает, из каких контейнеров и объектов состоит проект STEP 7. Внешний вид окна похож на Windows Explorer. Под контейнером здесь понимается объект, который в свою очередь может содержать контейнеры и/или объекты.

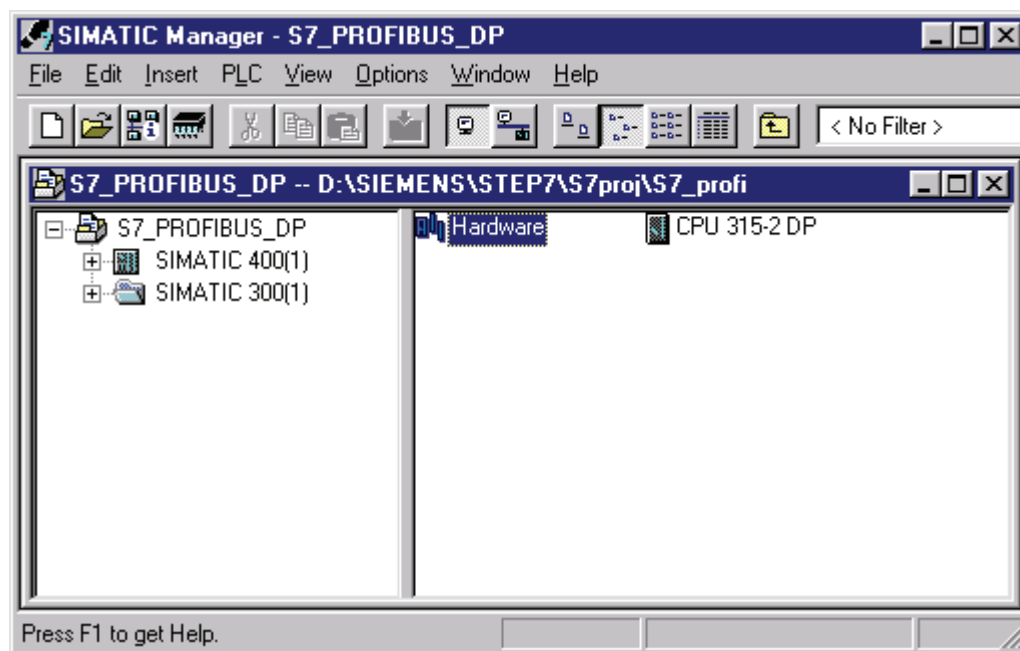


Рис 4.2 Пример контейнера и структуры объектов STEP 7

Объектно-ориентированная работа со STEP 7

При обработке объектов различных типов в SIMATIC Manager автоматически вызывается соответствующее приложение. Эта зависящая от типа связь объекта с соответствующим приложением делает возможным очень простой и сквозной образ действий при обработке проекта STEP 7. Все приложения, связанные с типом объекта, можно запустить или через вызов обрабатываемого объекта, или через контекстное меню “*Open*” (*Открыть*) (подвести указатель на объект в SIMATIC Manager и нажать правую клавишу мыши).

4.1.2 Проект STEP 7

Главный объект в SIMATIC Manager – проект. Внутри проекта все данные и программы, которые нужны для выполнения задачи автоматизации, представлены в виде древовидной структуры (проектной иерархии) (рис. 4.3). Внутри проекта объединяется следующая информация:

- Конфигурационные данные для структуры аппаратной части
- Параметры для установленных модулей
- Проектные данные для сетей и коммуникаций
- Программы для программируемых модулей

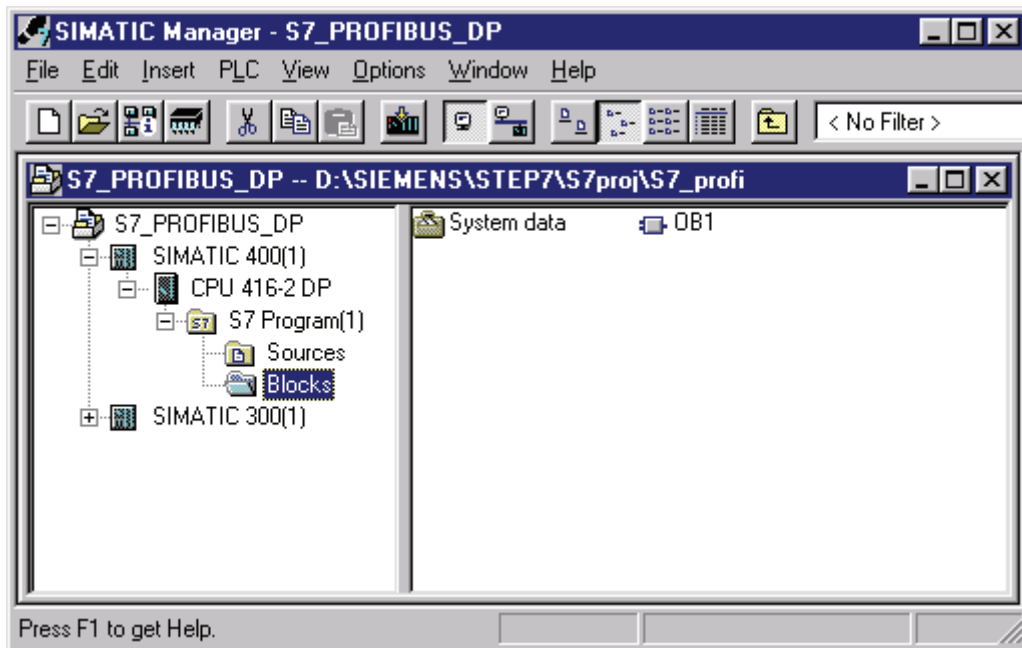


Рис. 4.3 Иерархия объектов в проекте STEP 7

4.2 Пример проекта с PROFIBUS-DP

В этой главе с помощью примера-проекта обсуждаются утилиты STEP 7, важные для связи с PROFIBUS-DP. В основном это *STEP 7 Manager* и *HW-Config*. Представляемый здесь способ действий при работе с проектом SIMATIC S7 должен помочь Вам просто и быстро начать работать с проектом.

В описанном ниже примере-проекте применяется станция S7-400 с CPU 416-2DP. Через встроенный DP-интерфейс подключены DP-Slave'ы – ET200B-16DI/16DO, ET200M и S7-300/CPU 315-2. Скорость передачи выбрана 1500 кбит/с.

4.2.1 Создание проекта STEP 7

При создании нового проекта внутри SIMATIC-Manager Вы можете поступать следующим образом:

- Через *File->New* (*Файл->Новый*) Вы получаете окно для создания нового проекта
- Через кнопку выбора “*New Project*” (“*Новый проект*”) Вы выбираете установку для создания нового проекта
- Вы задаете желаемое имя для нашего проекта, например, S7-PROFIBUS-DP и покидаете маску через ОК.

После ухода из окна с помощью ОК Вы попадаете опять в главное меню SIMATIC Manager. В проекте автоматически создается объект MPI (Multi Point Interface), который виден в правом окне проекта. Объект MPI создается в STEP7

автоматически и представляет стандартный интерфейс для PG и коммуникационный интерфейс для CPU.

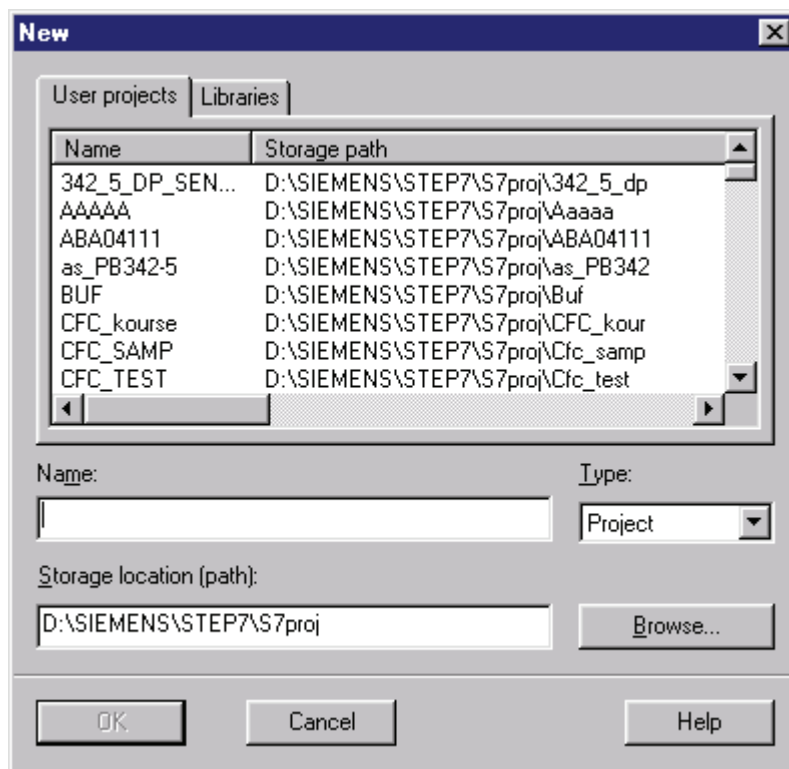


Рис 4.5 Окно для создания нового проекта

4.2.2 Вставка объекта в проект STEP 7

Выделите (маркируйте) проект и откройте с помощью правой клавиши мыши контекстное меню. Выберите через *Insert new object* (*Вставить новый объект*) SIMATIC 400 Station. Вновь вставленный объект появляется в правой половине проектного окна. Вы можете здесь, так же, как у всех других объектов, присвоить специфическое имя объекту. Затем Вы можете установить дальнейшие данные для каждого объекта через *Object properties* (*Свойства объекта*). Вставьте объект PROFIBUS в проекте, созданном в разделе 4.2.1.

4.2.3 Установки сети PROFIBUS

С помощью контекстного меню *“Open Object”* (*“Открыть объект”*) открывается утилита для проектирования NetPro. В ней Вы при выбранной подсети входите с помощью контекстного меню *“Object properties”* (*Свойства объекта*) в изображенную на рис.4.5 закладку *“Network Settings”* (*“Установки сети”*). Внутри этой закладки Вы можете сделать все основные сетевые установки для подсети PROFIBUS. Для проекта-примера подтвердите установки, предложенные по умолчанию с помощью кнопки OK.

На закладке *“Network Settings”* окна *“Properties PROFIBUS”* (*“Свойства PROFIBUS”*) можно установить следующие параметры:

Highest PROFIBUS Address (наивысший адрес PROFIBUS)

Этот параметр, обозначаемый в EN 50170 как HSA (Highest Station Address),

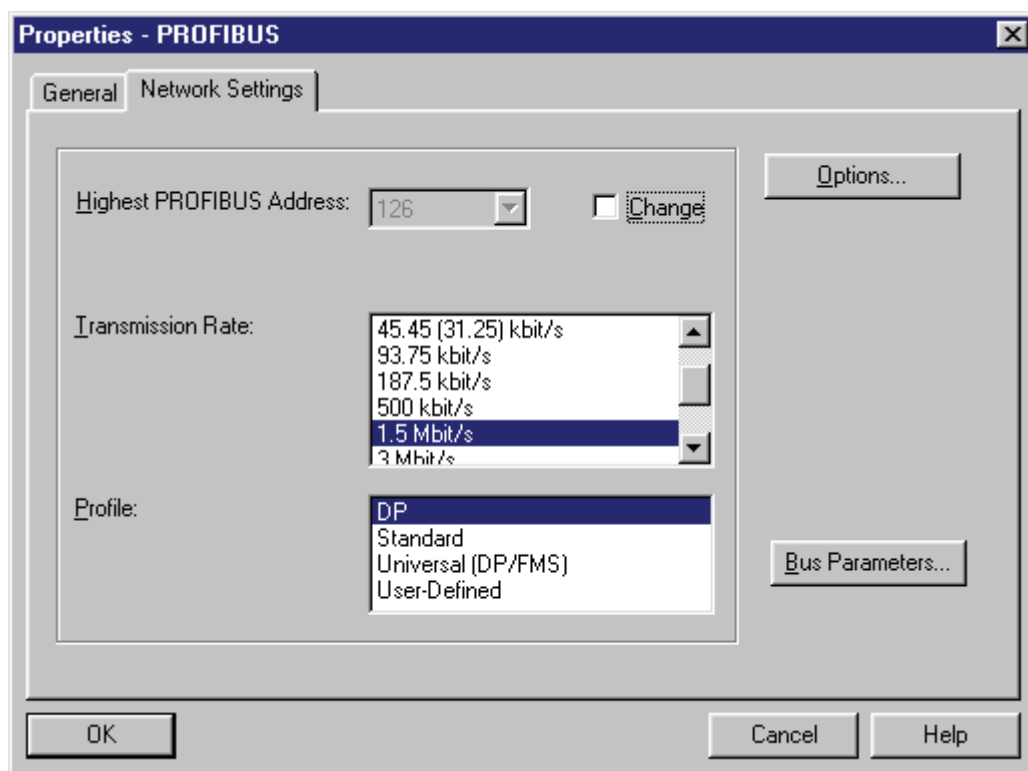


Рис. 4.5 Сетевые установки для PROFIBUS

служит для оптимизации управления доступом к сети в шинных конфигурациях с несколькими Master'ами. В шинной конфигурации PROFIBUS-DP с одним Master'ом установка по умолчанию не должна изменяться (HSA=126).

Transmission Rate (скорость передачи)

Устанавливаемая здесь скорость передачи имеет силу для всей подсети PROFIBUS. Это означает, что все участники, которые функционируют на этой PROFIBUS-подсети, должны поддерживать выбранную скорость. Может быть выбрана скорость передачи от 9,6 до 12000 кбит/с. В качестве установки по умолчанию предлагается величина 1500 кбит/с.

Profile (Профиль)

Профиль шины представляет собой стандартный набор шинных параметров для различных применений PROFIBUS. После установки профиля шины становятся доступными и могут быть изменены параметры шины PROFIBUS, которые зависят от конфигурации в STEP 7, профиля и скорости передачи. Шинный параметр справедлив для всех участников, подключенных к подсети PROFIBUS.

Для специальных применений существует возможность выбрать профиль **“User-Defined”** и после этого изменить все параметры шины. Однако пользоваться этой возможностью рекомендуется только специалисту. Устанавливаемый шинный профиль зависит от конфигурации PROFIBUS. При этом существуют следующие возможности:

Профиль “DP”

Этот профиль выбирается исключительно для “чистых” конфигураций SIMATIC S7 и SIMATIC M7 PROFIBUS, как Mono-Master, так и Multi-Master. Оптимально вычисленные для этого профиля шинные параметры учитывают также коммуникационную нагрузку для возможных позже коммуникационных связей, таких, как подключение PG на шину, B&B-службы (Bedienung&Beobachtung – нем.- обслуживание и наблюдения), а также ациклических FDL, FMS и S7-соединений.

При этом шинном профиле принимаются во внимание только те участники PROFIBUS, которые известны внутри соответствующей подсети PROFIBUS (проекта STEP 7), то есть которые запрограммированы.

Профиль “Standard”

При этом профиле существует возможность с помощью кнопки **“Options...”** (см. рис.4.5) учесть шинные параметры других участников, спроектированных не с помощью STEP 7 или не принадлежащих актуальному проекту STEP 7.

Пока не выбрана опция *“Include network configuration below”* (Принять во внимание следующую сетевую конфигурацию) (см. рис.4.9), параметры будут вычисляться по такому же, как в профиле “DP” оптимальному алгоритму. При выбранной опции параметры вычисляются по упрощенному алгоритму.

Таким образом, профиль “Standard” применяется для всех шинных конфигураций, охватываемых проектом STEP 7, а также специально для всех других многомастерных шинных конфигураций SIMATIC S7 и SIMATIC M7 (DP/FMS/FDL).

Профиль “Universal (DP/FMS)”

Этот профиль предназначен для установки параметров компонентов PROFIBUS из семейства SIMATIC S5, как, например, CP5431 или AG95U и должен выбираться тогда, когда должны одновременно использоваться в подсети PROFIBUS участники SIMATIC S5 и SIMATIC S7.

Bus Parameters (шинные параметры)

При нажатии на кнопку *“Bus Parameters...”* (*“Шинные параметры”*) (см. рис. 4.5) показываются шинные параметры, вычисленные STEP 7. При помощи известной из проекта шинной конфигурации и числа участников STEP 7 вычисляет значение для шинного параметра **“Ttr”** (Time target rotation) – заданного времени обращения маркера (токена) и параметр **“Response**

monitoring” (контроль ответа), который важен для PROFIBUS DP-Slave’ов (см. ниже).

Шинный параметр “Ttr”, вычисленный STEP 7 и представленный здесь, определяет не реальное время обращения маркера, а допустимое максимальное значение этого времени и, таким образом, не может использоваться для оценки времени реакции шинной системы.

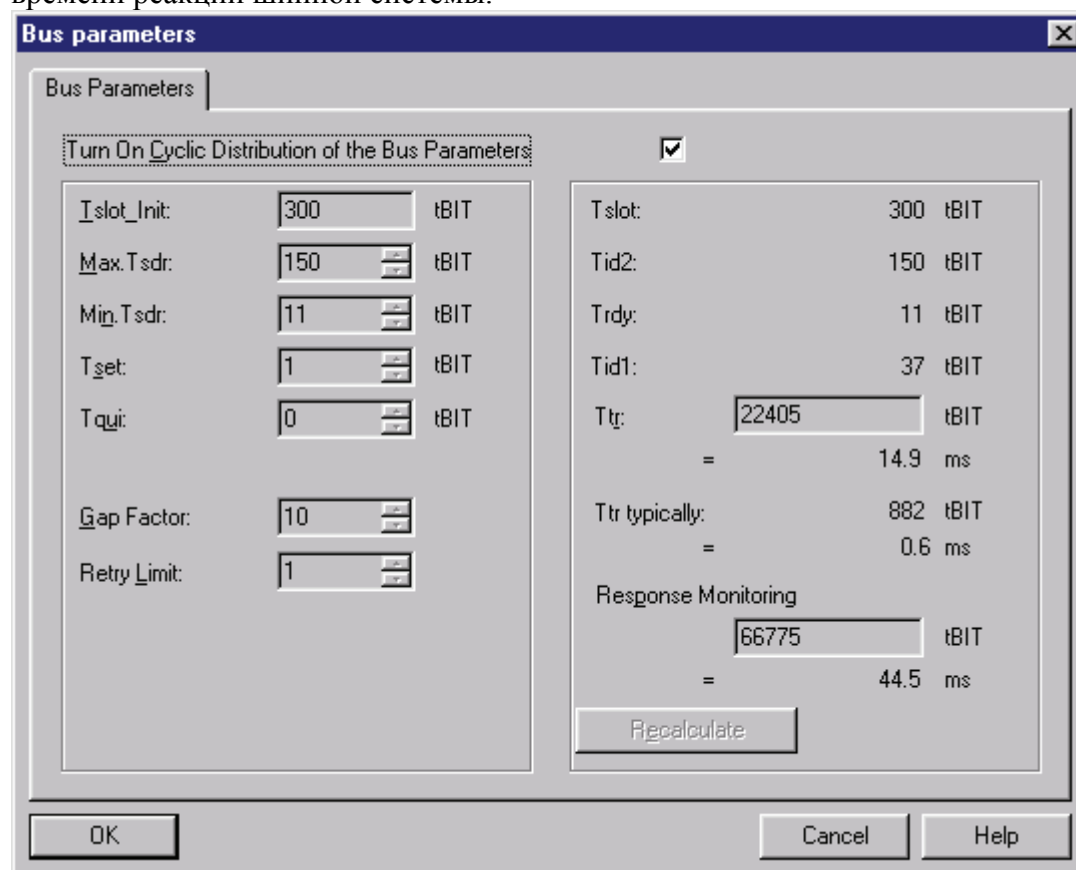


Рис. 4.6 Установки для шинных параметров

Показанные на рис.4.6 значения могут изменяться только при выбранном профиле протокола “User-Defined”. Это рекомендуется делать только экспертам, так как надежное выполнение функций PROFIBUS будет только в том случае, если эти параметры установлены правильно.

Все значения для шинных параметров даются в единицах tBIT (time Bit/время передачи бита)

В табл. 4.1 представлены времена передачи бита в зависимости от скорости передачи, они вычисляются по следующей формуле:

$$tBIT [\mu s] = 1 / \text{Mbit/s}$$

Табл.4.1 Время передачи бита в зависимости от скорости

Скорость передачи, кбит/с	tBIT [μs]
9,6	104,167
19,2	52,083
45,45	22,002
93,75	10,667
187,5	5,333
500	2,000
1.500	0,667
3.000	0,333
6.000	0,167
12.000	0,083

Опция *“Turn On Cyclic Distribution of the Bus Parameters”* (*“Циклически устанавливать шинные параметры”*) означает следующее:

при ее активизации установленные шинные параметры соответствующей подсети PROFIBUS циклически посылаются в определенные временные интервалы ко всем участникам данной сети. Передача данных происходит при этом через службу уровня 2 SDN (Send Data with No Acknowledge) с DSAP (Destination Service Access Point) 63 как Multicast-телеграмма.

Эта функция позволяет очень просто и комфортно подключать устройства программирования к сети PROFIBUS даже в том случае, если параметры сети не известны пользователю (см. также раздел 7.2 *“Установка online-интерфейса PG/PC”*).

Эта функция не должна быть активизирована, если выбран эквидистантный режим работы или на сети находятся другие участники, использующие DSAP 63 для Multicast-функций.

Options...Constant Bus Cycle Time (эквидистантность)

Эта опция поддерживается не каждым DP-Master’ом. Например, CPU 417-4 v1.0 имеет ее. Она служит для установки эквидистантного способа работы сети PROFIBUS.

Опция может быть вызвана с помощью кнопки *“Options...”* (см. рис.4.5) и выбора в открывшемся окне *“Options”* закладки *“Constant Bus Cycle Time”* (*“Постоянный шинный цикл”*) (эта закладка как раз и существует только у тех Master’ов, которые поддерживают эту опцию) (см. рис.4.7).

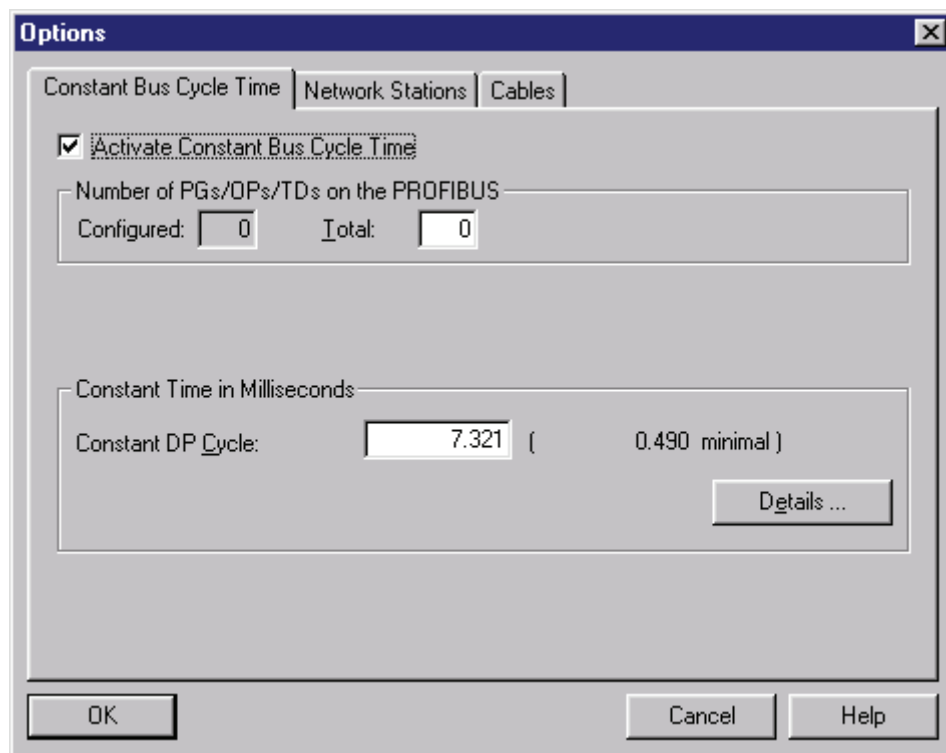


Рис. 4.7 Установки значений времени для постоянного по времени DP-цикла (основные значения)

Благодаря установке контрольного бокса *“Activate Constant Bus Cycle Time”* (*“Установить постоянный по времени цикл шины”*), Вы устанавливаете для подсети PROFIBUS постоянный цикл шины (см. раздел 2.3.3 *“Эквидистантный PROFIBUS-цикл”*). Это означает, что временные интервалы следующих друг за другом разрешений на посылки для DP-Master’a постоянны.

Пока проектирование эквидистантного шинного цикла возможно только в системе с одним мастером (класса 1). DP-Master’a (класса 1) - DP-Master’a, которым поставлены в соответствие DP-Slave’ы для циклического обмена входными/выходными данными.

Для спроектированной соответствующим образом с помощью STEP 7 конфигурации установки вычисленного и предустановленного значения времени для эквидистантного DP-цикла достаточно, чтобы развить внутри предоставленного времени циклический обмен пользовательскими данными с DP-Slave’ами и ациклический обмен данными для функционирования PG, OP и TD. В представленной на рис.4.7 закладке Вы можете через параметр *“Number of PGs/Ops/TDs on the PROFIBUS”* (*“Число PG/OP/TD на PROFIBUS”*) создать резерв для подключаемых позднее PG, OP и TD. Предлагаемое STEP 7 на этой закладке эквидистантное время может быть изменено. Причем увеличение предлагаемого времени не проблематично. При уменьшении до указанного в скобках минимального значения времени эквидистантного цикла, Вы должны понимать, что могут произойти ошибки, например, при выходе из строя DP-

Slave может произойти превышение установленного времени эквидистантного цикла за счет повторных посылок запросов к этому Slave. Далее, при такой установке другие активные участники, например, PG, сокращают до минимума предоставленное им время для ациклического обмена данными. Это может в неблагоприятных случаях привести к задержкам или отмене ациклических коммуникаций.

Кнопкой “Details...” (см. рис.4.7) Вы вызываете представленную на рис.4.8 маску для подробных установок.

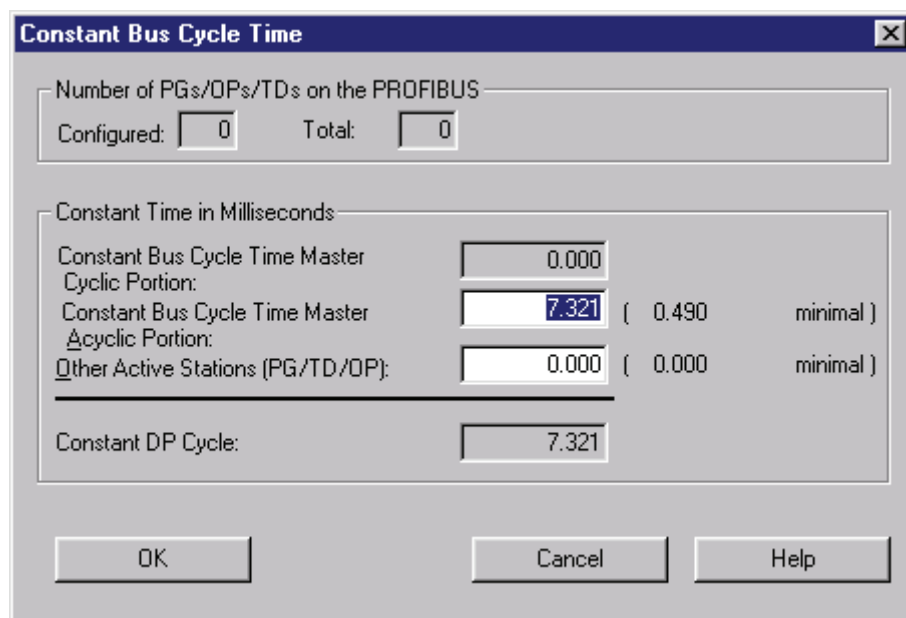


Рис. 4.8 Установки значений времени для постоянного по времени DP-цикла (детальные значения)

Здесь показаны отдельные интервалы времени, из которых состоит представленное эквидистантное время. Показанное время для циклической части постоянно и не может быть изменено. Однако здесь имеется возможность изменять ациклическую временную часть и часть, предоставляемую в распоряжение PG, OP и TD.

Options...Network Stations

Иногда невозможно зарегистрировать всех участников установки PROFIBUS в проекте STEP 7. Существует возможность учитывать остальных активных и пассивных участников на закладке, показанной на рис.4.9. Эта опция не возможна при выбранном профиле DP.

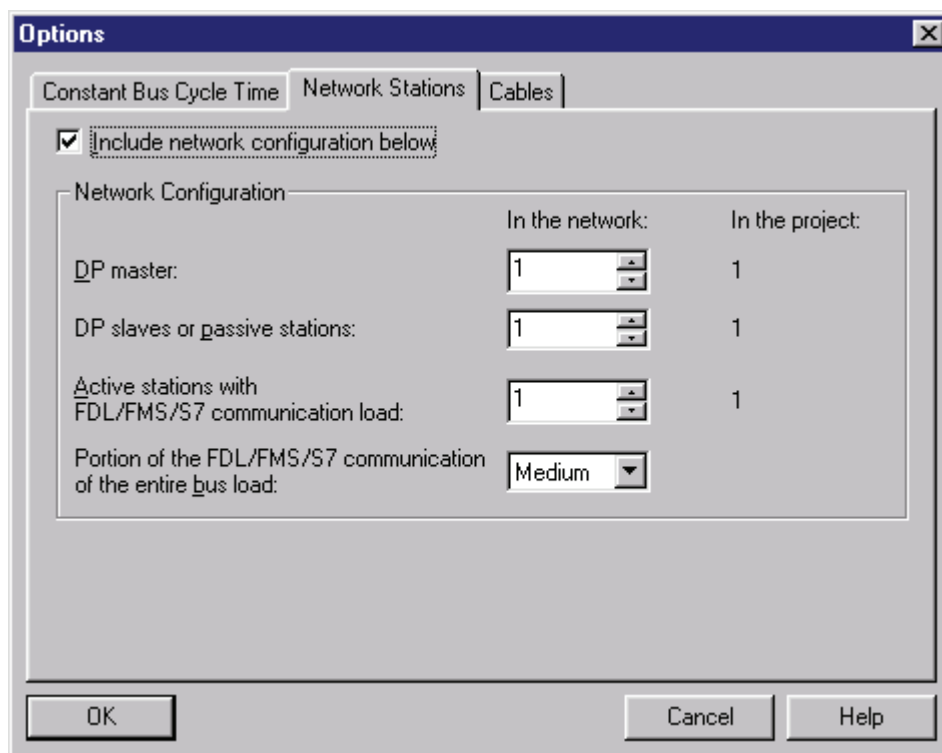


Рис. 4.9 Регистрация других участников для подсети PROFIBUS

Options...Cables

На вычисление параметров шины вместе с длиной проводов имеет влияние также использование RS-485 – повторителей или оптических соединительных модулей (OLM – Optical Link Module) при использовании световодной техники. С помощью закладки “Cables” (“Кабели”) Вы получаете маску для задания соответствующих величин (рис. 4.10).

4.2.4 Проектирование аппаратной конфигурации с помощью HW-Config

На следующем шаге проектирования примера (см. раздел 4.2.1) создается аппаратная часть системы автоматизации S7-400. Для этого Вы открываете в левой половине проекта контейнер S7-PROFIBUS-DP (название проекта). Затем пометьте объект SIMATIC 400(1) и запустите приложение HW-Config через контекстное меню Open Object или с помощью двойного “клика” на объекте Hardware в правой половине окна проекта. Вначале возникает разделенное пополам пустое окно станции для конфигурирования аппаратуры у станции SIMATIC S7.

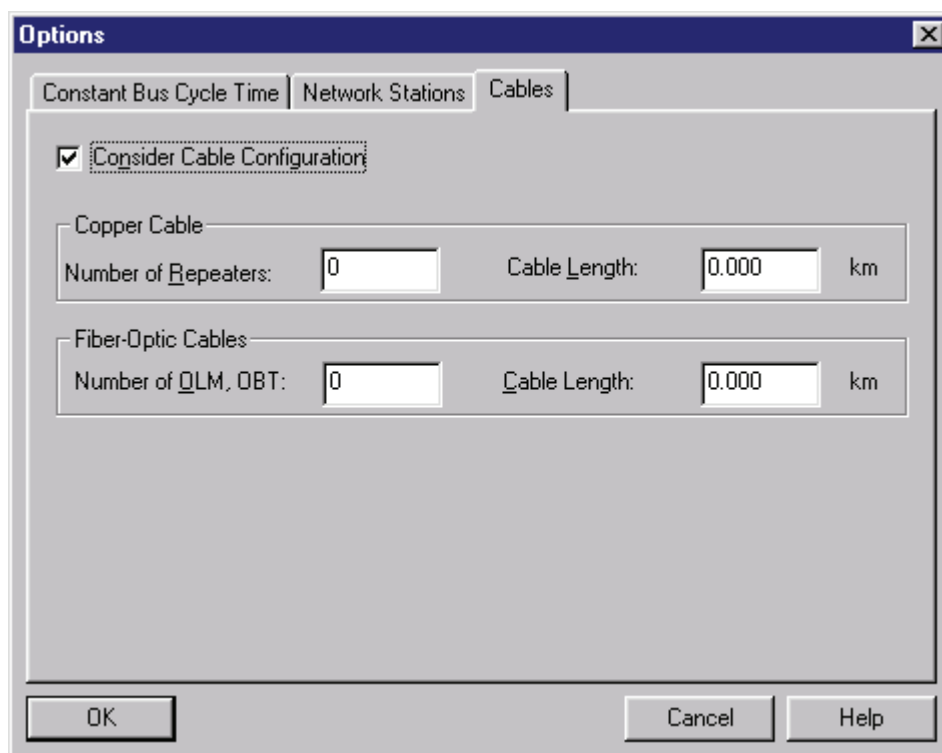


Рис 4.10 Маска “Cables” для учета повторителей, OLM и длины проводов

Проектирование носителя модулей

В HW-Config откройте Catalog (View->Catalog или соответствующая кнопка в панели инструментов). В каталоге откройте раздел SIMATIC 400, а в нем – раздел RACK-400. Выберите носитель модулей согласно заказному номеру и установите его в станцию методом Drag&Drop (перетащите его в верхнее или нижнее окно утилиты).

В появившемся носителе модулей на первое место (в первый слот) установите методом Drag&Drop источник питания. Источники питания находятся в разделе PS-400, выберите источник питания согласно заказному номеру.

Далее в третий слот установите CPU из раздела CPU-400. В нашем примере выберем CPU 416-2DP с заказным номером “6ES7 416-2XK00-0AB0”. Так как выбранный CPU имеет встроенный интерфейс DP, то Вам будет предложено назначить ему подсеть PROFIBUS и назначить адрес PROFIBUS (см. рис. 4.11). Сделайте это и перейдите в главное окно HW-Config с помощью кнопки OK.

4.2.5 Проектирование DP-Slave’ов

Рис.4.12 показывает окно станции HW-Config для спроектированной нами станции.

Спроектируем теперь DP-Slave’ы.

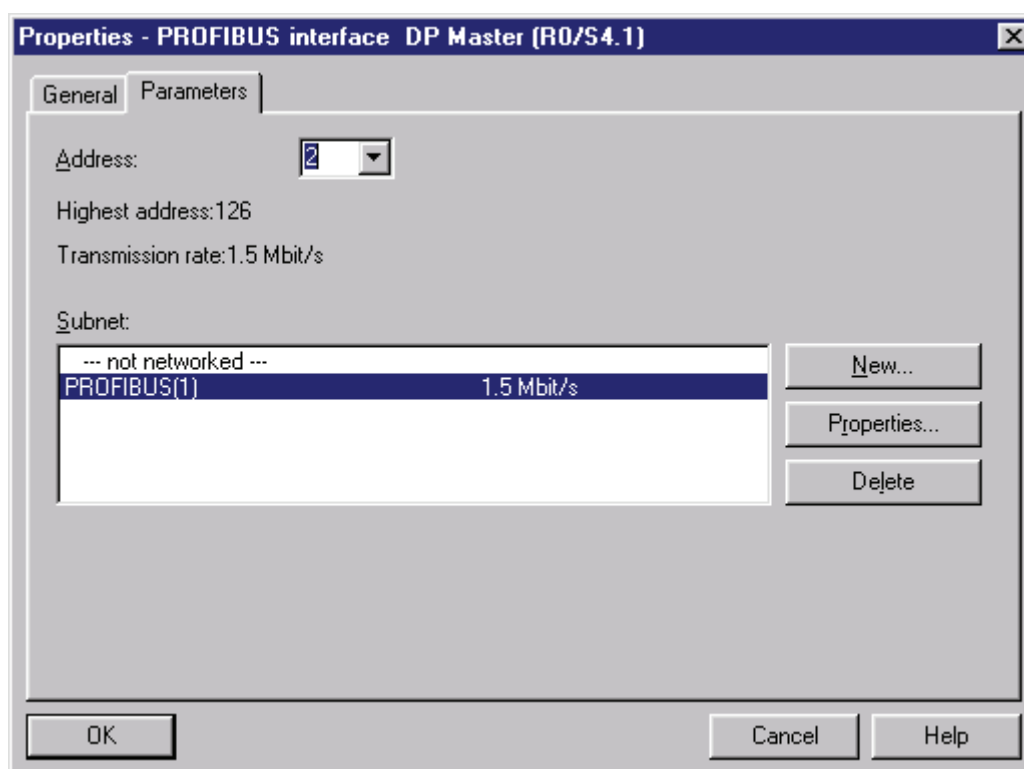


Рис. 4.11 Назначение адреса CPU

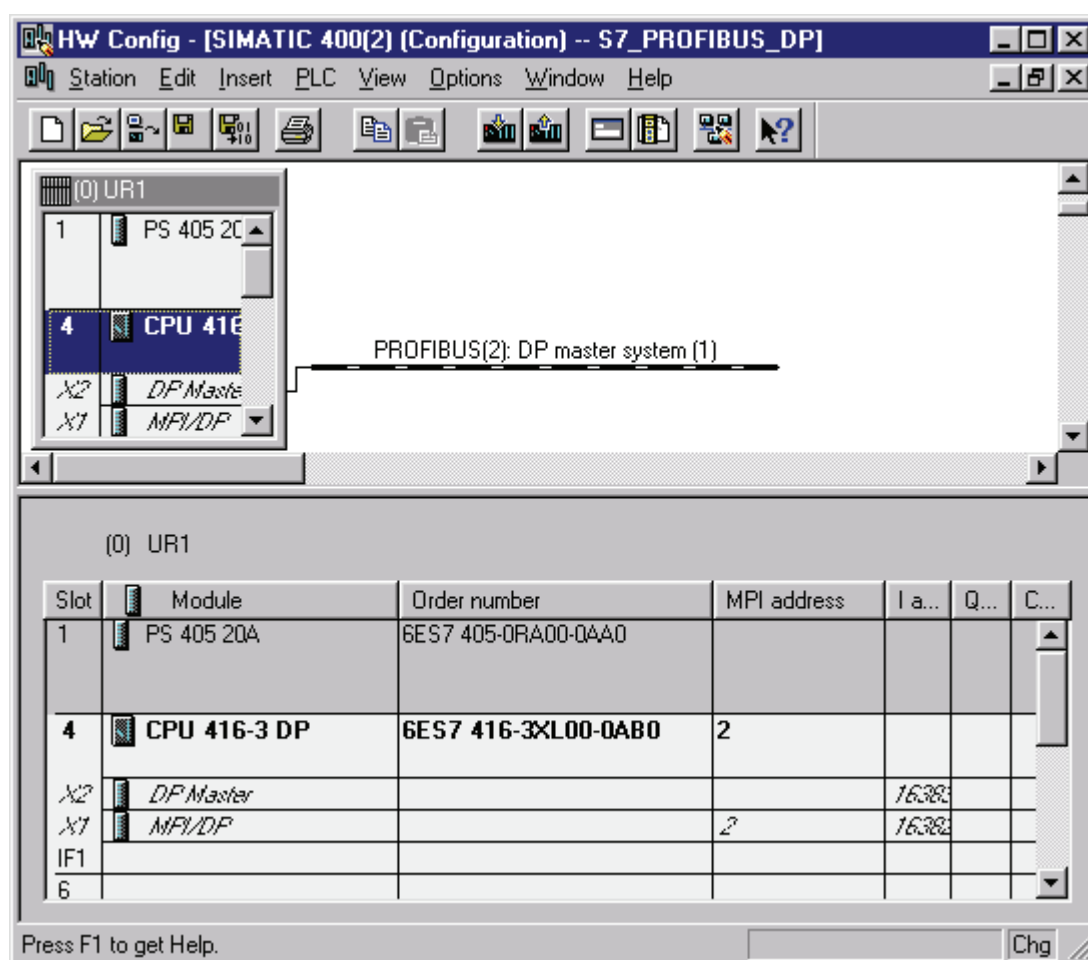


Рис. 4.12 Окно станции в HW-Config для системы DP-Master

Станция ET 200B

В разделе каталога PROFIBUS DP найдите раздел ET 200B, а в нем – соответствующий модуль, например, 16DI/16DO. Перетяните его на изображение подсети. При этом Вам будет предложено задать PROFIBUS-адрес станции.

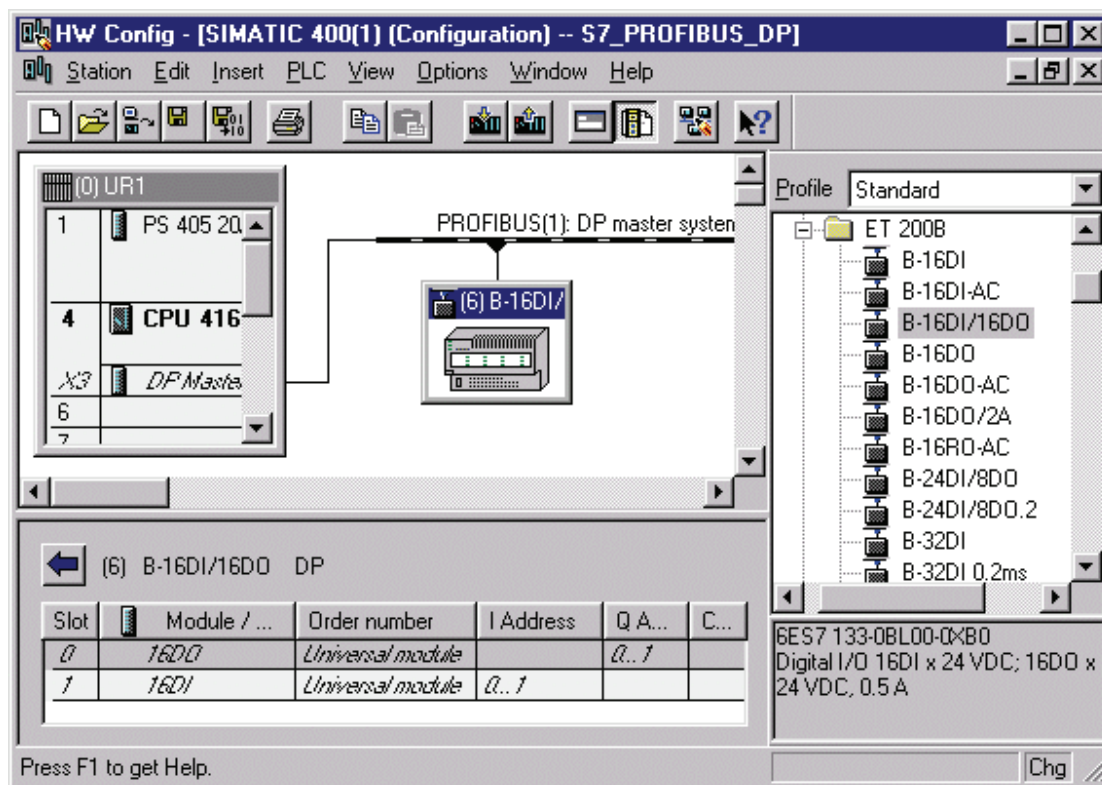


Рис.4.13 Окно HW-Config с ET200M DP-Slave

На рис.4.14 показано окно свойств DP-Slave. На этой закладке можно сменить PROFIBUS-адрес, диагностический адрес станции и т.д.

Диагностический адрес

Через диагностический адрес CPU сообщает о выходе из строя данного DP-Slave'a при помощи организационного блока OB86 "Сбой носителя модулей/DP-Slave".

В дальнейшем можно под этим адресом считать диагностику DP-Slave'a.

Способность SYNC/FREEZE

Здесь показывается, может ли выполнять DP-Slave команды DP-Master'a SYNC и/или FREEZE. Соответствующая информация берется инструментом проектирования из GSD-файла DP-Slave'a.

Response Monitoring

При включенном *Response Monitoring* DP-Master реагирует при выходе из строя обмена данными с DP-Master'ом в течение запроецированного времени (см. рис.4.6). После этого времени DP-Slave переходит в безопасное состояние, то есть все выходы устанавливаются на значение “0” или, если DP-Slave это поддерживает, выдаются заменяющие значения.

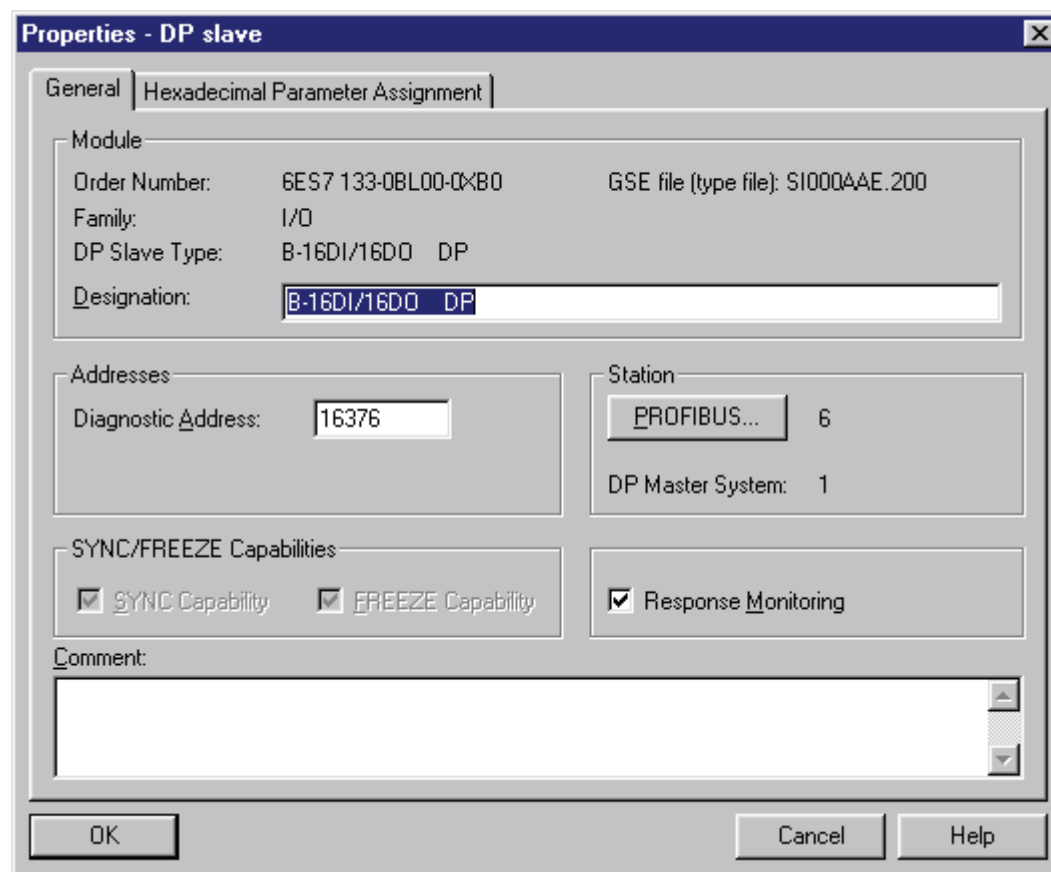


Рис.4.14 Маска свойств DP-Slave

Вы должны сознавать, что при выключенном *Response Monitoring* состояние установки может стать опасным. *Response Monitoring* может быть включено и выключено для каждого отдельного DP-Slave.

На закладке “Hexadecimal Parameter Assignment” окна “Properties – DP slave” задаются специфические для Slave данные параметрирования. Содержание и значение этих данных берется из документации на соответствующие DP-Slave’ы. Для ET 200B, спроектированной в примере, нет установок для данных параметров. Должно быть задано (установки по умолчанию) 5 байт с содержанием “0”. Данные, сохраняемые здесь, передаются DP-Slave’у с помощью телеграммы параметрирования. У S7-DP-Slave’ов параметры задаются в 16-ичном формате. Соответствующие установки для данных телеграмм параметрирования осуществляются внутри утилиты HW-Config прямо при проектировании DP-Slave.

Станция ET 200M

Для проектирования модульной станции ET 200M, содержащей 3 сигнальных модуля 8DI/8DO, AI2 x 12Bit и AO2 x 12Bit, сначала необходимо перетащить на шину PROFIBUS интерфейсный модуль IM153-2 и задать ему PROFIBUS-адрес. Далее отметить в верхнем окне IM153-2 и в нижнее окно установить методом Drag&Drop необходимые модули, содержащиеся в разделе каталога внутри раздела с названием соответствующего IM. Модули параметрируются обычным способом. На рис.4.15 показан результат.

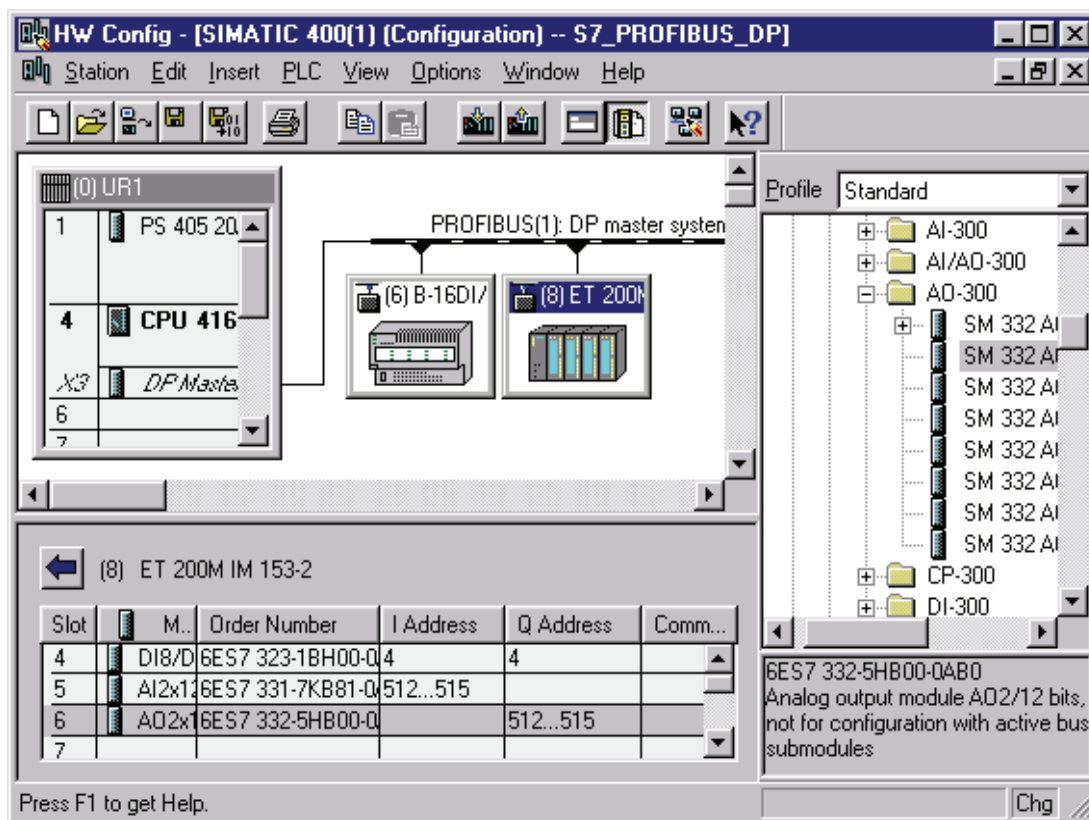


Рис. 4.15 Окно станции в HW-Config с ET200M-станцией

S7-300/CPU315-2DP как I-Slave

Перед подключением S7-300 к системе DP-Master он должен быть вначале создан внутри проекта. Вы вставляете в проект *SIMATIC 300 Station*, открываете HW-Config для этой станции и вставляете из RACK-300 носитель модулей (он там один), а в него – источник питания (не обязательно), CPU и сигнальные модули. CPU должен иметь встроенный интерфейс DP. При вставке CPU назначьте ту подсеть PROFIBUS, к которой станция должна быть подключена как Slave. Далее “кликните” мышью по *DP-Master*, откроется окно “*Properties – DP Master*” откройте закладку “*Operating Mode*” (“*Режим работы*”) и установите режим работы “*DP Slave*”. После этого название “*DP-Master*” изменится на “*DP-Slave*”. Далее перейдите на закладку “*Configuration*” (“*Конфигурация*”) и заполните маску, как показано на рис.4.16.

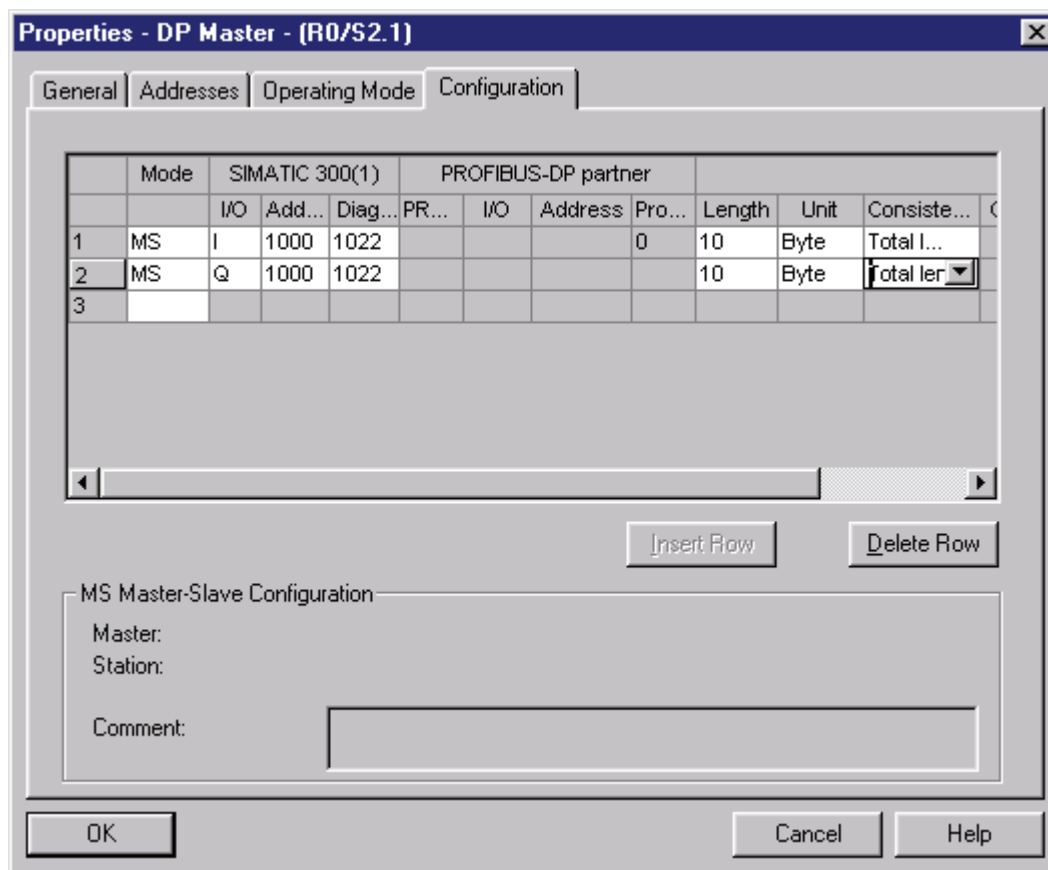


Рис. 4.16 HW-Config, "Properties DP-Master", закладка "Configuration" CPU315-2DP

В этой закладке Вы устанавливаете следующие параметры и свойства DP-интерфейса.

- Конфигурация входных/выходных областей в DP-Slave для коммуникаций Master – Slave.
- Конфигурация входных/выходных областей в DP-Slave для прямого обмена данными (перекрестная связь).
- Локальный диагностический адрес интерфейса DP-Slave (диагностический адрес на закладке "Addresses" при роде работы CPU "Slave" не важен).

С помощью кнопки ОК перейдите опять в окно HW-Config станции S7-300. Запомните конфигурацию станции S7-300 и перейдите в окно HW-Config для станции S7-400 (окно HW-Config для станции S7-400 должно быть открыто).

Для проектирования станции S7-300, как DP-Slave'a, откройте в Hardware Catalog'е папку "PROFIBUS-DP", а в ней подкаталог "Configured Station" и подключите объект "CPU31x-2DP" к DP-Master'у методом Drag&Drop. При этом появляется окно "DP slave properties" с открытой закладкой "Connection", где показан перечень соответствующих Slave'ов. Выделите нужный и нажмите кнопку "Connect" (см. рис.4.17). Выбранный Slave исчезает из списка.

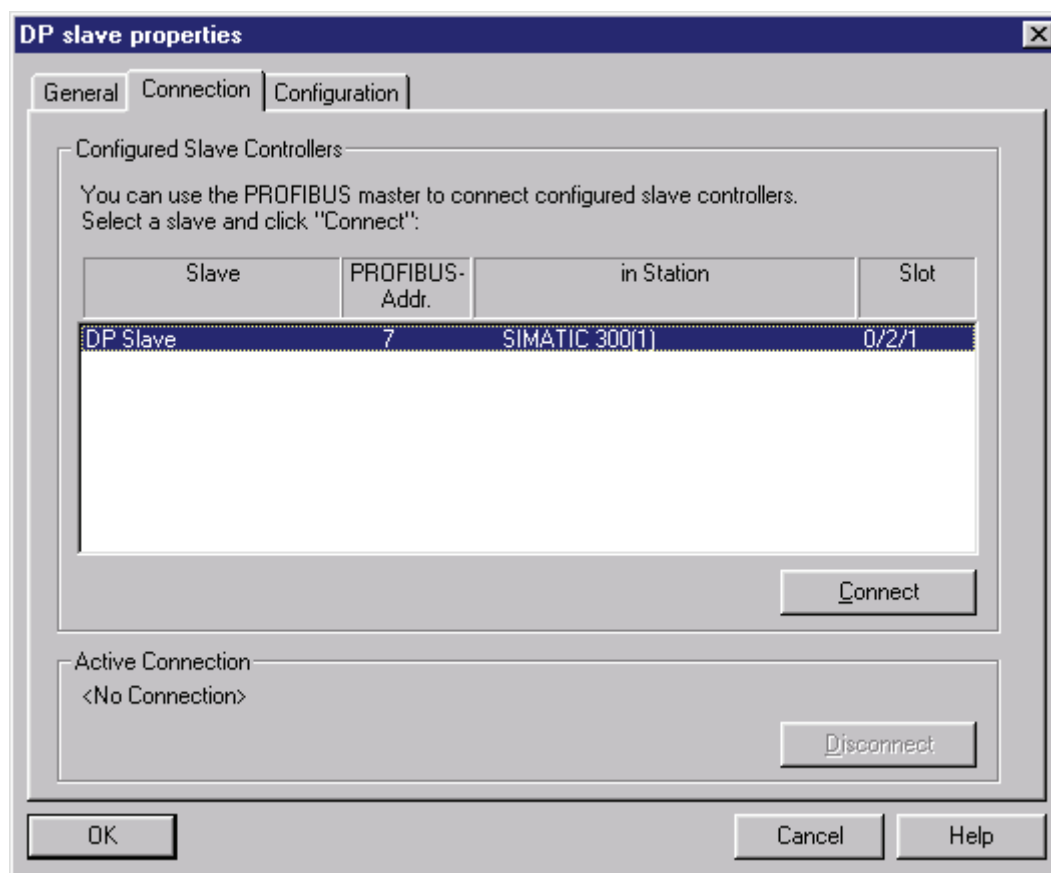


Рис. 4.17 HW-Config, “DP-slave properties”, закладка “Connection”

Перейдите на закладку “*Configuration*” и внесите туда параметры для Master’а, как это показано на рис.4.18.

Приведенные здесь области свободно выбраны для проекта-примера. Вы можете задать другие области и адреса. Нужно только обратить внимание на то, что выходная область DP-Master’а всегда соответствовала входной области DP-Slave и наоборот.

Вернитесь в главное окно станции SIMATIC S7-400 с помощью кнопки OK. Slave в главном окне получит PROFIBUS-адрес выбранного Slave (см. рис. 4.19).

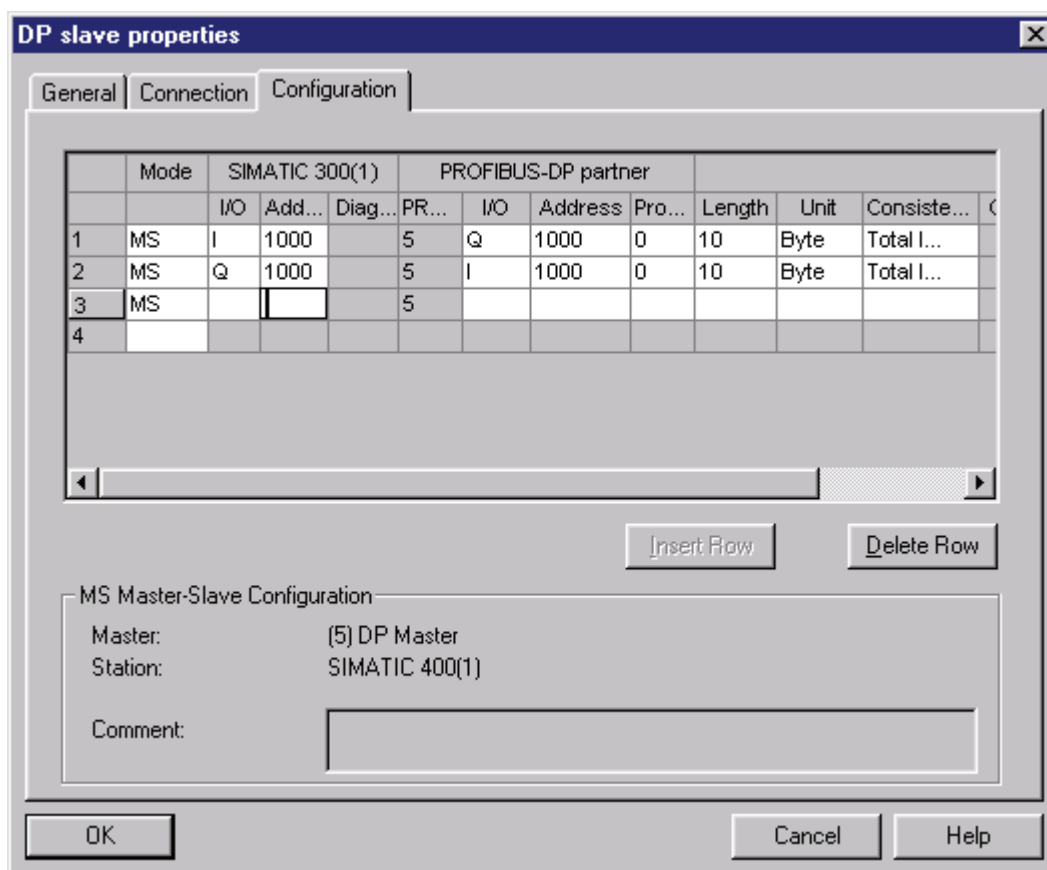


Рис. 4.18 HW-Config, “DP-slave properties”, закладка ”Configuration”

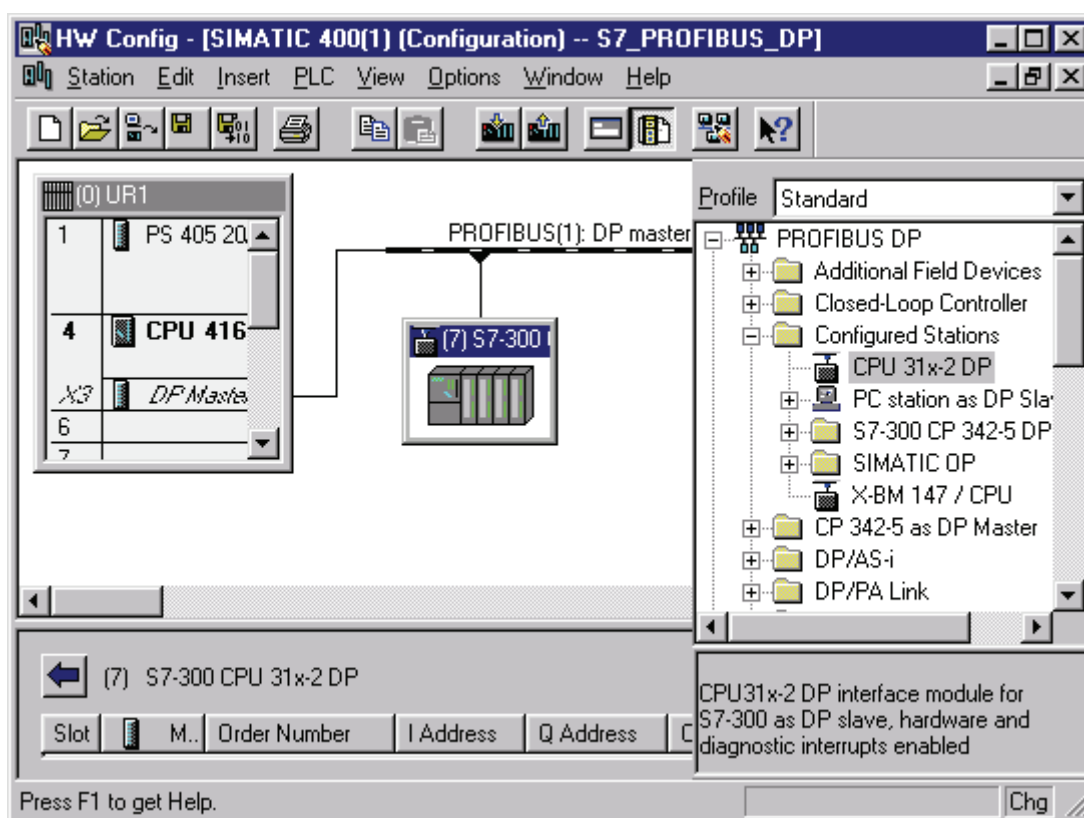


Рис. 4.19 HW-Config, окно станции SIMATIC 400

4.3 Использование коммуникационных процессоров CP443-5 Ext и CP342-5

Коммуникационный процессор CP443-5 Ext может быть только DP-Master'ом. Он ничем не отличается от встроенного интерфейса DP. В его окне свойств есть закладка *“Diagnostics”* (*“Диагностика”*). Она предназначена для диагностики с помощью пакета NCM Profibus (см. главу 7).

Коммуникационный процессор CP342-5 может быть как DP-Master'ом, так и DP-Slave'ом. На рис.4.20 показаны окна HW-Config для системы с CP342-5. Для работы с этим CP требуются специальные функции (см. далее).

В окне свойств CP342-5, на закладке *“Operation Mode”* можно выбрать вид работы CP342-5. Если выбран режим *“DP slave”*, то можно данный Slave сделать как активным (установлено по умолчанию), так и пассивным Slave'ом (см. гл.3, стр.5, раздел 3.2).

В окне свойств CP342-5, на закладке *“Options”* (см. рис.4.22) можно установить, где будут сохраняться системные блоки данных, относящиеся к CP: в папке *“Blocks”* CPU или в такой же папке CP. По умолчанию SDB сохраняются в CPU.

У CP342-5 в окне свойств есть также закладка *“Diagnostics”*, предназначенная для диагностики с помощью пакета NCM Profibus (см. далее).

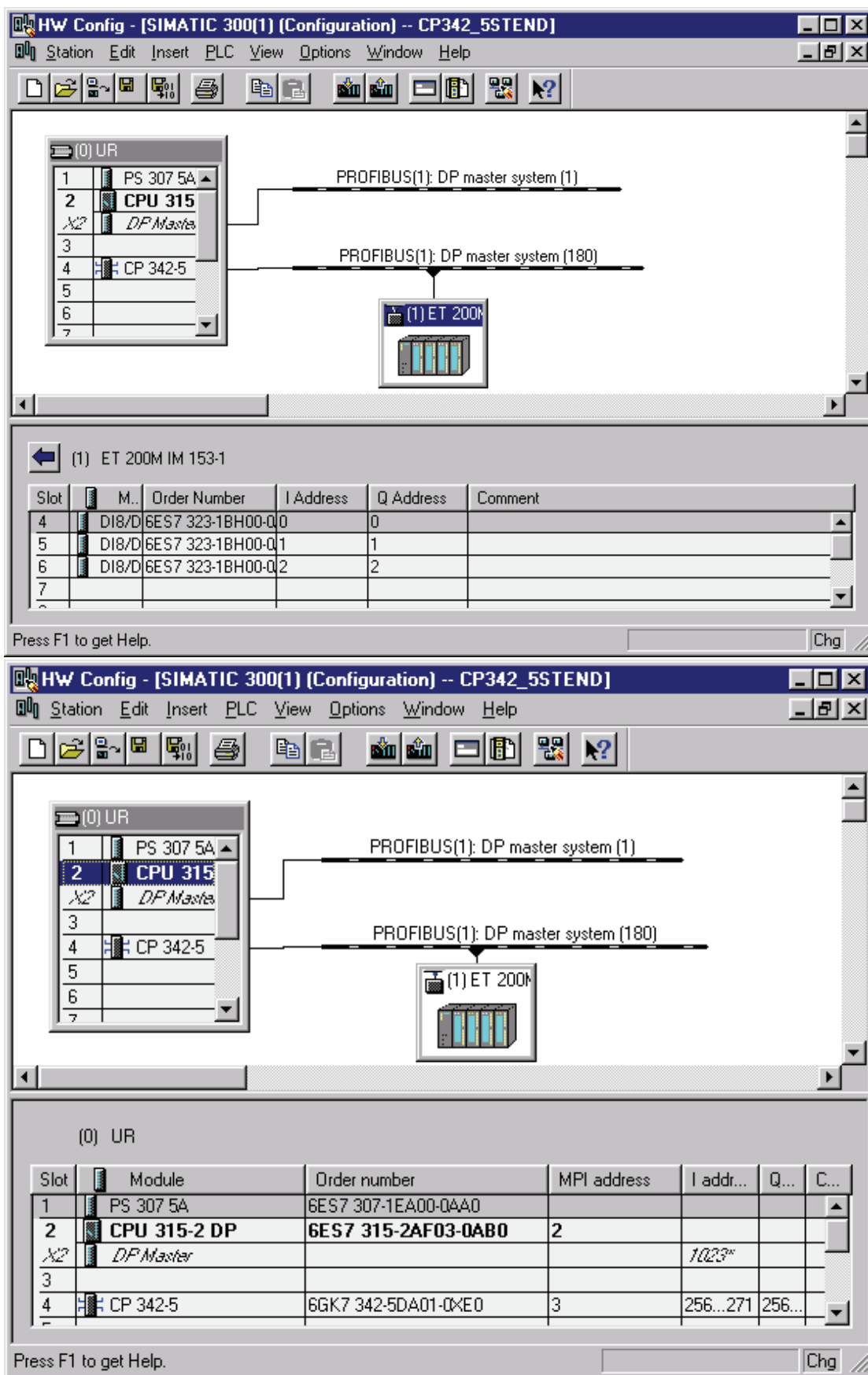


Рис. 4.20 Утилита HW-Config для PLC с CP342-5 - Master

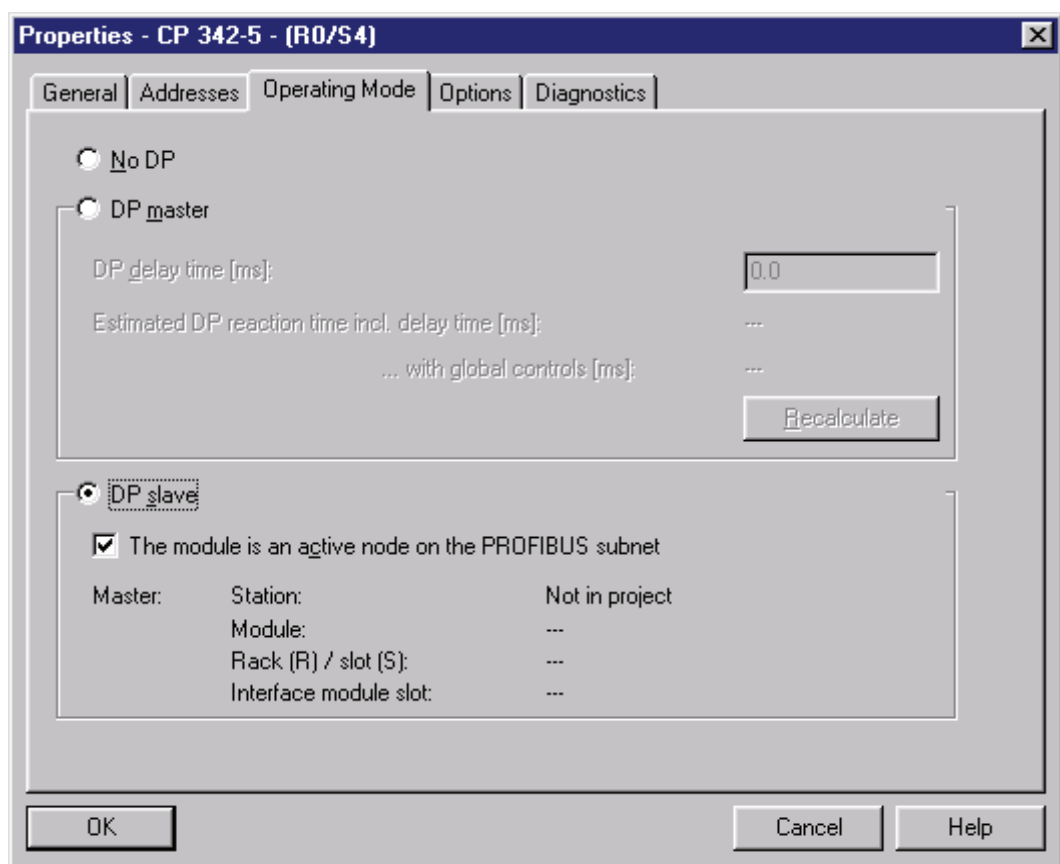


Рис.4.21 Свойства CP342-5: Operation Mode

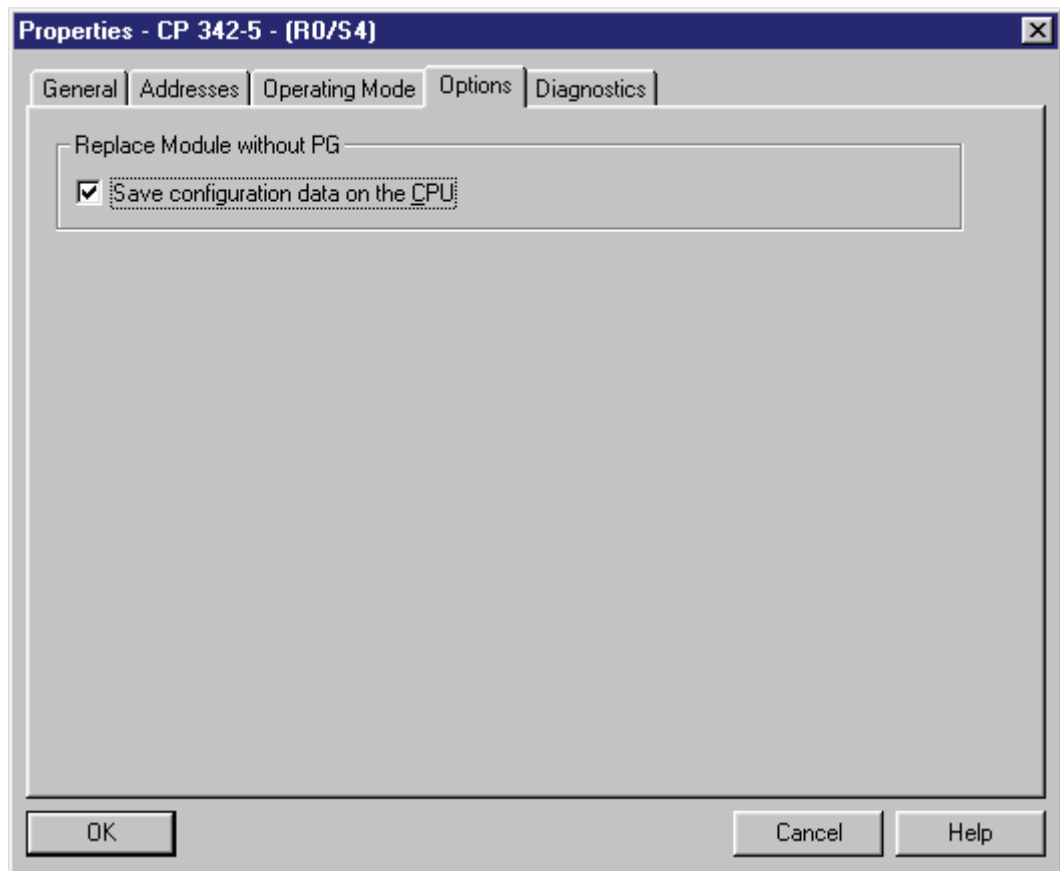


Рис. 4.22 Свойства CP342-5: Options

5. Пользовательское программирование DP-интерфейса

Введение

Децентрализованная периферия, подключенная к системе SIMATIC S7, обрабатывается с точки зрения пользовательской программы так же, как и центральная периферия. Исключение составляет децентрализованная периферия, подключенная к CP342-5. Обмен данными с DP-Slave происходит через области изображения входного/выходного процесса или с помощью прямого доступа к периферии из пользовательской программы.

Для обработки и оценки процессных и диагностических сигналов в распоряжении имеются соответствующие интерфейсы и функции. Возможно также непосредственное параметрирование DP-Slave'a из пользовательской программы.

Обмен данными с DP-Slave'ами, которые имеют сложные функции, может происходить на основании часто используемых консистентных данных не через простой доступ к периферии из программы пользователя. Для коммуникаций с этими DP-Slave'ами в системе SIMATIC S7 предусмотрены специальные системные функции.

Эта глава дает обзор основных DP-функций и интерфейсов программы пользователя в CPU SIMATIC S7. Она одновременно служит основой для понимания и реализации практической программы-примера в главах 6 и 7.

5.1 Основы пользовательского DP-интерфейса

5.1.1 Организационные блоки

Для обработки пользовательской программы CPU SIMATIC S7 имеет в своем распоряжении ряд организационных блоков (ОВ). ОВ – интерфейс между обрабатываемой пользовательской программой и операционной системой CPU. С помощью ОВ обрабатываются во время выполнения пользовательской программы специальные программные части, управляемые событиями. Так, например, при появлении сигналов от процесса, возбуждаемых S7-DP-Slave'ом, или при выходе из строя DP-Slave'a, операционная система S7-CPU каждый раз вызывает ОВ, зарезервированный для данного события. Таким образом, благодаря организационным блокам возможна обработка пользовательской программы, зависящая от событий. Так как вызов ОВ операционной системой при появлении определенного события является одновременно прерыванием обрабатываемого ОВ, обработка ОВ во всех S7-CPU определяется системой приоритетов (см. рис. 5.1). При этом “1” означает самый низкий приоритет, а “28” – самый высокий. Шкала приоритетов от 1 до 26 показана на рис. 5.1.

Каждый ОВ снабжается при вызове операционной системой 20-ю байтами в локальном стеке (переменными), которые предоставляют различную информацию. Значения локальных данных зависят от ОВ.



Рис. 5.1 Классы приоритетов OB

Объяснение поставляемых локальных данных осуществляется соответственно в описании соответствующих OB. Значения поставляемых локальных данных соответствующих OB разъяснены в разделах 5.2.1 и 5.2.7. Обозначения переменных соответствует стандартным обозначениям в STEP 7.

5.1.2 Основные системные функции для PROFIBUS DP

Благодаря вызову интегрированных в операционную систему S7-CPU функций (SFC – System Function Call) система SIMATIC S7 реализует ряд важных функций протокола PROFIBUS.

Общие значения отдельных параметров SFC

Ряд параметров SFC идентичны по своему значению для всех описываемых ниже SFC. Это относится, в частности, к входным параметрам SFC REQ, LADDR и выходным параметрам RET_VAL и BUSY.

Параметр SFC REQ

Некоторые SFC имеют для запуска системной функции входной параметр REQ. Если параметр REQ передает при вызове логическую “1” в SFC, то вызываемая

функция выполняется. Обратите внимание, что некоторые SFC асинхронны. Это означает, что функция обрабатывается через несколько вызовов SFC, то есть через несколько циклов CPU (учитывается параметром BUSY).

Параметр SFC BUSY

Выходной параметр BUSY показывает, завершился ли вызванный SFC. Пока параметр BUSY="1", вызываемая функция активна.

Параметр SFC LADDR

Через входной параметр LADDR в зависимости от вызываемого SFC задается либо спроектированный в HW-Config логический начальный адрес входного/выходного модуля, либо диагностический адрес DP-Slave'a. Обратите внимание, что этот адрес в HW-Config проектируется в десятичном формате, однако в блоке он задается в шестнадцатичном формате.

Параметр SFC RET_VAL

Выходной параметр RET_VAL возвращает код ошибки, которая произошла при вызове функции. Ошибки могут быть двух типов:

- Общие
- Специфические для данной функции

Какого типа произошла ошибка, можно определить по значению, возвращаемому в RET_VAL (рис. 5.1).

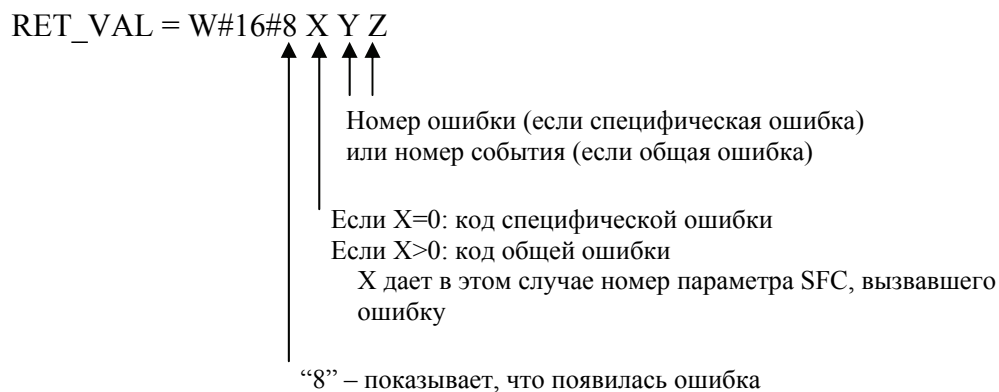


Рис. 5.2 Структура SFC-параметра RET VAL

В табл. 5.1 приведены коды общих ошибок для всех системных функций. Коды ошибок даются в 16-ичном коде.

Табл. 5.1 Коды общих ошибок в параметре RET_VAL

Код ошибки W#16#...	Пояснение
8x7F	Внутренняя ошибка. Этот код показывает внутреннюю ошибку в параметре x. Эта ошибка вызвана не пользователем и поэтому не может быть им устранена
8x22	Ошибка длины области при чтении параметра
8x23	Ошибка длины области при записи параметра. Эта ошибка показывает, что параметр x полностью или частично находится за границами области операнда или длина битовых полей у параметра типа данных ANY не делится на 8
8x24	Ошибка длины области при чтении параметра
8x25	Ошибка длины области при чтении параметра. Этот код ошибки показывает, что параметр x находится в области, которая недоступна системной функции. В описании каждой системной функции приводятся области, которые доступны функции.
8x28	Ошибка ориентации при записи параметра
8x29	Ошибка ориентации при записи параметра. Этот код ошибки показывает, что ссылка на параметр x – операнд, битовый адрес которого не равен “0”
8x30	Параметр находится в защищенном от записи глобальном DB
8x31	Параметр находится в защищенном от записи экземпляре DB. Этот код ошибки показывает, что параметр x находится в защищенном от записи блоке данных. Если блок данных открыт системной функцией, системная функция выдает всегда значение w#16#8x30
8x32	Параметр содержит слишком большой номер DB (ошибка номера DB)
8x34	Параметр содержит слишком большой номер FC (ошибка номера FC)
8x35	Параметр содержит слишком большой номер FB (ошибка номера FB). Этот код ошибки показывает, параметр x содержит номер блока, который больше максимально допустимого номера блока
8x3A	Параметр содержит номер DB, который не загружен
8x3C	Параметр содержит номер FC, которая не загружена
8x3E	Параметр содержит номер FB, который не загружен
8x42	Наступила ошибка доступа во время чтения системой входных параметров из области периферии
8x43	Наступила ошибка доступа во время записи системой выходных параметров из области периферии
8x44	Ошибка при n-м (n>1) доступе на чтение после наступления ошибки
8x45	Ошибка при n-м (n>1) доступе на запись после наступления ошибки. Эта ошибка показывает, что доступ к желаемому параметру запрещен.

Применяемые области памяти для параметров вызова SFC

Идентификаторы областей памяти, применяемые для параметров SFC, приведены в табл. 5.2.

Табл. 5.2 Области памяти для параметров SFC

Тип	Область памяти	Единица
I	Отображение процесса входов	Вход (бит)
		Входной байт (IB)
		Входное слово (IW)
		Входное двойное слово (ID)
Q	Отображение процесса выходов	Выход (бит)
		Выходной байт (QB)
		Выходное слово (QW)
		Выходное двойное слово (QD)
M	Область меркеров	Меркер (бит)
		Меркерный байт (MB)
		Меркерное слово (MW)
		Меркерное двойное слово (MD)
D	Блоки данных	Бит данных
		Байт данных (DBB)
		Слово данных (DBW)
		Двойное слово данных (DBD)
L	Локальные данные	Бит локальных данных
		Байт локальных данных (LB)
		Слово локальных данных (LW)
		Двойное слово локальных данных (LD)

5.1.3 Основы наборов данных SIMATIC S7

Системные данные и параметры сохраняются в S7-модулях как *наборы данных*. Отдельные наборы данных нумеруются от 0 до max 240, причем не каждый модуль располагает всеми наборами данных.

В зависимости от S7-модуля имеются области системных данных, в которые из пользовательской программы доступ может быть только на чтение или только на запись.

Табл. 5.3 показывает строение области системных данных, годных только для записи, определяет, как велики могут быть отдельные наборы данных и с помощью каких SFC они могут перенесены в модули.

Табл. 5.4 показывает строение области системных данных, годных только для чтения, определяет, как велики могут быть отдельные наборы данных и с помощью каких SFC они могут быть прочитаны.

Для каждой вновь запущенной передачи набора данных резервируются ресурсы CPU (память) для каждого асинхронно работающего SFC. При нескольких одновременно активных заданиях гарантируется, что все задания будут осуществлены и не будет их взаимного влияния друг на друга. Однако может быть только определенное число одновременно активных вызовов SFC. Максимальное число одновременно возможных вызовов SFC берется из рабочих характеристик CPU. Если достигается граница максимально определенных ресурсов, то через параметр RET_VAL выдается соответствующий код ошибки. В этом случае должна быть SFC запущена снова.

Табл. 5.3 Структура областей данных в S7-300 модулях, годных только для записи

Номер набора данных	Содержание	Размер	Ограничение	Могут быть описаны с помощью
0	Параметры	У S7-300: от 2 до 14 байт	Могут быть описаны только у S7-300	SFC56 WR_DPARM SFC57 PARM_MOD
1	Параметры	У S7-300: от 2 до 14 байт (DS0 и DS1 имеют вместе точно 16 байт)	-	SFC55 WR_PARM SFC 56 WR_DPARM SFC57 PARM_MOD
от 2 до 127	Пользовательские данные	до 240 байт	-	SFC55 WR_PARM SFC 56 WR_DPARM SFC57 PARM_MOD SFC 58 WR_REC
от 128 до 240	Параметры	до 240 байт	-	SFC55 WR_PARM SFC 56 WR_DPARM SFC57 PARM_MOD SFC 58 WR_REC

Табл. 5.4 Структура областей данных в S7-300 модулях, годных только для чтения

Номер набора данных	Содержание	Размер	Могут быть описаны с помощью
0	Специфич. для модуля диагностич. данные	4 байта	SFC51 RDSYSST (INDEX 00B1H) SFC59 RD_REC
1	Специфич. для модуля диагностич. данные (вкл. набор данных 0)	У S7-300: 16 байт У S7-400: от 7 до 220 байт	SFC51 RDSYSST (INDEX 00B2H и 00B3H) SFC 59 RD_REC
от 2 до 127	Пользовательские данные	до 240 байт	SFC59 RD_REC
от 128 до 240	Диагностические данные	до 240 байт	SFC59 RD_REC

Отдельные параметры в наборах данных могут быть *статическими* и *динамическими*. Статические параметры модулей, например, задержка входов

модуля дискретных входов, может быть изменены только инструментом проектирования STEP 7.

Динамические параметры модулей могут быть в противоположность к статическим параметрам, изменены во время работы с помощью вызова SFC, например, установлены новые граничные значения аналогового модуля входов.

5.2 Организационные блоки

5.2.1 Циклическая обработка главной программы (OB1)

Главная программа обрабатывается в OB1. В OB1 вызываются FB, SFB, FC и SFC. OB1 вызывается и обрабатывается циклически. Первым стартует OB1 после прохождения стартовых OB (OB100 – новый старт, OB101 – повторный старт). Если обработка OB1 закончена, операционная система переносит изображение процесса выходов в выходные модули. Перед новым стартом OB1 операционная система актуализирует изображение процесса входов: считывает значения входных модулей и записывает их в область отображения входов. Этот процесс постоянно повторяется. При этом говорится о циклической обработке. OB1 имеет самый низший приоритет из всех OB, работающих в реальном времени и, таким образом, может быть прерван любым другим OB.

S7-CPU предлагает пользователю контроль максимального времени цикла (времени обработки OB1), а также (только CPU S7-400) соблюдение минимального времени цикла для обработки OB1. Если запрограммировано минимальное время цикла, то операционная система CPU задерживает новый запуск OB1 пока не истечет запрограммированное время. Эти параметры могут быть установлены с помощью HW-Config в окне свойств CPU. Значения локальных данных OB1 приведены в табл. 5.5

Табл. 5.5 Локальные данные OB1

Переменная	Тип данных	Описание
OB1_EV_CLASS	BYTE	Класс события и идентификатор: B#16#11=active
OB1_SCAN_1	BYTE	B#16#01-окончание нового старта B#16#02-окончание повторного старта B#16#03-окончание свободного цикла
OB1_PRIORITY	BYTE	Класс приоритета "1"
OB1_OB_NUMBER	BYTE	Номер OB (01)
OB1_RESERVED_1	BYTE	Зарезервировано
OB1_RESERVED_2	BYTE	Зарезервировано
OB1_PREV_CYCLE	INT	Время работы предыдущего цикла (ms)
OB1_MIN_CYCLE	INT	Мин. время цикла с момента последнего запуска (ms)
OB1_MAX_CYCLE	INT	Макс. время цикла с момента последнего запуска (ms)
OB1_DATE_TIME	DT	Дата и время вызова OB

5.2.2 Сигналы от процесса (OB40 - OB47)

CPU SIMATIC S7-400 имеет в своем распоряжении до 8 независимых друг от друга OB (OB40 – OB47) для обработки сигналов от процесса. Для S7-DP-Slave, которые поддерживают сигналы от процесса, устанавливаются через HW-Config канал, граничные условия и номер OB для обработки сигналов от процесса.

Если DP-Slave возбудил сигнал от процесса, то он будет идентифицирован операционной системой CPU и запущен соответствующий ОБ обработки сигнала от процесса в зависимости от запроецированного класса приоритетов. После обработки пользовательской программы в ОБ для сигнала от процесса (окончания ОБ), будет квитировано DP-Slave'ом его сообщение о сигнале.

Если ОБ для сигнала от процесса еще активен, в то время, как поступает другой сигнал от процесса, то в системе S7-400 это событие регистрируется, запоминается и ОБ обрабатывается позднее. В системе S7-300 сигнал от процесса в этом случае теряется. Сигнал от процесса должен возникнуть вновь после квитирования предыдущего.

ОБ для сигнала от процесса снабжается временными переменными, занимающими 20 байт в локальном стеке. Среди этих переменных содержится, например, логический адрес блока, который выдал сигнал. Значения локальных данных приведены в таблице 5.6.

Табл. 5.6 Локальные данные OB40 – OB47

Переменная	Тип данных	Описание
OB4x_EV_CLASS	BYTE	Класс события и идентификатор: B#16#11=сигнал активен
OB4x_STRT_INF	BYTE	B#16#41 - сигнал через линию прерывания 1 Только для S7-400 B#16#42 - сигнал через линию прерывания 2 B#16#43 - сигнал через линию прерывания 3 B#16#44 - сигнал через линию прерывания 4
OB4x_PRIORITY	BYTE	Класс приоритета от “16” (OB40) до “23” (OB47) (по умолчанию)
OB4x_OB_NUMBER	BYTE	Номер ОБ (40 - 47)
OB4x_RESERVED_1	BYTE	Зарезервировано
OB4x_IO_FLAG	BYTE	B#16#54 = входной модуль B#16#55 = выходной модуль
OB4x_MDL_ADDR	WORD	Логический базовый адрес модуля, пославшего сигнал
OB4x_POINT_ADDR	DWORD	В дискр. модулях: битовое поле с состоянием входов модуля В аналоговом модуле (а также CP и FM) состояние сигнала модуля
OB4x_DATE_TIME	DT	Дата и время вызова ОБ

5.2.3 Сигнал состояния (OB55)

Операционная система S7-CPU вызывает OB55, если от одного установочного места (слота) DPV1-Slave'ов поступает сигнал состояния.

ОБ сигнала состояния имеется только у S7-CPU, поддерживающих DPV1. Для обработки сигнала состояния эти CPU имеют OB55. Если OB55 не запрограммирован, то CPU остается в состоянии RUN. Происходит лишь запись в диагностический буфер CPU.

ОБ сигнала состояния имеет 20 байт локальных данных, из которых, например, можно получить логический базовый адрес как слота, так и модуля, пославшего сигнал. Значения локальных данных пояснены в таблице 5.7.

Табл. 5.7 Локальные данные OB55

OB55_EV_CLASS	BYTE	Класс события и идентификаторы В#16#11 (приходящее событие)
OB55_STRT_INF	BYTE	В#16#55 (Требование запуска для OB55)
OB55_PRIORITY	BYTE	Параметрируемый класс приоритета. По умолчанию: 2
OB55_OB_NUMBER	BYTE	№ OB (55)
OB55_RESERVED_1	BYTE	Зарезервировано
OB55_IO_FLAG	BYTE	Входной модуль: В#16#54 Выходной модуль: В#16#55
OB55_MDL_ADDR	BYTE	Логический базовый адрес компонента, пославшего сигнал состояния
OB55_LEN	BYTE	Длина блока данных, который поставляет сигнал
OB55_TYPE	BYTE	Идентификатор для типа сигнала “Сигнал состояния”
OB55_SLOT	BYTE	Номер слота компонента, пославшего сигнал (модуль)
OB55_SPEC	BYTE	Спецификатор: Биты 0..1: Спецификатор сигнала Бит 2 : Add_Ack Биты 3..7:Номер последовательности
OB55_DATE_TIME	DT	Дата и время, когда был вызван OB

5.2.4 Сигнал модернизации (OB56)

Появление сигнала модернизации в SIMATIC S7-CPU может быть опознано с помощью вызова OB56. Сигнал модернизации имеют только S7-CPU, поддерживающие DPV1. Он вызывается операционной системой, когда слот или модуль DPV1-Slave’a посылает сигнал модернизации. В случае, если OB56 не запрограммирован, CPU остается в состоянии RUN. Происходит лишь запись в диагностический буфер CPU.

OB сигнала модернизации снабжен 20 байтами локальных данных, из которых, например, можно получить логический базовый адрес как слота, так и модуля, пославшего сигнал. Значения локальных данных пояснены в таблице 5.8.

Табл. 5.8 Локальные данные OB56

OB56_EV_CLASS	BYTE	Класс события и идентификаторы В#16#11 (приходящее событие)
OB56_STRT_INF	BYTE	В#16#56 (Требование запуска для OB56)
OB56_PRIORITY	BYTE	Параметрируемый класс приоритета. По умолчанию: 2
OB56_OB_NUMBER	BYTE	№ OB (56)
OB56_RESERVED_1	BYTE	Зарезервировано
OB56_IO_FLAG	BYTE	Входной модуль: В#16#54 Выходной модуль: В#16#55
OB56_MDL_ADDR	BYTE	Логический базовый адрес компонента, пославшего сигнал модернизации
OB56_LEN	BYTE	Длина блока данных, который поставляет сигнал
OB56_TYPE	BYTE	Идентификатор для типа сигнала “Сигнал модернизации”
OB56_SLOT	BYTE	Номер слота компонента, пославшего сигнал (модуль)
OB56_SPEC	BYTE	Спецификатор: Биты 0..1: Спецификатор сигнала Бит 2 : Add_Ack Биты 3..7:Номер последовательности
OB56_DATE_TIME	DT	Дата и время, когда был вызван OB

5.2.5 Сигнал, специфический для производителя (OB57)

Для распознавания сигнала, специфического для производителя, SIMATIC S7-CPU имеет OB57. Операционная система CPU вызывает OB57, когда слот DPV1-Slave'a посылает сигнал, специфический для производителя.

OB57 имеется только у S7-CPU, поддерживающих DPV1. Если OB57 не запрограммирован, то CPU остается в состоянии RUN. Происходит лишь запись в диагностический буфер CPU.

ОВ сигнала состояния имеет 20 байт локальных данных, из которых, например, можно получить логический базовый адрес как слота, так и модуля, вызвавшего сигнал. Значения локальных данных пояснены в таблице 5.9.

Табл. 5.9 Локальные данные OB57

OB57_EV_CLASS	BYTE	Класс события и идентификаторы B#16#11 (приходящее событие)
OB57_STRT_INF	BYTE	B#16#57 (Требование запуска для OB57)
OB57_PRIORITY	BYTE	Параметрируемый класс приоритета. По умолчанию: 2
OB57_OB_NUMBER	BYTE	№ OB (57)
OB57_RESERVED_1	BYTE	Зарезервировано
OB57_IO_FLAG	BYTE	Входной модуль: B#16#54 Выходной модуль: B#16#55
OB57_MDL_ADDR	BYTE	Логический базовый адрес компонента, пославшего сигнал, зависящий от производителя
OB57_LEN	BYTE	Длина блока данных, который поставляет сигнал
OB57_TYPE	BYTE	Идентификатор для типа сигнала "Сигнал , зависящий от производителя"
OB57_SLOT	BYTE	Номер слота компонента, пославшего сигнал (модуль)
OB57_SPEC	BYTE	Спецификатор: Биты 0..1: Спецификатор сигнала Бит 2 : Add_Ack Биты 3..7:Номер последовательности
OB57_DATE_TIME	DT	Дата и время, когда был вызван OB

5.2.6 Диагностические сигналы (OB82)

Чтобы распознавать и реагировать на диагностические события, CPU SIMATIC S7 имеет в распоряжении OB82. Он запускается, если способный к диагностике DP-Slave распознал некоторую ошибку (событие). Вызов OB82 операционной системой происходит как при приходящем, так и при уходящем событии. Предпосылкой для этого является то, что DP-Slave поддерживает эту функцию и диагностический сигнал был запараметрирован (деблокирован) при параметрировании DP-Slave'a в HW-Config.

Если OB82 не программируется, то CPU при наступлении диагностического события переходит в состояние STOP. Через OB82 DP-Slave'ы сигнализируют о актуальных диагностических событиях. Временные переменные OB82 объемом 20 байт (см. табл. 5.10) содержат, например, логический базовый адрес DP-Slave'a, содержащего ошибку, содержащий ошибку модуль DP-Slave'a, а также 4 байта диагностической информации.

Табл. 5.10 Локальные данные OB82

Переменная	Тип данных	Описание
OB82_EV_CLASS	BYTE	Класс события и идентификатор: В#16#38 – приходящее событие В#16#39 – уходящее событие
OB82_FLT_ID	BYTE	В#16#42 – код ошибки
OB82_PRIORITY	BYTE	Класс приоритетов “26” (по умолчанию для режима работы RUN) “28” (режим работы ЗАПУСК (STARTUP))
OB82_OB_NUMBER	BYTE	Номер OB (82)
OB82_RESERVED_1	BYTE	Зарезервировано
OB82_IO_FLAG	BYTE	В#16#54 = входной модуль В#16#55 = выходной модуль
OB82_MDL_ADDR	INT	Логический базовый адрес модуля, которым встретилась ошибка
OB82_MDL_DEFECT	BOOL	Повреждение в модуле
OB82_INT_FAULT	BOOL	Внутренняя ошибка
OB82_EXT_FAULT	BOOL	Внешняя ошибка
OB82_PNT_INFO	BOOL	Наличие ошибки канала
OB82_EXT_VOLTAGE	BOOL	Ошибка внешнего вспомогательного напряжения
OB82_FLD_CONNCTR	BOOL	Ошибка фронтштекера
OB82_NO_CONFIG	BOOL	Модуль не параметрирован
OB82_CONFIG_ERR	BOOL	Неверный параметр в модуле
OB82_MDL_TYPE	BYTE	Биты 0-3: класс модуля Бит 4 : имеющаяся канальная информация Бит 5 : имеющаяся пользовательская информация Бит 6 : диагностический сигнал
OB82_SUB_MDL_ERR	BOOL	Применяемый модуль неправильный/сбойный
OB82_COMM_FAULT	BOOL	Коммуникационная помеха
OB82_MDL_STOP	BOOL	Рабочее состояние (0:RUN, 1:STOP)
OB82_WTCH_DOG_FLT	BOOL	Контроль времени сработал
OB82_INT_PS_FLT	BOOL	Упало напряжение питания периферийного модуля
OB82_PRIM_BATT_FLT	BOOL	Батарея разряжена
OB82_BCKUP_BATT_FLT	BOOL	Общая буферизация вышла из строя
OB82_RESERVED_2	BOOL	Зарезервировано
OB82_RACK_FLT	BOOL	Выход из строя устройства расширения
OB82_PROC_FLT	BOOL	Выход из строя процессора
OB82_EPROM_FLT	BOOL	Ошибка EPROM
OB82_RAM_FLT	BOOL	Ошибка RAM
OB82_ADU_FLT	BOOL	Ошибка ADU/DAU
OB82_FUSE_FLT	BOOL	Выход из строя защиты
OB82_HW_INTR_FLT	BOOL	Потеря сигнала от процесса
OB82_RESERVED_3	BOOL	Зарезервировано
OB82_DATE_TIME	DT	Дата и время вызова OB

5.2.7 Сигналы снятия и установки модуля (OB83)

Снятие и установка центральных модулей распознается и сообщается в S7-CPU. Эта функция поддерживается также при децентрализованно установленных в S7-DP-Slave'ах модулях, как, например, ET 200M/IM153. Если при этом из модульного S7-DP-Slave'а извлечен модуль и S7-CPU находится в режиме RUN, то будет запущен OB83 и дополнительно сделана запись в

диагностический буфер CPU и в данные состояния модуля. Если S7-CPU находится в режиме STOP или в режиме STARTUP (ЗАПУСК), то имеет место только запись в диагностический буфер CPU и в данные состояния модуля.

Если запроецированный модуль устанавливается в состояние RUN, то CPU проверяет, совпадает ли тип модуля с запроецированным. Затем стартует OB83 и при соответствии типа модуля с типом, запроецированным в HW-Config, в CPU сохраняются данные параметрирования. При такой установке также имеется возможность параметрировать новые модули специальными пользовательскими данными параметрирования с помощью SFC перепараметрирования. В табл. 5.11 приведены локальные данные OB83.

Табл. 5.11 Локальные данные OB83

Переменная	Тип данных	Описание
OB83_EV_CLASS	BYTE	Класс события и идентификатор: B#16#38 – установка модуля B#16#39 – снятие модуля
OB83_FLT_ID	BYTE	Код ошибки (возможные значения: B#16#61, B#16#63, B#16#64, B#16#65)
OB83_PRIORITY	BYTE	Класс приоритетов “26” (по умолчанию для режима работы RUN) “28” (режим работы ЗАПУСК (STARTUP))
OB83_OB_NUMBER	BYTE	Номер OB (83)
OB83_RESERVED_1	BYTE	Зарезервировано
OB83_MDL_ID	BYTE	B#16#54 = область входной периферии (PI) B#16#55 = область выходной периферии (PQ)
OB83_MDL_ADDR	WORD	Логический базовый адрес соответствующего модуля.
OB83_RACK_NUM	WORD	№ носителя модулей, соотв. № DP-станции и идентификатор DP-Master’a системы (старший байт)
OB83_MDL_TYPE	WORD	Тип модуля
OB83_DATE_TIME	DT	Дата и время вызова OB

В зависимости от локальной переменной OB83_MDL_TYPE при несоответствии типа модуля выдается сообщение об ошибке, указанное в таблице 5.12.

Табл. 5.12 Код ошибки, сообщаемый через OB83_FLT_ID

Коди ошибки OB83_FLT_ID	Значение ошибки в зависимости от OB83_MDL_TYPE
B#16#61	Модуль установлен, тип модуля О.К. (при классе события B#16#38) Модуль снят, соответственно не отвечает (при классе события B#16#39) при OB83_FLT_ID = действительный тип модуля
B#16#63	Модуль установлен, однако неправильный тип модуля при OB83_FLT_ID = действительный тип модуля
B#16#64	Модуль установлен, однако он неисправен (идентификатор нечитаем) при OB83_FLT_ID = заданный тип модуля
B#16#65	Модуль установлен, однако ошибка при параметрировании модуля при OB83_FLT_ID = действительный тип модуля

5.2.8 Ошибка выполнения программы (OB85)

Вызов OB85 операционной системой S7-CPU происходит, если пользовательская программа вызывает блок, который не загружен или операционная система вызывает OB, который не запрограммирован. Однако

OB85 также вызывается, если имеет место ошибка доступа к периферии при актуализации общего изображения процесса. Это происходит, например, в случае, если запроецированные адреса для входов и выходов DP-Slave'a находятся внутри изображения процесса S7-CPU и DP-Slave вышел из строя. Если OB85 не запрограммирован, то S7-CPU переходит в STOP.

Таблица 5.13 показывает оригинальную структуру локальных данных OB85, таблица 5.14 – предложение для структурирования локальных данных OB85, чтобы сделать в пользовательской программе возможной простую оценку кодов ошибок. При применении децентрализованной периферии особенно важны 16-ичные коды ошибок “B1” и ”B2” переменной OB85_FLT_ID.

Табл. 5.13 Локальные данные OB85 (оригинальная структура)

Переменная	Тип данных	Описание
OB85_EV_CLASS	BYTE	Класс события и идентификатор, например, B#16#39 – ошибка актуализации процесса изображения
OB85_FLT_ID	BYTE	Код ошибки (возможные значения: B#16#A1, B#16#A2, B#16#A3, B#16#B1, B#16#B2)
OB85_PRIORITY	BYTE	Класс приоритетов “26” (по умолчанию для режима работы RUN) “28” (режим работы ЗАПУСК (STARTUP))
OB85_OB_NUMBER	BYTE	Номер OB (85)
OB85_RESERVED_1	BYTE	Зарезервировано
OB85_RESERVED_2	BYTE	Зарезервировано
OB85_RESERVED_3	INT	Зарезервировано
OB85_ERR_EV_CLASS	BYTE	Класс события, вызвавшего ошибку
OB85_ERR_EV_NUM	BYTE	Номер события, вызвавшего ошибку
OB85_OB_PRIOR	BYTE	Класс приоритета OB, который обрабатывался, когда встретилась ошибка
OB85_OB_NUM	BYTE	Номер OB, который обрабатывался, когда встретилась ошибка
OB85_DATE_TIME	DT	Дата и время вызова OB

Табл. 5.14 Локальные данные OB85 для программирования, зависящего от кода ошибки

Переменная	Тип данных
OB85_EV_CLASS	BYTE
OB85_FLT_ID	BYTE
OB85_PRIORITY	BYTE
OB85_OB_NUMBER	BYTE
OB85_DKZ23	BYTE
OB85_RESERVED_2	BYTE
OB85_Z1	WORD
OB85_Z23	DWORD
OB85_DATE_TIME	DT

Код ошибки, сообщаемый через OB85_FLT_ID в зависимости от записи в переменных OB85_DKZ23, OB85_Z1, OB85_Z23 приведен в табл. 5.15.

Табл. 5.15 Коды ошибок OB85_FLT_ID

Код ошибки OB85_FLT_ID	Значение ошибки
B#16#A1	Ваша программа или ОС (на основе Вашего проектирования с помощью STEP 7) генерирует стартовое событие для OB, который не загружен в CPU
B#16#A2	Ваша программа или ОС (на основе Вашего проектирования с помощью STEP 7) генерирует стартовое событие для OB, который не загружен в CPU. Через переменные OB85_Z1 и OB85_Z23 предоставляется следующая информация: OB85_Z1 : Класс события, возбудившего ошибку (значение прерванного программного уровня) OB85_Z23: <u>Старшее слово</u> : класс и номер события, вызвавшего ошибку <u>Младшее слово</u> : активный уровень программы и активный OB на момент появления ошибки.
B#16#A3	Ошибка обращения ОС к блоку. Через переменные OB85_Z1 и OB85_Z23 предоставляется следующая информация: OB85_Z1 : Детализирует идентификатор ошибки ОС. <u>Старший байт</u> : 1: встроенная функция 2: IEC-таймер <u>Младший байт</u> : 0: ошибка не обнаружена 1: блок не загружен 2: ошибка длины области 3: ошибка защиты записи OB85_Z23: <u>Старшее слово</u> : № блока <u>Младшее слово</u> : относительный адрес MC7-команды, вызвавшей ошибку. Тип блока следует взять из OB85_DKZ23: B#16#88 = OB B#16#8C = FC B#16#8E = FB B#16#8A = DB
B#16#B1	Ошибка доступа к периферии при актуализации изображения процесса входов.
B#16#B2	Ошибка доступа к периферии при переносе изображения процесса выходов в выходные модули. Через переменные OB85_Z1 и OB85_Z23 предоставляется следующая информация: OB85_Z1 : зарезервирование для внутреннего применения CPU OB85_Z23: номер периферийного байта, который вызвал ошибку доступа к периферии (PZF).

5.2.9 Выход из строя носителя модулей (OB86)

Выход из строя (приходящее событие) или восстановление (уходящее событие) устройств расширения, DP-Master-системы или DP-Slave'a сообщается операционной системой S7-CPU через OB86. Если OB86 не запрограммирован, S7-CPU переходит при наступлении события в STOP.

Таблица 5.16 показывает оригинальную структуру локальных данных OB86. Структура, представленная в таблице 5.17, является предложением по структурированию локальных данных OB86, чтобы в пользовательской программе можно было сделать простую, зависящую от кода, оценку. При

применении децентрализованной периферии особенно важны 16-ичные коды ошибок “C3”, “C4”, “C7” переменной OB86_FLT_ID.

Табл. 5.16 Локальные данные OB86

Переменная	Тип данных	Описание
OB86_EV_CLASS	BYTE	Класс события и идентификатор, например, B#16#39 – уходящее событие B#16#38 – приходящее событие
OB86_FLT_ID	BYTE	Код ошибки (возможные значения: B#16#C1, B#16#C2, B#16#C3, B#16#C4, B#16#C5, B#16#C6, B#16#C7)
OB86_PRIORITY	BYTE	Класс приоритетов “26” (по умолчанию для режима работы RUN) “28” (режим работы ЗАПУСК (STARTUP))
OB86_OB_NUMBER	BYTE	Номер OB (86)
OB86_RESERVED_1	BYTE	Зарезервировано
OB86_RESERVED_2	BYTE	Зарезервировано
OB86_MDL_ADDR	WORD	Зависит от кода ошибки
OB86_RACKS_FLTD	ARRAY[0..31] BOOL	Зависит от кода ошибки
OB85_DATE_TIME	DT	Дата и время вызова OB

Табл. 5.17 Локальные данные OB86 для программирования, зависящего от кода ошибки

Переменная	Тип данных
OB86_EV_CLASS	BYTE
OB86_FLT_ID	BYTE
OB86_PRIORITY	BYTE
OB86_OB_NUMBER	BYTE
OB86_RESERVED_1	BYTE
OB86_RESERVED_2	BYTE
OB86_MDL_ADDR	WORD
OB86_Z23	DWORD
OB86_DATE_TIME	DT

В таблице 5.18 даются значения кодов ошибок, сообщаемые через переменную OB86_FLT_ID в зависимости от переменной OB86_Z23.

Табл. 5.18 Код ошибки OB86_FLT_ID

Код ошибки OB86_FLT_ID	Значение ошибки
B#16#C1	<p>Неисправность носителя модулей OB86_MDL_ADDR: логический адрес модуля IM. Через переменную OB86_Z23 предоставляется следующая информация: OB86_Z23 содержит для каждого возможного носителя модулей один бит: бит 0: центральное устройство – всегда 0 бит 1: 1-е устройство расширения бит 21: 21-е устройство расширения биты 21-29: всегда 0 бит 30: неисправность носителя модулей в области SIMATIC S5 бит 31: всегда 0</p> <p><u>Примечание:</u> При приходящем событии индицируются носители модулей (соответствующие биты устанавливаются), которые обусловили вызов OB86. При этом носители модулей, вышедшие из строя, не индицируются. При уходящем событии индицируются восстановленные модули.</p>

Табл. 5.18 Продолжение

Код ошибки OB86_FLT_ID	Значение ошибки
B#16#C2	<p>Восстановление носителя модулей с идентификатором: “Выход из строя носителя модулей при отклонении истинной конфигурации от заданной”.</p> <p>OB86_MDL_ADDR: логический адрес модуля IM.</p> <p>Через переменную OB86_Z23 предоставляется следующая детализированная информация:</p> <p>OB86_Z23 содержит для каждого возможного носителя модулей один бит (см. код ошибки B#16#C1). Значение установленного бита:</p> <p>в соответствующем носителе модулей</p> <ul style="list-style-type: none"> - имеются модули с неправильным идентификатором типа; - отсутствуют запроектированные модули; - хотя бы один модуль неисправен.
B#16#C3	<p>Выход из строя DP-Master-системы у децентрализованной периферии. (Приходящее событие посылает код ошибки B#16#C3, уходящее - B#16#C4 и класс события B#16#38. Восстановление DP-станции также запускает OB86.)</p> <p>OB86_MDL_ADDR: логический базовый адрес DP-Master’a.</p> <p>Через переменную OB86_Z23 предоставляется следующая дополнительная детализированная информация:</p> <p>Биты 0-7 : зарезервированы</p> <p>Биты 8-15 : идентификатор (ID) DP-Master-системы</p> <p>Биты 16-31: зарезервированы</p>
B#16#C4 B#16#C5	<p>Выход из строя DP-станции</p> <p>Сбой в DP-станции</p> <p>OB86_MDL_ADDR: логический базовый адрес DP-Master’a.</p> <p>Через переменную OB86_Z23 предоставляется следующая дополнительная детализированная информация:</p> <p>OB86_Z23: Адрес соответствующего DP-Slave’a</p> <p>Биты 0-7 : № DP-станции</p> <p>Биты 8-15 : Идентификатор (ID) DP-Master’a</p> <p>Биты 16-30: Логический базовый адрес у S7-DP-Slave или диагностический адрес у DP-Normslave (стандартных DP-Slave)</p> <p>Бит 31 : I/O – идентификатор (идентификатор ввода/вывода)</p>
B#16#C6	<p>Восстановление носителя модулей, однако ошибка при параметрировании модуля</p> <p>OB86_MDL_ADDR: логический базовый адрес IM.</p> <p>Через переменную OB86_Z23 предоставляется следующая дополнительная детализированная информация:</p> <p>OB86_Z23 содержит для каждого возможного носителя модулей один бит:</p> <p>бит 0: центральное устройство – всегда 0</p> <p>бит 1: 1-е устройство расширения</p> <p>....</p> <p>бит 21: 21-е устройство расширения</p> <p>биты 21-29: всегда 0</p> <p>бит 30: неисправность носителя модулей в области SIMATIC S5</p> <p>бит 31: всегда 0</p> <p>Значение установленного бита:</p> <ul style="list-style-type: none"> - существуют модули с неправильным идентификатором - существуют модули с неправильными или незадаваемыми параметрами

Табл. 5.18 Продолжение

Код ошибки OB86 FLT ID	Значение ошибки
B#16#C7	<p>Восстановление DP-станции, однако ошибка при параметрировании модуля</p> <p>OB86_MDL_ADDR: логический базовый адрес DP-Master'a.</p> <p>Через переменную OB86_Z23 предоставляется следующая дополнительная детализированная информация:</p> <p>OB86_Z23: Адрес соответствующего DP-Slave'a</p> <p>Биты 0-7 : № DP-станции</p> <p>Биты 8-15 : Идентификатор (ID) DP-Master'a</p> <p>Биты 16-30: Логический базовый адрес S7-DP-Slave'a</p> <p>Бит 31 : I/O – идентификатор (идентификатор ввода/вывода)</p>

5.2.10 Ошибка доступа к периферии

Операционная система (ОС) S7-CPU вызывает OB122, когда происходит ошибка при доступе к входным / выходным данным периферийного модуля или DP-Slave. Если внутри пользовательской программы происходит обращение к несуществующему или вышедшему из строя DP-Slave'у, то ОС вызывает OB122. Если OB122 не запрограммирован, CPU переходит в состояние STOP. Таблица 5.19 показывает локальные данные OB122.

Табл. 5.19 Локальные данные OB122

Переменная	Тип данных	Описание
OB122_EV_CLASS	BYTE	Класс события и идентификатор, например, B#16#29
OB122_SW_FLT	BYTE	<p>Код ошибки</p> <p>B#16#42 = (у S7-300) ошибка доступа к периферии на чтение = (у S7-400) ошибка при первом обращении на чтение к периферии после появления ошибки.</p> <p>B#16#43 = (у S7-300) ошибка доступа к периферии на запись = (у S7-400) ошибка при первом обращении на запись к периферии после появления ошибки.</p> <p>B#16#44 = (только у S7-400) ошибка при n-м обращении на чтение (n>1) после появления ошибки</p> <p>B#16#45 = (только у S7-400) ошибка при n-м обращении на запись (n>1) после появления ошибки</p>
OB122_PRIORITY	BYTE	Класс приоритета OB, в котором встретилась ошибка
OB122_OB_NUMBER	BYTE	Номер OB (122)
OB122_BLK_TYPE	BYTE	Тип блока, в котором встретилась ошибка B#16#88=OB; B#16#8A=DB; B#16#8C=FC; B#16#8E=FB
OB122_MEM_AREA	BYTE	Область памяти и тип доступа: биты с 4 по 7 – способ доступа: 0-битовый, 1-байтовый, 2-к слову, 3-к двойному слову. Биты с 0 по 3 – область периферии: 0-область периферии, 1-PI, 2-PIQ.

Табл. 5.19 Продолжение

Переменная	Тип данных	Описание
OB122 _ MEM_ADDR	WORD	Адрес памяти, при обращении к которому произошла ошибка
OB122 _ BLK_NUM	WORD	№ блока с MC7-командой, вызвавшей ошибку
OB122 _ PRG_ADDR	WORD	Относительный адрес MC7-команды, вызвавшей ошибку
OB122 _ DATE_TIME	DT	Дата и время вызова OB

5.3 Функции для обмена DP-пользовательскими данными и функции для сигналов от процесса

5.3.1 Обмен консистентными DP-данными с помощью SFC14 DPRD_DAT и SFC15 DPWR_DAT

Для обмена областями DP- данных, которые имеют *консистентную* (единую) структуру с размером 3 или более 4 байт, нельзя использовать обычные команды обращения к байту, слову или двойному слову (см. также раздел 6.1). В этом случае обмен данными происходит с помощью SFC14 DPRD_DAT и SFC15 DPWR_DAT.

SFC14 DPRD_DAT

Область входных консистентных данных DP-Slave'а читается с помощью вызова SFC14 DPRD_DAT. SFC14 имеет приведенные в таблице 5.20 входные и выходные параметры, которые при вызове должны быть обеспечены соответствующим образом. Если DP-Slave имеет несколько консистентных входных модулей (блоков, областей), то должен для каждого из них использоваться отдельный вызов SEC14.

Табл. 5.20 Параметры для SFC14 DPRD DAT

Параметр	Описание (тип доступа)	Тип данных	Область памяти	Описание
LADDR	INPUT	WORD	I,Q,M,D,L;const	Задание (в 16-ичном формате) спроектированного в HW-Config начального адреса входного модуля DP-Slave
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Возвращаемое значение SFC
RECORD	OUTPUT	ANY	I, Q, M, D, L	Область для прочитанных пользовательских данных

Описание параметров

Параметр RECORD

Параметр RECORD описывает целевую область в S7-CPU для прочитанных из DP-Slave'а консистентных входных данных. Длина данных для параметра RECORD должна соответствовать длине данных входных модулей (областей) DP-Slave, спроектированных в HW-Config.

Внимание В этом параметре в типе данных ANY должен быть только тип BYTE.

Параметр RET_VAL

В таблице 5.21 представлены коды ошибок, возвращаемые в параметре RET_VAL функцией SFC14.

Табл. 5.21 Значения параметра RET_VAL у SFC14 DPRD_DAT

Код ошибки W#16#...	Комментарий
000	Ошибок нет
8090	Для заданного логического базового адреса не спроектирован модуль или ограничение на длину консистентных данных не соблюдено (не учтено)
8092	В параметре RECORD тип данных отличен от BYTE
8093	Для заданного в LADDR логического адреса не существует DP-модуля, из которой Вы можете читать консистентные данные.
80A0	Данный модуль неисправен
80B0	Выход из строя Slave при внешнем DP-подключении.
80B1	Длина целевой области не равна длине полезных данных, спроектированных в HW-Config
808x 80B2 80B3 80C0 80C2 80Fx 87xy	Системная ошибка при внешнем DP-подключении

SFC15 DPWR_DAT

Области выходных консистентных данных DP-Slave записываются с помощью вызова SFC15 DPWR_DAT в DP-Slave. SFC15 имеет входные и выходные параметр, приведенные в таблице 5.22.

Если DP-Slave имеет несколько консистентных выходных блоков, то перенос данных должен осуществляться для каждого такого блока вызовом SFC15.

Табл. 5.22 Параметры для SFC15 DPWR_DAT

Параметр	Описание (тип доступа)	Тип данных	Область памяти	Описание
LADDR	INPUT	WORD	I,Q,M,D,L;const	Задание (в 16-ичном формате) спроектированного в HW-Config начального адреса выходного модуля DP-Slave
RECORD	OUTPUT	ANY	I, Q, M, D, L	Область-источник для записываемых пользовательских данных
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Возвращаемое значение SFC

Описание параметров

Параметр RECORD

Параметр RECORD описывает область-источник для консистентных выходных данных, которые переносятся из S7-CPU в DP-Slave. Задание длины параметра RECORD должно соответствовать спроектированной в HW-Config длине выходного модуля DP-Slave.

Далее нужно обратить внимание, что этот параметр имеет тип ANY и допускает только тип данных BYTE.

Параметр RET_VAL

В таблице 5.23 представлены коды ошибок SFC15, возвращаемые в параметре RET_VAL.

Табл. 5.23 Коды специфических ошибок, возвращаемых в RET_VAL для SFC15 DPWR_DAT

Код ошибки W#16#...	Комментарий
0000	Нет ошибки
8090	Для заданного логического базового адреса модуль не спроектирован или не соблюдено ограничение на длину консистентных данных
8092	В параметре RECORD (тип ANY) указан тип, не совпадающий с BYTE
8093	Для заданного в LADDR логического адреса не существует DP-модуля, в который Вы можете записывать консистентные данные
80A1	Выбранный модуль неисправен
80B0	Неисправный Slave во внешнем DP-подключении
80B1	Длина заданной области-источника не равна спроектированной в HW-Config длине пользовательских данных
80B2	Системная ошибка при внешнем подключении DP
80B3	Системная ошибка при внешнем подключении DP
80C1	Данные, предварительно записанные в модуль, еще не обработаны
808x 80Fx 85xy 80C2	Системная ошибка при внешнем DP- подключении

5.3.2 Управляющие команды SYNC и FREEZE с помощью функции SFC11 DPSYC_FR

С помощью SFC11 DPSYC_FR можно передавать управляющие команды SYNC или FREEZE на один или несколько DP-Slave'ов. Они служат для того, чтобы синхронизировать обмен данными с определенными DP-Slave'ами. Соответствующие Slave'ы при этом объединяются при проектировании соответственно в SYNC-/FREEZE-группы.

Управляющие команды SYNC и FREEZE при этом посылаются на все DP-Slave'ы с помощью глобальных управляющих телеграмм (телеграмм Broadcast).

Управляющая команда SYNC

Управляющая команда SYNC производит “синхронизацию” выходов DP-Slave. Если DP-Slave находится в SYNC-режиме, то выходные данные, передаваемые с помощью Data_Exchange_Telegramm, заносятся в локальный буфер. При получении команды SYNC DP-Slave подключает на выходы данные, сохраненные в локальном буфере. Возможна одновременная активизация (синхронизация) выходных данных нескольких DP-Slave’ов.

Управляющая команда UNSYNC

Управляющая команда UNSYNC отменяет режим SYNC рассматриваемого DP-Slave’a. Таким образом данный DP-Slave вновь находится в циклическом обмене с DP-Master’ом.

Выходные данные, принятые с помощью Data_Exchange_Telegramm, будут немедленно передаваться на выходы DP-Slave.

Управляющая команда FREEZE

Управляющая команда FREEZE производит “замораживание” входов DP-Slave’ов. Если DP-Slave находится в режиме FREEZE, то актуальные данные, поступившие на входы данного DP-Slave’a, будут при приеме команды FREEZE от DP-Master’a сохранены в устройстве памяти DP-Slave’a и, таким образом, заморожены. DP-Master читает с помощью Data_Exchange_Telegramm замороженные данные исключительно из устройства памяти DP-Slave’a. После команды FREEZE актуальные входные данные будут прочитаны и скопированы в устройство памяти DP-Slave’a. DP-Master может читать эти данные до следующей команды FREEZE.

Эта управляющая команда позволяет, таким образом, одновременно (синхронно) принимать актуальные, поступившие на DP-Slave, входные данные.

Управляющая команда UNFREEZE

Управляющая команда UNFREEZE отменяет режим FREEZE на рассматриваемом DP-Slave. Таким образом он опять после этой команды находится в циклическом обмене данными с DP-Master’ом, то есть входные данные DP-Slave’a не запоминаются в “промежуточный” буфер и могут быть немедленно прочитаны DP-Master’ом.

Описание параметров

SFC11 обладает приведенными в таблице 5.24 входными и выходными параметрами.

Табл. 5.24 Параметры SFC11 DPSYC_FR

Параметр	Описание (тип доступа)	Тип данных	Область памяти	Описание
REQ	INPUT	BOOL	I,Q,M,D,L; const	REQ=1 – запуск задания SYNC/FREEZE
LADDR	INPUT	WORD	I,Q,M,D,L; const	Логический базовый адрес DP-Master'a
GROUP	INPUT	BYTE	I,Q,M,D,L; const	Выбор группы: если соответствующий бит равен 0, то группа не выбрана, если равен 1, то выбрана данная группа (см. табл.5.22)
MODE	INPUT	BYTE	I,Q,M,D,L; const	Идентификатор задания (ID) (кодирование согласно EN 50170 том 2) см. табл. 5.24
RET_VAL	OUTPUT	INT	I,Q,M,D,L	Возвращаемое значение SFC
BUSY	OUTPUT	BOOL	I,Q,M,D,L	BUSY=1 означает, что запущенная SFC11 еще не закончилась (задание SYNC/FREEZE еще не закончилось)

Параметр GROUP

DP-Slave'у уже при проектировании с помощью HW-Config должна быть назначена определенная группа.

Параметр GROUP определяет, какие группы должны рассматриваться с помощью SFC11. За одно задание может быть активизировано несколько групп. Значение “0” (все биты в поле) недопустимо. Соответствие битов в параметре GROUP номеру группы описано в таблице 5.25.

Табл. 5.25 Соответствие битов в параметре GROUP номерам групп

№ бита	7	6	5	4	3	2	1	0
GROUP	8	7	6	5	4	3	2	1

Параметр MODE

С помощью параметра MODE назначается и передается управляющая команда для группы. В таблице 5.26 показано назначение битов параметра отдельным управляющим командам.

Табл. 5.26 Параметр MODE функции SFC11 DPSYC FR

№ бита	7	6	5	4	3	2	1	0
MODE			SYNC	UNSYNC	FREEZE	UNFREEZE		

Во время одного вызова SFC11 можно актуализировать и посылать на CPU несколько управляющих команд. Возможные комбинации описаны в табл. 5.27. Таким образом с помощью одного вызова SFC11 можно посылать на DP-Slave несколько управляющих команд.

Табл. 5.27 Возможные комбинации параметра MODE SFC11 DPSYC FR

№ бита	7	6	5	4	3	2	1	0	W#16#...
MODE				UNSYNC					10
				UNSYNC		UNFREEZE			12
				UNSYNC	FREEZE				18
			SYNC						20
			SYNC			UNFREEZE			24
			SYNC		FREEZE				28
						UNFREEZE			4
					FREEZE				8

Параметр RET_VAL

Коды ошибок, возвращаемые в параметре RET_VAL для SFC11 представлены и прокомментированы в таблице 5.28.

Табл. 5.28 Коды ошибок SFC11 DPSYC FR, возвращаемые в RET_VAL

Код ошибки (W#16#...)	Комментарий
0000	Нет ошибок
7000	Первый вызов с REQ="0". SFC11 DPSYC_FR не активна; BUSY имеет значение "0".
7001	Первый вызов с REQ="1". Задание, указанное с помощью LADDR, GROUP и MODE, было запущено; BUSY имеет значение 1.
7002	Промежуточный вызов (REQ не имеет значения). Запущенное задание SYNC/FREEZE все еще активно. BUSY имеет значение 1.
8090	Модуль, выбранный с помощью LADDR, не является Master-устройством DP.
8093	Эта SFC не разрешена для модуля, выбранного с помощью LADDR (конфигурация или версия Master-устройства DP)
8094	Неверный параметр GROUP
8095	Неверный параметр MODE
80B0	Группа, выбранная с помощью GROUP, не конфигурирована
80B1	Группа, выбранная с помощью GROUP, не назначена этому CPU
80B2	Задание SYNC, указанное с помощью MODE, не разрешено в группе, выбранной с помощью GROUP
80B3	Задание FREEZE, указанное с помощью MODE, не разрешено в группе, выбранной с помощью GROUP
80C2	Временная нехватка ресурсов в Master-устройстве DP. Master-устройство DP в настоящее время обрабатывает максимальное для CPU количество заданий
80C3	Это задание SYNC/UNSYNC не может быть активизировано в настоящее время, так как в любой момент времени может быть запущено только одно задание SYNC/UNSYNC. Проверьте Вашу пользовательскую программу.
80C4	Это задание FREEZE /UNFREEZE не может быть активизировано в настоящее время, так как в любой момент времени может быть запущено только одно задание SYNC/UNSYNC. Проверьте Вашу пользовательскую программу.
80C5	Децентрализованная периферия недоступна. Отказ подсистемы DP.
80C6	Задание прервано из-за отключения входов/выходов посредством центрального процессора
80C7	Задание прервано из-за горячего или теплого рестарта в Master-устройстве DP.
8325	Параметр GROUP неверный
8425	Параметр MODE неверный

5.3.3 Запуск сигнала от процесса на DP-Master'е с помощью SFC7 DP_PRAL

Система управления SIMATIC S7-300, которая основывается на CPU 315-2DP и эксплуатируется на DP-шине, как I-Slave, может с помощью SFC7 DP_PRAL запустить сигнал от процесса на DP-Master-системе.

Благодаря вызову SFC7 DP_PRAL из пользовательской программы на станции с CPU 315-2DP, спроектированной как I-Slave, можно запустить на DP-Master'е (только S7-400 и S7-300 с CPU 315-2DP) сигнал от процесса (OB40...OB47). Через входной параметр SFC AL_INFO можно передать идентификатор сигнала, специфический для приложения. Этот идентификатор сигнала передается на DP-Master (переменная OB40_POINT_ADDR) и может быть оценен при обработке сигнального OB (OB40...OB47). Требуемый сигнал от процесса однозначно определяется через входные параметры IOID и LADDR. Таблица 5.29 показывает входные и выходные параметры SFC7.

Табл. 5.29 Параметры SFC7 DP PRAL

Параметр	Тип доступа	Область памяти	Тип данных	Описание
REQ	INPUT	I,Q,M,D,L, const	BOOL	Требование запуска сигнала от процессана DP-master'е (REQ=1 - запуск)
IOID	INPUT	I,Q,M,D,L, const	WORD	Идентификатор адресной области в памяти передачи (с точки зрения DP-Slave'a): B#16#54 = периферийный вход (PI) B#16#55 = периферийный выход (PQ) Идентификатором области, принадлежащей смешанному модулю, является идентификатор младшего из двух адресов. Если адреса одинаковы, то задайте B#16#54
LADDR	INPUT	I,Q,M,D,L, const	WORD	Начальный адрес адресной области в памяти передачи (с точки зрения DP-Slave). Если область принадлежит смешанному модулю, то задайте младший из 2-х адресов.
AL_INFO	INPUT	I,Q,M,D,L, const	WORD	Идентификатор прерывания. Он передается OB40, который запускается в DP-Master'е (переменная OB40_POINT_ADDR)
RET_VAL	OUTPUT	I,Q,M,D,L	INT	Возвращаемое значение
BUSY	OUTPUT	I,Q,M,D,L	BOOL	BUSY=1: запущенное аппаратное прерывание еще не было подтверждено DP-Master'ом.

SFC7 DP_PRAL обрабатывается *асинхронно*, это значит обработка распространяется на *несколько* вызовов SFC. Задание заканчивается, если сигнал от процесса после полной обработки соответствующего OB (OB40...OB47) будет квитирован DP-Master'ом.

Если CPU 315-2DP эксплуатируется как стандартный DP-Slave (DP-Normslave), то вызов SFC7 заканчивается, как только диагностический сигнал попадет на DP-Master. Таблица 5.30 показывает возможные коды ошибок SFC7, которые возвращаются в параметре RET_VAL.

Табл. 5.30 Специфические значения, возвращаемые параметром RET_VAL у SFC7 DP_PRAL

Коды ошибок W#16#...	Комментарий
0000	Задание выполнено без ошибок
7000	Первый вызов с REQ=0. Запрос на аппаратное прерывание не активен; BUSY имеет значение 0.
7001	Первый вызов с REQ=1. Запрос на аппаратное прерывание уже передан DP-Master'у. BUSY имеет значение 1.
7002	Промежуточный вызов (REQ не имеет значения) запускаемое аппаратное прерывание еще не было подтверждено DP-Master'ом. BUSY=1.
8090	Неправильный начальный адрес адресной области в памяти передачи
8091	Прерывание заблокировано (блокировка конфигурированная пользователем)
8093	Параметры IOID и LADDR адресуют модуль, который не способен к запросу аппаратного прерывания
80C6	Децентрализованная периферия в настоящий момент недоступна

5.4 Диагностика DP с помощью SFC

5.4.1 Чтение стандартных диагностических данных DP-Slave с помощью SFC13 DPNRM_DG

DP-Slave'ы предоставляют в распоряжение для распознавания и локализации ошибок диагностические данные. Принципиальная структура диагностических данных DP-Slave'ов установлена в стандарте EN 50170, том 2 PROFIBUS и представлена в таблице 5.31.

Табл. 5.31 Принципиальная структура диагностики DP-Slave

Байт	Значение
0	Состояние станции 1
1	Состояние станции 2
2	Состояние станции 3
3	Адрес PROFIBUS DP-Master'a
4	Идентификатор изготовителя (старший байт)
5	Идентификатор изготовителя (младший байт)
6...	Дополнительная диагностическая информация, относящаяся к данному DP-Slave'у

Дальнейшая детальная информация о диагностических данных содержится в разделе 7 “Диагностические функции”.

Считывание диагностических данных DP-Slave возможно с помощью SFC13 DPNRM_DG. SFC13 имеет входные и выходные параметры, приведенные в таблице 5.32.

Табл. 5.32 Параметры SFC13 DPNRM_DG

Параметр	Тип доступа	Тип данных	Область памяти	Описание
REQ	INPUT	BOOL	I,Q,M,D,L, const	Запрос на чтение (вызов чтения)
LADDR	INPUT	WORD	I,Q,M,D,L, const	Спроектированный в HW-Config диагностический адрес DP-Slave (в 16-ичной форме)
RET_VAL	OUTPUT	INT	I,Q,M,D,L	Возвращаемое значение SFC (Сообщение об ошибке или длина прочитанных данных в байтах)
RECORD	OUTPUT	ANY	I,Q,M,D,L	Целевая область для прочитанных диагностических данных
BUSY	OUTPUT	BOOL	I,Q,M,D,L	BUSY="1": процесс чтения не окончен

Обработка SFC13 происходит *асинхронно*, то есть выполнение функции может продолжаться несколько вызовов SFC и, таким образом, несколько циклов CPU.

Коды специфических ошибок SFC13, которые сообщаются с помощью параметра RET_VAL, являются составной частью кодов ошибок SFC59 (см. разделы 5.5.4 и 5.5.5).

Описание параметров

Параметр RECORD

Параметр RECORD описывает целевую область в CPU для прочитанных диагностических данных DP-Slave'a. Он имеет тип ANY и допускает в качестве типа данных только BYTE. (Например, P#M10.0 BYTE 15).

Если число байтов диагностических данных, которые читаются из DP-Slave, больше, чем в заданной целевой области, то диагностические данные будут отброшены и будет выдан соответствующий код ошибки с помощью параметра RET_VAL. Если длина читаемых данных меньше или равна длине, заданной в параметре RECORD, то диагностические данные будут перенесены в целевую область и в параметре RET_VAL будут сообщено действительное число прочитанных байтов. Минимальная длина читаемых диагностических данных составляет 6 байт, максимальная длина – 240 байт. Если DP-Slave имеет больше, чем 240 байт диагностических данных (допустимо до 244 байт) и если в параметре RECORD зарезервирована целевая область этой длины, то будут в целевую область перенесены первые 240 байт и установлен бит “Overflow” (OV). Если DP-Slave дает больше, чем 240 байт диагностических данных и если длина, заданная в параметре RECORD меньше, чем 240 байт, то диагностическая телеграмма будет отброшена.

Системные ресурсы для SFC13 в системе SIMATIC S7-400

При новом вызове SFC13 DPNRM_DG для асинхронного задания в S7-CPU-400 занимают ресурсы (место в памяти). При нескольких одновременно активных заданиях гарантируется, что все задания будут выполнены и не будет их влияния друг на друга. Однако одновременно может быть активно только определенное число SFC13-заданий. Максимальное число возможных SFC-заданий берется из технических данных соответствующего S7-CPU-400. Если

граница максимального числа ресурсов, занятых SFC-заданиями, превышена, то через параметр RET_VAL выдается соответствующий код ошибки. В этом случае SFC должна быть запущена снова.

5.4.2 Прием сигнала от DP-Slave'a с помощью SFB54 RALRM

SFB54 RALRM принимает сигнал с информацией от блока, соответственно, модуля DP-Slave'a.

Информация для выходных параметров SFB54 предоставляется в распоряжение от источника сигнала, которая находится в стартовой информации OB, а также от посылающего сигнал DP-Slave'a.

SFB54 должен вызываться только внутри OB сигнала, который операционная система CPU запускает на основании события, посылаемого от DP-Slave'a.

Интерфейс SFB54 RALRM идентичен определенному в стандарте PNO АК 1131 FB RALRM. Входные и выходные параметры SFB54 представлены в табл. 5.33.

Табл. 5.33 Параметры SFB54 RALRM

Параметр	Объявлен.	Тип	Обл. памяти	Описание
MODE	INPUT	INT	I,Q,M,D,L,const	Режим работы SFB54
F_ID	INPUT	DWORD	I,Q,M,D,L,const	Указывается (в 16-ичной форме) начальный адрес модуля DP-Slave, спроектированный в HW-Config, от которого ожидается сигнал
MLEN	INPUT	INT	I,Q,M,D,L,const	Максимальная длина принимаемой информации в байтах.
NEW	OUTPUT	BOOL	I,Q,M,D,L	TRUE=новый сигнал принят
STATUS	OUTPUT	DWORD	I,Q,M,D,L	Код ошибки SFB или DP-Master'a
ID	OUTPUT	DWORD	I,Q,M,D,L	Логический начальный адрес блока, посылающего сигнал
LEN	OUTPUT	INT	I,Q,M,D,L	Длина принятой информации
TINFO	OUTPUT	ANY	I,Q,M,D,L	<i>task information</i> Область для стартовой информации OB и управляющей информации
AINFO	OUTPUT	ANY	I,Q,M,D,L	<i>alarm information</i> Заголовочная информация и дополнительная информация сигнала

Описание параметров

Параметр MODE

SFB54 может вызываться в различных рабочих режимах (MODE). В таблице 5.34 описаны различные значения параметра MODE.

Табл. 5.34 Структура управляющей информации

MODE	Значение
0	SFB54 показывает блок/модуль, пославший сигнал в выходном параметре ID и возвращает в выходном параметре NEW значение TRUE
1	SFB54 описывает все выходные параметры, независимо от посылающего прерывание блока/модуля.
2	SFB54 проверяет, послал ли прерывание компонент, указанный во входном параметре F_ID. <ul style="list-style-type: none"> Если да: параметр NEW получает значение TRUE и все другие выходные параметры получают соответствующие значения. Если нет: параметр NEW получает значение FALSE

Параметр TINFO

Параметр TINFO задает целевую область для стартовой информации ОВ и управляющей информации. Если целевая область выбрана слишком малой, то SFB54 не может записать всю информацию.

Стартовая информация ОВ, в котором вызвана SFB54, заносится в байты с 0 до 19, управляющая информация заносится в байты с 20 по 27.

Структура управляющей информации для прерывания от децентрализованного блока представлена в таблице 5.35.

Табл. 5.35 Структура управляющей информации

№ байта в TINFO	Тип данных	Значение
20	BYTE	Идентификатор DP-Master System (возможные значения от 1 до 255)
21	BYTE	Адрес DP-Slave'a
22	BYTE	Биты с 0 до 3: тип Slave'a: 0000=DP-Slave 0001=DPS7-Slave 0010=DPS7V1-Slave 0011=DPV1-Slave с 0100 = зарезервировано Биты с 4 по 7: тип профиля 0000=DP с 0001= зарезервировано
23	BYTE	Биты с 0 по 3: тип информационного прерывания 0000 = прозрачное (Transparent) (Прерывание происходит от спроектированного децентрализованного модуля) 0001 = представитель (Прерывание исходит не от DPV1-Slave'a, соотв., не от спроектированного слота) 0010 = прерывание генерируется в CPU с 0011 = зарезервировано Биты с 4 по 7: версия структуры 0000 = инициализирующее значение с 0001 = зарезервировано
24	BYTE	Флаги интерфейса DP-Master Бит 0=0 : прерывание от встроенного DP-Master'a Бит 0=1: прерывание от внешнего DP-Master'a Биты с 1 по 7 - зарезервированы
25	BYTE	Флаги интерфейса DP-Slave'a Бит 0: EXT_DIAG_Bit из диагностической телеграммы (см. п 7.3.1: бит 3 первого байта) или 0, или этот бит при прерывании не существует Биты с 1 по 7 = зарезервированы
26 - 27	WORD	PROFIBUS-идентификационный номер Slave'a

Параметр AINFO

Параметр AINFO определяет целевую область для телеграммы заголовка и дополнительной информации прерывания. Если целевая область выбрана слишком малой, то SFB54 может не внести всю информацию. Поэтому длина AINFO должна задаваться по меньшей мере MLEN байт.

Информация заголовка вносится в байты с 0 по 3, дополнительная информация прерывания – с байта 4 до максимум 63 байта (для прерываний от децентрализованной периферии.)

Структура информации заголовка для прерывания от децентрализованного блока представлена в таблице 5.36.

Табл. 5.36 Структура информации заголовка

№ байта в AINFO	Тип данных	Значение
0	BYTE	Длина принятой информации прерывания в байтах (с 4 по 63)
1	BYTE	Идентификатор для типа прерывания 1: Диагностическое прерывание 2: Прерывание от процесса 3: Прерывание удаления 4: Прерывание вставки 5: Прерывание состояния 6: Прерывание модернизации 31: Выход из строя прибора расширения DP-Master System'ы или DP-станции с 32 по 126: Прерывание, специфическое для производителя
2	BYTE	№ слота компонента, пославшего прерывание
3	BYTE	Спецификатор 0: нет дальнейшей информации 1: приходящее событие, слот неисправен 2: уходящее событие, слот восстановлен (больше не неисправен) 3: уходящее событие, слот все еще неисправен

Целевые области TINFO и AINFO

В зависимости от ОВ, в котором вызывается SFB54, целевые области TINFO и AINFO будут заполнены только частично. Какая информация будет вноситься, определено в таблице 5.37

Табл. 5.37 Доступность информации прерывания

Прерывание	ОВ	TINFO старт. информ. ОВ	TINFO управл. информ	AINFO информ. заголовок	AINFO доп. информ. прерывания
от процесса	4х	да	да	да	Как поставляется DP-Slave'ом
состояния	55	да	да	да	да
модернизации	56	да	да	да	да
зависящее от производит.	57	да	да	да	да
диагностическое	82	да	да	да	Как поставляется DP-Slave'ом
установки/съем	83	да	да	да	Как поставляется DP-Slave'ом
выхода из строя станции	86	да	да	нет	нет

Параметр STATUS

Выходной параметр STATUS содержит информацию об ошибке. Если его интерпретировать как ARRAY[1..4] OF BYTE, то информация об ошибке содержится в структуре, представленной в таблице 5.38.

Табл. 5.38 Представление выходного параметра STATUS

Элемент массива	Имя	Значение
STATUS[1]	Function_Num	<ul style="list-style-type: none"> В#16#00, в случае отсутствия ошибки Идентификатор функции из DPV1-PDU: в случае ошибки используется В#16#80. В случае отсутствия элемента протокола DPV1 используется В#16#C0.
STATUS[2]	Error Decode	Местоположение идентификатора ошибки
STATUS[3]	Error Code 1	Идентификатор ошибки
STATUS[4]	Error_Code_2	Расширение идентификатора ошибки, специфическое для изготовителя

Местоположение идентификатора ошибки, вносимое в STATUS[2], представлено в таблице 5.39.

Идентификатор ошибки из STATUS[3] представлен в таблице 5.40.

Код ошибки в STATUS[4] при DPV1-ошибке подается от DPV1-Master'а к CPU. Если нет DPV1-ошибки, то там содержится значение "0".

Табл. 5.39 Идентификатор ошибки в STATUS[2]

Error Decode (В#16#...)	Источник	Значение
с 00 по 7F	CPU	Нет ошибок или нет предупреждений
80	DPV1	Ошибки по IEC 61158-6
с 81 по 8F	CPU	В#16#8 показывает ошибку в X-м (8X) вызываемом параметре
FE, FF	DP-профиль	Ошибка, специфическая для производителя.

Табл. 5.40 Идентификатор ошибки в STATUS[3]

Error Decode (В#16#...)	Error_Code_1 (В#16#...)	Комментарий согласно DPV1	Значение
00	00		Нет ошибок, нет предупреждений
70	00	reserved, reject (зарезервировано, отклоняется)	Первый вызов, передача данных не активна
70	01	reserved, reject (зарезервировано, отклоняется)	Первый вызов, передача данных запускается
70	02	reserved, reject (зарезервировано, отклоняется)	Промежуточный вызов, передача данных активна
80	90	reserved, pass	Логический начальный адрес недействителен
80	92	reserved, pass	Недопустимый тип в указателе ANY
80	93	reserved, pass	DP-компонент, адресованный посредством ID, соотв., F-ID, не конфигурирован
80	A0	read error	Негативное квитирование при чтении из блока
80	A1	write error	Негативное квитирование при записи в блок
80	A2	module failure	Ошибка DP-протокола на уровне 2, возможно дефект аппаратуры
80	A3	reserved, pass	Ошибка DP-протокола при Direct-Data-Link-Mapper или User-Interface/User, возможно дефект аппаратуры
80	A4	reserved, pass	Нарушена коммуникация на С-шине

Табл. 5.40 Идентификатор ошибки в STATUS[3] (Продолжение)

Error_Decode (B#16#...)	Error_Code_1 (B#16#...)	Комментарий согласно DPV1	Значение
80	A5	reserved, pass	—
80	A7	reserved, pass	Заняты DP-ресурсы
80	A8	Version conflict	Конфликт версий
80	A9	feature not supported	Свойства не поддерживаются
80	от AA до AF	user specific	Специфические для DP-Master'a
80	B0	invalid index	Модуль (блок) не знает записи данных: запись данных с номером 256 не допустима
80	B1	write length error	Ошибка длины в AINFO
80	B3	type conflict	Существующий блок отличен от заданного
80	B4	invalid area	Недействительная область
80	B5	state conflict	Конфликт состояния
80	B6	access denied	Отказано в доступе
80	B7	invalid range	Недопустимая область
80	B8	invalid parameter	Недопустимый параметр
80	B9	invalid type	Недопустимый тип
80	от BA до BF	user specific	Специфические для DP-Master'a
80	C0	read constrain conflict	Блок ведет чтение набора данных, но еще не существует данных для чтения
80	C1	write constrain conflict	Данные предшествующего задания на запись для того же набора данных еще не обработаны блоком
80	C2	resource busy	В данный момент модуль обрабатывает максимально возможное количество заданий для CPU
80	C3	resource unavailable	Необходимые ресурсы в данный момент заняты
80	Dx	user specific	Специфические для DP-Slave'a (см. описание DP-Slave'a)
81	от 00 до FF		Ошибка в первом вызываемом параметре (у SFB54: MODE)
81	00		Недопустимый режим работы
82	от 00 до FF		Ошибка во втором вызываемом параметре
↓		↓	↓
88	от 00 до FF		Ошибка в 8-ом вызываемом параметре
88	01		Неверный синтаксис
88	23		Превышение границ или целевая область слишком мала
88	24		Идентификатор области неверный
88	32		№ DB/DI вне пользовательской области
88	3A		№ DB/DI=NULL в идентификаторе области или заданный DB/DI не существует
8A	от 00 до FF		Ошибка в 10-ом вызываемом параметре
↓		↓	↓
8F	от 00 до FF		Ошибка в 15-ом вызываемом параметре
FE, FF	от 00 до FF		Ошибки, специфические для производителя

5.4.3 Важный для распределенной периферии список состояний системы (SZL – нем., SSL – англ.)

Список состояний системы (SZL) описывает актуальные состояния системы автоматизации (информационная функция). SZL – *виртуальный* список, то есть он собирается для системы и выдается только по требованию.

Список состояний системы содержит информацию о:

- *Системных данных*
Системные данные – постоянные и параметрируемые данные CPU. Они описывают структуру CPU, состояние классов приоритетов, а также коммуникации.
- *Диагностических данных состояния CPU*
Диагностические данные состояния описывают актуальные состояния всех компонентов, которые могут контролироваться через системную диагностику.
- *Диагностических данных модулей*
Назначенные CPU модули со способностью к диагностике, обладают диагностическими данными, которые хранятся непосредственно в модуле.
- *Диагностическом буфере*
Диагностический буфер содержит диагностические записи в порядке их появления.

5.4.4 Структура подписка SZL

Подписка SZL всегда состоит из заголовка и собственно требуемых записей данных.

Заголовок подписка содержит SZL-ID (идентификатор SZL), индекс, длину данных в байтах и количество записей данных, содержащихся в этом подписке. Запись данных подписка имеет определенную длину. Она зависит от информации, содержащейся в подписке.

5.4.5 Чтение подписка SZL с помощью SFC51 RDSYSST

С помощью SFC51 RDSYSST (ReaD SYStem STatus) может быть прочитан список состояний системы или частичный список состояний системы. Параметры SFC51 SZL_ID (SSL_ID – англ.) и INDEX определяют при этом, какой частичный список должен быть прочитан. Параметры вызова SFC51 RDSYSST приведены в таблице 5.41.

Табл. 5.41 Параметры SFC51 RDSYSST

Параметр	Тип доступа	Тип данных	Область памяти	Описание
REQ	INPUT	BOOL	I,Q,M,D,L, const	REQ="1": запуск обработки
SZL_ID	INPUT	WORD	I,Q,M,D,L, const	Идентификатор списка состояний системы или частичного списка, который надо читать
INDEX	INPUT	WORD	I,Q,M,D,L, const	Тип или номер объекта в частичном списке
RET_VAL	OUTPUT	INT	I,Q,M,D,L	Возвращаемое значение SFC
BUSY	OUTPUT	BOOL	I,Q,M,D,L	BUSY="1": чтение еще завершилось
SZL_HEADER	OUTPUT	STRUCT	I,Q,M,D,L	См. описание параметра SZL_HEADER
DR	OUTPUT	ANY	I,Q,M,D,L	Целевая область для прочитанного списка SZL или прочитанного частичного списка SZL: <ul style="list-style-type: none"> Если Вы считали информацию только заголовка списка SZL, то Вы должны оценивать не DR, только SZL_HEADER В противном случае LENGTHDR*N_DR=число байтов, выведенных в DR

Описание параметров

Параметр SZL_ID

Каждый SZL-подпись внутри SZL имеет свой номер (SZL_ID). Рисунок 5.3 показывает структуру SZL_ID.



Рис. 5.3 Структура SZL_ID

Возможные SZL_ID приведены в разделе 5.4.4 в таблице 5.33. С помощью указания ID можно вызвать или полный подписк или фрагмент подписка. Возможные фрагменты подписков твердо определены и тоже характеризуются номером. Номер фрагмента подписка и его значение зависят от запрошенного частичного списка.

SZL_ID содержит первые 4 бита, как идентификатор класса модуля. Этим задается тип модуля, из которого должен быть прочитан частичный список или, соответственно, фрагмент частичного подписка. Пример для идентификаторов классов модулей приведен в таблице 5.42.

Табл. 5.42 Идентификаторы классов модулей

Идентификатор (двоичный)	Тип модуля (класс модуля)
0000	CPU
1000	FM
1100	CP
0100	IM
0101	Аналоговые модули
1111	Цифровые модули

SZL_ID состоит также из номера подписка и номера фрагмента подписка.

Параметр INDEX

Параметр INDEX используется, если для определенного подписка, соответственно, фрагмента подписка, необходимо задание идентификатора типа объекта или номер объекта. Если SFC не нуждается в этой информации, то его содержание не важно.

Параметр SZL_HEADER

Параметр SZL_HEADER имеет следующую структуру

```

SZL_HEADER : STRUCT
    LENGTHDR: WORD
    N_DR      : WORD
END_STRUCT

```

При этом в поле LENGTHDR считывается длина записи данных в байтах, в поле N_DR – число записей данных в области прочитанных записей данных (число прочитанных записей данных).

Параметр RET_VAL

Параметр RET_VAL возвращает коды ошибок, описанные в таблице 5.43.

Табл. 5.43 Коды ошибок, возвращаемые параметром RET_VAL
SFC51 RDSYSST

Код ошибки W#16#...	Коментарий
0000	Нет ошибок
0081	Поле результата слишком коротко (тем не менее передается столько записей, сколько возможно. Заголовок SZL указывает это количество)
7000	Первый вызов с REQ=0: передача данных не активна, BUSY имеет значение 0.
7001	Первый вызов с REQ=0: передача данных завершена, BUSY имеет значение 1.
7002	Промежуточный вызов (REQ не имеет значения): передача данных уже активна; BUSY=1.

Табл. 5.43 Продолжение

Код ошибки W#16#...	Коментарий
8081	Поле результата слишком коротко (недостаточно места для одной записи данных)
8082	SZL_ID неверен или неизвестен в CPU или SFC
8083	Неправильный или неразрешенный INDEX
8085	Из-за проблемы в системе информация в настоящее время недоступна (напр., из-за недостатка ресурсов)
8086	Запись данных не может быть прочитана из-за ошибки системы (шина, модули, OS)
8087	Запись данных не может быть прочитана, потому что модуль не существует или не квитирует
8088	Запись данных не может быть прочитана, потому что фактический идентификатор типа отличается от ожидаемого
8089	Запись данных не может быть прочитана, так как модуль не обладает диагностическими свойствами
808A	Тип данных для параметра DR недопустим (допустимые типы данных BOOL, BYTE, CHAR, WORD, DWORD, INT, DINT) или битовый адрес не равен 0.
80A2	Ошибка протокола DP (ошибка уровня 2). (Временная ошибка)
80A3	Ошибка протокола DP у пользовательского интерфейса/пользователя (Временная ошибка)
80A4	Ошибка связи в коммуникационной шине (ошибка между CPU и внешним интерфейсным модулем DP).
80C5	Децентрализованная периферия недоступна. (Временная ошибка)
80C6	Передача записи данных прекратилась из-за прерывания класса приоритета.

5.4.6 Имеющиеся в распоряжении SZL-подписки

В таблице 5.44 даются SZL-подписки, которые можно выбрать. В модуле всегда предоставляется в распоряжение набор из возможных подписков, зависящий от типа модуля.

Табл. 5.44 Имеющиеся в распоряжении SZL-подписки

Подсписок	SZL-ID (W#16#...)
Список всех SZL-ID модуля	xy00
Идентификация модуля	xy11
Характеристики CPU	xy12
Области памяти пользователя	xy13
Системные области	xy14
Типы модулей	xy15
Классы приоритетов	xy16
Список доступных SDB с № < 1000	xy17
Максимальное расширение периферии у S7-300	xy18
Состояние светодиодов модулей	xy19
Соотношение ошибок, сигналов (alarm'ов)	xy21
Состояние сигналов (Alarmstatus)	xy22
Классы приоритетов	xy23
Состояние функционирования	xy24
Коммуникации: характеристики	xy31

Табл. 5.44 Продолжение

Подсписок	SZL-ID (W#16#...)
Коммуникации: данные о состоянии	xy32
Список диагностики участников	xy33
Список стартовой информации	xy81
Список стартовых событий	xy82
Информация о состоянии модулей	xy91
Информация о состоянии модулей/станций	xy92
Диагностический буфер в CPU	xyA0
Диагностическая информация модулей (DS0)	00B1
Диагностическая информация модулей (DS1) через географический адрес	00B2
Диагностическая информация модулей (DS1) через логический адрес	00B3
Диагностические данные DP-Slave'a	00B4

5.4.7 Особенности SFC51 RDSYSST

SFC51 обрабатывается в нормальном случае асинхронно. При вызове SFC51 в OB82 (OB диагностики) с SZL-ID W#16#00B2 или W#16#00B3 и заданным адресом модуля в параметре INDEX, который вызвал диагностический сигнал, SFC51 выполняется немедленно, то есть синхронно.

Для каждого асинхронно работающего SFC51 в CPU занимают ресурсы (место в памяти), которые используются для выполнения задания. Для нескольких “одновременно” активных заданий гарантируется, что все они будут выполнены и не будет их влияния друг на друга.

“Одновременно”, однако, может быть активным определенное число вызовов SFC51. Максимальное число возможных SFC-вызовов можно взять из технических характеристик соответствующего CPU. Если граница максимального числа предоставленных в распоряжение ресурсов превышена, то в параметре RET_VAL выдается соответствующий код ошибки. В этом случае SFC должен быть вызван снова. Всегда можно считать одним вызовом SFC51 RDSYSST одновременно только один подсписок.

5.5 Запись и чтение записей данных / параметров

5.5.1 Запись динамических параметров с помощью SFC55 WR_PARM

С помощью SFC55 WR_PARM (Write PARaMeter) переносится запись данных RECORD в адресуемый S7-модуль. Параметры, которые с помощью SFC55 переносятся в модуль, не переписывают параметры этого модуля, которые сохраняются в системных блоках данных (SDB) CPU.

Передаваемые записи данных не могут быть статическими записями данных (например, SDB0). Если отмечены записи, сохраняемые в SDB100 – SDB129, “статический бит” не может быть установлен.

Параметры вызова SFC55 WR_PARM представлены в таблице 5.45

Табл. 5.45 Параметры SFC55 WR_PARM

Параметр	Тип доступа	Тип данных	Область памяти	Описание
REQ	INPUT	BOOL	I,Q,M,D,L, const	REQ="1": Требование на запись
IOID	INPUT	BYTE	I,Q,M,D,L, const	Идентификатор адресуемой области: B#16#54 – периферийные входы B#16#55 – периферийные выходы Если адреса входов и выходов совпадают – то B#16#54
LADDR	INPUT	WORD	I,Q,M,D,L, const	Заданный в HW-Config логический адрес для данного модуля (в 16-ичном формате)
RECNUM	INPUT	BYTE	I,Q,M,D,L, const	Номер записи данных
RECORD	INPUT	ANY	I,Q,M,D,L	Запись данных
RET_VAL	OUTPUT	INT	I,Q,M,D,L	Возвращаемое значение SFC
BUSY	OUTPUT	BOOL	I,Q,M,D,L	BUSY="1": процесс записи еще не завершен

Описание параметров

Параметр IOID

Параметр задает идентификатор адресной области модуля, которая адресуется с помощью LADDR. Если речь идет о смешанном модуле, например, о модуле со входами и выходами, то идентификатор области задается для младшего периферийного адреса. Если адреса входов и выходов совпадают, то в этом случае задается идентификатор для периферийных входов – B#16#54.

Параметр LADDR

Заданный в HW-Config логический адрес для данного модуля (в 16-ичном формате). Если рассматривается смешанный модуль, то задается наименьший из двух адресов.

Параметр RECORD

С помощью параметра RECORD с типом данных ANY задается передаваемая в CPU динамическая запись данных.

Соответствующая запись данных считывается при первом вызове SFC. Если передача записи данных длится дольше, чем цикл CPU, то при последующих вызовах CPU содержание параметра RECORD неважно.

Параметр RET_VAL

Через выходной параметр RET_VAL индицируется успешная или ошибочная обработка SFC55. *Возможные коды ошибок одинаковы для SFC55 и SFC56.*

В информации об ошибках переноса данных (коды ошибок W#16#8xyz), которые не обусловлены ошибочной параметризацией входных и выходных параметров SFC, различают два случая:

- Временные ошибки (коды ошибок W#16#80A2 –80A4 и 80Cx):

Эти ошибки устраняются новым вызовом SFC. Пример временной ошибки – сообщение “W#16#80C3”, которое означает, что необходимый ресурс (память) в момент вызова занят другой функцией.

- Постоянные ошибки (коды ошибок W#16#809x, 80A1, 80Bx, 80Dx):

Постоянные ошибки должны устраняться. Новый вызов SFC целесообразен, если сообщенные ошибки устранены. Пример для постоянной ошибки - неверное задание длины параметра RECORD (W#16#80B1).

Таблица 5.46 показывает специальные коды ошибок для SFC55, SFC56 и SFC57.

Табл. 5.46 Специальные коды ошибок для SFC55, SFC56 и SFC57.

Код ошибки W#16#...	Коментарий	Ограничения
7000	Первый вызов с REQ="0": передача данных не активна. BUSY имеет значение "0".	-
7001	Первый вызов с REQ="1": передача данных запускается. BUSY имеет значение "1".	Децентрализованная периферия
7002	Второй вызов (значение REQ не важно): передача данных уже активна; BUSY имеет значение "1".	Децентрализованная периферия
8090	Заданный логический адрес недействителен: не существует сопоставления в SDB1/SDB2x или нет базового адреса	-
8092	В параметре с типом данных ANY задан тип, отличный от BYTE.	Только в S7-400 для SFC55 "WR_PARM"
8093	Для выбранного посредством LADDR и IOID модуля эти SFC недопустимы (допустимы модули S7-300 – у S7-300, модули S7-400 – у S7-400, модули S7-DP у S7-300 и S7-400)	-
80A2	Ошибка DP-протокола на уровне 2 (Layer 2); возможно дефект аппаратуры.	Децентрализованная периферия
80A3	Ошибка DP-протокола при Direct-Data-Link-Mapper или в User-Interface; возможно дефект аппаратуры.	Децентрализованная периферия
80A4	Коммуникационная шина (K-Bus) неисправна	Ошибка встречается между CPU и внешним DP-подключением
80B0	SFC невозможна для данного типа модуля, так как модуль не распознает запись данных	-
80B1	Неправильная длина передаваемой записи данных	-
80B2	Спроектированный слот не занят	-
80B3	Фактический тип модуля не совпадает с заданным типом модуля в SDB1	-
80C1	Данные предшествующего задания на запись в модуль для того же самого набора данных еще не обработаны модулем	-

Табл. 5.46 Продолжение

Код ошибки W#16#...	Коментарий	Ограничения
80C2	В данный момент времени модуль модуль обрабатывает максимально возможное для данного CPU число заданий	-
80C3	В данный момент времени ресурсы (память и т.п.) заняты	-
80C4	Коммуникационная ошибка: - ошибка четности - SW-Ready не установлено - ошибка в приводимой длине блока - ошибка контрольной суммы на стороне CPU - ошибка контрольной суммы на стороне модуля	-
80C5	Децентрализованная периферия недоступна	Децентрализованная периферия
80C6	Передача записи данных прервана операционной системой для обработки программы с более высоким приоритетом	Децентрализованная периферия
80D0	В соответствующем SDB нет записи для модуля	-
80D1	Номер записи данных в принадлежащем SDB для модуля не спроектирован (номер записи данных 241 отклоняются STEP 7)	-
80D2	Модуль является непараметризуемым согласно идентификатору типа	-
80D3	Нельзя получить доступ к SDB, так как он не существует	-
80D4	Внутренняя ошибка структуры SDB: внутренний указатель SDB указывает за пределы SDB.	Только у S7-300
80D5	Набор данных является статическим	Только у SFC55 WR_PARM

Указание для вызова SFC55 в системе S7-400:

- Код общей ошибки W#16#8544 показывает, что доступ к младшему байту в записи данных области памяти входов/выходов был закрыт. Передача данных, однако, продолжается.
- SFC55 – SFC59 могут также возвращать код ошибки W#16#80Fx. Это говорит о том, что появившаяся ошибка не точно локализована.

5.5.2 Запись предварительно определенных записей данных/параметров из SDB с помощью SFC56 WR_DPARM

С помощью SFC56 WR_DPARM (Write Default PARaMeter) переносятся статические или динамические записи данных с номером из RECNUM из SDB (SDB100 – SDB103 у S7-300, SDB100 – SDB129 у S7-400) S7-CPU в модуль, адресованный через параметры LADDR и IOID. Таблица 5.47 показывает входные и выходные параметры SFC56 WR_DPARM.

Табл. 5.47 Параметры SFC56 WR_DPARM

Параметры	Тип доступа	Тип данных	Область памяти	Описание
REQ	INPUT	BOOL	I,Q,M,D,L, const	REQ="1": требование на задание

Табл. 5.47 Продолжение

Параметры	Тип доступа	Тип данных	Область памяти	Описание
IOID	INPUT	BYTE	I,Q,M,D,L, const	Идентификатор адресуемой области: B#16#54=периферийные входы B#16#55= периферийные выходы
LADDR	INPUT	WORD	I,Q,M,D,L, const	Заданный в HW-Config логический адрес для данного модуля (в 16-ичном формате)
RECNUM	INPUT	BYTE	I,Q,M,D,L, const	Номер записи данных
RET_VAL	OUTPUT	INT	I,Q,M,D,L	Возвращаемое значение SFC
BUSY	OUTPUT	BOOL	I,Q,M,D,L	BUSY="1": процесс записи не закончен

Описание параметров**Параметр IOID**

Параметр задает идентификатор области памяти, адресуемой с помощью LADDR. Если речь идет о смешанном модуле, как, например, модуль со входами и выходами, то идентификатор задается для младшего периферийного адреса. Если адреса входов и выходов совпадают, то в этом случае задается идентификатор входов B#16#54.

Параметр LADDR

Заданный в HW-Config логический адрес для данного модуля (в 16-ичном формате). Если рассматривается смешанный модуль, то задается меньший из двух адресов.

Параметр RET_VAL

В параметре RET_VAL возвращается код ошибки. Возможные коды ошибок SFC56 соответствуют значениям RET_VAL для SFC55, приведенным в таблице 5.46.

5.5.3 Запись всех предопределенных записей данных/параметров из SDB с помощью SFC57 PARM_MOD

С помощью SFC57 PARM_MOD (PARaMetrire MODule) можно передавать все статические или динамические записи данных модуля, спроектированного в HW-Config в соответствующие SDB (SDB100 – SDB103 для S7-300 и SDB100 – SDB129 для S7-400) для адресованного модуля. Таблица 5.48 показывает входные и выходные параметры SFC57 PARM_MOD.

Табл. 5.48 Параметры SFC57 PARM_MOD

Параметр	Тип доступа	Тип данных	Область памяти	Описание
REQ	INPUT	BOOL	I,Q,M,D,L, const	REQ="1": требование на запись
IOID	INPUT	BYTE	I,Q,M,D,L, const	Идентификатор адресуемой области: B#16#54=периферийные входы B#16#55= периферийные выходы
LADDR	INPUT	WORD	I,Q,M,D,L, const	Заданный в HW-Config логический адрес для данного модуля (в 16-ичном формате)
RET_VAL	OUTPUT	INT	I,Q,M,D,L	Возвращаемое значение SFC
BUSY	OUTPUT	BOOL	I,Q,M,D,L	BUSY="1": процесс записи не закончен

Описание параметров**Параметр IOID**

Параметр задает идентификатор области памяти, адресуемой с помощью LADDR. Если речь идет о смешанном модуле, как, например, модуль со входами и выходами, то идентификатор задается для младшего периферийного адреса. Если адреса входов и выходов совпадают, то в этом случае задается идентификатор входов B#16#54.

Параметр LADDR

Заданный в HW-Config логический адрес для данного модуля (в 16-ичном формате). Если рассматривается смешанный модуль, то задается меньший из двух адресов.

Параметр RET_VAL

В параметре RET_VAL возвращается код ошибки. Возможные коды ошибок SFC57 соответствуют значениям RET_VAL для SFC55, приведенным в таблице 5.46.

5.5.4 Запись наборов данных/параметров с помощью SFC58 WR_REC

SFC58 WR_REC (Write RECoRD) переносит заданную с помощью параметра RECORD запись данных в модуль, адресованный с помощью LADDR и IOID. В отличие от SFC55 с помощью SFC58 можно передавать только записи данных с номерами от 2 до 240. Входные и выходные параметры SFC58 WR_REC даны в таблице 5.49.

Описание параметров**Параметр IOID**

Параметр задает идентификатор области памяти, адресуемой с помощью LADDR. Если речь идет о смешанном модуле, как, например, модуль со входами и выходами, то идентификатор задается для младшего периферийного адреса. Если адреса входов и выходов совпадают, то в этом случае задается идентификатор входов B#16#54.

Табл. 5.49 Параметры SFC58 WR_REC

Параметр	Тип доступа	Тип данных	Область памяти	Описание
REQ	INPUT	BOOL	I,Q,M,D,L, const	REQ="1": требование на запись
IOID	INPUT	BYTE	I,Q,M,D,L, const	Идентификатор адресуемой области: B#16#54=периферийные входы B#16#55= периферийные выходы
LADDR	INPUT	WORD	I,Q,M,D,L, const	Заданный в HW-Config логический адрес для данного модуля (в 16-ичном формате)
RECNUM	INPUT	BYTE	I,Q,M,D,L, const	Номер записи данных (допустимые значения от 2 до 240)
RECORD	OUTPUT	ANY	I,Q,M,D,L	Запись данных: только с типом данных BYTE
RET_VAL	OUTPUT	INT	I,Q,M,D,L	Возвращаемое значение SFC
BUSY	OUTPUT	BOOL	I,Q,M,D,L	BUSY="1": процесс записи не закончен

Параметр LADDR

Заданный в HW-Config логический адрес для данного модуля (в 16-ичном формате). Если рассматривается смешанный модуль, то задается меньший из двух адресов.

Параметр RET_VAL

В параметре RET_VAL возвращаются коды ошибок, приведенные в таблице 5.50.

Табл. 5.50 Специфические коды ошибок SFC58 WR_REC

Код ошибки W#16#...	Комментарий	Ограничения
7000	Первый вызов с REQ="0": передача данных не активна. BUSY имеет значение "0".	-
7001	Первый вызов с REQ="1": передача данных запускается. BUSY имеет значение "1".	Децентрализованная периферия
7002	Второй вызов (значение REQ не важно): передача данных уже активна; BUSY имеет значение "1".	Децентрализованная периферия
8090	Заданный логический адрес недействителен: не существует сопоставления в SDB1/SDB2х или нет базового адреса	-
8092	В параметре с типом данных ANY задан тип, отличный от BYTE.	Только в S7-400
8093	Для выбранного посредством LADDR и IOID модуля эта SFC недопустима (допустимы модули S7-300 – у S7-300, модули S7-400 – у S7-400, модули S7-DP у S7-300 и S7-400)	-
80A0	Отрицательное квитирование при чтении из модуля (модуль во время чтения отсутствует или неисправен)	Только к SFC59 RD_REC

Табл. 5.50 Продолжение

Код ошибки W#16#...	Комментарий	Ограничения
80A1	Отрицательное квитирование при записи в модуль (модуль во время чтения отсутствует или неисправен)	Только к SFC58 WR_REC
80A2	Ошибка DP-протокола на уровне 2 (Layer 2); возможно дефект аппаратуры.	Децентрализованная периферия
80A3	Ошибка DP-протокола при Direct-Data-Link-Mapper или в User-Interface; возможно дефект аппаратуры.	Децентрализованная периферия
80A4	Коммуникационная шина (K-Bus) неисправна (Помехи в K-шине)	Ошибка встречается между CPU и внешним DP-подключением
80B0	Возможные причины: - вызов SFC для данного модуля невозможен - номер записи больше 240 недопустим - в случае SFC58 WR_REC записи данных 0 и 1 не разрешены	-
80B1	Информация о длине в параметре RECORD неверна - у SFC58 WR_REC длина записи данных неверна - у SFC59 RD_REC данные больше длины записи данных (возможно только при применении старых S7-300 FM и S7-300 CPU)	-
80B2	Спроектированный слот не занят	-
80B3	Фактический тип модуля не совпадает с заданным типом модуля в SDB1	-
80C0	- SFC59 RD_REC: модуль ведет запись данных, но здесь еще нет данных для чтения - SFC13 DPNRM_DG: нет в наличии данных диагностики	Только у SFC59 RD_REC или у SFC13 DPNRM_DG
80C1	Данные предшествующего задания на запись в модуль для того же самого набора данных еще не обработаны модулем	-
80C2	В данный момент времени модуль модуль обрабатывает максимально возможное для данного CPU число заданий	-
80C3	В данный момент времени ресурсы (память и т.п.) заняты	-
80C4	Коммуникационная ошибка: - ошибка четности - SW-Ready не установлено - ошибка в приводимой длине блока - ошибка контрольной суммы на стороне CPU - ошибка контрольной суммы на стороне модуля	-
80C5	Децентрализованная периферия недоступна	Децентрализованная периферия
80C6	Передача записи данных прервана операционной системой для обработки программы с более высоким приоритетом	Децентрализованная периферия

В S7-400 SFC58 может также возвращать коды ошибок W#16#80Fx. Это говорит о том, что появившаяся ошибка не точно локализована.

5.5.5 Чтение записи данных с помощью SFC59 RD_REC

С помощью SFC59 RD_REC (ReaD RECoRD) читается запись данных RECNUM (0-240) из адресованного модуля и сохраняется в целевой области, заданной

параметром RECORD. Входные и выходные параметры SFC59 RD_REC описаны в таблице 5.51.

Табл. 5.51 Параметры SFC59 RD_REC

Параметр	Тип доступа	Тип данных	Область памяти	Описание
REQ	INPUT	BOOL	I,Q,M,D,L, const	REQ="1": требование на запись
IOID	INPUT	BYTE	I,Q,M,D,L, const	Идентификатор адресуемой области: B#16#54=периферийные входы B#16#55= периферийные выходы
LADDR	INPUT	WORD	I,Q,M,D,L, const	Заданный в HW-Config логический адрес для данного модуля (в 16-ичном формате)
RECNUM	INPUT	BYTE	I,Q,M,D,L, const	Номер записи данных (допустимые значения от 2 до 240)
RET_VAL	OUTPUT	INT	I,Q,M,D,L	Возвращаемое значение SFC
BUSY	OUTPUT	BOOL	I,Q,M,D,L	BUSY="1": процесс записи не закончен
RECORD	OUTPUT	ANY	I,Q,M,D,L	Целевая область для прочитанной записи данных

Описание параметров

Параметр IOID

Параметр задает идентификатор области памяти, адресуемой с помощью LADDR. Если речь идет о смешанном модуле, как, например, модуль со входами и выходами, то идентификатор задается для младшего периферийного адреса. Если адреса входов и выходов совпадают, то в этом случае задается идентификатор входов B#16#54.

Параметр LADDR

Заданный в HW-Config логический адрес для данного модуля (в 16-ичном формате). Если рассматривается смешанный модуль, то задается меньший из двух адресов.

Параметр RET_VAL

Если во время обработки функции наступает ошибка, то параметр RET_VAL содержит код ошибки. Возможные коды ошибок соответствуют кодам ошибок SFC58, которые приведены в таблице 5.39. У S7-400 SFC59 может также возвращать коды ошибок W#16#80Fx. Они означают, что ошибка точно не локализована.

Параметр RECORD

Содержащаяся информация о длине в выходном параметре RECORD специфицирует длину записи данных для чтения из выделенной записи данных. Это означает, что заданная здесь информация о длине не может быть больше, чем действительная длина записи данных. Задание длины в RECORD должно быть точно равно величине действительно читаемой записи данных.

Далее нужно обратить внимание на асинхронную обработку SFC59: параметр RECORD при всех (следующих) вызовах содержит информацию о длине. В качестве типа данных допустим *только* тип BYTE.

5.5.6 Чтение наборов данных с помощью SFB52 RDREC

С помощью SFB52 “RDREC” (ReaD RECoRD) читаются записи данных INDEX (область 0-255) из блока (компонента или модуля) DP-Slave’a, адресованного посредством ID.

Прочитанные данные помещаются в область, определенную с помощью параметра RECORD.

Параметр MLEN определяет, сколько байт максимально читается из компонента. Целевая область RECORD должна при этом быть выбрана длиной не менее MLEN байт.

В случае, если при передаче данных наступает ошибка, это будет сообщено через выходной параметр ERROR. Выходной параметр STATUS содержит в этом случае информацию об ошибке.

Интерфейс SFB52 “RDREC” идентичен с интерфейсом FB “RDREC”, определенным в стандарте PNO AK 1131. Входные и выходные параметры SFB52 “RDREC” определены в таблице 5.52.

Табл. 5.52 Параметры SFB52 RDREC

Параметр	Тип доступа	Тип данных	Область памяти	Описание
REQ	INPUT	BOOL	I,Q,M,D,L,const	REQ=1: требование на запись
ID	INPUT	DWORD	I,Q,M,D,L,const	Логический адрес компонента DP-Slave’a (блока, соотв., модуля)
INDEX	INPUT	INT	I,Q,M,D,L,const	Номер записи данных
MLEN	INPUT	INT	I,Q,M,D,L,const	Максимальная длина читаемой информации в байтах
VALID	OUTPUT	BOOL	I,Q,M,D,L	Новая запись данных принята и законна (справедлива, вступила в силу)
BUSY	OUTPUT	BOOL	I,Q,M,D,L	BUSY=1: процесс записи не окончен
ERROR	OUTPUT	BOOL	I,Q,M,D,L	ERROR=1: в процессе записи произошла ошибка
STATUS	OUTPUT	DWORD	I,Q,M,D,L	Идентификатор вызова в байтах 2 и 3 (W#16#7001 соотв., W#16#7002) или код ошибки
LEN	OUTPUT	INT	I,Q,M,D,L	Длина прочитанной информации записи данных
RECORD	OUTPUT	ANY	I,Q,M,D,L	Запись данных

Описание параметров

Параметр VALID

Значение TRUE в выходном параметре показывает, что запись данных успешно перенесена в целевую область RECORD. В этом случае выходной параметр LEN содержит длину прочитанных данных в байтах.

Параметр RECORD

Благодаря асинхронной обработке SFB52 нужно обратить внимание, что актуальный (фактический) параметр RECORD при всех вызовах, принадлежащих одному и тому же заданию, имеет одно и то же значение.

Параметр STATUS

В выходном параметре STATUS содержится информация об ошибке. Точные объяснения параметров будут даны в конце главы 5.5.7.

5.5.7 Запись наборов данных с помощью SFB53 WDREC

С помощью SFB53 WRREC (WRiteRECORD) передается компоненту (блоку, соотв., модулю), адресованному с помощью ID, набор данных RECORD с номером INDEX (диапазон от 0 до 255).

Длина передаваемого набора данных в байтах определяется с помощью параметра LEN. Область-источник RECORD должна при этом иметь длину не менее LEN байт.

Если при передаче набора данных наступает ошибка, это будет передано через выходной параметр ERROR. Выходной параметр STATUS содержит в этом случае информацию об ошибке.

Интерфейс SFB53 “WRREC” идентичен с интерфейсом FB “WRREC”, определенным в стандарте PNO AK 1131. Входные и выходные параметры SFB53 “WRREC” определены в таблице 5.53.

Табл. 5.53 Параметры SFB53 WDREC

Параметр	Тип доступа	Тип данных	Область памяти	Описание
REQ	INPUT	BOOL	I,Q,M,D,L,const	REQ=1: требование на запись
ID	INPUT	DWORD	I,Q,M,D,L,const	Логический адрес компонента DP-Slave'a (блока, соотв., модуля)
INDEX	INPUT	INT	I,Q,M,D,L,const	№ набора данных
LEN	INPUT	INT	I,Q,M,D,L,const	Макс. длина набора данных для передачи в байтах
DONE	OUTPUT	BOOL	I,Q,M,D,L	Набор данных передан
BUSY	OUTPUT	BOOL	I,Q,M,D,L	BUSY=1: процесс записи еще не окончен
ERROR	OUTPUT	BOOL	I,Q,M,D,L	ERROR=1: в процессе записи произошла ошибка
STATUS	OUTPUT	DWORD	I,Q,M,D,L	Идентификатор вызова в байтах 2 и 3 (W#16#7001 соотв., W#16#7002) или код ошибки
RECORD	OUTPUT	ANY	I,Q,M,D,L	Набор данных

Описание параметров

Параметр DONE

Значение TRUE в выходном параметре DONE показывает, что набор данных передан требуемому DP-Slave'у.

Параметр RECORD

Нужно обратить внимание, что из-за асинхронной обработки SFB53 актуальный (фактический) параметр RECORD при всех вызовах, принадлежащих одному и тому же заданию, содержит одно и то же значение. То же самое справедливо для актуального параметра LEN.

Параметр STATUS

Выходной параметр STATUS содержит информацию об ошибке. Если он интерпретируется как ARRAY[1..4] OF BYTE, то информация об ошибке содержится в структуре, представленной в таблице 5.5.4.

С помощью выходного параметра BUSY и байтами 2 и 3 выходного параметра STATUS индицируется состояние задания. При этом байты 2 и 3 STATUS'a соответствуют выходному параметру RET_VAL асинхронно работающих SFC (см. также таблицу 5.50).

Табл. 5.54 Представление выходного параметра STATUS

Элемент массива	Имя	Значение
STATUS[1]	Function_Num	B#16#00: нет ошибки. Идентификатор функции из DPV1-PDU: в случае ошибки используется B#16#80. В случае отсутствия элемента протокола DPV1 используется B#16#C0.
STATUS[2]	Error Decode	ID местоположения ошибки
STATUS[3]	Error Code 1	Идентификатор ошибки
STATUS[4]	Error Code 2	Специфический для пользователя идентификатор ошибки

Идентификатор местоположения ошибки вносится в STATUS[2] и представлено в таблице 5.55.

Идентификатор ошибки из STATUS[3] представлен в таблице 5.56.

Код ошибки в STATUS[4] попадает при DPV1-ошибке из DP-Master'a в CPU. Если нет DPV1-ошибки, то это значение устанавливается в "0", с исключением для SFB52:

- STATUS[4] содержит длину целевой области из RECORD в случае MLEN больше длины целевой области из RECORD.
- STATUS[4]=MLEN в случае, когда действительная длина набора данных меньше MLEN и MLEN меньше длины целевой области из RECORD.

Табл. 5.55 Идентификатор ошибки в STATUS[2]

Код ошибки (B#16#...)	Источник	Значение
От 00 до 7F	CPU	Нет ошибок или нет предупреждений
80	DPV1	Ошибка по IEC 61158-6
От 81 до 8F	CPU	B#16#8x показывает ошибку в x-м вызываемом параметре SFB
FE, FF	DP-профиль	Ошибка, специфическая для производителя

Табл. 5.56 Идентификатор ошибки в STATUS[3]

STATUS[2] Error Decode (B#16#...)	STATUS[3] Error_Code_1 (B#16#...)	Комментарий согласно DPV1	Значение
00	00		Нет ошибок, нет предупреждений
70	00	Reserved, reject	Первый вызов; передача набора данных неактивна
70	01	Reserved, reject	Первый вызов; передача набора данных запущена
70	02	Reserved, reject	Промежуточный вызов; передача набора данных уже активна
80	90	Reserved, pass	Логический начальный адрес недействителен
80	92	Reserved, pass	Недопустимый тип в ANY-указателе
80	93	Reserved, pass	DP-компонент, адресованный посредством ID, соотв., F_ID, не конфигурирован
80	A0	Error read	Отрицательное квитирование при чтении из блока
80	A1	Write error	Отрицательное квитирование при записи в блок
80	A2	Module failure	Ошибка DP-протокола на уровне 2, возможно дефект аппаратуры
80	A3	Reserved, pass	Ошибка DP-протокола при Direct-Data-Link-Mapper или User-Interface/User, возможно дефект аппаратуры
80	A4	Reserved, pass	Нарушена коммуникация по C-шине
80	A5	Reserved, pass	---
80	A7	Reserved, pass	Занят DP-ресурс
80	A8	Version conflict	Конфликт версий
80	A9	Feature not supported	Свойство не поддерживается
80	От AA до AF	User specific	Специфично для DP-Master'a
80	B0	Invalid index	Блоку неизвестен набор данных: набор данных с номером >=256 недопустим
80	B1	Write length error	Ошибка длины в AINFO
80	B2	Invalid slot	Спроектированный слот не занят
80	B3	Type conflict	Тип фактического блока не совпадает с заданным
80	B4	Invalid area	Недействительная область
80	B5	State conflict	Конфликт состояния
80	B6	Access denied	Доступ запрещен
80	B7	Invalid range	Недействительная область
80	B8	Invalid parameter	Недействительный параметр
80	B9	Invalid type	Недействительный тип
80	От BA до BF	User specific	Специфично для DP-Master'a

Табл. 5.55 Продолжение

STATUS[2] Error_Decode (B#16#...)	STATUS[3] Error_Code_1 (B#16#...)	Комментарий согласно DPV1	Значение
80	C0	Read constrain conflict	Блок читает набор данных, но прочитанных данных еще нет тут
80	C1	Write constrain conflict	Данные предшествующего задания на запись в блок для того же самого набора данных еще не обработаны блоком
80	C2	Resource busy	Блок в данный момент обрабатывает максимально возможное количество заданий для CPU
80	C3	Resource unavalable	Необходимые рабочие ресурсы в данный момент заняты
80	Dx	User specific	Специфические для DP-Slave См. описание DP-Slave'a
81	От 00 до FF		Ошибка в первом параметре (у SFB54: MODE)
81	00		Недопустимый режим работы
82	От 00 до FF		Ошибка во втором параметре
↓	↓		↓
88	От 00 до FF		Ошибка в восьмом параметре (у SFB54: TINFO)
88	01		Неверный синтаксический идентификатор
88	23		Переполнение разрядной сетки или целевая область мала
88	32		Номер DB/DI вне пользовательской области
88	3A		Номер DB/DI = NULL у идентификатора области DB/DI или заданный DB/DI не существует
89	От 00 до FF		Ошибка в девятом параметре (у SFB54: AINFO)
89	01		Неверный синтаксический идентификатор
89	23		См. 88-23
89	24		См. 88-24
89	32		См. 88-32
89	3A		См. 88-3A
8A	От 00 до FF		Ошибка в десятом параметре
↓	↓		↓
8F	От 00 до FF		Ошибка в пятнадцатом параметре
FE, FF	От 00 до FF		Специфическая для профиля ошибка

6. Пример пользовательского обмена данными с помощью PROFIBUS-DP

Введение

Децентрализованная периферия, подключенные через DP-интерфейс к системе SIMATIC S7, обрабатывается так же, как и периферия, находящаяся в центральной корзине или корзинах расширения. В зависимости от предоставленных при проектировании в HW-Config адресов обмен входными и выходными данными происходит или прямо через изображение процесса, или через соответствующие команды доступа к периферии. Однако в системе SIMATIC S7 существуют специальные системные функции для работы с DP-периферией.

- Для обмена данными со сложными DP-Slave'ами, которые имеют *консистентные* входные/выходные данные, в системе SIMATIC S7 предусмотрены SFC14 DRRD_DAT и SFC15 DPWR_DAT.
- Для запуска сигналов от процесса на DP-Master'е из системы S7-300, используемой как I-Slave, предусмотрена функция SFC7 DP_PRAL.
- Параметры модулей S7-DP-Slave'а можно читать и записывать из пользовательской программы с помощью вызовов специально для этого предусмотренных функций.
- С помощью SFC11 DPSYC_FR можно синхронизировать запись выходов и фиксировать входы DP-Slave'ов.

В следующей главе на основе практического примера описывается организация обмена данными с DP-Slave'ами в системе SIMATIC S7. В основу конфигурации установки положена конфигурация примеров из главы 4, созданная с помощью HW-Config. Для понимания описываемого примера применения необходимы базовые знания в программировании на STL (AWL), так как примеры этой главы даются в представлении STL.

6.1 Обмен данными с помощью команд доступа к периферии

Как представлено на рисунке 6.1, S7-CPU может получать доступ к периферии с помощью команд доступа через изображение процесса или команд прямого доступа к периферии в форме байта, слова или двойного слова.

Однако как только с помощью DP-Save'а обрабатывается информация длиной 3 байта или более, чем 4 байта и установлена консистентность "Total length", нельзя обмениваться входными/выходными данными через изображение процесса или через соответствующие команды прямого доступа.

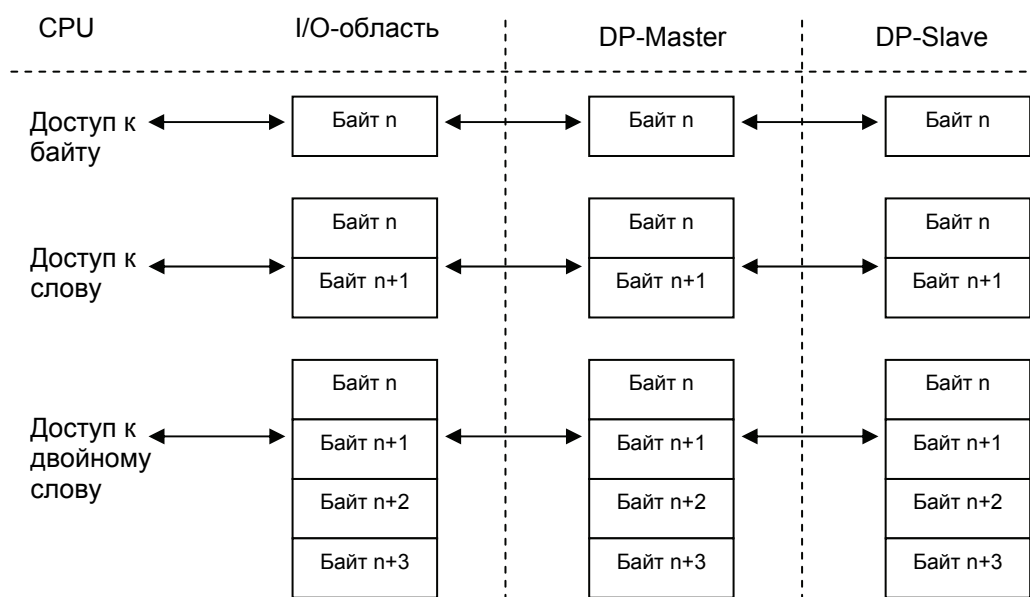


Рис. 6.1 Обмен входными / выходными данными с помощью STEP7 через команды доступа к периферии

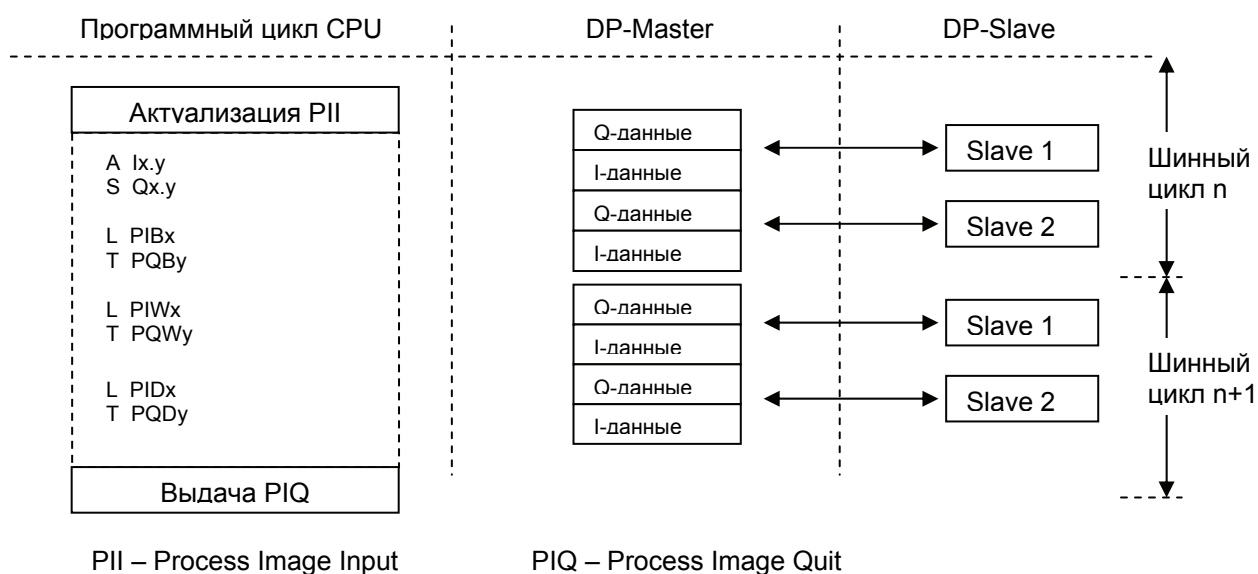


Рис. 6.2 Входные/выходные данные DP-Slave. Актуализация и доступ

Как показано на рисунке 6.2, актуализация DP-входных/выходных данных определяется исключительно циклическим обменом данными (шинным циклом) DP-Master'а с DP-Slave'ами. При известных обстоятельствах это может означать, что между двумя доступами к периферии в пользовательской программе область периферии DP-Slave'а может быть актуализирована (изменена), то есть, например, данные, прочитанные в первом и во втором циклах, относятся к *разным моментам времени*. По этой причине гарантируется консистентность данных только для тех периферийных

структур и областей, к которым пользовательская программа обращается без прерывов с помощью команд для байтов, слов или двойных слов.

6.2 Обмен консистентными данными с помощью SFC14 DPRD_DAT и SFC15 DPWR_DAT

DP-Slave'ы, которые реализуют сложные функции, не обходятся обычными простыми структурами данных. Для структур данных, применяемых в этих DP-Slave'ах и определяемых, например, областями параметров регулятора или привода, необходимы большие области входных/выходных данных. Подобные области входных/выходных данных, которые содержат единую информацию и не могут разделяться на байты, слова или двойные слова, должны обрабатываться как консистентные данные (см. раздел 2.2.2 “Консистентные данные”). У некоторых входных/выходных модулей можно с помощью конфигурационной телеграммы установить консистентные области входных/выходных данных с длиной максимум 64 байта или, соответственно, слова (128 байт). Обмен данными с консистентными областями данных DP-Slave'a осуществляется с помощью SFC14 DRRD_DAT и SFC15 DPWR_DAT.

Рисунок 6.3 показывает принцип работы SFC14 DRRD_DAT и SFC15 DPWR_DAT.

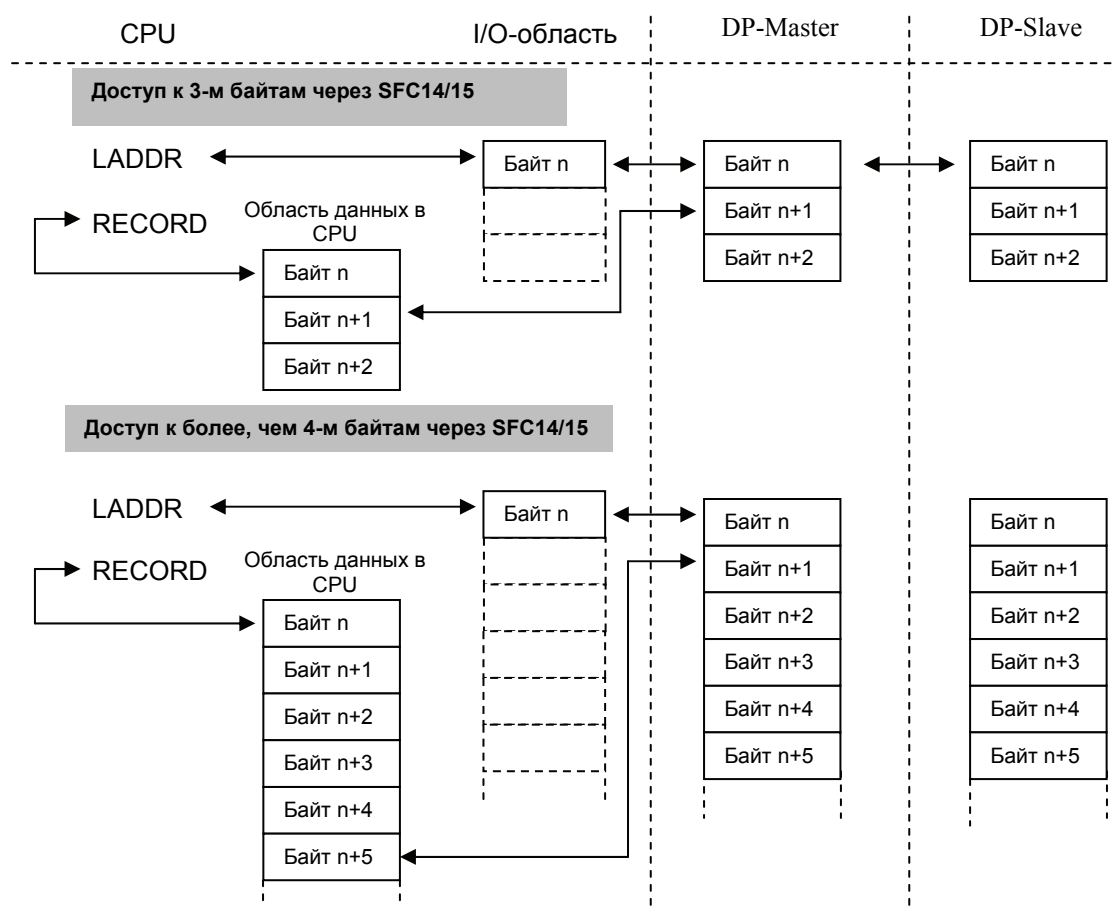


Рис. 6.3 Обмен входными/выходными данными через SFC14 и SFC15

Параметр SFC LADDR служит здесь как указатель на область входных данных для чтения или область выходных данных записи. В этом параметре вызова SFC задается начальный адрес области входных или выходных данных в 16-ичном формате, спроектированный в HW-Config.

Параметр SFC RECORD задает область-источник или область-цель для данных в CPU. Описания входных и выходных параметров, а также специфические возвращаемые значения (коды ошибок) параметра RET_VAL содержатся в разделе 5.3.

Следующий пример показывает применение SFC14 и SFC15. Он базируется на проекте-примере, описанном в разделе 4.2.5 “S7-300/CPU315-2DP как I-Slave” и ограничивается участием станции S7-DP-Master (S7-400) вместе со станцией S7-300, как I-Slave. Поэтому Вы теперь должны стереть спроектированные станции ET200M и ET200B, соединить DP-интерфейсы S7-300 и S7-400 кабелем PROFIBUS. В примере следует исходить из того, что обе системы управления стерты и находятся в состоянии RUN (ключ в положении RUN-P). Оба участника в примере имеют входные/выходные данные с длиной 10 байт с консистентностью “Total length” (см. раздел 4.2.5, рис.4.18). Это означает, что для обмена входными/выходными данными между DP-Slave’ом (I-Slave’ом) и DP-Master’ом должны применяться SFC14 и SFC15.

6.2.1 Пользовательская программа для I-Slave (S7-300 с CPU315-2DP)

Точно так же, как у S7-DP-Master’а, у I-Slave’а в нашем примере обмен консистентными входными/выходными данными, длиной в 3 байта или, как в нашем примере, более, чем 4 байта, должен происходить с помощью SFC14 и SFC15. Обратите внимание при этом, как показано на рисунке 6.4, что выходные данные, передаваемые в DP-Master’е через SFC15, в I-Slave’е читаются как входные данные через SFC14. С входными данными DP-Master’а все происходит соответственно наоборот.

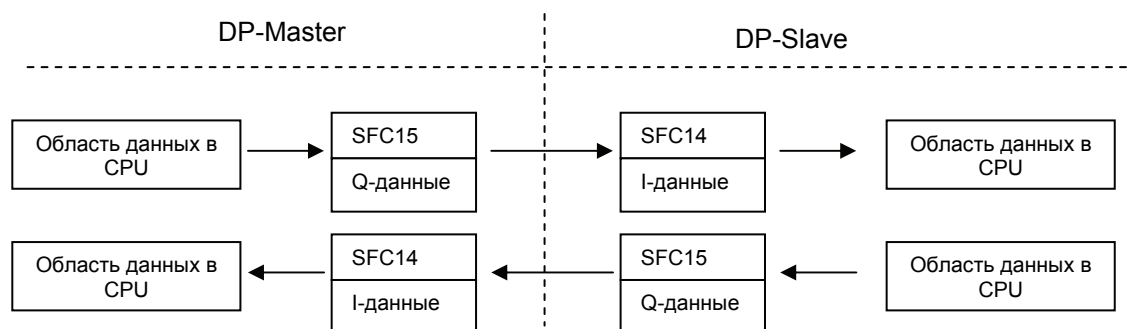


Рис. 6.4 Обмен входными/выходными данными с I-Slave в проекте-примере через SFC14 и SFC15

Так как CPU SIMATIC S7-300 не распознает ошибок адресации, можно помещать входные/выходные данные, принимаемые и, соответственно, посылаемые с помощью SFC в программе-примере в области отображения процесса, не занятые в конфигурации CPU315-2DP модулями, например, IB100 – 109 и QB100 – 109. На основании этого к этим данным можно в

пользовательской программе получить доступ с помощью простых команд для байта, слова, двойного слова.

Для создания необходимой пользовательской программы Вы должны действовать, как указано ниже:

- Выберите, как показано на рисунке 6.5, в SIMATIC Manager с открытым проектом S7-PROFIBUS-DP, контейнер SIMATIC 300(1), потом – контейнер Blocks. В контейнере объектов уже существует организационный блок OB1 и созданные в HW-Config системные данные SDB (системные блоки данных).

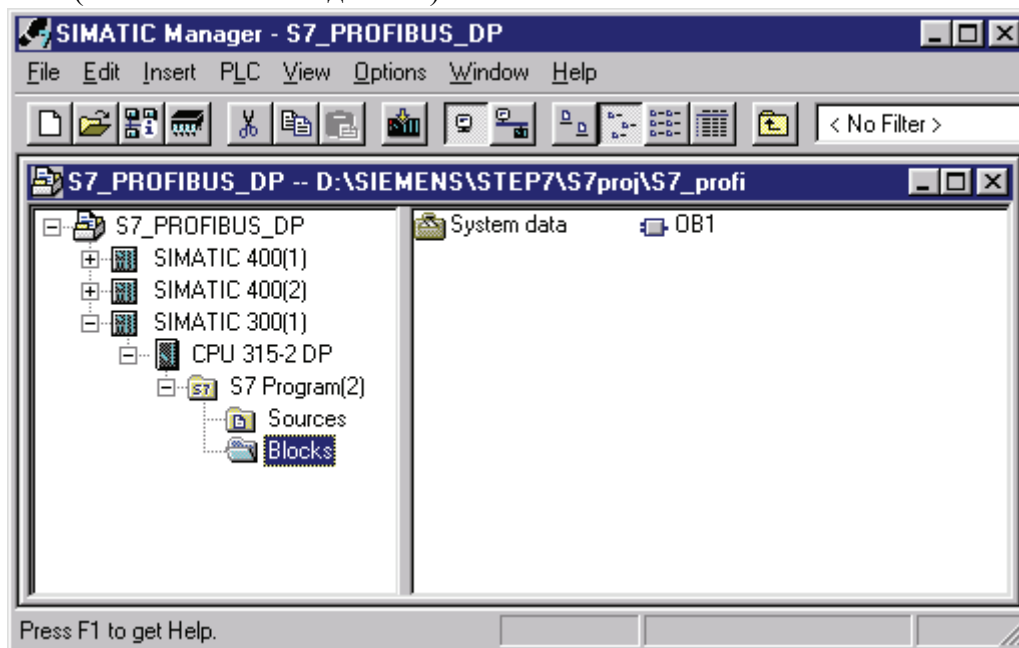


Рис. 6.5 SIMATIC Manager с открытым окном контейнера объектов *Blocks*

- Двойным щелчком на OB1 откройте этот блок в LAD/FBD/STL-редакторе в представлении STL.
- Введите в программном редакторе команду “CALL SFC14” и нажмите кнопку “Enter”. SFC14 DPRD_DAT появится со своими входными/выходными параметрами. Снабдите формальные входные/выходные параметры фактическими, как показано на рисунке 6.6. Вызовите также SFC15 и снабдите входные/выходные параметры соответственно. Блоки SFC14 и SFC15 можно скопировать в проект из библиотеки стандартных функций (... \SIEMENS\STEP7\S7libs\STDLIB30).

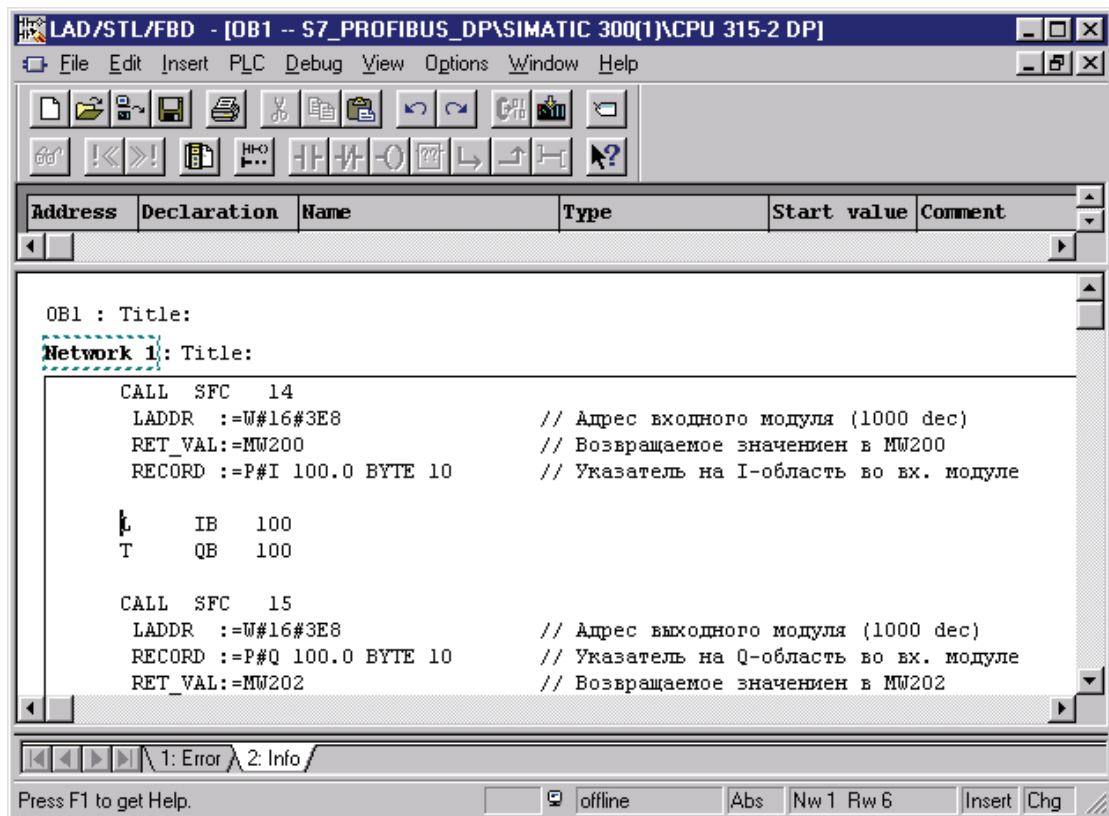


Рис. 6.6 Программный редактор STL с OB1 программы- примера для CPU315-2DP

- Чтобы в примере было просто контролировать обмен данными в Master'е, передайте, как показано на рисунке 6.6, с помощью команд загрузки и передачи первый принимаемый байт (IB100) в первый посылаемый байт (QB100). Таким образом посланный из DP-Master'а первый байт попадает в область входных данных I-Slave, а оттуда – в область выходных данных I-Slave, которая передается обратно в DP-Master.
- Запомните OB1. Теперь в папке блоков помимо блока OB1 и папки System Data, содержатся блоки SFC14 и SFC15.

Чтобы избежать перехода CPU в STOP во время работы нашего примера из-за отсутствия OB в I-Slave'е, которые операционная система вызывает при смене рабочего состояния DP-Master'а или при выходе его из строя, нужно создать соответствующие OB ошибок.

- Смена рабочего состояния CPU DP-Master'а с RUN на STOP вызывает OB82 (диагностические сигналы) в I-Slave. Поэтому создайте в CPU I-Slave'а OB82.
- Далее, при выходе из строя DP-Master'а в I-Slave'е вызывается OB86 (выход из строя носителя модулей). Чтобы избежать останова CPU по этой причине, создайте OB86.
- Передайте все блоки в CPU.
- После окончания процесса передачи CPU315-2DP должен опять переключен в состояние RUN. Светодиоды CPU315-2DP для DP-интерфейса показывают следующее состояние:
 - светодиод "SF DP" – горит;

- светодиод “BUSF” – мигает.

6.2.2 Пользовательская программа для DP-Master’a (S7-400 с CPU416-2DP)

Для создания программы DP-Master’a для программы-примера откройте в проекте контейнер объектов “Blocks”, содержащийся в контейнере объектов SIMATIC 400(1). Откройте OB1 и вызовите, как показано на рисунке 6.7, SFC14 и SFC15.

Чтобы при работе программы-примера в DP-Master’e избежать останова CPU из-за отсутствия ОБ диагностики и ошибок, создайте в CPU OB82 и OB86. Как области данных для входных/выходных данных I-Slave в примере должны применяться блоки данных DB10 и DB20. Эти DB должны иметь соответствующую длину.

- Создайте DB10 и DB20.
- Внутри каждого блока создайте по переменной с типом ARRAY[1..10] OF BYTE.
- Загрузите OB82, OB86, DB10, DB20 в CPU.
- После загрузки переведите CPU в состояние RUN-P. Светодиоды “SF DP” и “BUSF” не должны светиться или мигать. Обмен начался.

```
CALL SFC 14
  LADDR :=W#16#3E8           // Адрес входного модуля (1000 dec)
  RET_VAL:=MW200             // Возвращаемое значение в MW200
  RECORD :=P#DB10.DBX 0.0 BYTE 10 // Указатель на область данных для
                                // входных данных

CALL SFC 15
  LADDR :=W#16#3E8           // Адрес выходного модуля (1000 dec)
  RECORD := P#DB20.DBX 0.0 BYTE 10 // Указатель на область данных для выходных данных
  RET_VAL:=MW202             // Возвращаемое значение в MW202
```

Рис.6.7 STL-программа DP-Master’a в STEP 7 с программой в OB1 для обмена данными через SFC14 и SFC15

6.2.3 Проверка обмена данными между DP-Master’ом и DP-Slave’ом

Чтобы протестировать обмен входными/выходными данными, выберите в SIMATIC Manager при имеющейся MPI-связи между PG/PC и CPU416-2DP online-вид для проекта. Откройте таблицу Monitor/Modify Variables. В ней в качестве операндов задайте DB10.DBB0 и DB20.DBB0 (DB10.DBB0 – 1-й байт выходных данных I-Slave; DB20.DBB0 DBB0 – 1-й байт входных данных I-Slave).

6.3 Сигнал процесса с помощью S7-300 как I-Slave: создание и обработка

Как и центральная периферия, DP-Slave’ы, а также отдельные модули внутри DP-Slave’a, могут генерировать сигналы от процесса при условии, что DP-

Slave'ы и входные/выходные модули в состоянии это делать. Так, например, можно с помощью аналогового модуля, способного генерировать сигналы от процесса, прерывать пользовательскую программу и запускать ОВ сигнала от процесса при выходе процесса за заданные границы.

Следующий пример описывает, как станция S7, которая используется, как I-Slave на шине PROFIBUS-DP, порождает сигнал от процесса. После этого описывается, как сигнал от процесса распознается и обрабатывается в DP-Master'е (S7-400).

6.3.1 Генерирование сигнала от процесса станцией S7-300, используемой как I-Slave

Как показано на рисунке 6.10, можно с помощью вызова SFC7 DP_PRAL на станции с CPU315-2DP, спроектированной как I-Slave, возбуждать на DP-Master'е (может быть только с CPU S7-400 или S7-315-2DP) сигнал от процесса.

Требуемый сигнал от процесса однозначно определяется относящимися к модулю входными параметрами SFC7 IOID и LADDR. В нашем примере сигнал от процесса должен быть возбужден для выходного модуля, спроектированного в I-Slave'е с адресом "1000". Так как для нашего примера представляет интерес только вызов сигнала от процесса в I-Slave и обработка его в DP-Master'е, SFC7 будет запущена циклически.

Затем в DP-Master должен быть передан с помощью входного параметра SFC7 AL_INFO специфический для приложения идентификатор сигнала I-Slave'а (в примере этот идентификатор – "ABCD") в первой половине и "счетчик сигналов" (счетчик выполненных заданий) во второй половине двойного слова. Этот идентификатор сигнала передается с сообщением о сигнале от процесса на DP-Master и там во время обработки OB40 записывается в его локальную переменную OB40_POINT_ADDR.

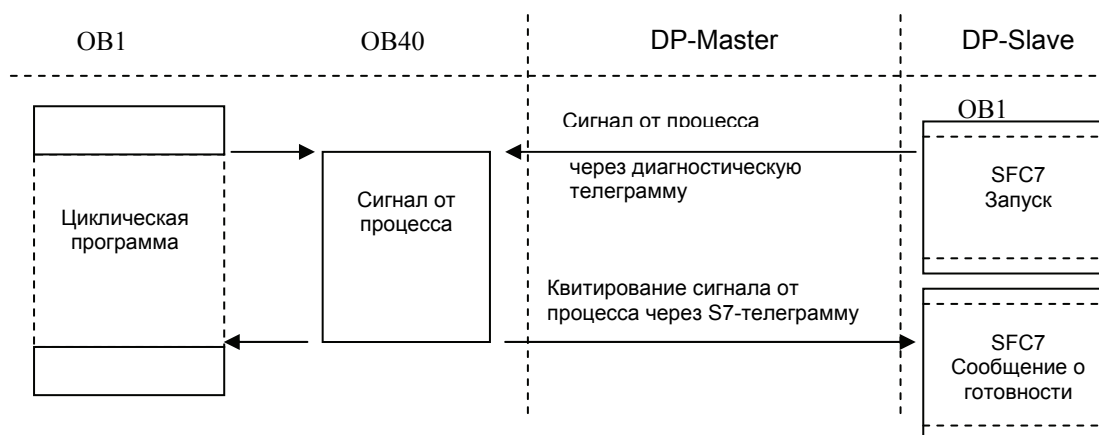


Рис. 6.10 Возбуждение сигнала от процесса с помощью S7-300 (CPU315-2DP), как I-Slave

Для вызова сигнала от процесса напишите в OB1 для SIMATIC 300 программу, приведенную на рисунке 6.11 и загрузите ее в CPU, который находится в состоянии STOP.

6.3.2 Обработка сигнала от прцесса с помощью S7-400 как DP-Master'a

Сигнал от процесса, запущенный в примере от I-Slave через PROFIBUS, идентифицируется CPU DP-Master'a и благодаря операционной системе запускается принадлежащий ему OB40. С помощью сигнала от процесса можно с помощью локальных данных OB40 (см. раздел 5.1.2) определить через логический базовый адрес модуля, вызвавшего сигнал, причину сигнала, а у сложных модулей также состояние сигнала и идентификатор сигнала. После обработки пользовательской программы в OB40, I-Slave, возбудивший сигнал, квитирует этот сигнал. При этом меняется состояние сигнала выходного параметра – BUSY SFC7 с “1” на ”0”.

```
L   W#16#ABCD      // Часть идентификатора сигнала
T   MW 104

CALL "DP_PRAL"
  REQ  :=M100.0
  IOID :=B#16#55     // Адресная область модуля ("55"=выходы)
  LADDR :=W#16#3EF   // Начальный адрес модуля (1000d)
  AL_INFO:=MD104     // Специфический адрес модуля
  RET_VAL:=MW102
  BUSY :=M100.1

A   M 100.1         // Запуск (циклический), если SFC7 "свободна"
BEC
=   M 100.0         // Вызов нового сигнала от прцесса

L   MW 106          // Увеличение счетчика сигналов
+   1
T   MW 106
```

Рис. 6.11 Пример программы в DP-Slave (S7-300) для возбуждения сигнала от процесса

Для обработки сигнала от процесса в DP-Master'e поместите в контейнер объектов "Blocks" в SIMATIC 400(1) примера OB40 с STL-программой, изображенной на рисунке 6.12.

```
L   #OB40_MDL_ADDR // Логи ческий базовый адрес модуля
T   MW 10

L   #OB40_POINT_ADDR // Специфический для приложения
                        // идентификатор сигнала в I-Slave
T   MD 12
```

Рис. 6.12 Программа DP-Master'a S7-400 для обработки сигнала от процесса

Сохраните OB40 и загрузите его в CPU.

С помощью команд загрузки и передачи, изображенных на рисунке 6.12, Вы копируете базовый адрес периферийного модуля, вызвавшего сигнал, в MW10 и специфический для приложения идентификатор сигнала в MD12. С помощью функции STEP7 Monitor/Modify Variables Вы можете позднее контролировать обе эти меркерные области при обработке сигнала от процесса.

После загрузки OB40 в CPU416-2DP, Вы можете CPU S7-300 опять перевести в состояние RUN. (После этого обе системы управления находиться в состоянии RUN).

6.3.3 Тестирование обработки сигнала от процесса в DP-Master'е

Для тестирования обработки сигнала от процесса в DP-Master'е выберите при имеющейся MPI-связи между PG/PC и CPU 416-2DP в SIMATIC Manager online-представление для проекта-примера.

Выберите с помощью двойного щелчка контейнер объектов “Blocks”, содержащийся в контейнере объектов SIMATIC 400(1). Двойным щелчком вызовите для блока OB40 online-представление. Вы можете наблюдать обработку сигнала от процесса в DP-Master'е с помощью функции *Debug->Monitor*.

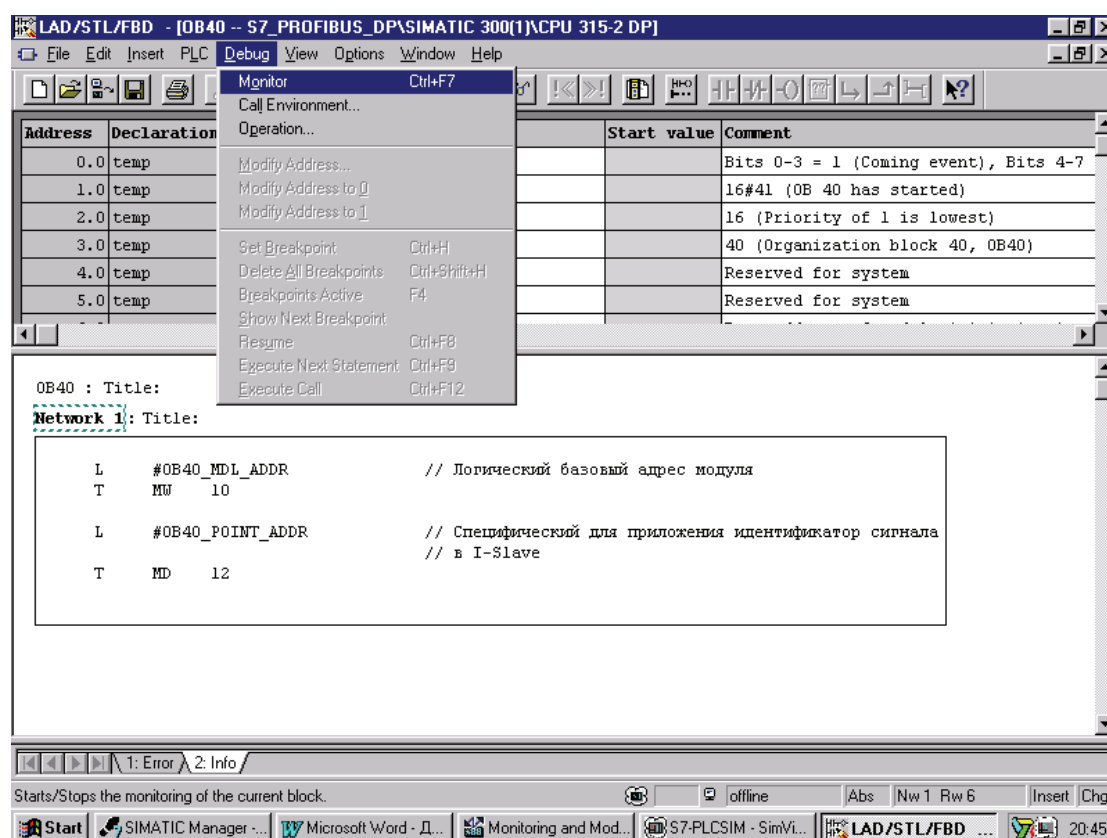


Рис.6.13 Функция состояния для OB40 в программе-примере

6.4 Передача записей (наборов) данных и параметров

Благодаря возможности передавать из пользовательской программы наборы данных в модули SIMATIC S7, можно в рабочем состоянии перепараметризовать эти модули. Перенос наборов данных возможен как для модулей в центральной стойке, так и для децентрализованных модулей S7. Переносимые наборы данных подразделяются на динамические наборы данных, которые как правило предоставляются в распоряжение пользовательской программой, и статические наборы данных, которые создаются с помощью HW-Config и сохраняются в SDB. Для передачи наборов данных в S7-модули

SIMATIC S7 предоставляет в распоряжение различные SFC (см. также раздел 5.5).

В ниже описываемом примере для записей наборов данных/параметров в модули применяются SFC55 WR_PARM и SFC56 WR_DPARM. Как показано на рисунке 6.14, можно передавать в S7-модули с помощью SFC55 (свободно устанавливаемые) динамические наборы данных, с помощью SFC56 – созданные в HW-Config и сохраняемые в SDB “статические наборы данных”. Эти наборы данных при запуске системы автоматически переносятся в соответствующий модуль.

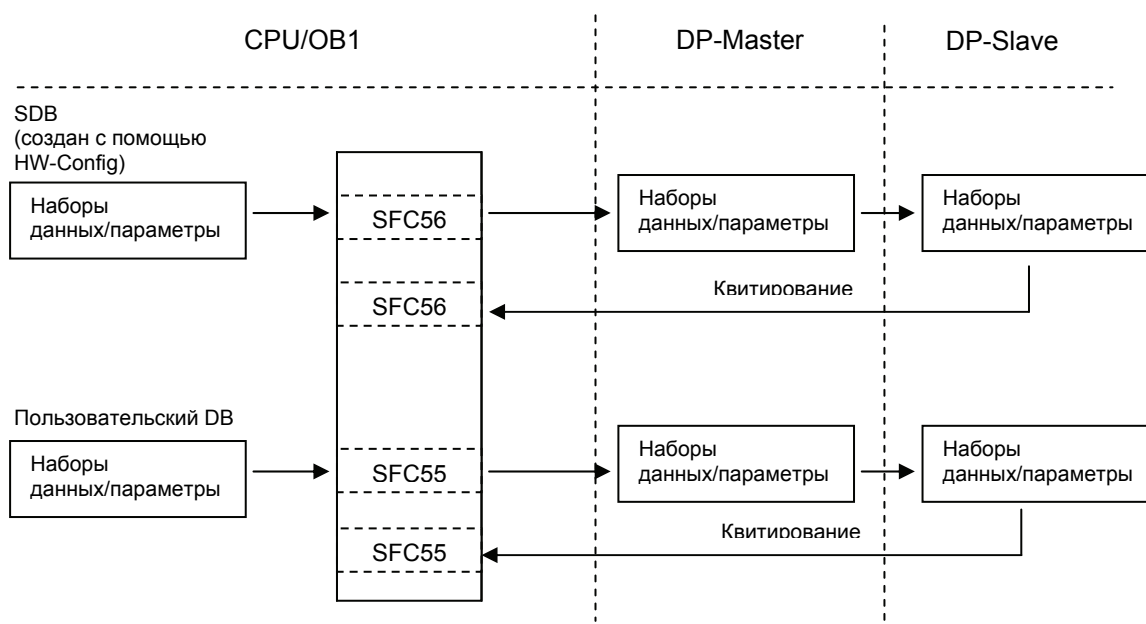


Рис. 6.14 Передача наборов данных с помощью SFC55/SFC56 в S7-модули

В примере применения нужно перепараметризовать спроектированный в разделе 4.2.5 аналоговый модуль в станции ET200M: первоначально заданный диапазон измерений $\pm 10V$ с помощью SFC55 изменить на $\pm 2,5V$. Затем это перепараметрирование должно быть отменено с помощью SFC56 и модуль должен опять работать с параметрами, заданными при проектировании в HW-Config. Эти функции могут быть важными на практике, например, чтобы при достижении определенных состояний процесса или области измеряемых значений временно достигать более точного разрешения измеряемых значений.

6.4.1 Структура набора данных (DS1) для аналоговых входных модулей SIMATIC S7-300

В применяемом в примере аналоговом модуле речь идет о модуле SIMATIC S7-300 “SM331 AI2x12Bit” с двумя аналоговыми каналами, которые имеют разрешение от 12 до 14 бит. Таблица 6.1 показывает имеющиеся наборы данных для аналогового входного модуля SIMATIC S7-300. Набор данных № 0 (DS0) можно с помощью SFC только читать и, таким образом, не может быть передан с помощью SFC55.

Табл. 6.1 Наборы данных и параметры аналогового входного модуля в SIMATIC S7-300

Параметр	№ набора данных	Параметрируемость с SFC55
Диагностика: сборная диагностика	0	нет
Диагностика, включая проверку на обрыв провода	0	нет
Деблокировка сигнала от граничных значений	1	да
Деблокировка диагностических сигналов	1	да
Подавление частоты помех	1	да
Тип измерений	1	да
Область измерений	1	да
верхняя граница	1	да
нижняя граница	1	да

Рисунок 6.15 показывает в деталях структуру набора данных DS1 для параметров аналогового входного модуля в SIMATIC S7-300. Благодаря параметрам, сохраняемым в этом наборе данных, можно деблокировать сигналы, выбирать время интегрирования для подавления частоты помехи, а также тип измерений и, если это необходимо, верхнюю и нижнюю границы для области измерений группы каналов аналоговых входов. DS1 имеет длину 14 байт.

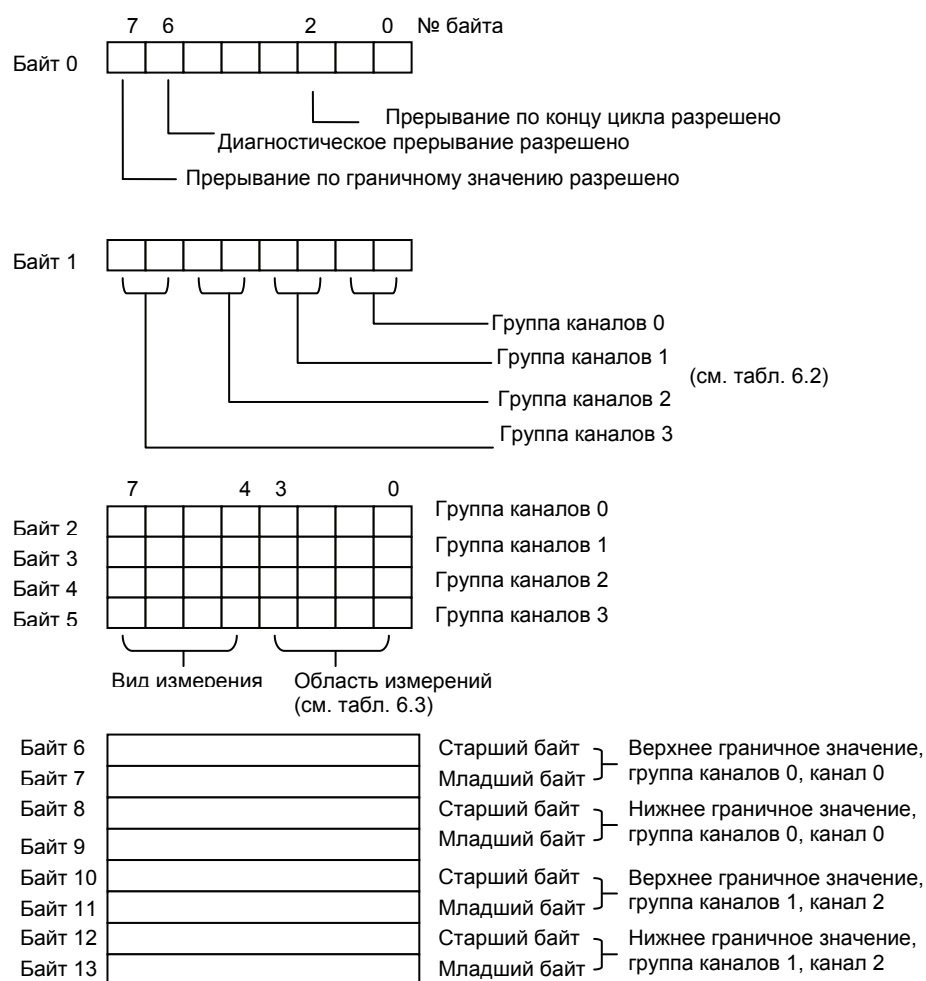


Рис. 6.15 Структура набора данных DS1 для аналогового входного модуля S7-300

В таблице 6.2 представлены возможные установки времени интегрирования для подавления частоты помехи в аналоговом входном модуле.

Табл. 6.2 Установки времени интегрирования для аналогового входного модуля в S7-300

Подавление частоты помехи	Время интегрирования	Установка
400 Hz	2,5 ms	2#00
60 Hz	16,7 ms	2#01
50 Hz	20,0 ms	2#10
10 Hz	100,0 ms	2#11

Таблица 6.3 показывает для выбранного типа измерения “напряжение” устанавливаемые области измерения для аналогового входного модуля S7-300.

Табл. 6.3 Устанавливаемые области измерения для аналогового входного модуля S7-300

Тип измерения	Установка	Область измерений	Установка
Напряжение	2#0001	$\pm 80 \text{ mV}$	2#0001
		$\pm 250 \text{ mV}$	2#0010
		$\pm 500 \text{ mV}$	2#0011
		$\pm 1 \text{ V}$	2#0100
		$\pm 2,5 \text{ V}$	2#0101
		$\pm 5 \text{ V}$	2#0110
		от 1 до 5 V	2#0111
		$\pm 10 \text{ V}$	2#1001
		$\pm 25 \text{ V}$	2#1010
		$\pm 50 \text{ V}$	2#1011

При создании проекта-примера установите в HW-Config значения для применяемого аналогового входного модуля в ET200M:

Диагностика: общая диагностика “on”
 Тип измерений: напряжение (U)
 Область измерений: +/-10 V
 Время интегрирования: 20 ms

6.4.2 Пример применения: параметрирование аналогового входного модуля с помощью SFC55 WR_PARM

Рассмотренный ниже пример применения использования SFC55 относится к разделу 4.2.5 “ET200M”. Здесь, однако, применяются только станция S7-DP-Master (S7-400) и станция DP-Slave ET200M. Поэтому в проекте-примере Вы должны с помощью HW-Config удалить спроектированные ранее станции ET200B и S7-300. Соедините DP-интерфейсы станций S7-400 и ET200M PROFIBUS-кабелем и включите источник питания. В примере исходим из того, что у DP-Master’а очищена память и он находится в состоянии RUN (ключ в

положении RUN-P). Далее, считаем, что для ET200M установлен PROFIBUS-адрес “5”.

Создайте в контейнере объектов Blocks, содержащемся в контейнере объектов SIMATIC 400(1), блок данных DB30, представленный в таблице 6.4. Сохраните этот блок и закройте окно редактора с этим блоком.

Табл. 6.4 Набор данных для аналогового входного модуля для параметрирования области измерений на +/- 2,5 V

№ байта	Имя	Тип	Начальное значение	Комментарий
0.0	STRUCT			
+0.0	AlarmEnable	BYTE	B#16#00	Граничное значение - /диагностический сигнал
+1.0	IntTime	BYTE	B#16#02	Время интегрирования 20 ms
+2.0	M_Kgr_0	BYTE	B#16#15	Группа каналов 0 (напряжение +/- 2,5 V)
+3.0	M_Kgr_1	BYTE		Группа каналов 1 (не важно)
+4.0	M_Kgr_2	BYTE		Группа каналов 2 (не важно)
+5.0	M_Kgr_3	BYTE		Группа каналов 3 (не важно)
+6.0	Ogr_Kgr_0H	BYTE		Граничные значения не важны, так как они не деблокированы
+7.0	Ogr_Kgr_0L	BYTE		
+8.0	Ugr_Kgr_0H	BYTE		
+9.0	Ugr_Kgr_0L	BYTE		
+10.0	Ogr_Kgr_1H	BYTE		Не имеется
+11.0	Ogr_Kgr_1L	BYTE		Не имеется
+12.0	Ugr_Kgr_1H	BYTE		Не имеется
+13.0	Ugr_Kgr_0L	BYTE		Не имеется
=14.0	END_STRUCT			

Введите представленный на рисунке 6.16 вызов SFC55 WR_PARM в OB1 и запомните его.

```

CALL "WR_PARM"
  REQ :=M30.0           // Запуск задани
  IOID :=B#16#54        // Идент. периф. модуля входов
  LADDR :=W#16#200      // Адрес входного модуля (512 dec)
  RECNUM :=B#16#1       // Номер набора данных (DS1)
  RECORD :=P#DB30.DBX 0.0 BYTE 14 // Указатель на DS1 в DB30
  RET_VAL:=MW32
  BUSY :=M30.1

  A   M   30.1          // Подготовка к повторному запуску
  R   M   30.0          // задания

```

Рис. 6.16 Вызов SFC55 для перепараметрирования входного аналогового модуля

Загрузите OB1 через MPI-интерфейс.

После процесса загрузки CPU416-2DP находится в состоянии RUN и светодиоды для DP-интерфейса не светятся и не мигают ("SF DP" и "BUSF"). Это имеет силу также для индикаторных светодиодов станции ET200M. Если это так, начинается корректный обмен пользовательскими данными между DP-Master'ом и станцией ET200M.

6.4.3 Тестирование перепараметрирование аналогового входного модуля с помощью SFC55 WR_PARM

С помощью функции STEP7 *Monitor/Modify Variables* (см. раздел 6.2.3) Вы можете вызвать перепараметрирование области измерений входного аналогового модуля в ET200M с +/-10 V на +/-2,5 V с помощью запрограммированной SFC55 и контролировать обработку этой SFC.

Задайте в таблице переменных в качестве операндов переменные MB30 (M30.0 = REQ и M30.1 = BUSY) и MW32 (RET_VAL). Для MB30 задайте значение B#16#01. Активизируйте показ значений. Значение для MB30 пока B#16#00, а значение RET_VAL – W#16#7000. Активизируйте задание значений для MB30, при этом стартует SFC55.

Если функция обрабатывается без ошибок, в обеих переменных после процесса управления находятся выходные значения.

Замечание: Приведенное здесь перепараметрирование входного аналогового модуля *пропадет* после нового запуска DP-Master-системы. При новом запуске Master'а аналоговый модуль параметрируется статическими наборами данных, хранящихся в SDB в DS1.

6.4.4 Перепараметрирования аналогового входного модуля с помощью SFC56 WR_DPARM

Для передачи первоначально созданных при проектировании с помощью HW-Config параметров модуля, которые хранятся в наборе данных DS1 во входной аналоговый модуль входов, используется SFC56 WR_DPARM. Эта SFC передает в модуль заранее определенную и хранящуюся в CPU в соответствующем SDB DS1.

Для этого создайте изображенный на рисунке 6.17 вызов SFC56 WR_DPARM в представлении STL в OB1 для SIMATIC 400(1). Сохраните блок и закройте окно с этим блоком в редакторе LAD/FBD/STL.

```
CALL "WR_DPARM"
  REQ      := M40.0           // Запуск задани
  IOID     := B#16#54         // Идент. периф. модуля входов
  LADDR    := W#16#200        // Адрес входного модуля (512 dec)
  RECNUM   := B#16#1          // Номер набора данных (DS1)
  RET_VAL  := MW42
  BUSY     := M40.1

  A   M   40.1                // Повторный запуск задания
  R   M   40.0
```

Рис. 6.17 Вызов SFC56 WR_DPARM в OB1

Вернитесь снова в SIMATIC Manager и передайте все блоки в CPU, используя MPI-связь между PG/PC и контроллером.

После процесса передачи должен CPU416-2DP находиться в рабочем состоянии RUN и светодиоды, отвечающие за интерфейс DP (“SF DP” и “BUSF”) не должны гореть или мигать. Светодиоды на станции ET200M тоже не должны гореть. Если это так, то начинается корректный обмен пользовательскими данными между DP-Master’ом и станцией ET200M.

6.4.5 Тестирование перепараметрирования аналогового входного модуля с помощью SFC56 WR_DPARM

С помощью функции STEP7 *Monitor/Modify Variables* Вы можете теперь контролировать процесс восстановления параметров аналогового входного модуля в ET200M.

Задайте для этого в таблице переменных обе переменные MB40 (M40.0 = REQ и M40.1 = BUSY) и MW42 (RET_VAL). Для MB40 задайте значение B#16#01. Активизируйте наблюдение переменных. Значение для MB40 должно пока быть равно B#16#00, а значение для MW42 (RET_VAL) – B#16#7000. Активизируйте теперь заданные управляющие значения для MB40. Этим Вы запустите SFC56. Если функция обрабатывается без ошибок, в обеих переменных после процесса управления находятся выходные значения.

6.5 Вызов (запуск) управляющих DP-функций SYNC/FREEZE

Управляющие команды SYNC (синхронизация выходов) и FREEZE (замораживание выходов) предлагают пользователю возможность корректировать обмен данными с несколькими Slave’ами. DP-Master с соответствующей функциональной возможностью может одновременно посылать управляющие команды (телеграммы Broadcast) SYNC и/или FREEZE. DP-Slave’ы при этом объединяются в SYNC –и FREEZE-группы. Для Master-системы может быть образовано максимум 8 групп. Каждый DP-Slave может быть назначен только одной группе.

Управляющая команда SYNC позволяет пользователю синхронизировать выходы нескольких Slave’ов. С получением управляющей команды SYNC подключают рассматриваемые DP-Slave’ы на выходы к DP-Master’у данные, сохраненные в буфере передачи после последней телеграммы Data_Exchange. Это позволяет осуществить одновременную активизацию (синхронизацию) выходных данных нескольких DP-Slave’ов. На рисунке 6.18 показано принципиальное протекание команды.

С помощью команды UNSYNC отменяется режим SYNC на рассматриваемом DP-Slave. После этой команды DP-Slave находится снова в циклическом обмене данными, то есть посылаемые от DP-Master’а данные *немедленно* подключаются на выходы.

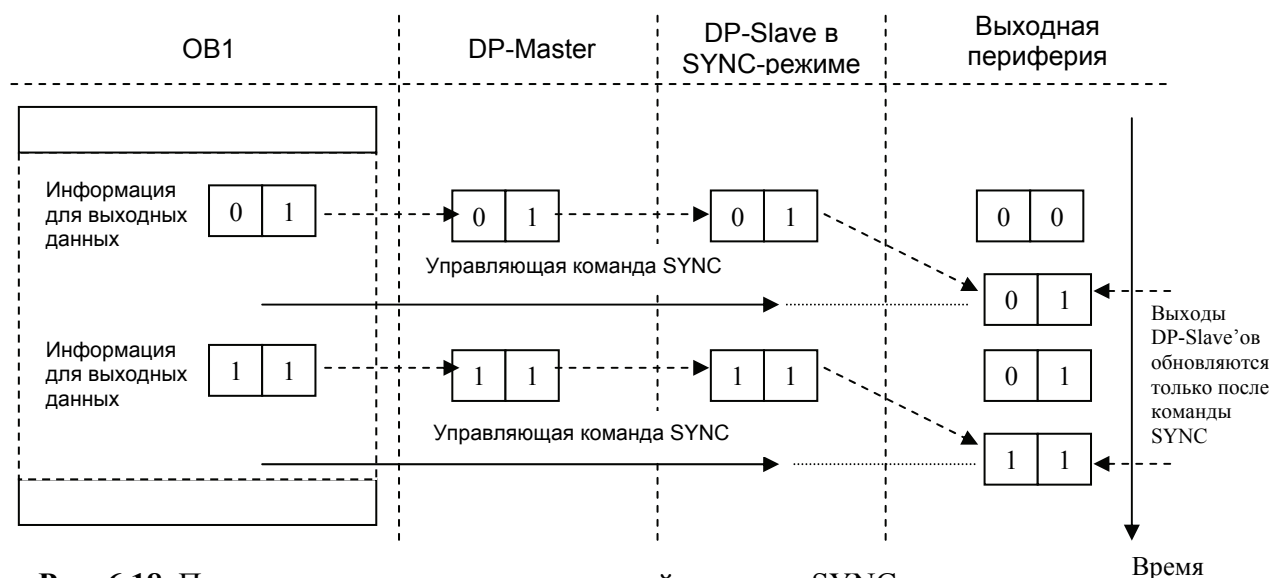


Рис. 6.18 Принцип выполнения управляющей команды SYNC

Управляющая команда FREEZE позволяет пользователю “замораживать” входные данные DP-Slave’a. Если группе DP-Slave’ов послана команда FREEZE, то одновременно замораживаются сигналы всех входах, так что вслед за этим DP-Master может их читать. Актуализация входных данных DP-Slave’ов происходит после нового приема команды FREEZE. Рисунок 6.19 иллюстрирует выполнение команды FREEZE.

Управляющая команда UNFREEZE отменяет режим FREEZE на рассматриваемом DP-Slave так, что он опять переходит в циклический режим обмена данными с DP-Master’ом. Входные данные от DP-Slave’a немедленно актуализируются и могут быть сразу же прочитаны DP-Master’ом.

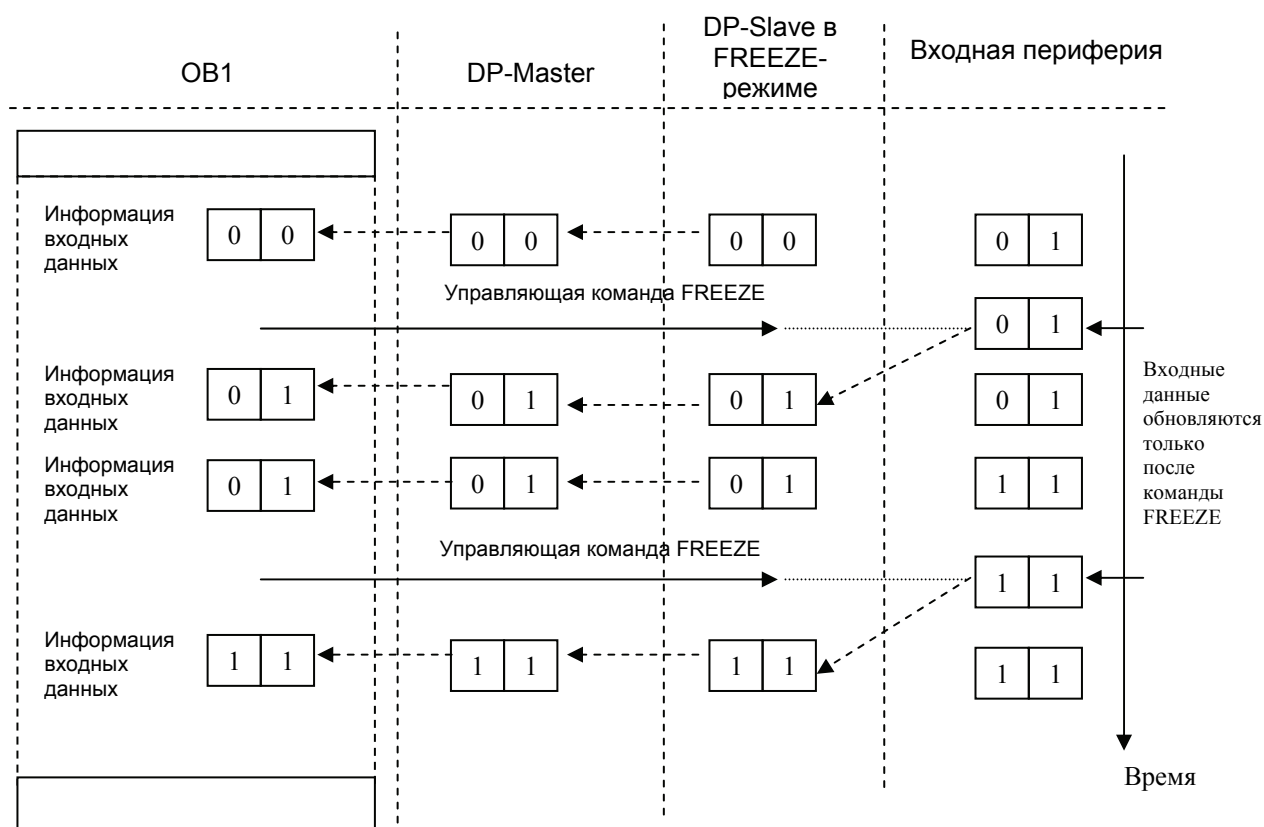


Рис. 6.19 Принцип выполнения управляющей команды FREEZE

6.5.1 Пример применения команд SYNC/FREEZE с DP-Master'ом IM467

С помощью ниже следующего примера практического применения поясняется использование управляющих команд.

Чтобы создать конфигурацию установки, откройте вначале SIMATIC Manager и выберите *File->New*. Создайте новый проект с именем “SYNCFR” и нажмите кнопку OK. Вставьте затем с помощью *Insert->Station->SIMATIC400-Station* новую станцию S7-400.

Вставьте теперь из каталога аппаратных средств носитель модулей (Rack) “UR2”. Разместите на 1-е место источник питания “PS407 10A”. При выборе CPU нужно обратить внимание на то, чтобы он поддерживал функции SYNC и FREEZE. Поэтому выберите, например, CPU 416-1 с заказным номером 6ES7416-1XJ02-0AB0 и разместите его на месте номер 3.

Для проектирования модуля DP-Master'а (IM467) перейдите в аппаратном каталоге для SIMATIC 400 к подкаталогу “IM-400”. Выберите там модуль IM 467 с заказным номером 6ES7467-5GJ01-0AB0 и установите его на место номер 4 (рисунок 6.20).

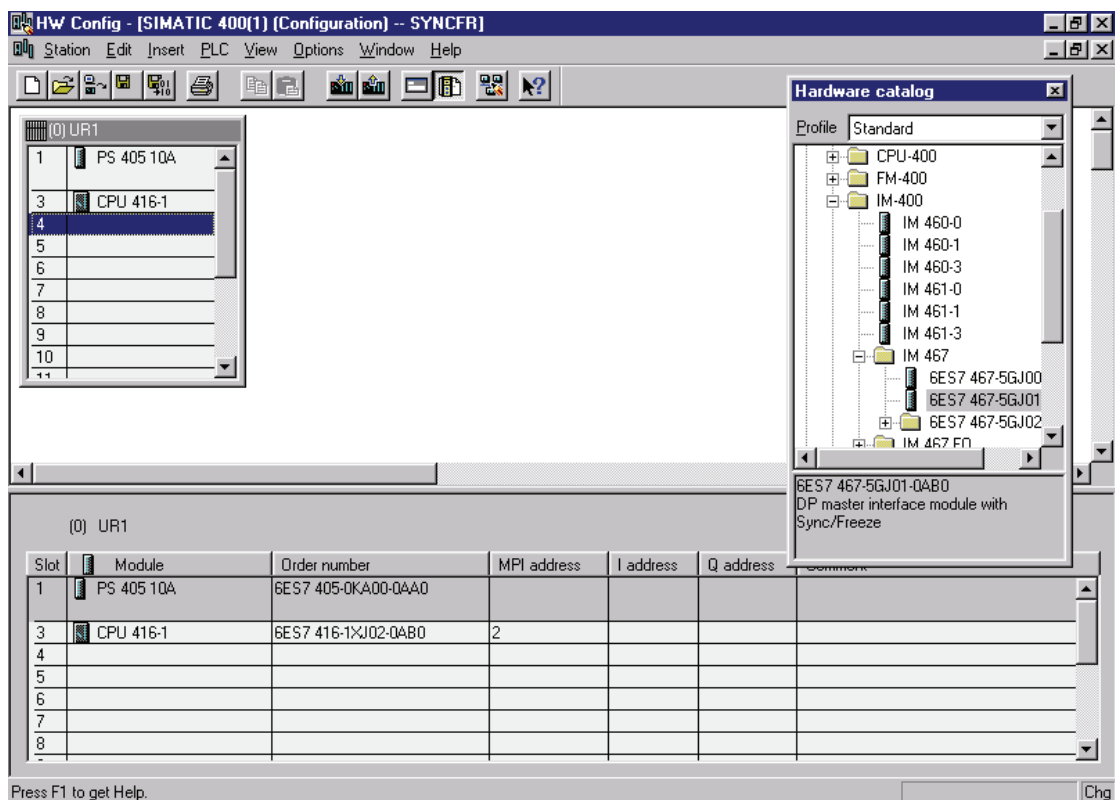


Рис. 6.20 Выбор IM467 из Hardware catalog

При размещении модуля в носителе модулей автоматически появляется окно “Properties-PROFIBUS interface IM 467”, закладка “Parameters”. Выберите “New” и подтвердите выбор в диалоговом боксе с помощью кнопки “OK”. Таким образом создается новая подсеть PROFIBUS со скоростью 1,5 МБод и профилем шины DP. Выберите для IM 467 предлагаемый адрес участника “2”. Закройте окно с помощью “OK”. Модуль IM 467 вставлен на установочное место “4” и графически показана DP-Master-система для IM 467 (рисунок 6.21).

В качестве Slave’а спроектируйте теперь станцию ET200B, которая поддерживает управляющие команды SYNC и FREEZE. Откройте для этого в аппаратном подкаталоге для модулей PROFIBUS-DP и выберите из подкаталога “ET200B” модуль “B-16DI”. Перетяните модуль на графически представленную DP-Master-систему IM 467. При этом откроется окно “Properties-PROFIBUS Interface B-16DI”. Выберите в качестве адреса PROFIBUS “3” и покиньте окно с помощью OK.

Перетяните теперь из аппаратного каталога *PROFIBUS-DP->ET200B* на Master-систему IM 467 модуль “B-16DO”. Задайте в следующем окне “ Properties-PROFIBUS Interface B-16DO” PROFIBUS-адрес “4” и закройте окно с помощью “OK”.

Таким образом DP-Master-система IM 467 для примера полностью спроектирована.

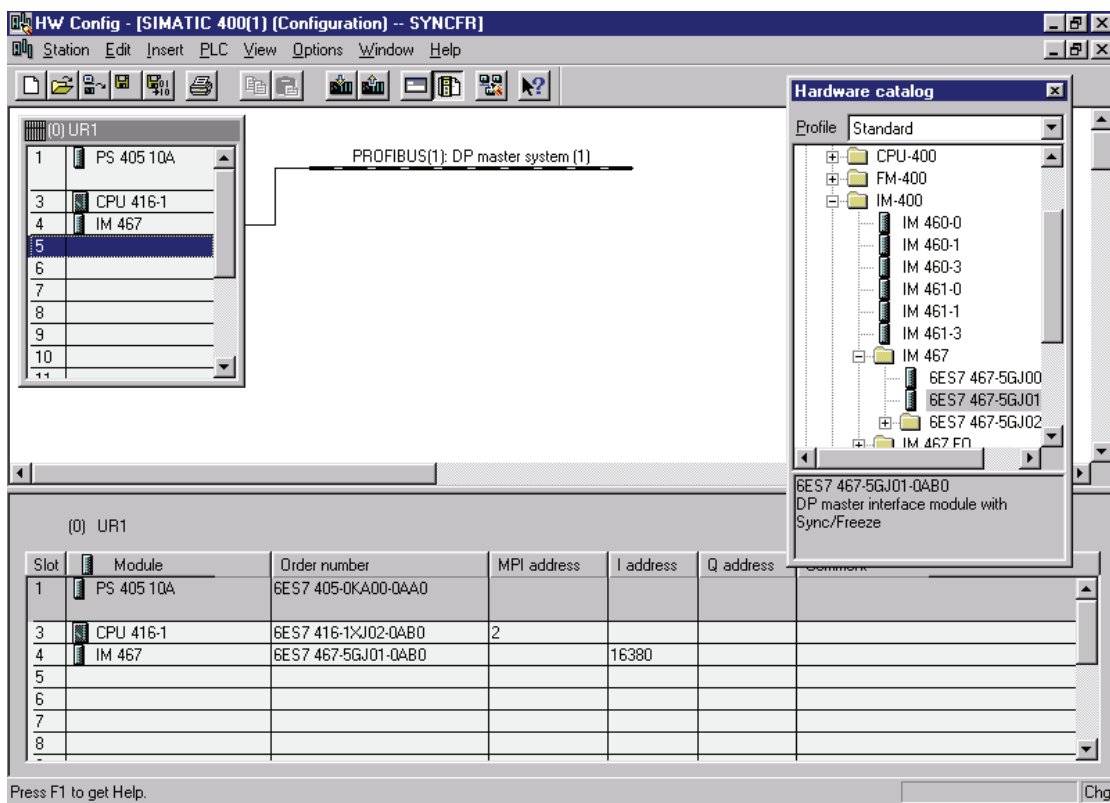


Рис.6.21 Аппаратная конфигурация с IM467

Теперь должны быть определены установки для функций SYNC-/FREEZE.

Выберите для этого с помощью щелчка мышью графически представленную DP-Master-систему PROFIBUS(1). Появляется окно “Properties-DP master system”, закладка “Group assignment”. В этом окне можно назначить различные группы DP-Slave’ов, способных к командам SYNC-/FREEZE (рисунок 6.22). В первой колонке таблицы показаны DP-Slave’ы, спроектированные для DP-Master-системы (в скобках указан адрес PROFIBUS). В колонках с 1 по 8 показано 8 возможных групп, к которым могут быть отнесены DP-Slave’ы.

Выберите на закладке “Group assignment” вначале “Properties”, чтобы установить свойства используемых групп. Окно “Properties” открыто. В колонке “Comment”, Вы можете установить для каждой группы дополнительный текст (комментарий/обозначение группы). В колонке “Properties” Вы выбираете, какие функции назначены группе. Параметрируйте группу так, как показано на рисунке 6.23. Группа 1 при этом запараметрирована как FREEZE-группа, группа 2 – как SYNC-группа. Покиньте окно с помощью “OK”.

Вы вновь находитесь в окне “Properties-DP master system”, закладка “Group assignment”. Откройте (с помощью щелчка мышью) теперь станцию B-16DI. Теперь Вы можете назначить DP-Slave’ов группе 1. После этого отметьте курсором станцию B-16DO и укажите ей группу 2 (рисунок 6.24). Сохраните установки с помощью OK.

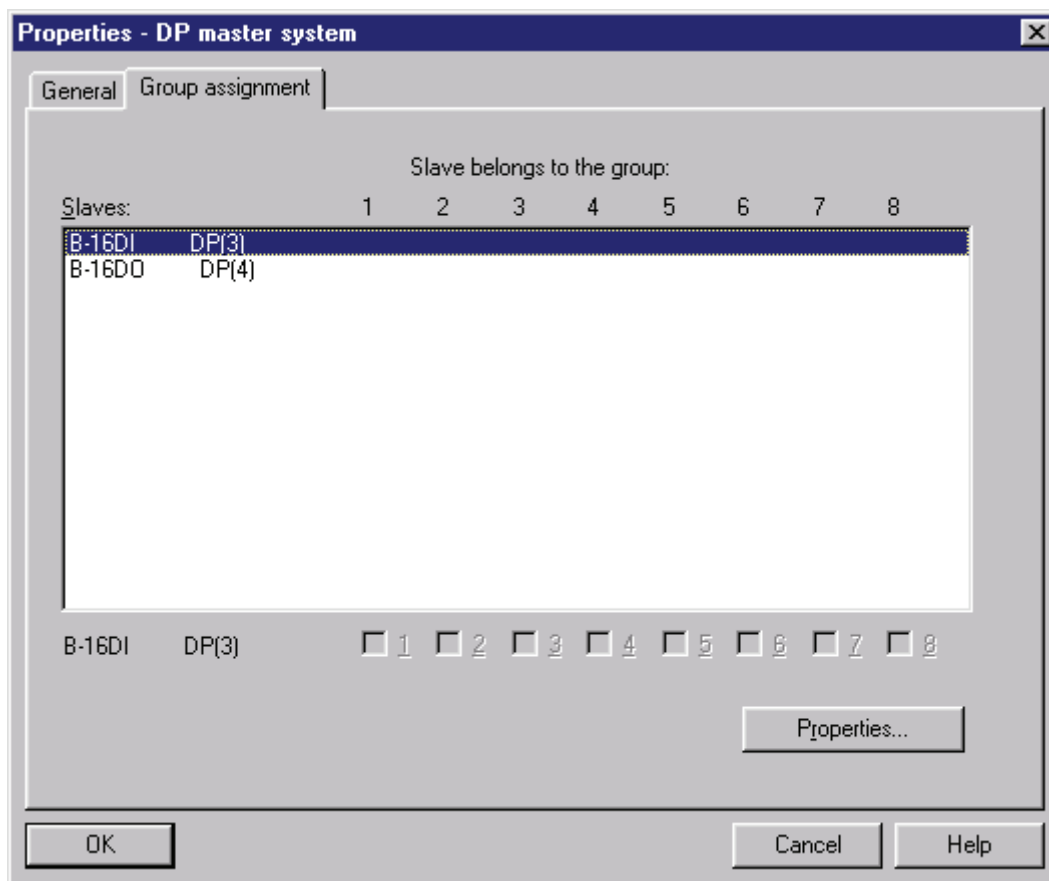


Рис. 6.22 Назначение групп в *HW Config*

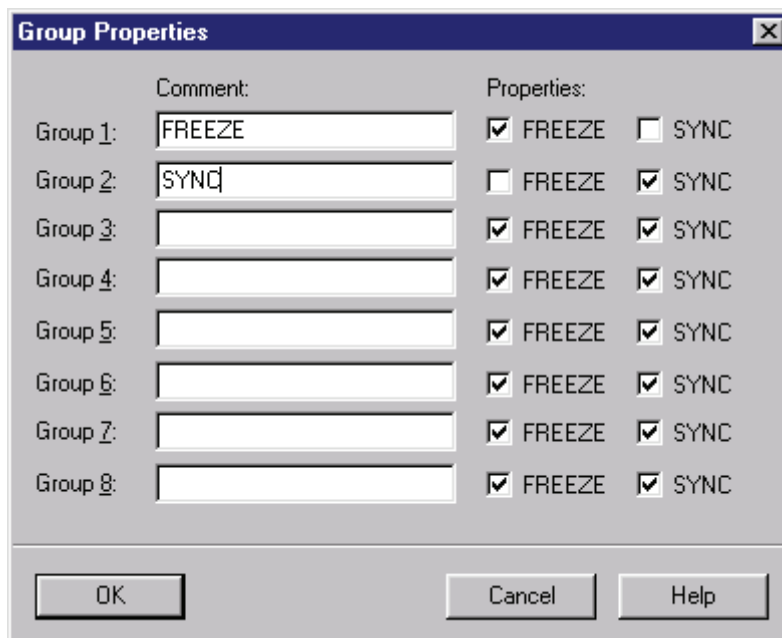


Рис.6.23 Свойства групп в *HW Config*

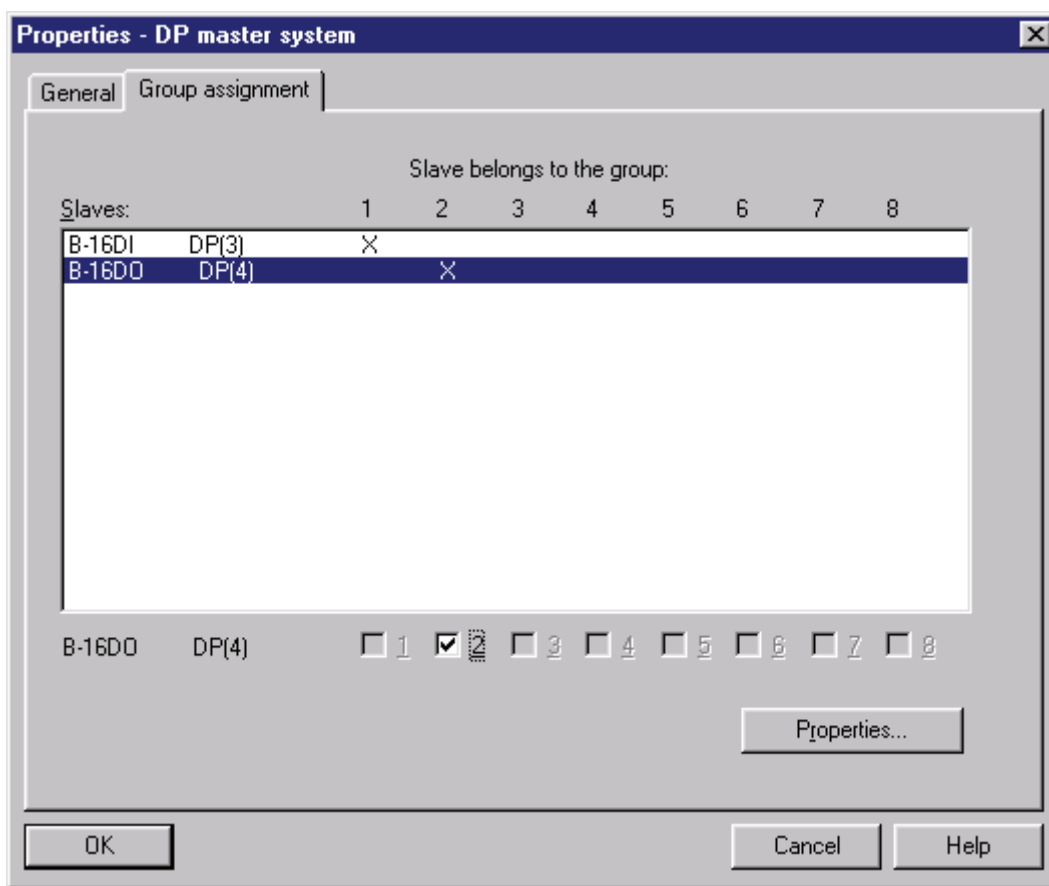


Рис.6.24 Проектирование групп для модулей ET 200B

Выберите теперь *STATION->Save and Compile*. Переведите проектируемую станцию в STOP и загрузите аппаратную конфигурацию в CPU S7-400.

Свяжите кабелем PROFIBUS IM 467 с обоими модулями ET200B и переведите CPU 416-1 в состояние RUN-P. CPU находится в состоянии RUN. Все красные светодиоды ошибок должны потухнуть. Закройте утилиту HW-Config.

6.5.2 Создание пользовательской программы для функций SYNC-/FREEZE

Теперь нужно запрограммировать функции SYNC-/FREEZE с помощью функции SFC11. В качестве примера запрограммируем SFC11 в OB1 и вызовем ее с помощью смены сигнала (фронта).

Выберите с помощью двойного щелчка находящийся в правом окне SIMATIC Manager CPU 416-1. Откройте объект, а в нем – папку “S7-Programm(1)”. В папке “S7-Programm(1)” откройте папку “Blocks”, в которой по умолчанию содержится OB1 (рисунок 6.25).

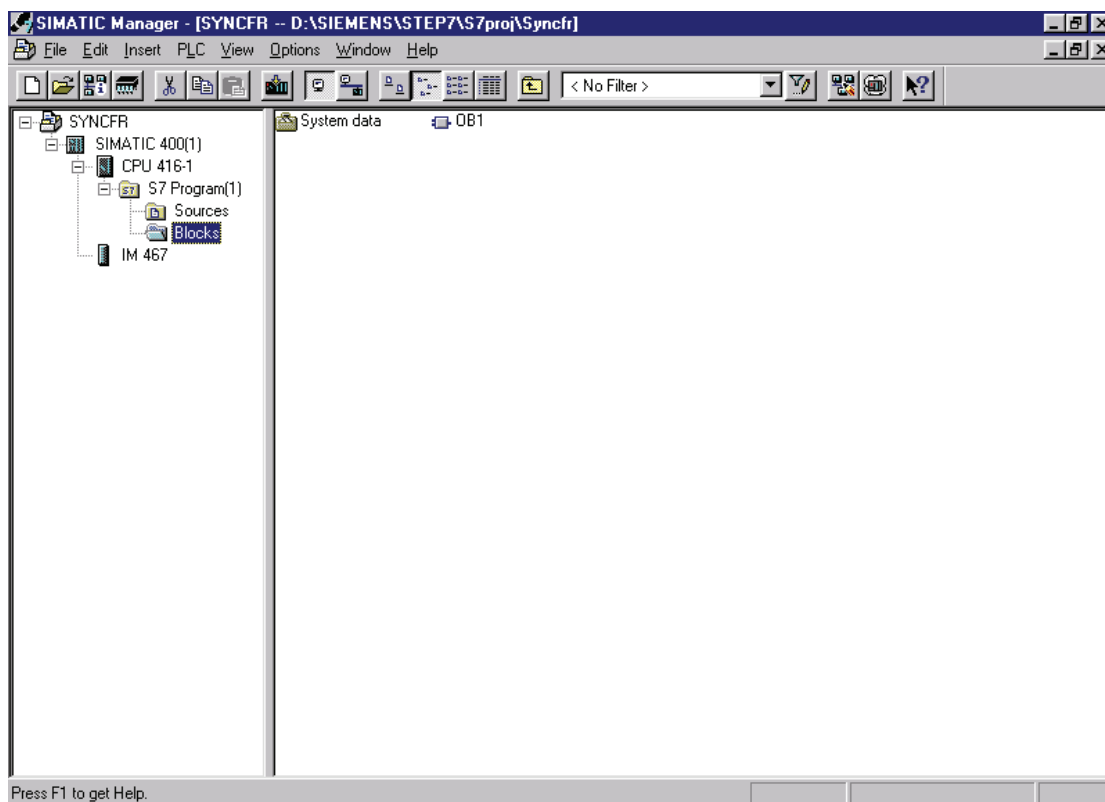


Рис.6.25 *SIMATIC Manager* с открытым контейнером блоков

Двойным щелчком откройте OB1. Появляется окно “ Properties OB1”. Нажмите на OK – запускается редактор LAD/FBD/STL для программирования OB1 в STL-представлении.

Чтобы установить SFC11 из “Standard Library” выберите *View->Catalog*. Появляется каталог блоков. Выберите в нем *Library->Standard Library->System Function Blocks*. Там есть SFC11 DPSYC_FR (рисунок 6.26).

Перетяните SFC11 в первую сеть OB1 и дополните STL-программу листингом, показанным на рисунке 6.27.

Сохраните и загрузите OB1 в CPU 416-1. Далее можно наблюдать и диагностировать программу с помощью Monitor/Modify Variables.

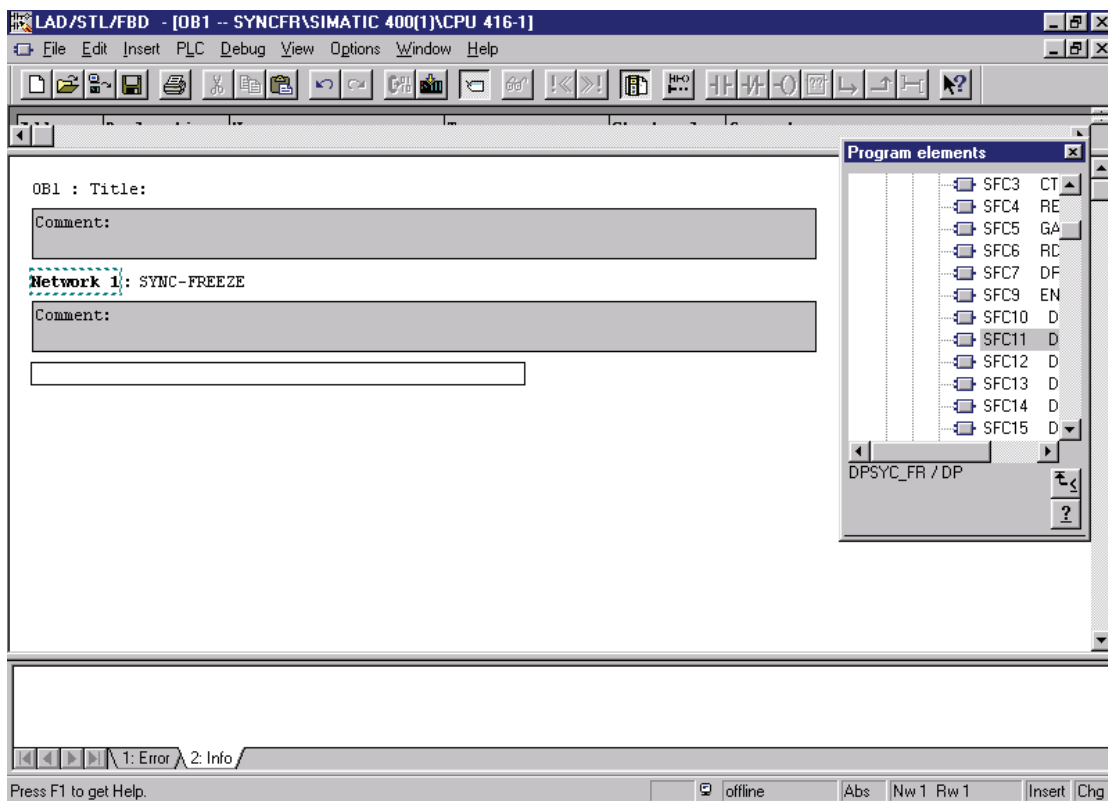


Рис.6.26 LAD/STL/FBD-редактор с каталогом программных элементов

```

A   M   10.0      //Оценка фронта для SFC11
FP  M   10.4      //Фронт положительный ???
=   M   10.2      //Маркер запуска(устанавливается на 1 цикл OB1)

GO1: CALL "DPSYC_FR"
      REQ :=M10.2      //Маркер запуска
      LADDR :=W#16#200 //Входной адрес IM467 (512dez)
      GROUP :=B#16#1   //Выбрана группа 1
      MODE  :=B#16#8   //Выбран режим FREEZE
      RET_VAL:=MW12     //RET_VAL в MW12
      BUSY  :=M10.3     //Флаг BUSY в M10.3

A   M   10.3      //SFC11 готова ?? Если нет, тогда
JC  GO1          //переход на метку GO1

A   M   10.1      //Оценка фронта для SFC11
FP  M   10.5      //Фронт положительный ???
=   M   10.6      //Маркер запуска(устанавливается на 1 цикл OB1)

GO2: CALL "DPSYC_FR"
      REQ :=M10.6      //Маркер запуска
      LADDR :=W#16#200 //Входной адрес IM467 (512dez)
      GROUP :=B#16#2   //Выбрана группа 2
      MODE  :=B#16#20  //Выбран режим SYNC
      RET_VAL:=MW14     //RET_VAL в MW14
      BUSY  :=M10.7     //Флаг BUSY в M10.7

A   M   10.7      //SFC11 готова ?? Если нет, тогда
JC  GO2          //переход на метку GO2

```

Рис. 6.27 Листинг OB1 с SFC11 *DPSYC_FR*

Для этого откройте утилиту Monitor/Modify Variables и в таблицу переменных введите строки, показанные на рисунке 6.28.

Address	Symbol	Monitor	Format	Monitor Value	Modify Value
QB 0	---		HEX		
IB 0	---		HEX		
M 10.0	---		BIN		2#1
M 10.1	---		BIN		2#1

Рис. 6.28 Таблица переменных для тестирования SFC11 *DPSYC FR*

После запуска шинной системы DP все Slave'ы находятся в циклическом обмене данными. С установкой меркеров 10.0 и 10.1 в состояние "1" запускаются управляющие команды SYNC и FREEZE.

ET200B/16DI находится теперь в режиме FREEZE, а ET200B/16DO – в режиме SYNC. Изменения входных сигналов в станции ET200B/16DI теперь "сами по себе" в CPU не передаются. В окне Monitor/Modify Variables не будут показываться изменения входных сигналов. Значение, которое записывается для QB0, также не выдается на выходы ET200B/16DO. Только когда меркеры запуска задания M10.0 и M10.1 переходят из состояния "0" в состояние "1", управляющие команды SYNC и FREEZE снова запускаются. Тем самым выдаются на ET200B/16DO переданные значения и считываются из ET200B/16DI актуальные входные данные.

Заметьте, что во время выполнения SFC11 (BUSY="1") выходы DP-Slave'ов, для которых используется SFC11, не могут изменяться пользовательской программой. Поэтому рекомендуется SFC11 программировать или в цикле (опрос BUSY) или использовать функцию "Part Process Image".

6.6 Обмен данными с помощью перекрестной связи

Функция перекрестной связи позволяет осуществить прямую передачу входных данных одного DP-Slave'а к другому DP-Slave'у или DP-Master'у (класса 2). Это делается возможным благодаря тому, что DP-Slave посылает DP-Master'у свои телеграммы-ответы (Response-телеграммы) не через связь one-to-one, а через связь one-to-many (рисунок 6.29).

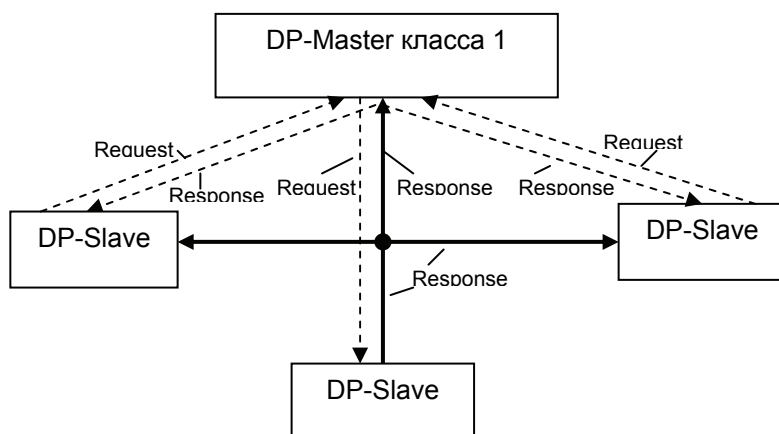


Рис. 6.29 Response-телеграмма DP-Slave при перекрестной связи

Проектирование перекрестной связи осуществляется в HW-Config и возможно только для таких DP-участников (Master/Slave), которые поддерживают эту функцию.

В описанном ниже примере с помощью S7-300 CPU315-2DP, как DP-Master'а и DP-Slave'ов, показана возможность обмена данными Slave'а со Slave'ом и Slave'а с Master'ом.

Чтобы создать необходимую конфигурацию установки, откройте вначале SIMATIC Manager и выберите *File->New*. Задайте для проекта имя "Querverkehr" и покиньте окно через "OK". Затем вставьте с помощью *Insert->Station->SIMATIC 300-Station* новую станцию S7-300, которой дайте имя "DP-Master". Тем же способом затем вставьте еще три станции с именами "I-Slave 5", "I-Slave 6" и "DP-Master/Inputs" (см. рисунок 6.30).

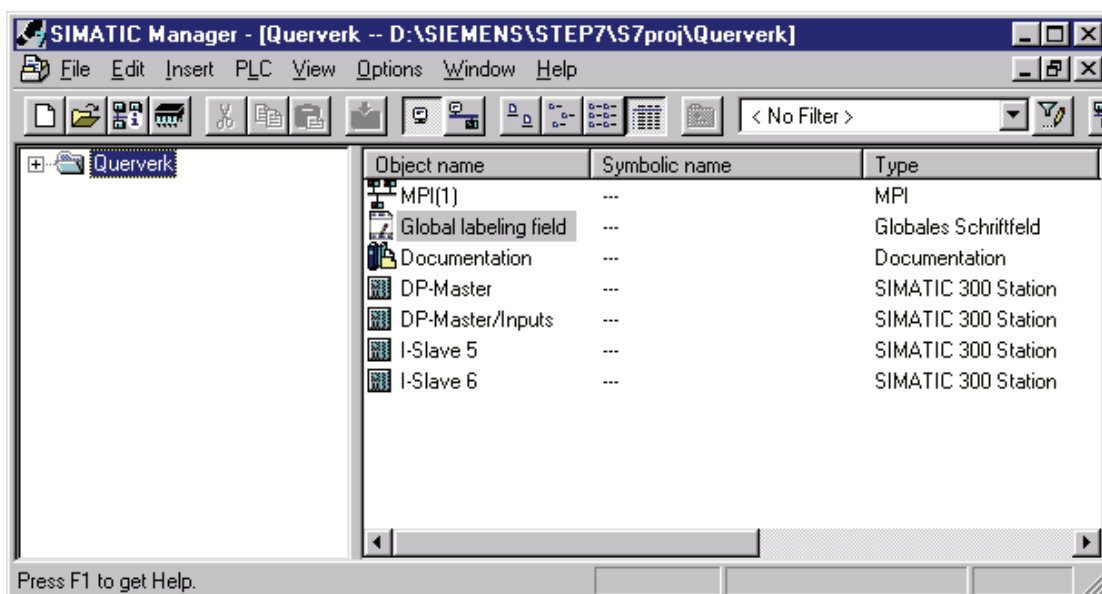


Рис. 6.30 Проект "Querverk" с принадлежащими ему станциями SIMATIC 300

Откройте теперь с помощью двойного щелчка на контейнере объектов “I-Slave 5” первую S7-300-DP-Slave-станцию. В правом окне SIMATIC Manager появляется объект Hardware. Откройте двойным щелчком на нем Hardware-конфигурацию для соответствующей станции SIMATIC 300.

Вставьте теперь из Hardware Catalog SIMATIC 300 Rack-300. На установочном месте 1 разместите источник питания “PS 307-2A”. При выборе CPU необходимо обратить внимание, чтобы он поддерживал перекрестную связь. Поэтому выберите CPU 315-2DP с заказным номером 6ES7315-2AF03-0AB0 и разместите его на установочном месте 2.

При размещении CPU в носителе модулей автоматически возникает окно “Properties-PROFIBUS interface DP-Master”, закладка “Parameters”. Измените предустановленный PROFIBUS-адрес на “5” и выберите новую подсеть. Квитируйте следующее окно “Properties-PROFIBUS”, закладку “General” с помощью “OK”. Подтвердите следующую закладку “Parameters” с помощью “OK”. Таким образом создана новая PROFIBUS-подсеть со скоростью передачи 1,5 МБод и с профилем “DP”. Двойным щелчком на DP-Master-интерфейсе CPU 315-2DP вызовите окно “Properties DP Master”. Установите там в закладке “Operation Mode” DP-интерфейс CPU на “DP-Slave”. Выберите теперь закладку “Configuration”. Здесь установите в форме таблицы все необходимые параметры для обмена данными для I-Slave’a. В столбце “Mode” установите I/Q-области, данные которых должны обмениваться или через коммуникационную связь “MS” (Master-Slave) или через перекрестную связь “DX” (Direct Data Exchange). Внесите представленные на рисунке 6.31 параметры и покиньте окно через OK. Запомните HW-Config для данного Slave.

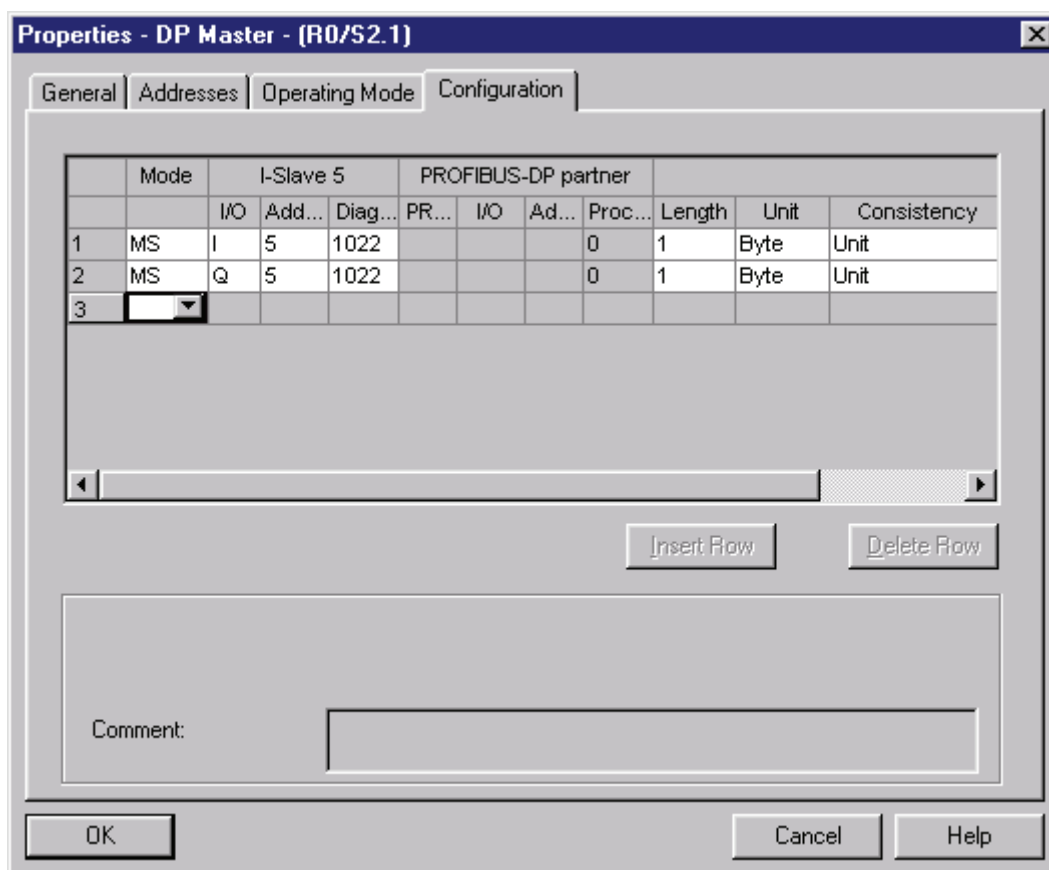


Рис. 6.31 “Конфигурация” I-Slave 5

Спроектируйте теперь в SIMATIC Manager тем же способом I-Slave 6. Установите для него PROFIBUS-адрес 6 и подключите Slave на уже имеющуюся PROFIBUS-подсеть “PROFIBUS(1)”. Устанавливаемые в закладке “Configuration” значения представлены на рисунке 6.32. Сохраните проектирование для I-Slave 6.

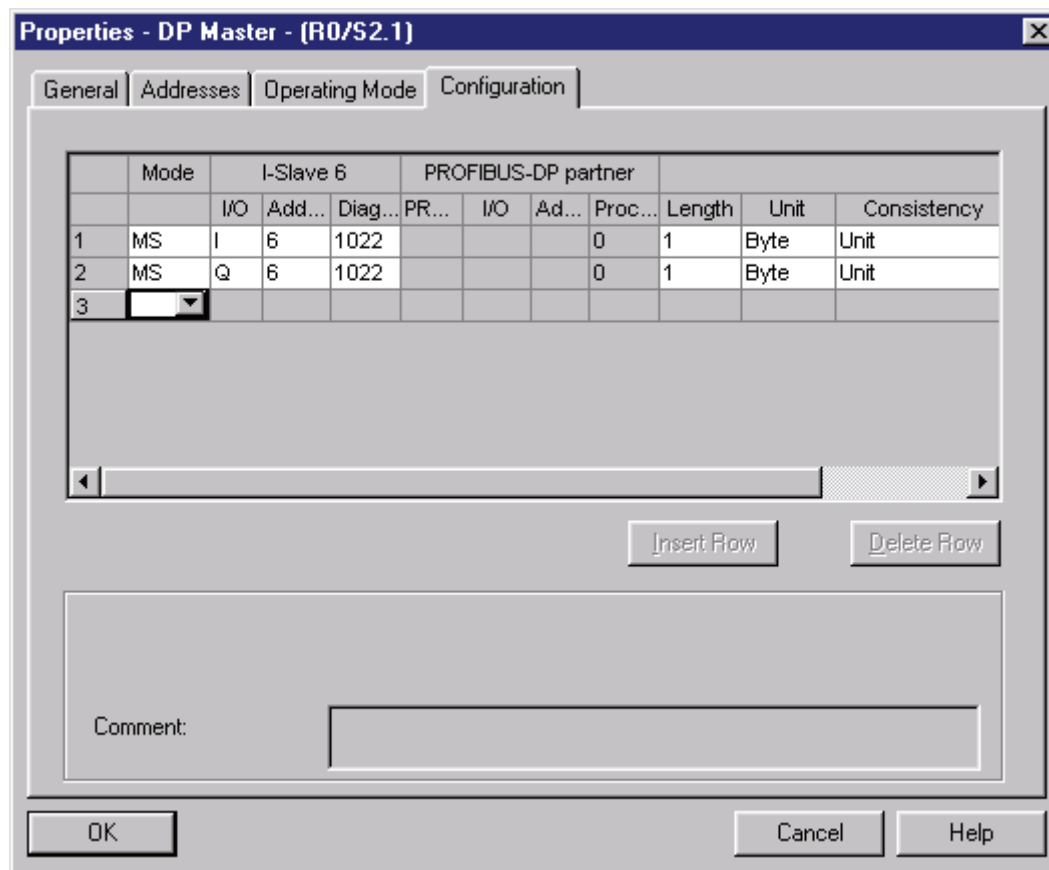


Рис.6.32 “Конфигурация” I-Slave 6

Тем же способом спроектируйте теперь аппаратную конфигурацию для станции S7-300 “DP-Master”. Задайте для этой станции PROFIBUS-адрес 2 и свяжите Master’а с уже имеющейся PROFIBUS-подсетью “PROFIBUS(1)”. Так как здесь речь идет о DP-Master’е, то оставьте вид работы “DP-Master”.

На следующем шаге присоедините обе спроектированные DP-Slave-станции “I Slave 5” и “I Slave 6” к подсети PROFIBUS DP Master’а. Для этого откройте Hardware Catalog, раздел “ PROFIBUS DP”, подкаталог “Configured Stations” и подключите CPU 31x-2DP посредством Drag&Drop к DP-Master-системе. В открывающемся окне “DP-Slave properties” (рисунок 6.33) выберите в закладке “Connection” станцию “I-Slave 5” и свяжите ее с помощью кнопки “Connect” с DP-Master-системой.

Дополните затем в закладке “Configuration” конфигурацию входов/выходов, как показано на рисунке 6.34 для “I-Slave 5” (группа столбцов “PROFIBUS-DP Partner”). Покиньте окно “DP-Slave properties” с помощью кнопки “OK”.

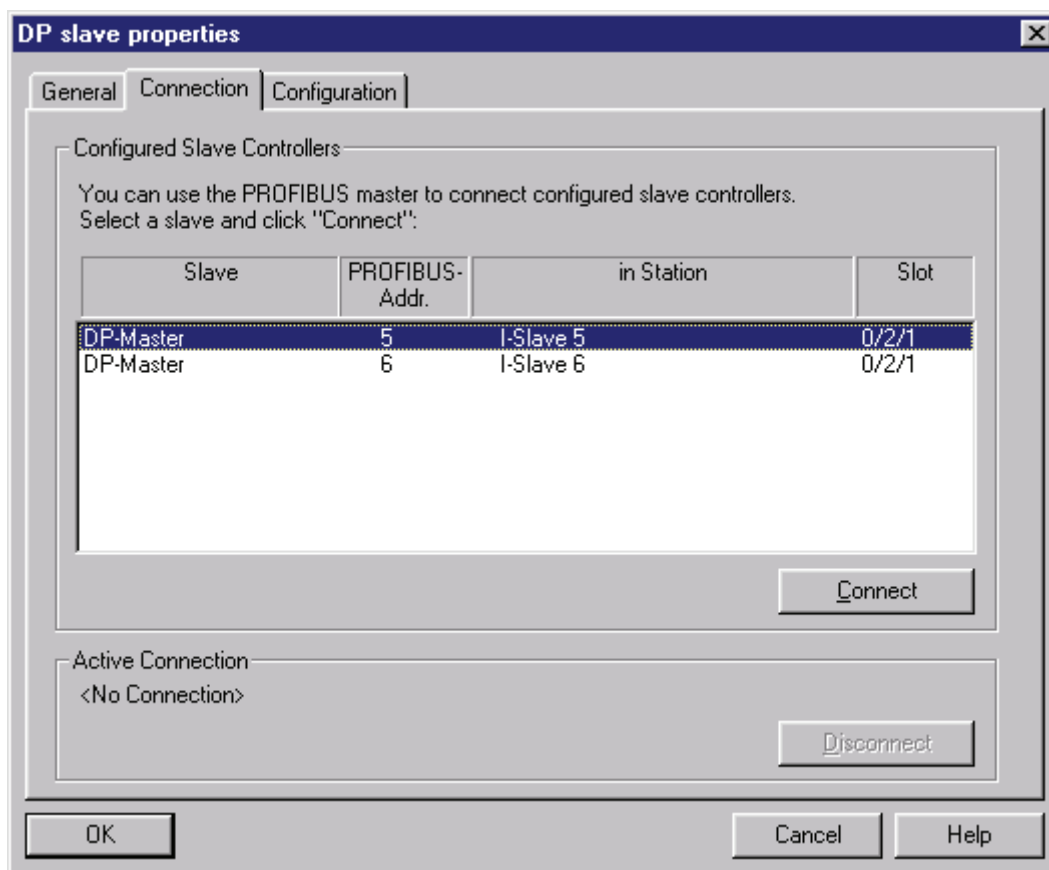


Рис. 6.33 Присоединение I-Slave 5 к подсети PROFIBUS

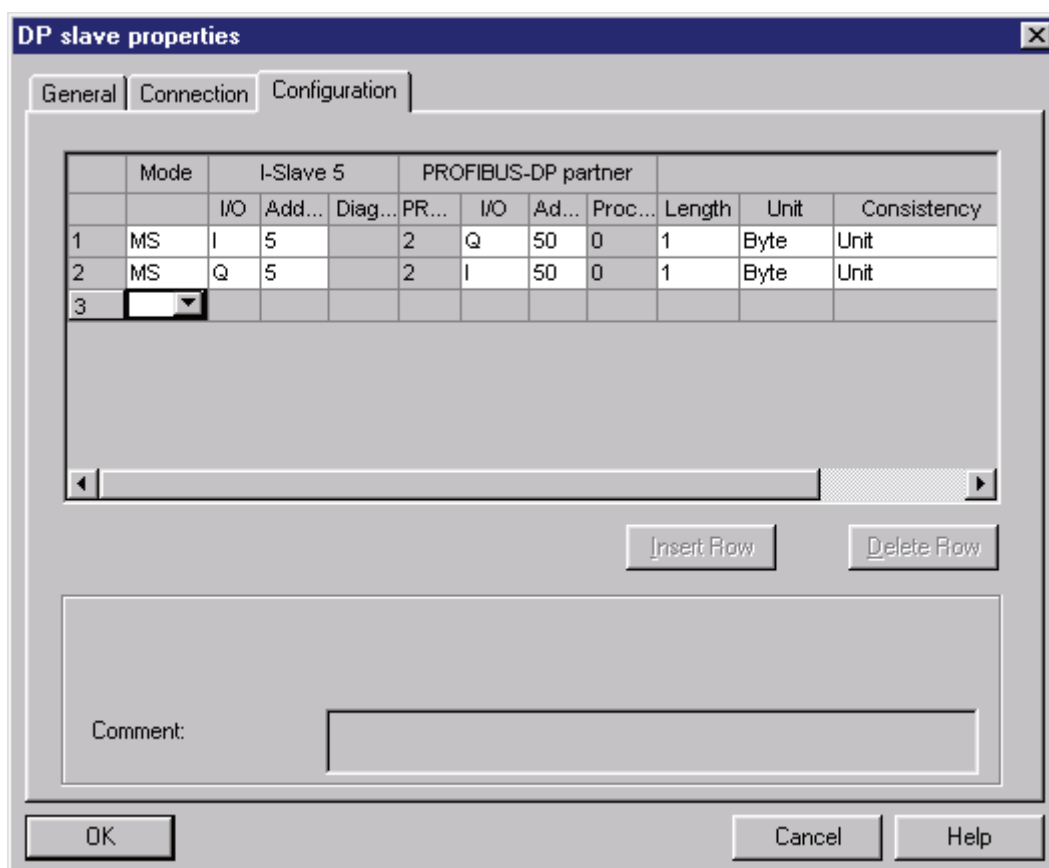


Рис. 6.34 I/O-конфигурация I-Slave 5

Подключите тем же способом станцию “I-Slave 6” к DP-Master-системе и дополните I/Q-конфигурацию, как показано на рисунке 6.35.

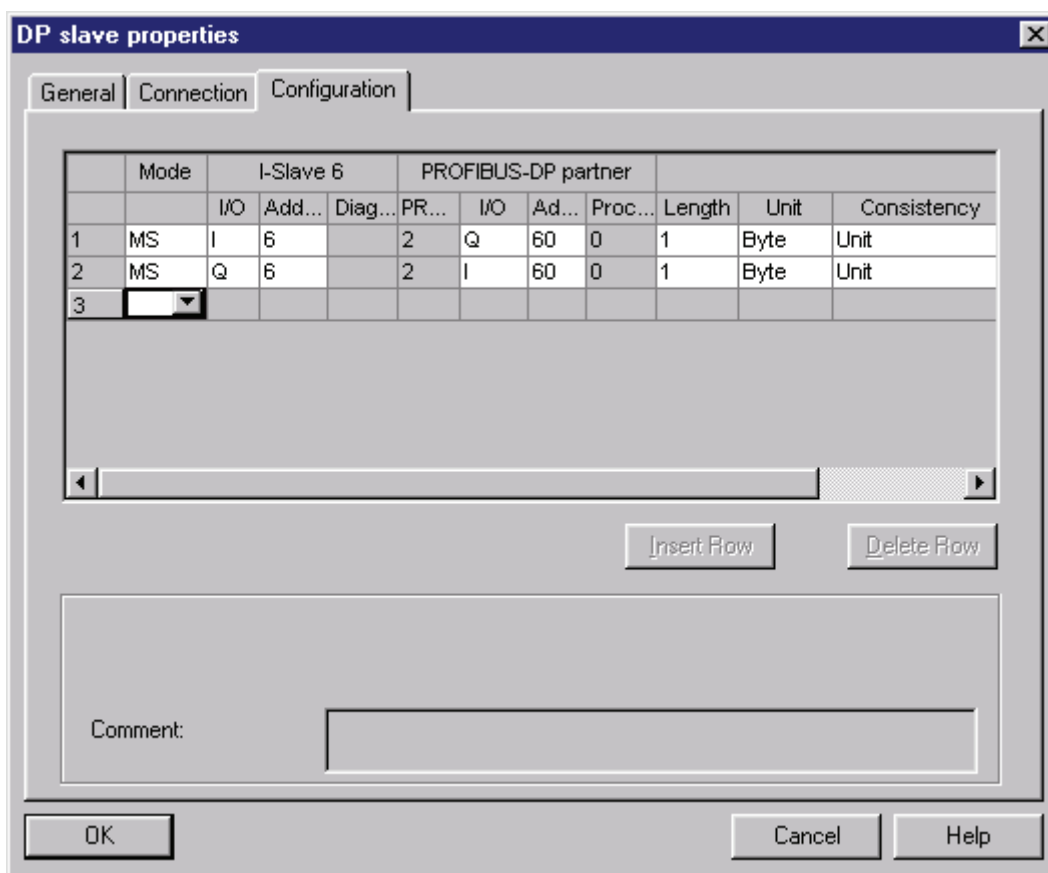


Рис. 6.35 I/O-конфигурация I-Slave 6

С помощью следующих шагов спроектируйте перекрестную связь от I-Slave 5 к I-Slave 6 и обратно. Откройте с помощью двойного щелчка в Hardware Config DP-Master закладку “Configuration” для “I-Slave 5”. В “DP-Slave properties” внесите в третью строку в столбце “Mode” для перекрестной связи “DX” и дополните, как показано на рисунке 6.36, параметры для перекрестной связи с I-Slave 6. Закройте окно через OK.

Для перекрестной связи от I-Slave 6 к I-Slave 5 поступайте таким же образом. Откройте закладку “Configuration” через двойной щелчок на I-Slave 6 и внесите параметры, приведенные на рисунке 6.37.

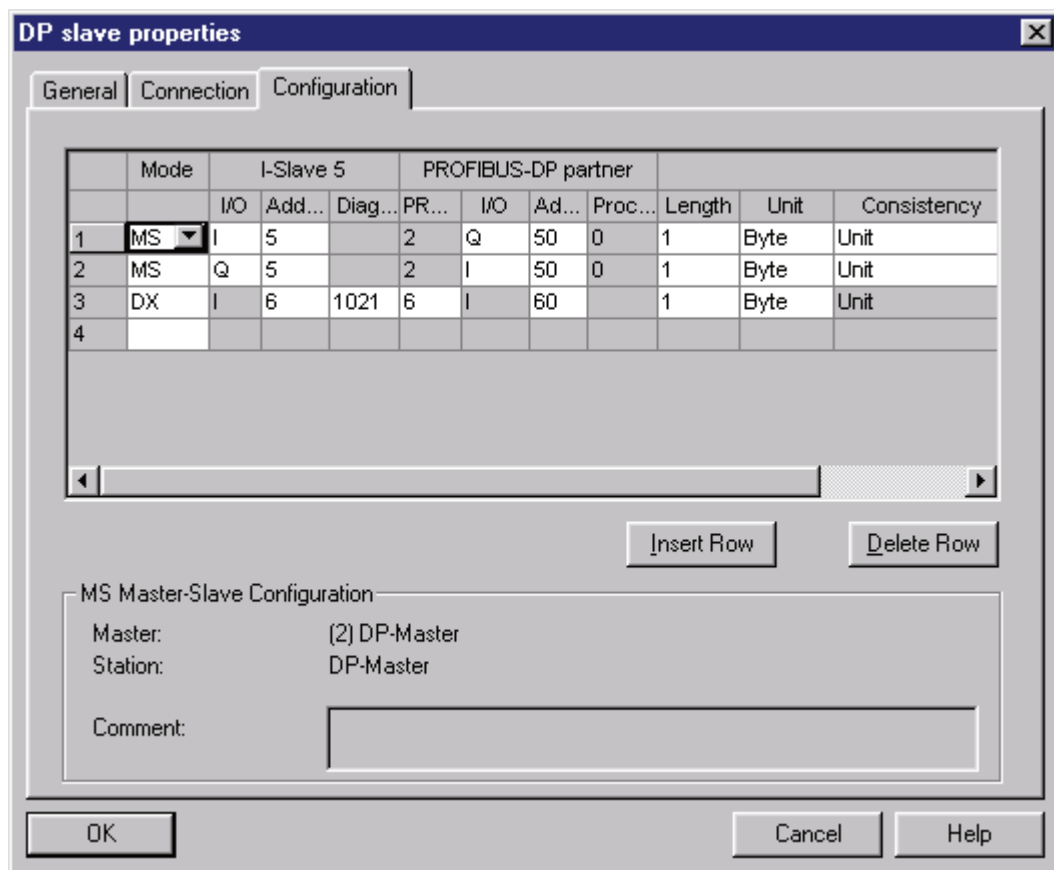


Рис. 6.36 Перекрестная связь I-Slave 5 с I-Slave 6

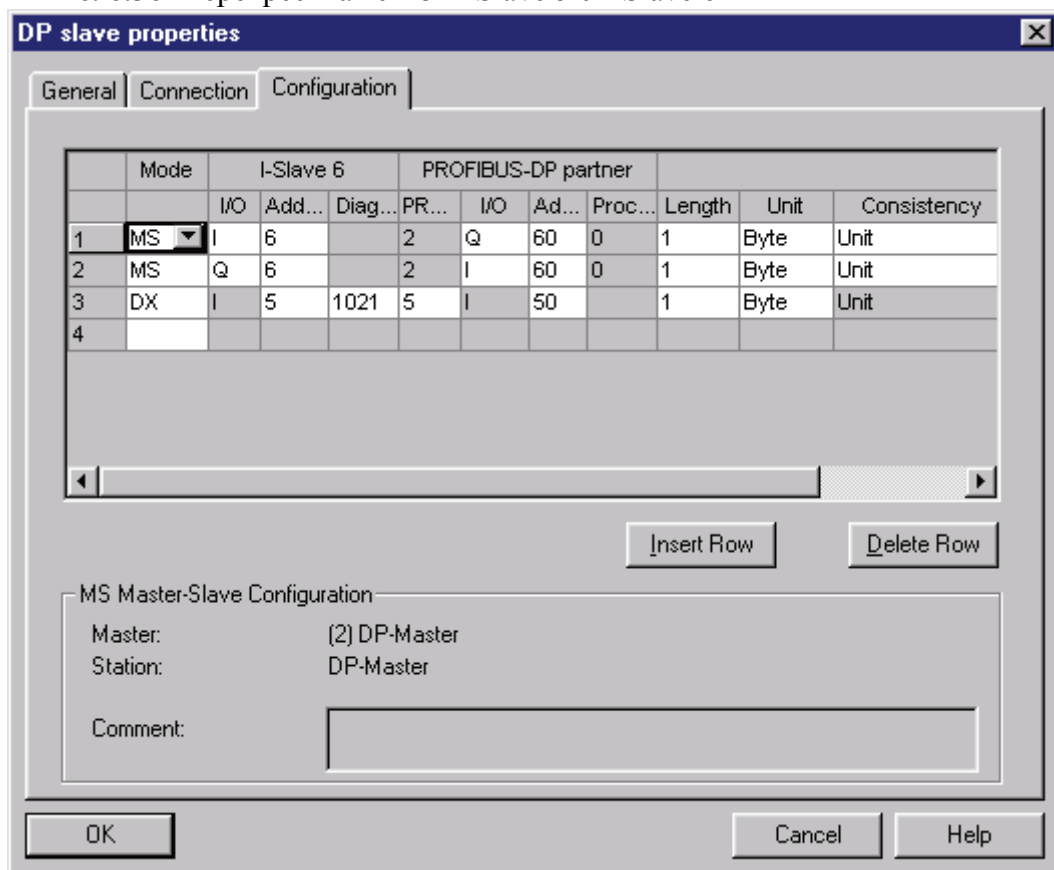


Рис. 6.37 Перекрестная связь I-Slave 6 с I-Slave 5

У перекрестной связи, спроектированной до сих пор, речь шла о связи Slave-Slave. Другой вариант для перекрестной связи представляет связь Slave к Master'у. Однако здесь имеется в виду не параметрированный мастер (мастер класса 1), а мастер класса 2, которому не принадлежат данные Slave'ы.

В примере этот мастер реализуется через станцию S7-300 “DP-Master/Inputs”. Создайте с помощью SIMATIC Manager DP-Master-станцию. В качестве CPU используйте также CPU S7-300: CPU315-2DP. Задайте этому мастеру шинный адрес 3 и свяжите его с уже имеющейся подсетью PROFIBUS. Через двойной щелчок на DP-Master-интерфейсе этой станции вызовите через окно “DP-Master properties” закладку “Configuration”. Внесите там представленные на рисунке 6.38 две пассивные перекрестные связи (режим “DX”) для I-Slave 5 и I-Slave 6 и покиньте окно через ОК.

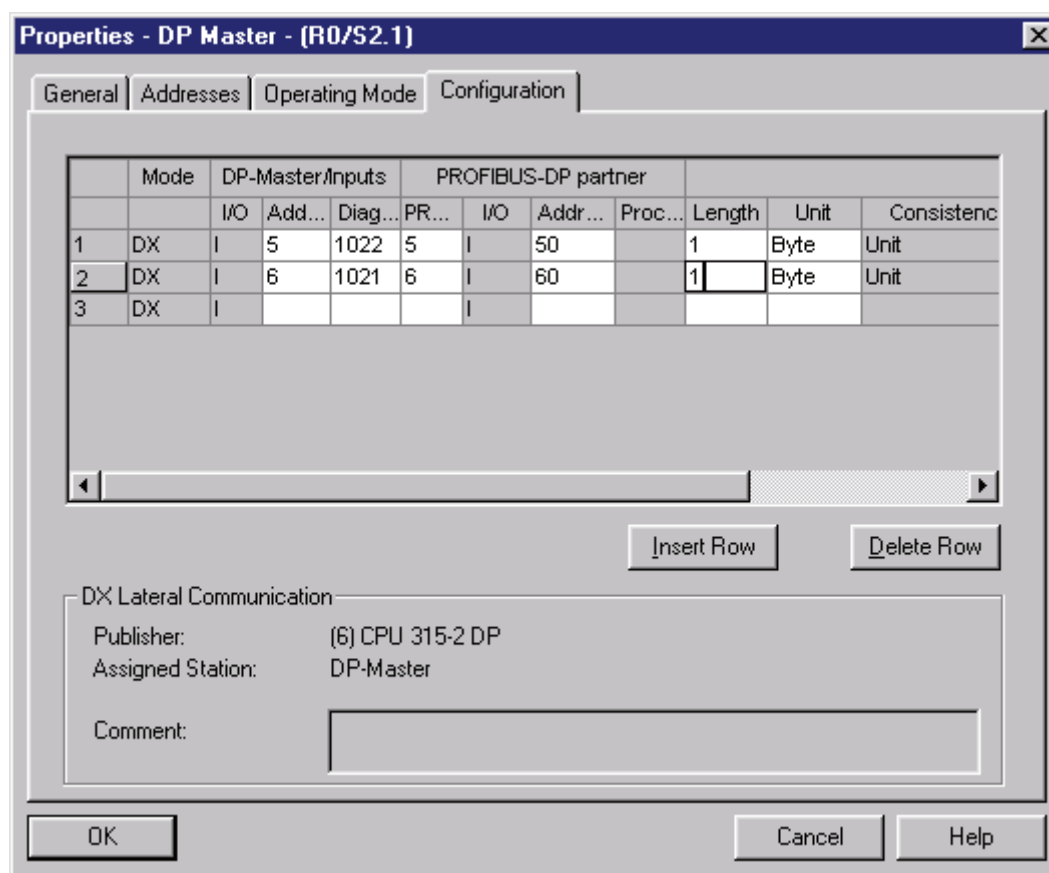


Рис. 6.38 Перекрестные связи DP-Master/Inputs

Тем самым этот DP-Master получает в свое распоряжение входные состояния соответствующих DP-Slave'ов. После сохранения и компиляции, загрузите проект в отдельные S7-300-станции. После этого Вы можете тестировать обмен данными через спроектированные I/Q-адреса с помощью STEP7-утилиты Monitor/Modify Variables (см. раздел 6.2.3).

6.7 Обмен данными с помощью коммуникационного процессора CP 342-5

Ранее было сказано, что коммуникационный процессор CP 342-5 имеет особенности при использовании. Эти особенности заключаются в том, что обмен между этим CP и CPU не является “прозрачным”, то есть требуются специальные функции для обмена данными.

CP 342-5 может работать на шине PROFIBUS как Master, Slave и как активный Slave. Рассмотрим эти возможности.

6.7.1 CP 342-5 как DP-Master

Создайте новый проект в SIMATIC Manager с именем “CP342-5 als DP-Master”. Присоедините на шину DP-Master-системы CP 342-5 станцию ET200M с входным и выходным модулями (см. рисунок 6.39).

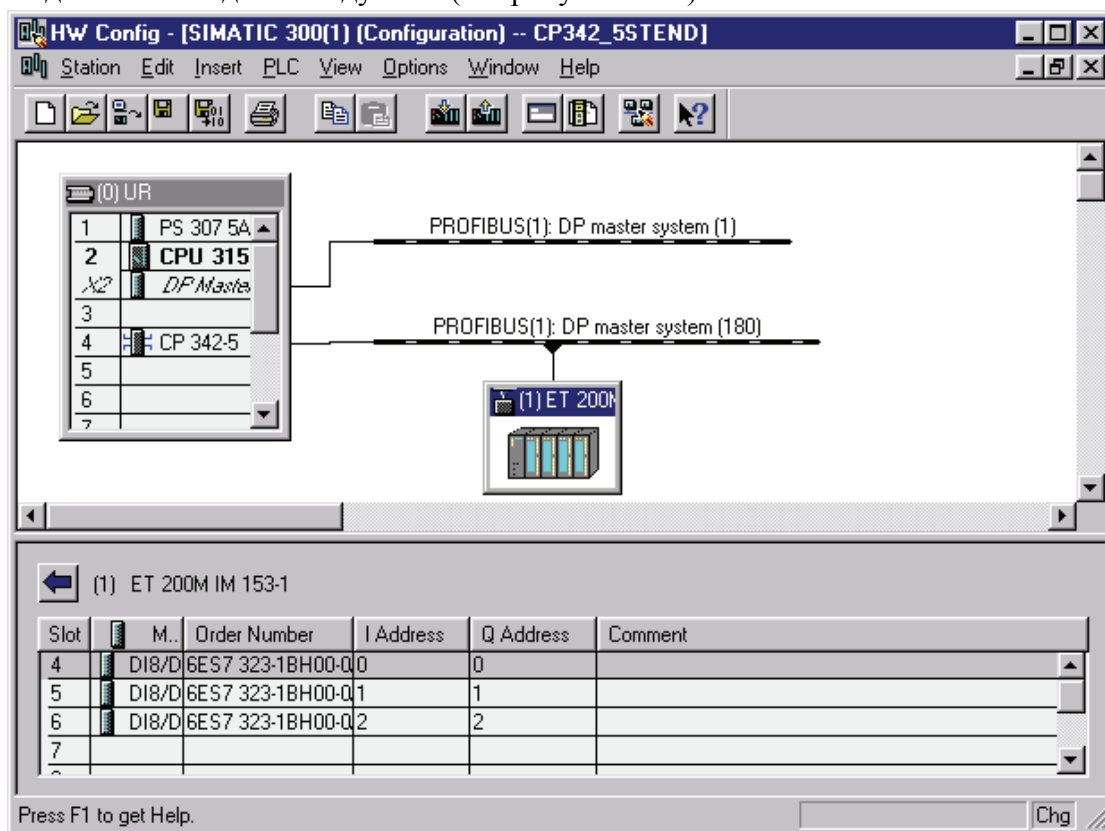


Рис. 6.39 Утилита HW-Config для PLC с CP342-5 - Master

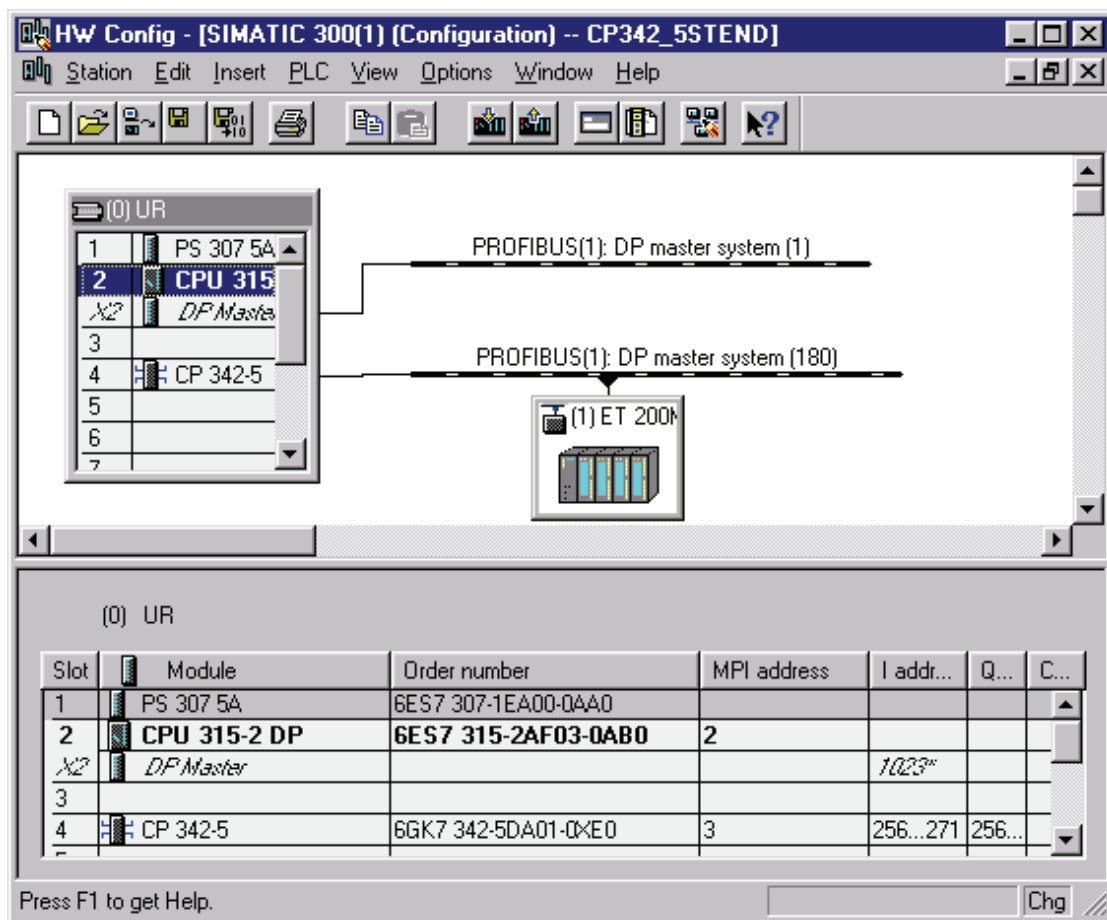


Рис. 6.39 Продолжение

Сохраните эту конфигурацию и скомпилируйте ее.

Напишите пользовательскую программу. Для этого в OB1 вставьте из Standard Library->Communication Blocks функции FC1 "DP_SEND" и FC2 "DP_RECV", как показано на рисунке 6.40.

```
CALL "DP_SEND"
  CPLADDR:=W#16#100 // Адрес CP 342-5 из HW-Config (256 dez)
  SEND :=P#M 0.0 BYTE 3 //Область для отправки в периферию
  DONE :=M60.0
  ERROR :=M60.1
  STATUS :=MW55

CALL "DP_RECV"
  CPLADDR :=W#16#100 // Адрес CP 342-5 из HW-Config (256 dez)
  RECV :=P#M 5.0 BYTE 3//Область для приема из периферии
  NDR :=M30.0
  ERROR :=M30.1
  STATUS :=MW40
  DPSTATUS:=MB50
```

Рис. 6.40 Программа для записи и чтения Slave'ов через CP342-5

Функция “DP_SEND” должна обновить всю область PIQ в CP, занятую периферией, то есть область, указанная в параметре SEND должна иметь размер не менее, чем старший адрес периферии, но не более, чем 240 байт (рисунок 6.41).

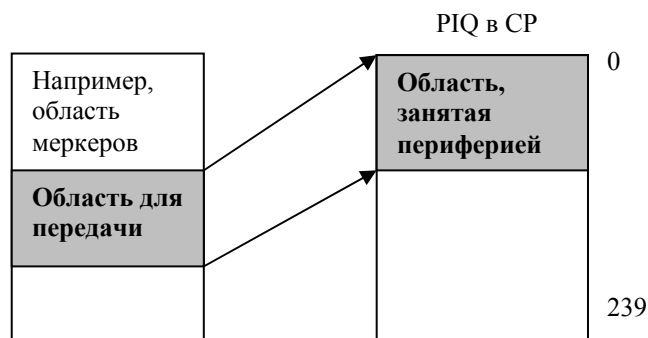


Рис.6.41 Принцип передачи данных из CPU в CP 342-5 с помощью функции “DP_SEND”

Функция “DP_RECV” может читать данные из области ПИ CP 342-5 произвольно, то есть не обязательно читать всю область ПИ, занятую периферией.

Загрузите конфигурацию и пользовательскую программу в CPU. Протестируйте обмен. Для этого в утилите Monitor/Modify Variables создайте таблицу с 6-ю строками: MB0, MB1, MB2, MB5, MB6, MB7. Измените значения MB0, MB1, MB2 и наблюдайте реакцию DP-Slave. Подайте значения на входной модуль DP-Slave – наблюдайте изменения MB5, MB6, MB7.

6.7.2 CP 342-5 как DP-Slave

Создайте конфигурацию, как показано на рисунке 6.42. В качестве DP-Slave используйте станцию S7-300 с CP 342-5. Режим работы для CP 342-5 установите “DP-Slave”. Подключите DP-Slave. Для этого перетяните из аппаратного каталога в HW-Config, раздел “PROFIBUS DP”, подраздел “Configured Station” CP 342-5, согласно его заказному номеру и подключите его к DP-Master-системе. Установите для CP 342-5 необходимое количество фиктивных входных и выходных модулей, как показано на рисунке 6.42.

Обмен данными между CPU и CP будет происходить через эту фиктивную периферию. Это их общая периферия.

Напишите пользовательскую программу в OB1 из CPU DP-Slave как показано на рисунке 6.43. Загрузите скомпилированную конфигурацию и пользовательскую программу в систему. Тестируйте обмен данными с помощью Monitor/Modify Variables: изменяйте MB10 в Slave’е и наблюдайте реакцию в MB20 в Master’е, соответственно, MB30 в Slave’е – реакцию в MB11 в Master’е.

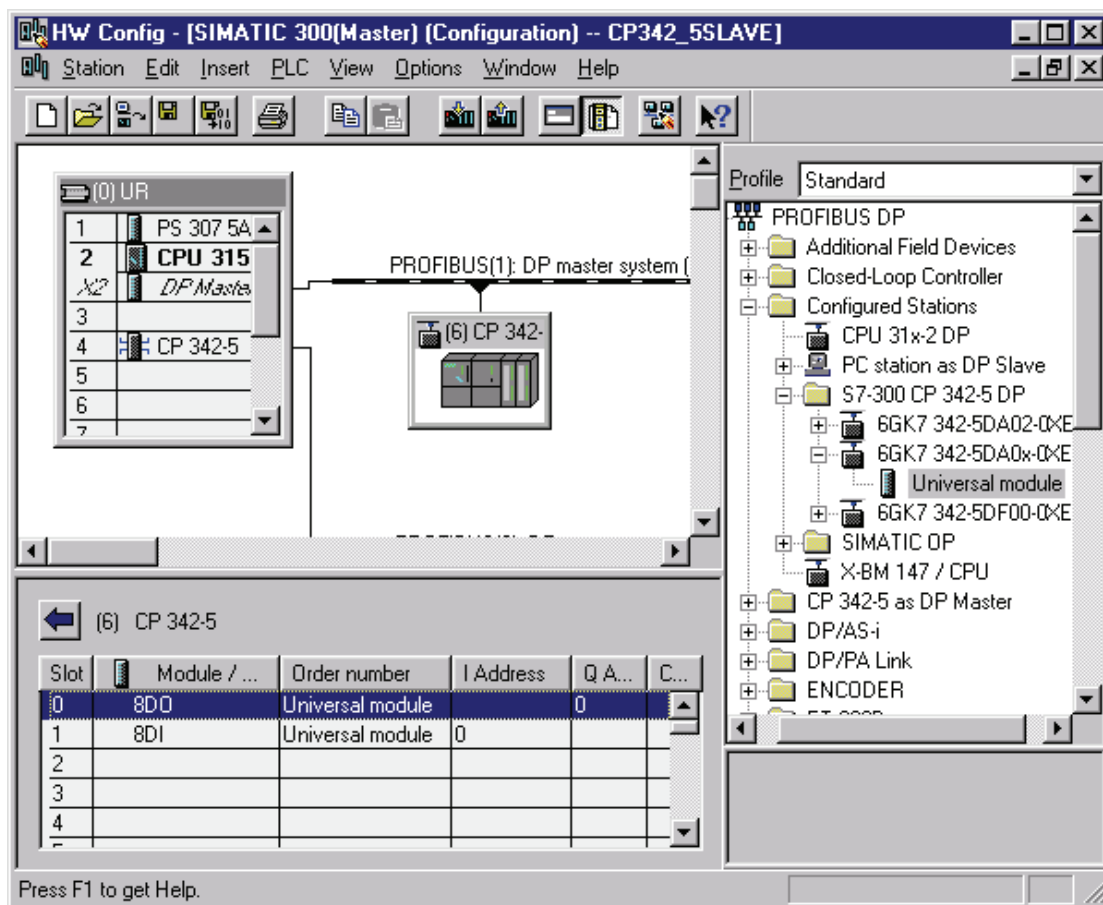


Рис. 6.42 CP342-5 как DP-Slave

Master

```
L IB0 // адрес фиктивного
// входа
T MB20

L MB30
T QB0 // адрес
//фиктивного выхода
```

Slave

```
CALL "DP_SEND"
  CPLADDR:=W#16#100 // Адрес CP 342-5 из HW-Config (256 dez)
  SEND :=P#M 10.0 BYTE 1 //Область для послылки в периферию
  DONE :=M60.0
  ERROR :=M60.1
  STATUS :=MW55

CALL "DP_RECV"
  CPLADDR :=W#16#100 // Адрес CP 342-5 из HW-Config (256 dez)
  RECV :=P#M 11.0 BYTE 1//Область для приема из периферии
  NDR :=M30.0
  ERROR :=M30.1
  STATUS :=MW40
  DPSTATUS:=MB50
```

Рис. 6.43 Программа для записи и чтения данных через CP342-5 как DP-Slave.

6.7.3 CP 342-5 как активный DP-Slave

В разделе 3.2 “Дальнейшие коммуникационные возможности DP-интерфейса” было сказано о возможности создания *активного* DP-интерфейса. При этом активный DP-Slave работает следующим образом: мастер, которому принадлежит данный Slave, работает с ним так же, как с обычным Slave’ом, но активный DP-Slave включается в логическое маркерное кольцо и, получив маркер, может осуществлять коммуникации с помощью FDL- или S7-функций с любыми мастерами. Рассмотрим эту возможность. Активным DP-интерфейсом может быть CP 342-5, кроме того, он поддерживает FDL- и S7-функции.

Создайте конфигурацию, как показано на рисунке 6.42, только Slave CP342-5 подключите к DP-интерфейсу CP342-5, установленного в станции-мастере. В окне свойств Slave CP342-5 должна быть установлена опция “The module is an active node on the PROFIBUS subnet”, как показано на рисунке 6.44. (Она установлена по умолчанию).

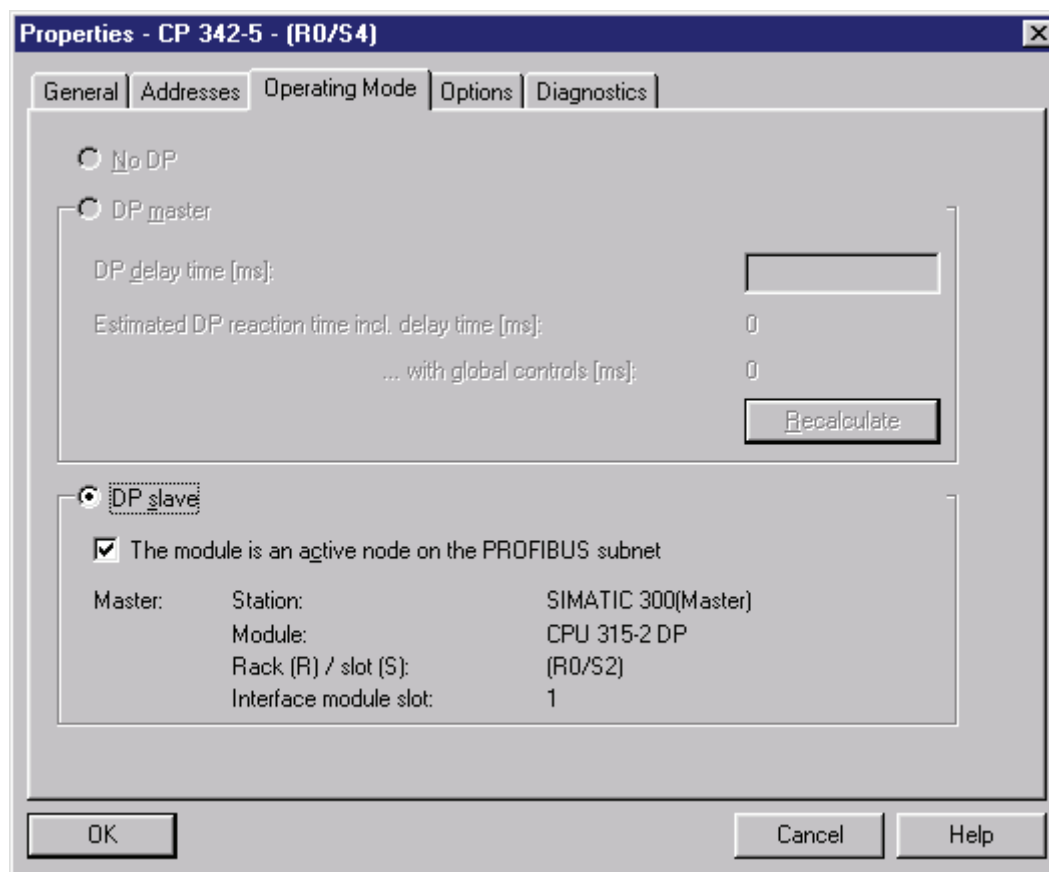


Рис. 6.44 CP342-5: активный DP-Slave

Установите Slave CP342-5 фиктивную периферию: один модуль 8DI и один модуль 8DO.

В OB1 Master’а и Slave’а напишите вызовы функций “DP_SEND” и ”DP_RECV”, как показано на рисунке 6.45.

Master

```
L    7
T    MB10

CALL "DP_SEND"
  CPLADDR:=W#16#100 // Адрес CP 342-5 из HW-Config (256 dez)
  SEND  :=P#M 10.0 BYTE 1 //Область для посылки в периферию
  DONE  :=M60.0
  ERROR :=M60.1
  STATUS:=MW55

CALL "DP_RECV"
  CPLADDR:=W#16#100 // Адрес CP 342-5 из HW-Config (256 dez)
  RECV   :=P#M 21.0 BYTE 1//Область для приема из периферии
  NDR    :=M30.0
  ERROR  :=M30.1
  STATUS :=MW40
  DPSTATUS:=MB50
```

Slave

```
CALL "DP_SEND"
  CPLADDR:=W#16#100 // Адрес CP 342-5 из HW-Config (256 dez)
  SEND  :=P#M 11.0 BYTE 1 //Область для посылки в периферию
  DONE  :=M60.0
  ERROR :=M60.1
  STATUS:=MW55

L    8
T    MB20
CALL "DP_RECV"
  CPLADDR:=W#16#100 // Адрес CP 342-5 из HW-Config (256 dez)
  RECV   :=P#M 20.0 BYTE 1//Область для приема из периферии
  NDR    :=M30.0
  ERROR  :=M30.1
  STATUS :=MW40
  DPSTATUS:=MB50
```

Рис. 6.45 Программы для обмена данными Master'а и Slave'а через DP-интерфейс

Проверка этой коммуникации осуществляется так же, как в предыдущем разделе. Master передает число 7 (из MB10), Slave принимает его в MB11; Slave передает число 8 (из MB20), Master его принимает в MB21.

Создадим теперь FDL-коммуникации. Сначала сконфигурируем их с помощью утилит STEP7.

Откройте утилиту NetPro. Открывшееся окно показано на рисунке 6.46. Щелкните дважды мышью на первой строке таблицы соединений, предварительно выделив с помощью курсора CPU любой станции.

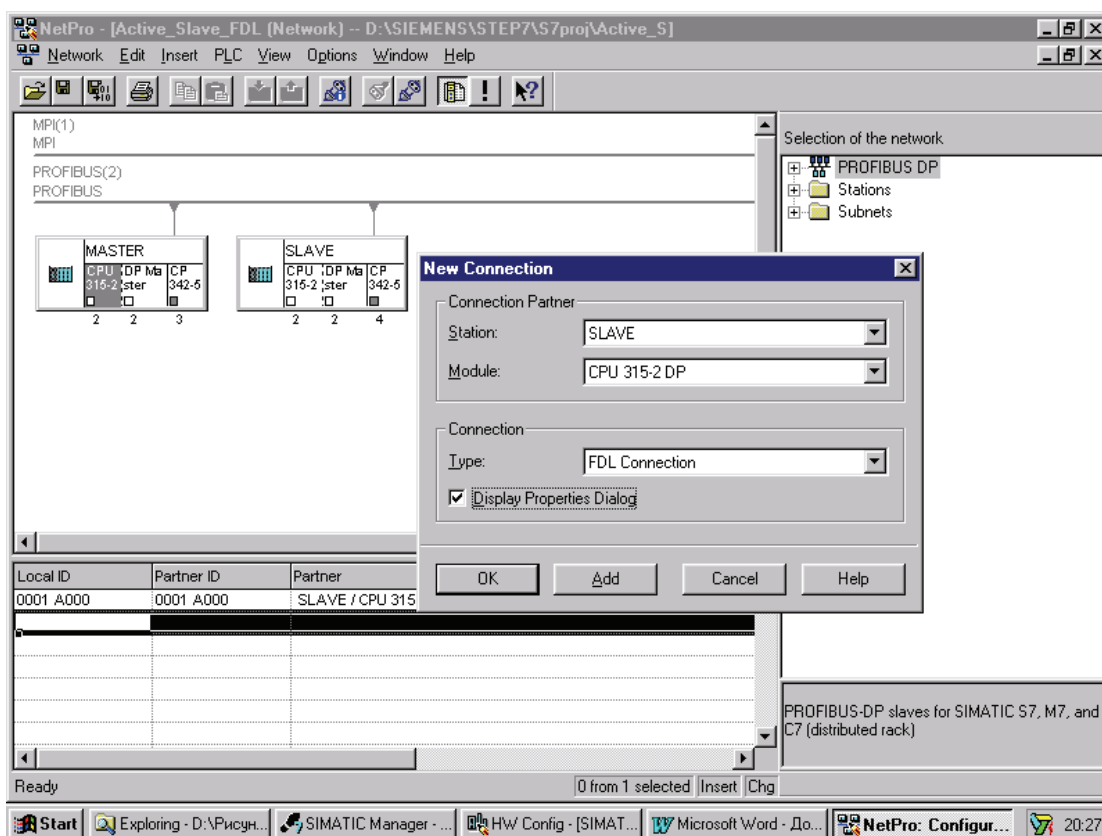


Рис. 6.46 FDL-соединение: определение соединения

В появившемся окне установите тип соединения “FDL Connection”. Установите опцию “Display Properties Dialog” и нажмите кнопку “OK”. Появляется окно, показанное на рисунке 6.47.

В этом окне показана часть интерфейса функции для FDL-коммуникаций, с параметрами ID и LADDR. Значения этих параметров, указанные в этом окне, должны иметь все FDL-функции, использующие соединение, имя которого указано в этом же окне.

Нажав кнопку “Route”, можно увидеть маршрут передачи данных (рисунок 6.48).

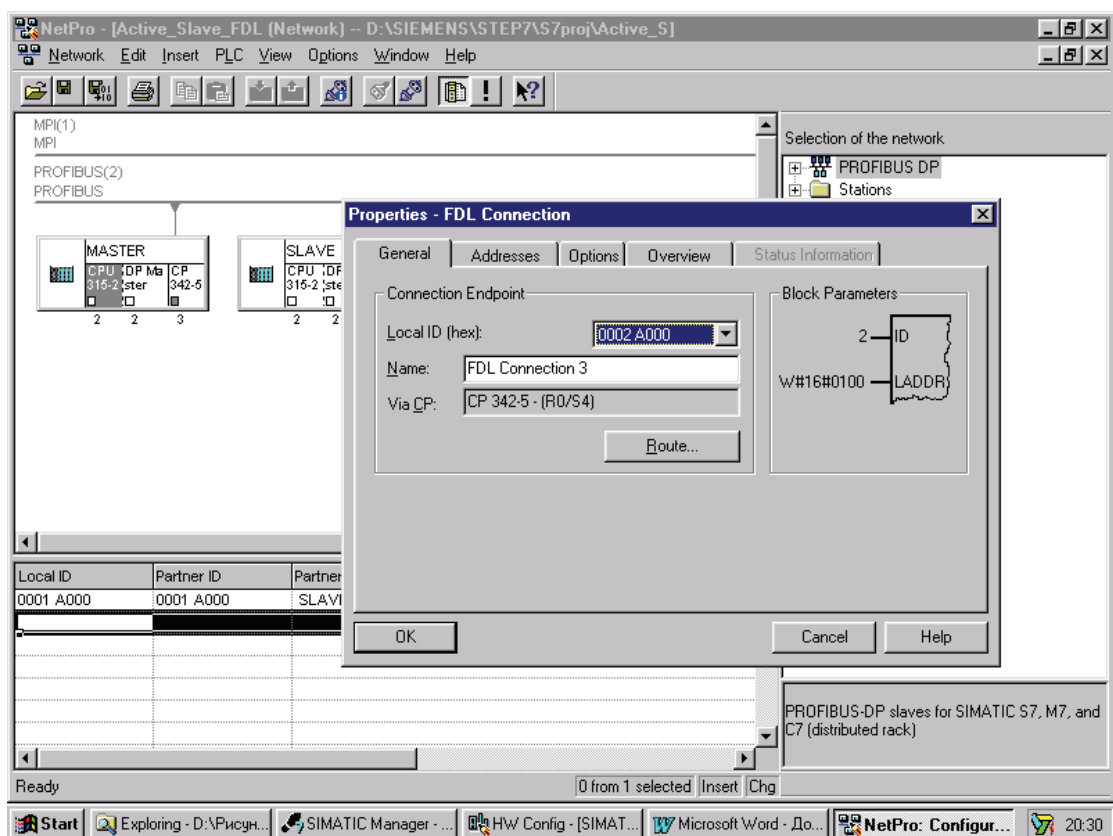


Рис. 6.47 FDL-соединение: параметры функций

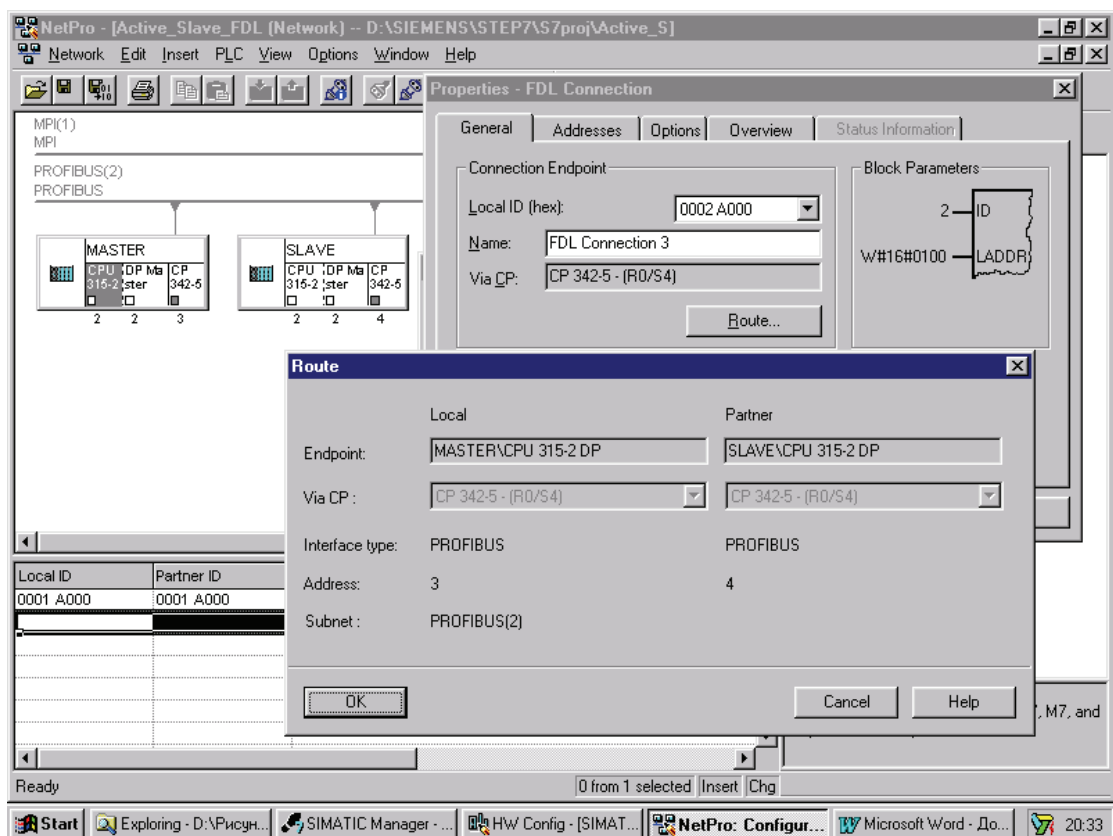


Рис. 6.48 FDL-соединение: маршрут

Закройте все окна с помощью “OK”. Окно в NetPro будет выглядеть, как показано на рисунке 6.49.

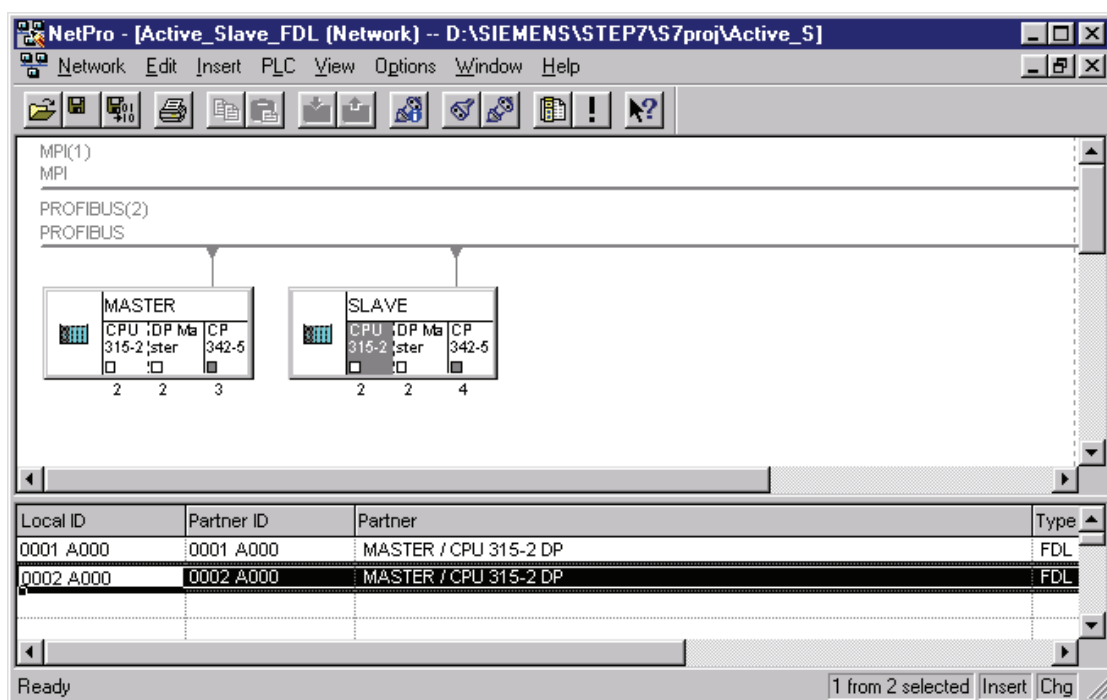


Рис. 6.49 FDL-соединение: окончание конфигурирования

В таблице соединений появляется новое соединение. Двойной щелчок по этой строке при выделенном CPU станции, открывает окно, как показано на рисунке 6.47. Так можно определить параметры функции для данного CPU. Скомпилируйте конфигурацию и загрузите ее в обе станции. Закройте утилиту NetPro.

Теперь создадим пользовательскую программу в OB1. Необходимые функции содержатся в библиотеке “SIMATIC_NET_CP”. Это функции “AG_SEND” и “AG_RCV”. Вызовите их в соответствующих OB1, как показано на рисунке 6.50.

OB1, DP-Master	OB1, DP-Slave
<pre> L 5 T MB 70 CALL "AG_SEND" ACT :=TRUE ID :=1 LADDR :=W#16#100 SEND :=P#M 70.0 BYTE 1 LEN :=1 DONE :=M50.0 ERROR :=M50.1 STATUS:=MW52 </pre>	<pre> CALL "AG_RECV" ID :=1 LADDR :=W#16#100 RCV :=P#M 70.0 BYTE 1 NDR :=M50.0 ERROR :=M50.1 STATUS:=MW52 LEN :=MW54 </pre>

Рис. 6.50 FDL-соединение: пользовательская программа

Загрузите OB1 в соответствующие CPU.

Проверка коммуникации производится с помощью утилиты Monitor/Modify Variables. Как видно из программы, в нашем случае Master записывает число 5 в MB70 и посылает MB70 Slave'у. Slave принимает это значение тоже в MB70.

Замечание 1. FDL-коммуникации можно было создать между двумя Master'ами.

Замечание 2. Обмен данными между двумя DP-Master'ами или DP-Master'ом и активным DP-Slave'ом можно было осуществить с помощью S7-функций.

7. Функции диагностики для PROFIBUS-DP

Введение

Диагностические средства, представленные в SIMATIC S7 для PROFIBUS DP, служат для того, чтобы можно было обнаружить и локализовать ошибки в автоматической системе. Диагностические функции могут, кроме того, применяться, как функции контроля (наблюдения), которые в пользовательской программе могут работать автоматически. Целевое использование имеющихся средств диагностики и диагностических функций способствует быстрому поиску и ликвидации ошибок.

Диагностические возможности, которые предоставляются в распоряжение при реализации DP-установки с помощью SIMATIC S7, классифицируются следующим образом:

- Диагностика через локальные индикаторы. Для этого имеются индикаторы на CPU, который является DP-Master'ом и на отдельных DP-Slave'ах.
- Диагностика с помощью online-функций STEP7. STEP7 предоставляет в распоряжение пользователя ряд online-диагностических функций, как, например,
 - Accessible Nodes;
 - Diagnosing Hardware;
 - Module Information.
- Диагностика S7-DP-Slave'ов через пользовательскую программу полностью увязана с концепцией диагностики SIMATIC S7, которая представляет программе пользователя соответствующий интерфейс для сообщений об ошибках и выходах из строя. Далее можно с помощью SFC получить детальную информацию о причинах ошибок и состояний системы.
- Диагностика с помощью PROFIBUS-Monitor (только под Windows95/98). Для поиска сложных ошибок или проблем при передаче данных предоставляется в распоряжение PROFIBUS-Monitor. Этот инструмент позволяет производить выделение и оценку коммуникационных телеграмм PROFIBUS.

Следующая глава описывает принадлежащие SIMATIC S7 важные диагностические средства и функции. Далее при помощи ряда практических примеров обсуждается применение диагностических средств и SFC.

7.1 Диагностика с помощью индикаторов SIMATIC S7 CPU, интерфейсов DP-Master'а и DP-Slave'а

Благодаря индикаторам S7-300 и S7-400 CPU, которые находятся на лицевой панели CPU, позволяют определить актуальное состояние CPU, то есть интерфейса PROFIBUS-DP. При помощи этих индикаторных светодиодов можно в случае ошибки проводить первичную диагностику.

Индикаторы светодиодов подразделяются на

- светодиоды общего состояния и светодиоды ошибок CPU;

- светодиоды ошибок, которые важны для DP-интерфейса.

7.1.1 Индикаторы S7-300

Общие индикаторы CPU 315-2DP

Общие индикаторы состояния и ошибок для CPU S7-300-2DP прокомментированы в таблице 7.1. Последовательность светодиодов в таблице 7.1 соответствуют расположению индикаторов на CPU.

Табл. 7.1 Общие индикаторы CPU 315-2DP

Обозначение	Значение	Комментарий
SF (красный)	Общая ошибка	Светодиод светится при <ul style="list-style-type: none"> • Hardware-ошибке • Firmware-ошибке • Ошибке программирования • Ошибке параметрирования • Ошибке вычисления • Временной ошибке • Ошибке Memory Card • Выходе из строя батареи, соотв., ее отсутствии • Ошибке периферии (только внешней) Замечание: Для дальнейшего определения ошибки Вы должны использовать PG и прочитать диагностический буфер CPU
BATF (красный)	Ошибка батареи	Светодиод светится, если батарея дефектная, отсутствует или разряжена
DC5V (зеленый)	Питание DC5V	Светодиод светится, если внутреннее напряжение DC5V CPU и шины S7-300 в порядке
FRCE (желтый)	Режим FORCE	CPU находится в режиме FORCE
RUN (зеленый)	Состояние RUN	<ul style="list-style-type: none"> • Светодиод мерцает с частотой 2 Гц во время запуска CPU по крайней мере 3 с (запуск CPU может быть короче). Во время запуска CPU дополнительно светится индикатор STOP. • Светодиод светится, если CPU находится в состоянии RUN
STOP (желтый)	Состояние STOP	<ul style="list-style-type: none"> • Светодиод светится, если CPU не обрабатывает пользовательскую программу • Светодиод мерцает с периодом 1 с, если CPU требует очистки (сброса)

Индикаторы DP-интерфейса CPU 315-2DP

Значения индикаторов для интерфейса PROFIBUS-DP зависят от вида работы DP-интерфейса. Виды работы бывают

- “DP-Master”;
- “DP-Slave”.

Индикаторные светодиоды CPU 315-2DP при режиме работы “DP-Master”

Если CPU 315-2DP используется как DP-Master, то имеют значение индикаторные светодиоды интерфейса PROFIBUS-DP, представленные в таблице 7.2.

Табл. 7.2 Индикаторные светодиоды CPU 315-2DP в режиме работы “DP-Master”

SF DP	BUSF	Значение	Мероприятие
Выкл.	Выкл.	<ul style="list-style-type: none"> • Проектирование в порядке • Все спроектированные Slave исправны 	
Светиться	Светиться	<ul style="list-style-type: none"> • Шинная ошибка (физич. ошибка) • Ошибка DP-интерфейса • Различные скорости в многомастерной системе 	<ul style="list-style-type: none"> • Проверьте шинный кабель на короткое замыкание или разрыв • Оцените диагностику. Проектируйте заново или корректируйте проект
Светиться	Мерцает	<ul style="list-style-type: none"> • Выход из строя станции • По крайней мере один назначенный Slave неисправен 	Проверьте подключен ли шинный кабель к CPU 315-2DP, соответствующая шина разорвана. Ждите, пока CPU 315-2DP стартует. Если светодиод не прекращает мерцать, проверьте DP-Slave или оцените диагностику DP-Slave.
Светиться	Выкл.	Ошибочное проектирование (также, если CPU спроектирован не как DP-Master)	Оцените диагностику. Проектируйте заново или корректируйте проект.

Индикаторные светодиоды CPU 315-2DP в режиме работы “DP-Slave”

Если CPU 315-2DP используется, как DP-Slave, то индикаторные светодиоды PROFIBUS-DP-интерфейса представлены в таблице 7.3.

Табл. 7.3 Индикаторные светодиоды CPU 315-2DP в режиме работы “DP-Slave”

SF	DP BUSF	Значение	Мероприятие
Выкл.	Выкл.	Проектирование в порядке	
Не важно	Мерцает	<p>CPU 315-2DP не верно параметрирован. Нет обмена данными между DP-Master’ом и CPU 315-2DP. Причины:</p> <ul style="list-style-type: none"> • Нарушен контроль времени • Прерваны коммуникации через PROFIBUS-DP • Адрес PROFIBUS неверно параметрирован 	<ul style="list-style-type: none"> • Проверьте CPU 315-2DP • Проверьте, правильно ли установлен шинный штекер • Проверьте, не разорван ли шинный кабель в DP-Master’ом • Проверьте конфигурирование и параметрирование
Не важно	Светиться	Короткое замыкание шины	Проверьте структуру шины
Светиться	Не важно	<ul style="list-style-type: none"> • Ошибочное программирование • Нет обмена данными с DP-Master’ом 	<ul style="list-style-type: none"> • Проверьте проектирование • Оцените диагностические сигналы, соотв. содержимое диагностического буфера

7.1.2 Индикаторы CPU S7-400 с DP-интерфейсом

В таблице 7.4 даны индикаторы и их значения CPU S7-400 с PROFIBUS-интерфейсом. Индикаторы в таблице представлены в той последовательности, как они расположены на CPU.

Табл. 7.4 Индикаторы CPU S7-400 с DP-интерфейсом

CPU		DP-интерфейс	
Индикатор	Значение	Индикатор	Значение
INTF (красный)	Внутренняя ошибка	DP INTF (красный)	Внутренняя ошибка в DP-интерфейсе
EXTF (красный)	Внешняя ошибка	DP EXTF (красный)	Внешняя ошибка в DP-интерфейсе
FRCE (желтый)	Режим FORCE	BUSF	Шинная ошибка в DP-интерфейсе
CRST (желтый)	Новый старт		
RUN (зеленый)	Режим RUN		
STOP (желтый)	Режим STOP		

Общие светодиодные индикаторы CPU S7-400 с интерфейсом DP-Master'a

Светодиодные индикаторы для сообщения о состоянии CPU S7-400 с интерфейсом DP-Master'a прокомментированы в таблице 7.5.

Табл. 7.5 Светодиодные индикаторы для сообщения о состоянии CPU S7-400 с интерфейсом DP

Светодиод			Значение
RUN	STOP	CRST	
Светится	Выкл.	Выкл.	CPU в режиме RUN
Выкл.	Светится	Выкл.	CPU в состоянии STOP. Программа пользователя не обрабатывается. Возможны повторный старт или новый старт. Если состояние STOP вызвано ошибкой, то дополнительно светятся индикаторы ошибок (INTF и EXTF)
Выкл.	Светится	Светится	CPU в состоянии STOP. В качестве запуска возможен только новый старт
Мигает (0,5 Hz)	Светится	Выкл.	Состояние HALT. Запущена тестовая функция PG
Мигает (2 Hz)	Светится	Светится	Проводится повторный старт
Мигает (2 Hz)	Светится	Выкл.	Проводится новый старт
Не важно	Мигает (0,5 Hz)	Не важно	CPU требует сброса
Не важно	Мигает (2 Hz)	Не важно	Идет сброс

Возникающая ошибка или выполняющаяся специальная функция индицируется светодиодными индикаторами, которые прокомментированы в таблице 7.6.

Табл. 7.6 Светодиодные индикаторы для ошибок и специальных функций CPU S7-400 с DP-интерфейсом

Светодиод			Значение
INTF	EXTF	FRCE	
Светится	Не важно	Не важно	Распознана внутренняя ошибка (ошибка программирования или параметрирования)
Не важно	Светится	Не важно	Распознана внешняя ошибка (ошибка, причина которой лежит не в CPU)
Не важно	Не важно	Светится	На этом CPU с PG запущена функция "FORCE". Переменные программы пользователя получают постоянные значения и не могут изменяться программой пользователя

Светодиодные индикаторы DP-интерфейса CPU S7-400

Значения светодиодных индикаторов интерфейса PROFIBUS-DP CPU S7-400 представлены в таблице 7.7.

Табл. 7.7 Светодиодные индикаторы DP-интерфейса S7-400

Светодиод			Значение
DP INTF	DP EXTF	BUSF	
Светится	Не важно	Не важно	Распознана внутренняя ошибка (ошибка программирования или параметрирования)
Не важно	Светится	Не важно	Распознана внешняя ошибка (ошибка, причина которой лежит не в CPU, а в DP-Slave)
Не важно	Не важно	Мерцает	Один или более DP-Slave на PROFIBUS не отвечают
Не важно	Не важно	Светится	Распознана шинная ошибка DP-интерфейса (например, обрыв кабеля или неверные шинные параметры)

7.1.3 Индикаторы DP-Slave

DP-Slave также имеют индикаторы, с помощью которых индицируется рабочее состояние и, при необходимости, ошибки DP-Slave. Число и значение индикаторов зависит от типа применяемого Slave, и поэтому должны быть взяты из технической документации на соответствующий Slave.

Индикаторы DP-Slave'ов, которые применяются в примере проектирования (раздел 4.2.5), описаны ниже.

Индикаторы ET200B 16DI/16DO

В таблице 7.8 прокомментированы значения индикаторов ET200B 16DI/16DO.

Табл. 7.8 Индикаторы состояния ошибок ET200B 16DI/16DO

Светодиод	Оптич. сигнал	Значение
RUN	Светится (зеленый)	ET200B находится в рабочем состоянии (ист. питан. вкл. переключат. STOP/RUN в положении RUN)
BF	Светится (красный)	<ul style="list-style-type: none">• Контроль времени превышен• Не параметрировано
DIA	Светится (красный)	У модулей цифровых выходов отсутствует DC24V; по крайней мере у одного выхода короткое замыкание или сбой питающего напряжения
L1+	Светится (зеленый)	Напряжение для группы каналов "0" приложено (при нарушении защиты, соответственно при падении напряжения ниже допустимого, типично +15,5 V, загорается индикаторный светодиод)
L2+	Светится (зеленый)	То же для группы каналов "1"

Светодиодные индикаторы ET200M/ IM 153-2

В таблице 7.9 прокомментированы значения светодиодных индикаторов ET200M/ IM 153-2

Табл. 7.9 Индикаторы состояния и ошибок ET 200M/IM 153-2

Светодиод			Значение	Действия
ON (зеленый)	SF (красный)	BF (красный)		
Выкл.	Выкл.	Выкл.	Нет напряжения или дефект аппаратуры в IM 153-2	Проверить DC24V источника питания
Светится	Не важно	Мигает	IM 153-2 неверно параметрирован и обмен данными между DP-Master'ом и IM 153-2 не происходит. Возможные причины: <ul style="list-style-type: none"> • Контроль времени превышен • Прерван шинные коммуникации через PROFIBUS-DP с IM 153-2 	Проверьте DP-адрес. Проверьте IM 153-2. Проверьте, правильно ли установлен шинный штекер. Проверьте, не разорван ли кабель, соединяющий с DP-Master'ом Выкл. и вкл. напряжение DC24V в источнике питания. Проверьте конфигурацию и параметрирование
Светится	Не важно	Светится	Поиск модуля или недоступен DP-адрес	Установите для IM 153-2 доступный DP-адрес (от 1 до 125) или проверьте структуру шины
Светится	Светится	Не важно	Запроектированная структура ET200M не совпадает с действительной структурой. Ошибка в подключенном модуле S7-300 или дефект IM 153-2	Проверьте структуру ET200M: дефектен ли модуль или он не запроектирован. Проверьте проектирование. Смените S7-300 или IM 153-2
Светится	Выкл.	Выкл.	Обмен данными между DP-Master'ом и ET200M происходит. Заданная и действительная конфигурации ET200M совпадают.	

7.2 Диагностика с помощью online-функций STEP 7

Базовый пакет STEP 7 предоставляет в распоряжение пользователю различные online-функции для диагностики. Эта глава описывает эти функции и их использование на примере установки PROFIBUS-DP.

7.2.1 Функция Accessible Nodes в SIMATIC Manager

С помощью функции Accessible Nodes в SIMATIC Manager можно проверить, какие активные и пассивные шинные участники находятся на сети MPI или PROFIBUS. Также можно благодаря этой функции проводить для участников, подключенных к сети MPI или PROFIBUS, диагностику без данных соответствующего проекта STEP 7.

Чтобы использовать эту диагностическую online-функцию, должны быть установлены скорость интерфейса PG/PC, такая же, как у сети PROFIBUS (при MPI – 187,5 кБод) и шинный пофиль. Online-интерфейс PG/PC при старте функции пассивно связывается с шиной и проверяет, совпадает ли его скорость

передачи со скоростью передачи шины PROFIBUS. Если это не так, выдается соответствующее сообщение об ошибке. То же происходит, если подключенный PG/PC не имеет уникального шинного адреса. Если установлено совпадение скоростей и PG/PC имеет уникальный адрес, то он принимается в логическое маркерное кольцо.

С помощью MPI/ISA-карты можно установить скорость передачи только до 1,5 Мбод. Чтобы функция выполнялась при более высоких скоростях, необходимо использовать другие коммуникационные карты, например, CP5411 (ISA-карта), CP5511 (PCMCIA-карта) или CP5611 (PCI-карта). Все названные устройства полностью поддерживаются базовым пакетом STEP 7, то есть не требует использования дополнительных средств.

При активизации функции Accessible Nodes с помощью команды меню SIMATIC Manager появляется окно, в котором показаны все доступные программируемые модули (CPU, FM, CP) со своими MPI или PROFIBUS шинными адресами. Показываются также MPI- и PROFIBUS-участники, которые проектируются не с помощью STEP 7 (например, панель оператора). Шинный участник, с которым PG или PC связан напрямую, отмечен словом “direct” рядом с шинным адресом (см. рисунок 7.1).

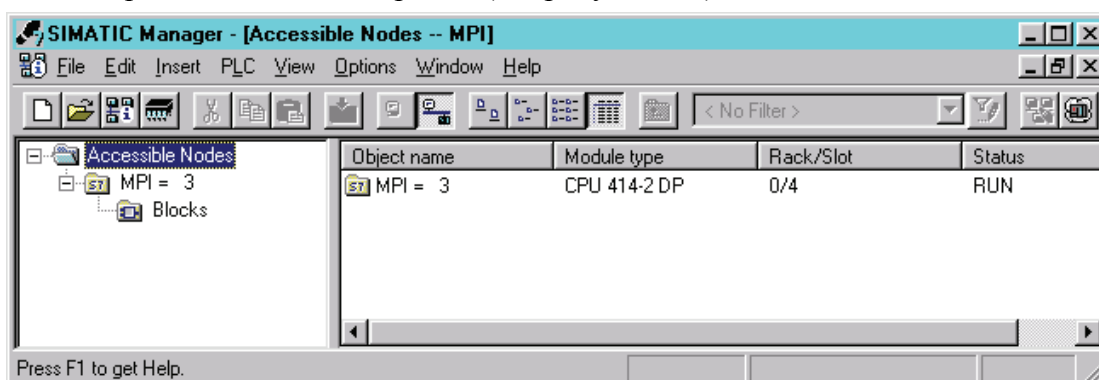


Рис. 7.1 Функция Accessible Nodes с MPI-узлами

Диагностическая возможность Accessible Nodes предлагает прежде всего быстрый доступ к программируемому модулю.

Обратите внимание, что изменения в online-представлении (например, выход из строя участника) автоматически не актуализируется в открытом окне Accessible Nodes. Актуализация достигается или с помощью функциональной клавиши “F5”, или с помощью *View->Update*.

Если MPI-участник выбран с помощью правой клавиши мыши, то открывается контекстное меню. Если выбрать CPU, то открывается следующее подменю. Для диагностики имеют значение следующие пункты контекстного меню:

- Monitor/Modify Variables. Выбор этой функции запускает утилиту STEP 7 Monitoring and Modifying Variables. Вслед за этим Вы можете наблюдать и управлять переменными.
- Module Information (см. раздел 7.2.3).
- Diagnose Hardware (см. раздел 7.2.4).

Установка online-интерфейса PG/PC

Выберите в SIMATIC Manager *Option->Set PG/PC Interface*. Открывается окно, показанное на рисунке 7.2.

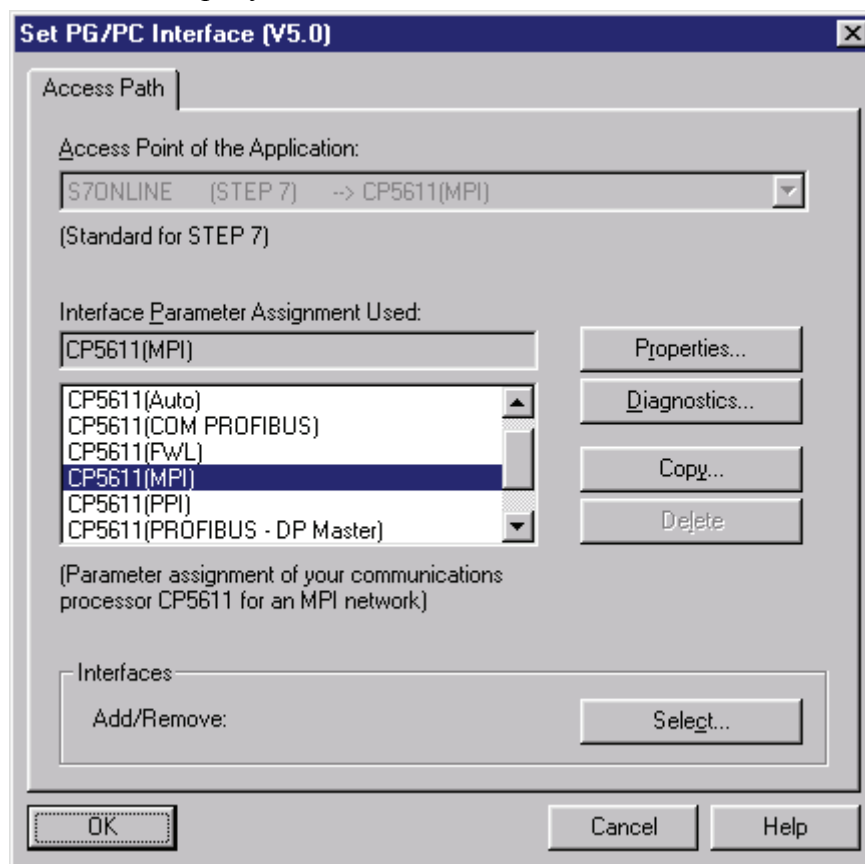


Рис. 7.2 Установка PG/PC интерфейса

Используйте PG740 со встроенной MPI-картой, выберите в качестве модуля “MPI/ISA on Board (PROFIBUS)”. Войдите с помощью кнопки “Properties” в окно установки деталей модуля и выберите не занятый адрес PROFIBUS, с которым PG должен работать. Установите скорость передачи на значение, которое имеет Ваша установка, и сравните “Highest Station Address” и “Profile” со значениями, имеющимися на установке. Квитируйте заданные параметры с помощью “OK”.

Соедините MPI/DP-интерфейс Вашего PG с сетью PROFIBUS. Обратите внимание, что при подключении PG к PROFIBUS применяется активный кабель (PROFIBUS-кабель со встроенным репитером). Иначе Вы можете при подключении Вашего кабеля вызвать помехи.

Если PG/PC физически связан с PROFIBUS, щелкните затем на кнопке Accessible Noes. Теперь PG “слушает” шину и отображает всех подключенных к ней участников. Здесь также показан тип станции, то есть активный это участник (DP-Master) или пассивный (DP-Slave). В случае, если программатор подключен к некоторому узлу напрямую, то у этого узла будет надпись “direct” (рис. 7.3).

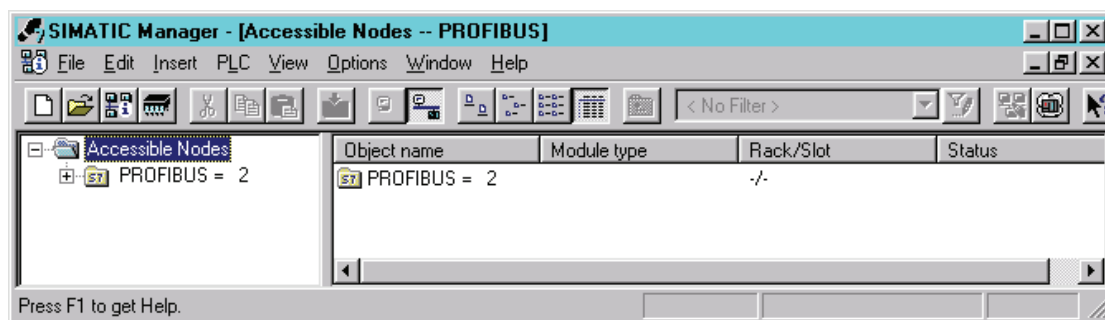


Рис. 7.3 Функция Accessible Nodes через PROFIBUS

Функция Accessible Nodes может, например, применяться, чтобы контролировать установленные PROFIBUS-адреса DP-Slave или если предполагается обрыв шины. В этом случае можно установить достижимы ли модули и, если да, то какие.

Дальнейшая диагностика подключенных PROFIBUS-участников возможна только тогда, когда выбранные участники поддерживают диагностические функции STEP 7. Например, S7-CPU с интерфейсом PROFIBUS-DP поддерживает эти диагностические функции.

Щелчок на PROFIBUS-DP-адресе CPU открывает обычное контекстное меню. Через пункт меню CPU здесь можно также открыть диагностические функции, такие, как Monitor/Modify Variables, Module Information и т.д.

Двойной щелчок на PROFIBUS-адресе CPU открывает объект и показывает контейнер блоков CPU. Опять с помощью двойного щелчка на контейнере модулей в правой половине окна SIMATIC Manager появляются применяемые блоки. Их можно открыть, изменить и опять загрузить в CPU для проверки. Конечно, в этом случае невозможно символическое программирование, так как для этого необходим offline-проект STEP 7.

7.2.2 ONLINE-функции в SIMATIC Manager

Если Вы загрузили проект STEP 7 в контроллер, Вы можете в SIMATIC Manager при MPI-связи с помощью команды “online” перевести проект из offline-представления в online-представление, чтобы, например, открыть блоки STEP 7 с символическими именами. Эта функция возможна также при поддержке PG/PC через PROFIBUS. Откройте проект и установите интерфейс Вашего PG/PC, как это описано в главе 7.2.1, чтобы видеть актуальные значения на Вашей установке. Доступ к целевой системе вместе с управлением проектом тогда можно осуществить с помощью сконфигурированной аппаратуры или без нее.

Чтобы осуществить доступ с помощью сконфигурированной аппаратуры, откройте соответствующий проект и выберите для online-представления станции команду меню *View->Online*.

Выполните затем двойной щелчок на станции, которую Вы хотите открыть online, чтобы видеть содержащиеся в ней модули. Автоматически появится

окно, в котором посредством закладок установите свойства соединения, как, например, PROFIBUS-адрес выбранного участника, а также номер слота (см. рисунок 7.4).

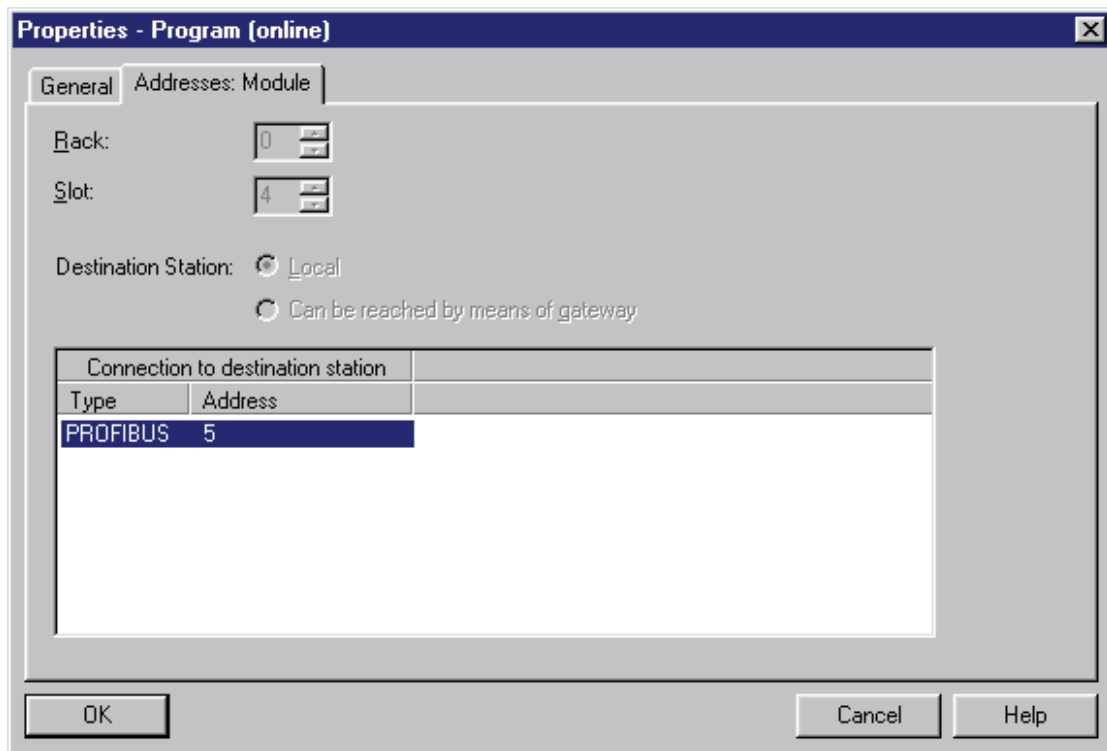


Рис. 7.4 Свойства соединения (свойства папки S7-Programm в online-режиме)

Задайте данные Вашего партнера, соответственно Вашего CPU и покиньте диалоговый бокс с помощью “ОК”. Опрос происходит только при первом online-доступе. Вводимые данные запоминаются в проекте STEP 7 и, таким образом, не должны при каждом online-доступе обновляться на экране. С помощью двойного щелчок на модуле, с которым Вы хотите установить связь, будет создана связь, принимающая во внимание установки и может иметь место online-диагностика через PROFIBUS-интерфейс.

Для доступа без сконфигурированной аппаратной части, то есть без конфигурации аппаратуры в offline-проекте, откройте соответствующее окно проекта. Выберите для online-представления команду меню *View->Online*. Установите курсор на папку S7-Programm, расположенную непосредственно под проектом, с помощью правой клавиши мыши выберите в контекстном меню команду Object Properties. В открывшемся окне перейдите на закладку Addresses и задайте в окне для PROFIBUS-адреса адрес CPU, к которому Вы хотите получить доступ. Закройте диалоговое окно кнопкой “ОК”. Связь установлена и Вы можете тестировать STEP 7-программу в режиме online.

7.2.3 Диагностика с помощью функции Module Information

С помощью вызова функции Module Information открывается окно с несколькими закладками, которые показывают актуальную информацию о модулях. Объем этой информации зависит от типа выбранного модуля. Наряду

с информацией на закладке показано состояние модуля. Если выбран не S7-CPU, то выдается состояние, которое он имеет с точки зрения CPU (например, ОК, ошибка, модуль отсутствует).

Таблица 7.10 показывает, какие закладки имеются для отдельных модулей в окне “Module Information”.

Табл. 7.10 Информация о состоянии модулей для каждого типа модуля

Название закладки окна	CPU или M7-FM	Модули, способные к системной диагностике	Модули с диагностикой	Модули без диагностики	Стандартные DP-Slave
General	X	X	X	X	X
Diagnostic Buffer	X	X			
Memory	X				
Scan Cycle Time	X				
Time System	X				
Performance Data	X				
Communication	X				
Stacks	X				
Diagnostic Alarm		X	X		
DP-Slave-Diagnose					X

Способностью к системной диагностике обладают, например, FM-модули (функциональные модули). Способностью к диагностике – в основном аналоговые сигнальные модули. Модули без диагностики – это в основном цифровые сигнальные модули.

Окно “Module Information” может быть открыто несколькими путями:

- С помощью функции “Accessible Nodes” из SIMATIC Manager щелкните на желаемом партнере правой клавишей мыши, после этого выберите в контекстном меню *CPU->Module Information*.
- С помощью функции SIMATIC Manager “Online”. Откройте проект в Online-режиме, желаемая станция появится в левой половине окна SIMATIC Manager. Двойным щелчком открывается станция и выбирается программируемый модуль, то есть CPU. С помощью щелчка правой кнопкой мыши и контекстного меню *CPU->Module Information* открывается функция “Module Information”.
- С помощью функции “Diagnosing Hardware” (см. раздел 7.2.4)

Рисунок 7.5 показывает открытое окно “Module Information”, закладку “General”.

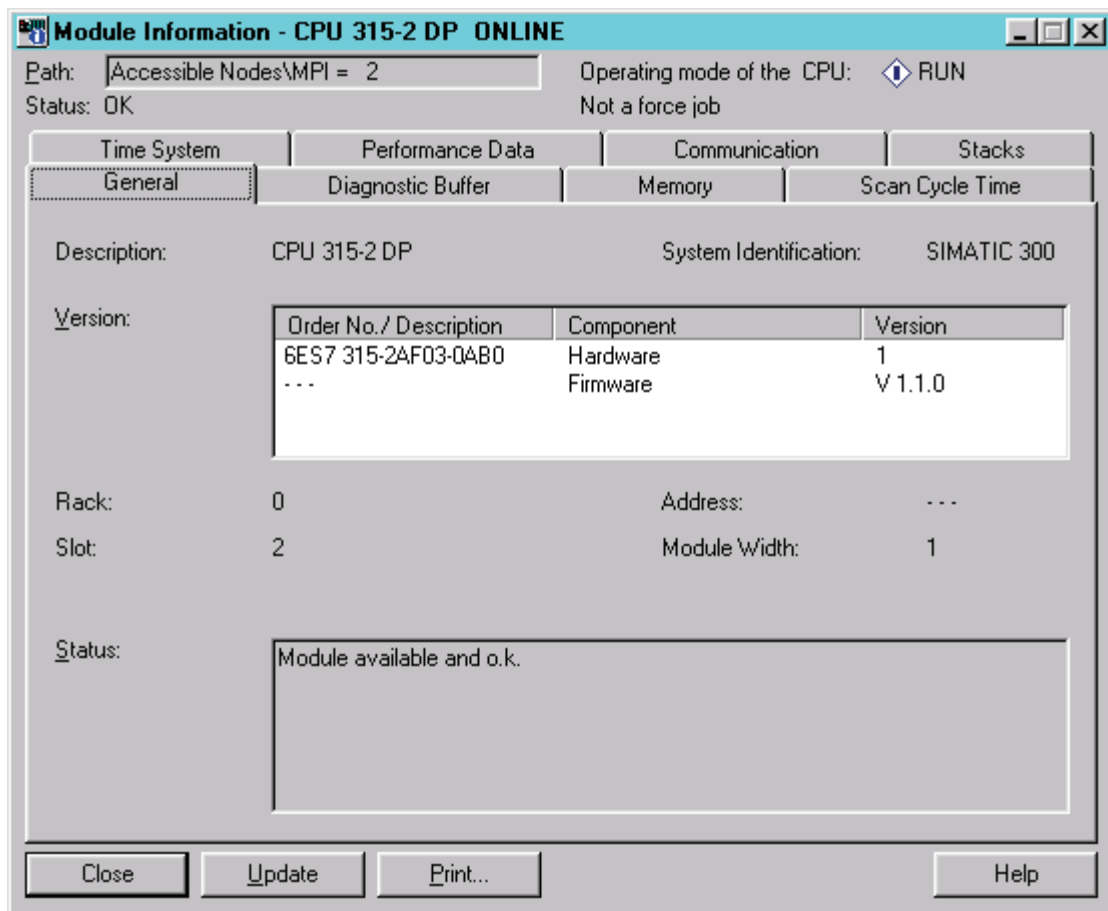


Рис. 7.5 Диалоговое окно “Module Information” (для CPU)

Отдельные закладки содержат различную информацию. Таблица 7.11 показывает список возможных закладок этого диалогового окна и их возможное использование. В конкретных случаях показываются только те закладки, которые имеют смысл для выбранного модуля.

Следующая информация показывается в каждой закладке:

- Online-путь к выбранному модулю;
- Рабочее состояние CPU, которому принадлежит модуль;
- Состояние выбранного модуля (например, имеет ошибку, ОК);
- Рабочее состояние выбранного модуля (например, RUN, STOP), в той мере, в какой выбранный модуль этим располагает.

Табл. 7.11 Обзор информация закладок окна “Module Information”

Название закладки окна	Информация	Использование
General	Идентификационные данные выбранного модуля, напр., тип, заказной номер, номер слота в стойке, состояние	Online-информация установленного модуля может быть сравнена с данными запроецированного модуля
Diagnostic Buffer	Обзор событий в диагностическом буфере	Для оценки причины останова CPU
Memory	Текущая загрузка рабочей и загрузочной памяти выбранного CPU	Для оценки возможности переноса в CPU новых или обновленных блоков
Scan Cycle Time	Длительность наиболее короткого, длинного и последнего циклов выбранного CPU или M7-FM	Для контроля параметрирования минимального и максимального времени цикла, а также для контроля актуального времени цикла
Time System	Актуальное время, рабочее время и информация для синхронизации часов	Для показа времени и даты модуля и для контроля синхронизации времени
Performance Data	Расширение памяти, области операндов и блоки, имеющиеся в выбранном CPU/FM. Показ всех видов блоков, которые имеются в выбранном модуле. Список OB, SFC и SFB, которые могут применяться в данном модуле	Применяется для и во время создания пользовательской программы и для проверки совместимости существующей программы пользователя с данным модулем
Communication	Скорость передачи, обзор соединений, коммуникационная нагрузка, а также максимальная величина телеграмм	Для установления, как много и какие могут быть связи CPU или M7-FM
Stacks	Показывается содержание B-Stack'a, I-Stack'a и L--Stack'a. Дополнительно Вы можете перейти в редактор блоков	Для установления причин останова и для корректировки блоков
Diagnostic Alarm	Диагностические данные выбранного модуля	Для определения причин выхода из строя модуля
DP-Slave-Diagnose	Диагностические данные выбранного DP-Slave'a по EN 50170	Для определения причин ошибок DP-Slave'a

При каждой смене закладки окна “Module Information” будут считываться новые данные из модуля. Во время показа закладки ее состояние автоматически не актуализируется. При нажатии на кнопку “Update” будут прочитаны новые данные из модуля без смены закладки.

Далее описываются важнейшие закладки окна “Module Information”.

Diagnostic Buffer

Закладка “Diagnostic Buffer” считывает системную диагностику модуля (например, CPU) из его диагностического буфера. В диагностический буфер заносятся все диагностические события в порядке их наступления с подробной информацией (рисунок 7.6).

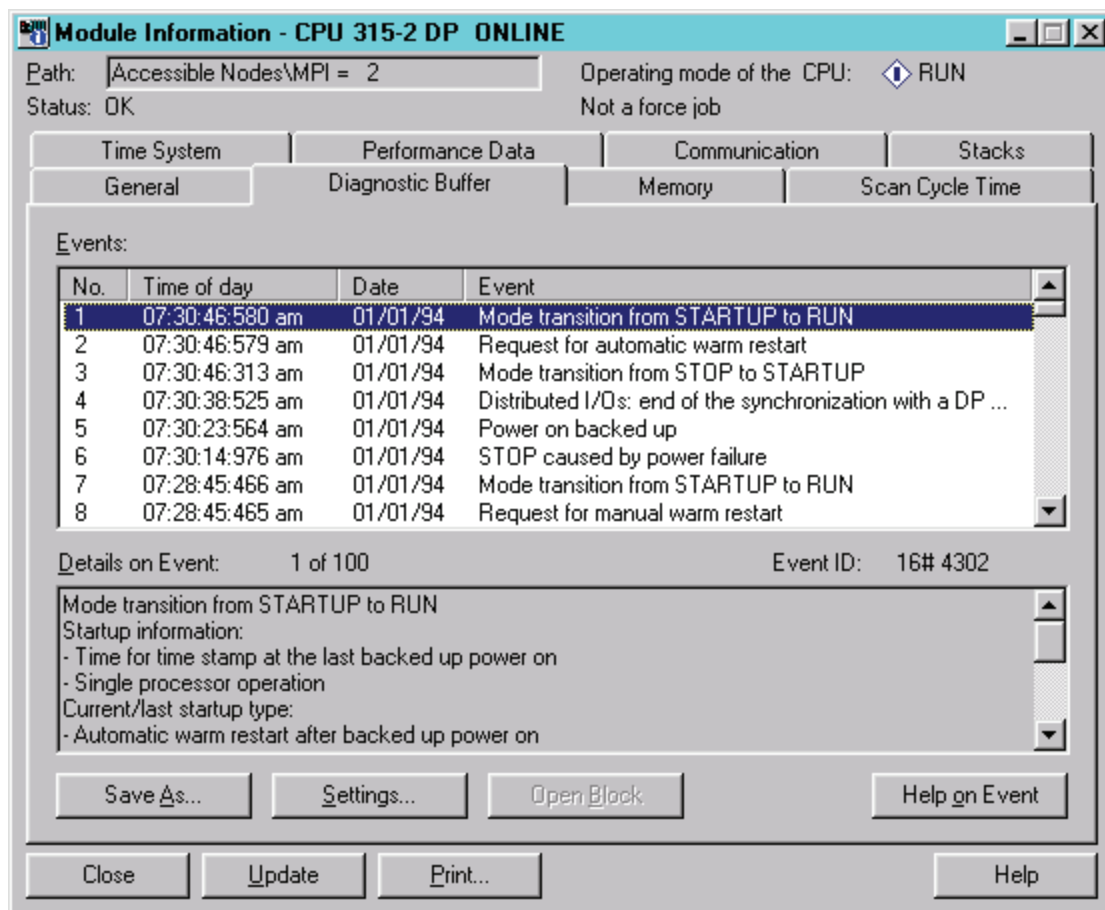


Рис. 7.6 Зкладка “Diagnostic Buffer” регистра “Module Information”

При сбросе CPU содержимое диагностического буфера сохраняется.

В качестве диагностических событий считываются, например, ошибки в модуле, системные ошибки в CPU, смена рабочего режима (например, RUN на STOP), а также ошибки программы пользователя.

Ошибки в системе могут оцениваться благодаря диагностическому буферу спустя длительное время после их появления, чтобы установить причину перехода в STOP или чтобы знать порядок наступления диагностических событий.

Если событие маркировать мышью, то можно с помощью кнопки “Help on Event” получить дополнительную информацию. Для записи в диагностический буфер, которая ссылается на место ошибки (тип блока, номер блока, относительный адрес), можно открыть соответствующий блок, чтобы устранить причину ошибки. Курсор в этом случае устанавливается прямо на команде, являющейся причиной события.

Диагностический буфер – кольцевой буфер. Каждый модуль имеет определенное максимальное число записей. Если максимальное число записей достигнуто, то при новой записи в буфер самая старая запись стирается. Все

записи сдвигаются вниз, а на место стертой записи заносится новая. Благодаря этому актуальная запись всегда находится на первом месте.

Diagnostic Alarm

На закладке “Diagnostic Alarm” показывается для модуля, способного к диагностике, информация о наступивших повреждениях модуля. В окне “Standard Diagnosis of the Module” представлены внутренние и внешние повреждения и сопутствующая диагностическая информация (см. рисунок 7.7).

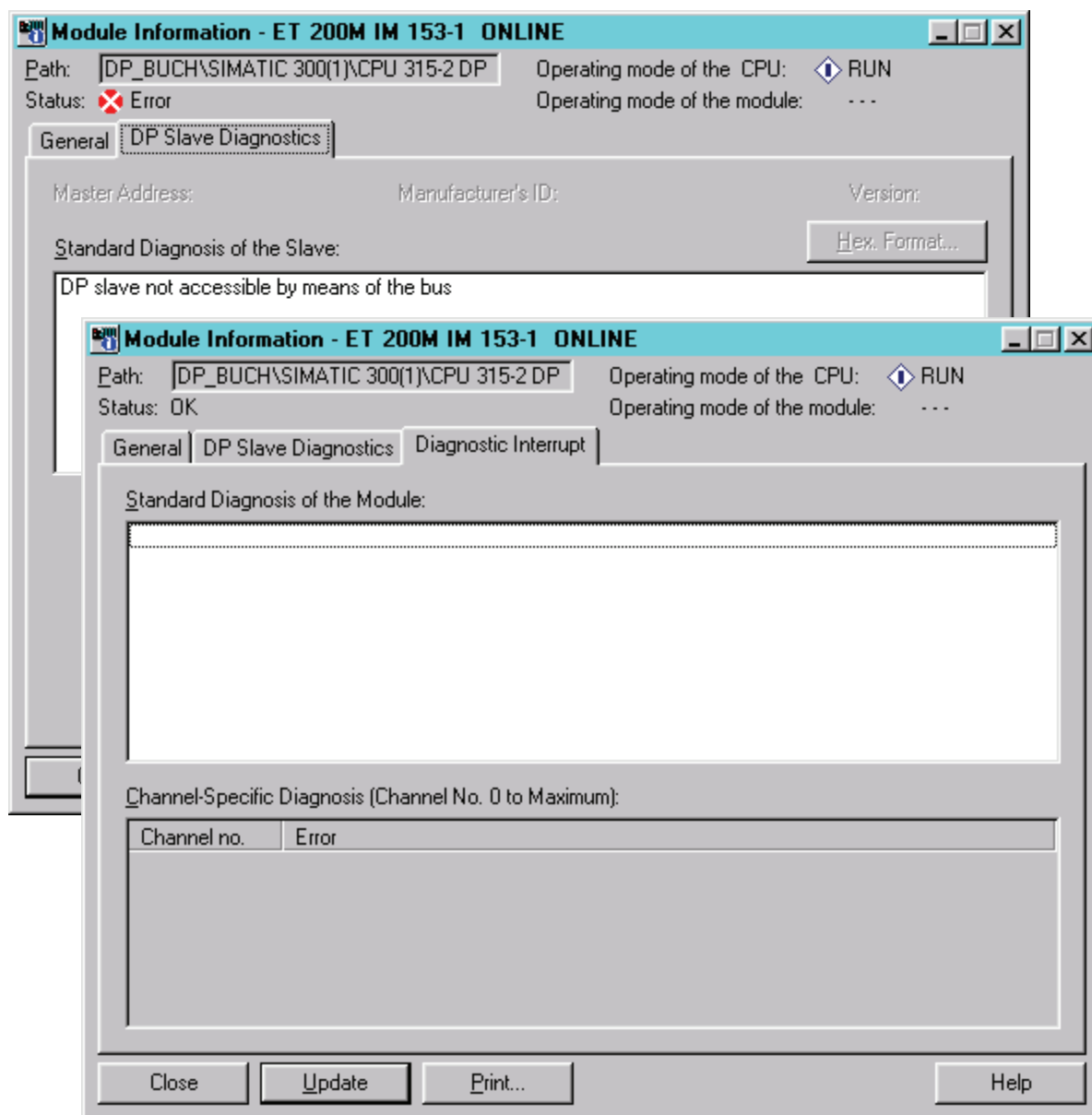


Рис. 7.7 Диалоговые регистры “DP Slave Diagnostics” и “Diagnostic Interrupt”

В окне “Channel-Specific Diagnosis” показываются диагностические данные появившихся канальных ошибок, например:

- Ошибка параметрирования;
- Обрыв провода.

Диагностика DP-Slave

Закладка “DP Slave Diagnosis” информирует Вас о диагностических данных DP-Slave’a, которые имеют структуру, согласно EN 50170 (см. рисунок 7.8).

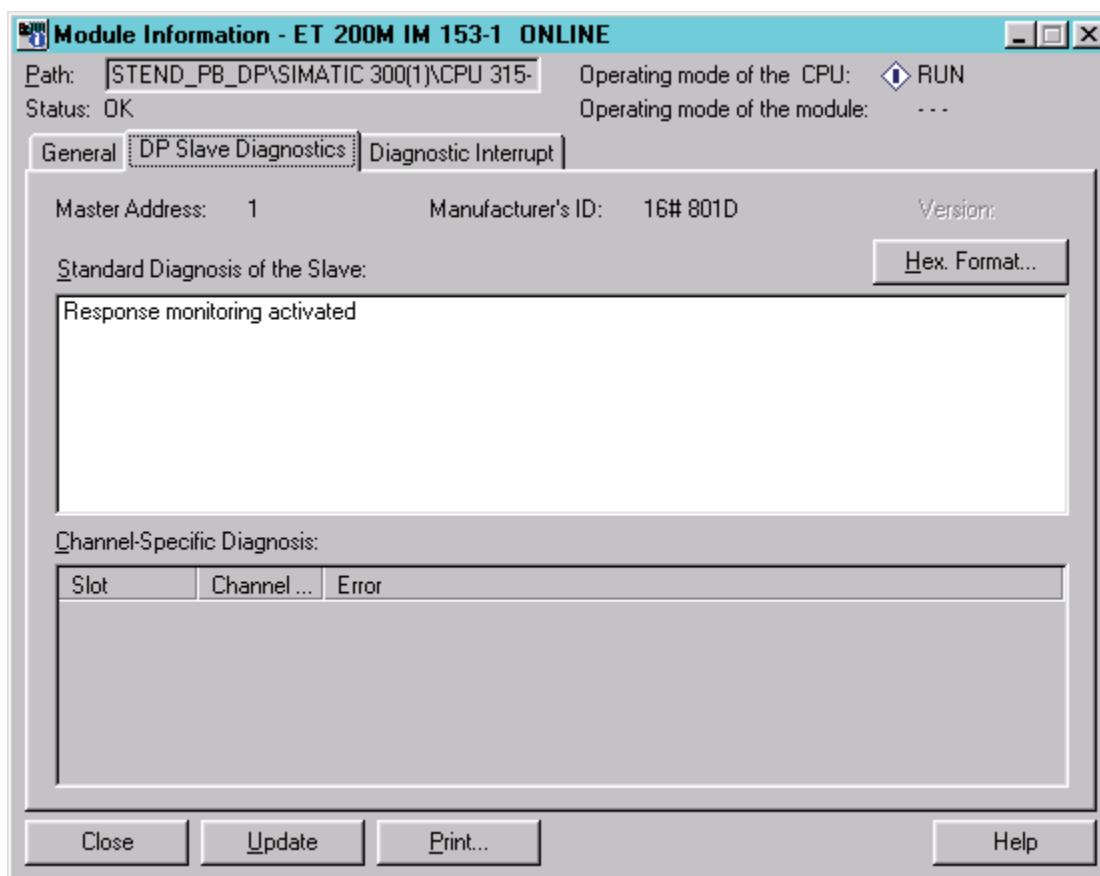


Рис.7.8 Диалоговое окно “DP Slave Diagnostics”

В окне “Standard Diagnosis of the Slave” показывается общая и зависящая от прибора диагностическая информация для Slave’a.

- Общая диагностическая информация для DP-Slave’a. Информация относится к корректной работе или ошибкам DP-Slave’a. К последним относятся, например, ошибки конфигурирования и параметрирования.
- Диагностическая информация DP-Slave’a, зависящая от устройства. Показываемые диагностические записи определяются на основе специфического для прибора GSD-файла (Geräte Stamm Daten – нем.). Если диагностические сообщения не содержатся в GSD-файле, диагностика не может выдаваться в виде ясных текстов.

В окне “Channel-Specific Diagnosis” показываются относящиеся к каналу диагностические записи для конфигурированного модуля DP-Slave. Для каждого вносимого диагностического сообщения выдается точно канал, послуживший причиной. Канал однозначно описывается благодаря номеру слота модуля и номеру канала.

Специфические для прибора диагностические тексты определяются с помощью GSD-файла. Если диагностические сообщения не содержатся в GSD-файле, диагностика не дает ясных текстов. С помощью кнопки “Hex. Format” можно всю диагностическую телеграмму выдать в 16-ичном представлении.

7.2.4 Диагностика с помощью функции “Diagnosing Hardware”

Функция “Diagnosing Hardware” может быть вызвана различными способами.

- Через окно “Accessible Nodes”, вызов желаемого партнера с помощью правой клавиши мыши, потом – в контекстном меню *PLC->Diagnosing Hardware*.
- Через функцию “Online” в SIMATIC Manager. Переключить проект в online-представление, щелкнуть на желаемой станции правой кнопкой мыши и затем – в контекстном меню *PLC->Diagnosing Hardware*.

Появляется окно “Diagnosing Hardware - properties”. В этом окне показываются символы для состояний блоков. Если, например, модуль поврежден, то в быстром просмотре рядом с CPU показывается также символом DP-Slave (рисунок 7.9).

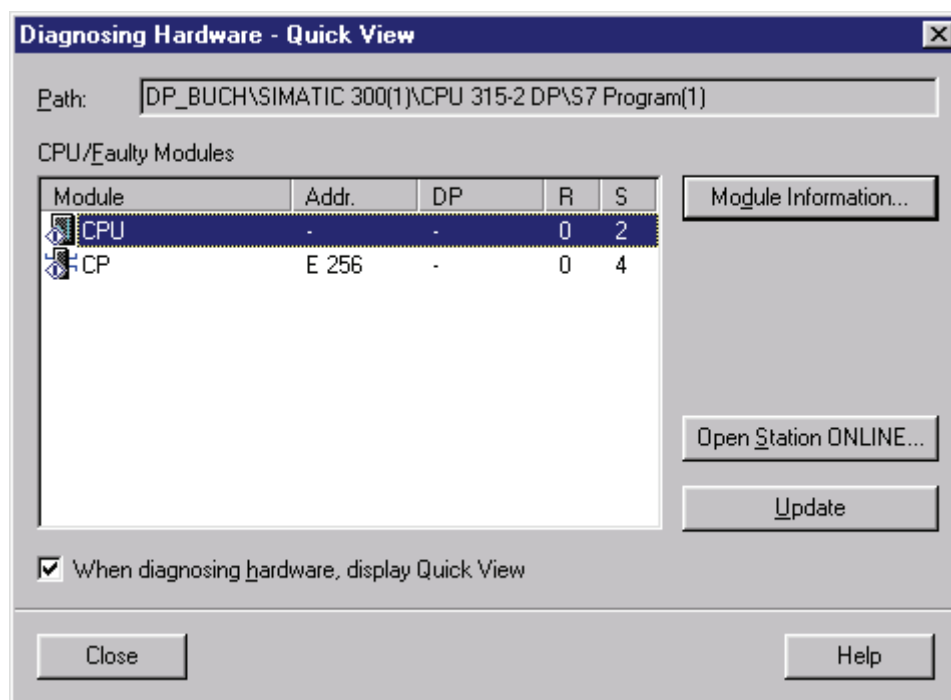


Рис. 7.9 Диалоговое окно “Diagnosing Hardware – Quick View”

В таблице 7.12 описаны общие символы. Ошибки у модулей с диагностикой отображаются соответствующим символом, если диагностические прерывания деблокированы.

Табл. 7.12 Общее описание диагностических символов

Диагностический символ	Значение
Красная диагональная полоса перед символом блока	Отличие заданной от действительной конфигурации: спроектированный модуль не представлен или установлен другой модуль
Красный круг с белым крестом	Модуль неисправен. Возможные причины: распознан диагностический сигнал или ошибка доступа к периферии
Изображение модуля с пониженным контрастом	Диагностика невозможна, так как нет online-соединения или CPU не получает диагностической информации от модуля (напр., от источника питания).
Красная скоба над модулем	В этом модуле проводится форсирование переменных, т.е. переменные заданы постоянными, программа их значения не может менять. Идентификатор для FORCE может появиться также в связи с другими символами

Диалоговое окно “Diagnosing Hardware - properties” предлагает благодаря трем кнопкам различные функции на выбор (рисунок 7.9).

Через кнопку “Module Information” открывается соответствующее окно. Через кнопку “Update” можно актуализировать содержание диалогового окна “Diagnosing Hardware – Quick View”. Через кнопку “Open Station ONLINE” загружается аппаратная конфигурация выбранной станции.

При этом проверяется конфигурация каждого модуля. Неисправный модуль или модуль с ошибкой помечается соответствующим символом (рисунок 7.10).

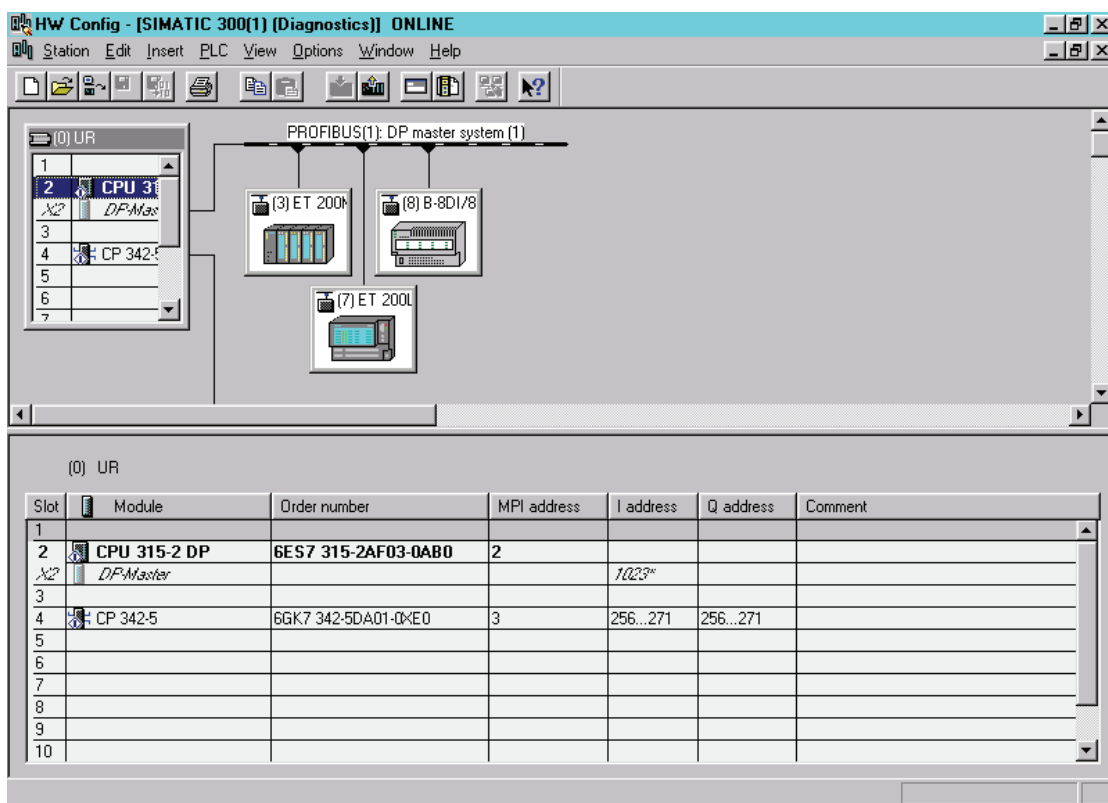


Рис.7.10 Загрузка конфигурации через “Diagnosing hardware ”

Дальнейшую диагностику можно провести, если щелкнуть по модулю правой клавишей мыши и в контекстном меню выбрать “Module Information”.

7.3 Диагностика через программу пользователя

Система автоматизации SIMATIC S7 предлагает ряд диагностических функций, которые могут выполняться из программы пользователя. При использовании диагностических функций можно при неисправности установки зарегистрировать причину неисправности и соответствующим образом реагировать в программе пользователя. В следующей главе показан пример только одной возможной диагностической функции, которая относится к примеру из раздела 4.2.5.

7.3.1 Диагностика DP-Slave с помощью SFC 13 DPNRM_DG

Чтобы можно было диагностировать DP-Slave, можно прочитать с помощью SFC13 DPNRM_DG стандартную диагностику DP-Slave. Представленные данные соответствуют диагностическим данным по EN 50170.

Максимальная длина телеграммы, которую может прочитать SFC13, ограничена 240 байтами, хотя максимальная длина телеграммы от Slave’а по EN 50170 ограничена 244 байтами. В этом случае устанавливается в читаемых диагностических данных “Overflow-Bit”. Общая структура диагностических данных представлена на рисунке 7.11.

Использовать SFC13 можно в циклической программе (OB1), в OB диагностических сигналов (OB82), а также при выходе из строя станции, соответственно, при восстановлении станции (OB86). Обратите внимание, что процесс чтения SFC13 выполняет асинхронно, это означает, что процесс чтения требует несколько вызовов системной функции после запуска (REQ=1), чтобы прочитать диагностические данные DP-Slave’а и записать их в область, заданную в параметре RECORD.

Чтобы при неисправностях или выходе из строя, при которых вызываются OB82 и OB86, можно было читать актуальные диагностические данные DP-Slave’а, SFC13 должна вызываться повторно в цикле, пока выходной параметр BUSY не просигнализирует, что перенос данных завершен.

Рисунок 7.12 показывает вызов SFC13 в OB82, с помощью которого могут быть зарегистрированы неисправности ET200B 16DI/16DO. Приходящие и уходящие причины неисправностей оцениваются отдельно и заносятся в две различных области данных. SFC13 работает благодаря программированию в цикле. Сигналом выхода из цикла служит появление на выходе BUSY сигнала “0”. Способ вызова SFC13 представлен на рисунке 7.13.

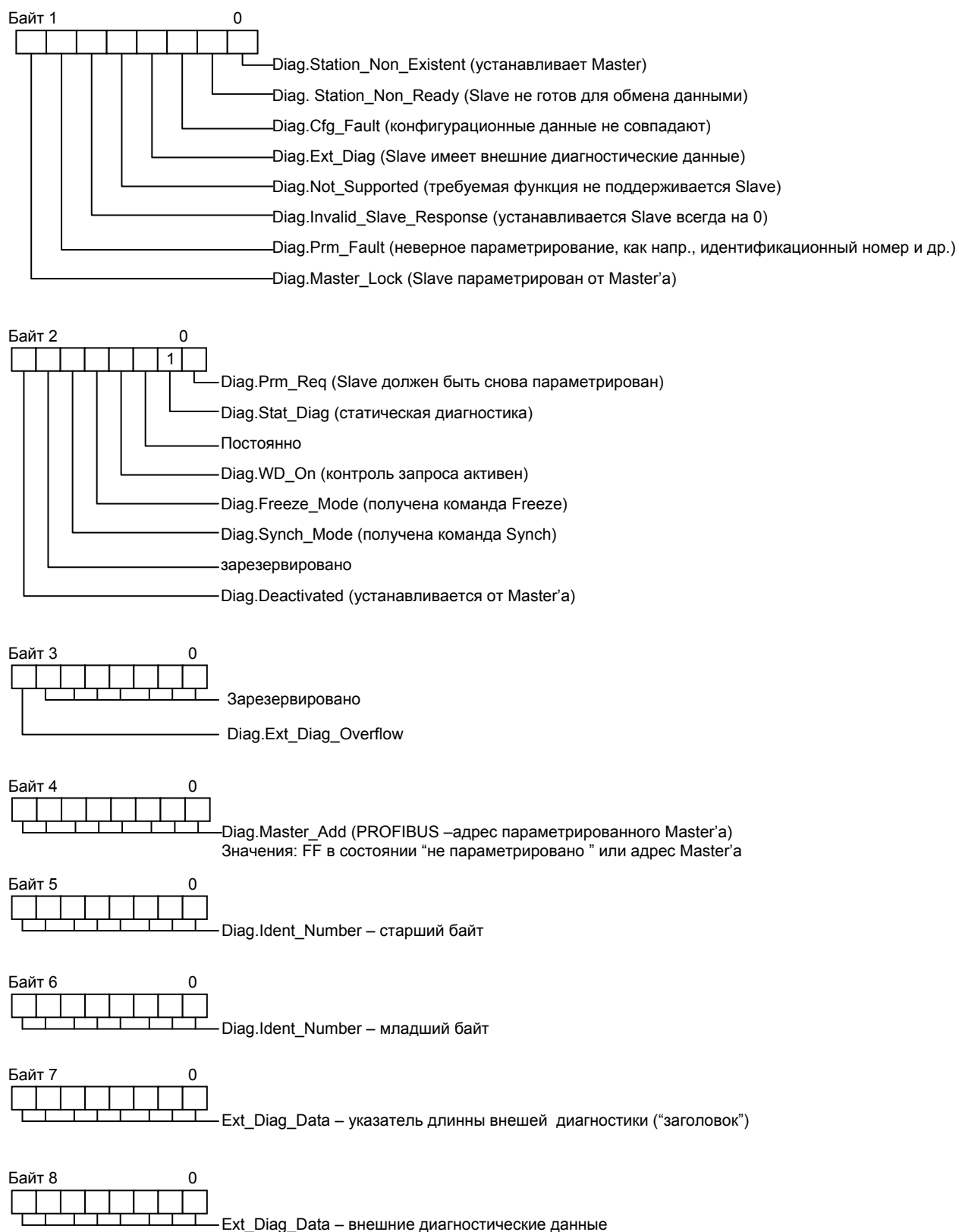


Рис. 7.11 Общая структура диагностических данных по EN 50170

```

L   #OB82_EV_CLASS //Загрузка класса события
L   B#16#39         //Проверка на “приходящее”
==|
JC   GO1

//Часть программы для чтения “уходящего” диагностического события

GO2: CALL "DPNRM_DG"
      REQ :=TRUE
      LADDR :=W#16#1FFC //Диагностич. адрес станции ET200B
      RET_VAL:=MW240
      RECORD :=P#DB13.DBX 100.0 BYTE 32 //Диагностич. данные в DB13 с DBB100
      BUSY :=M230.0

      A   M   230.0
      JC   GO2
      BEU

// Часть программы для чтения “приходящего” диагностического события

GO1: CALL "DPNRM_DG"
      REQ :=TRUE
      LADDR :=W#16#1FFC // Диагностич. адрес станции ET200B
      RET_VAL:=MW240
      RECORD :=P#DB13.DBX 0.0 BYTE 32 // Диагностич. данные в DB13 с DBB0
      BUSY :=M230.0

      A   M   230.0
      JC   GO1

```

Рис. 7.12 Вызов SFC13 *DPNRM_DG* в OB82

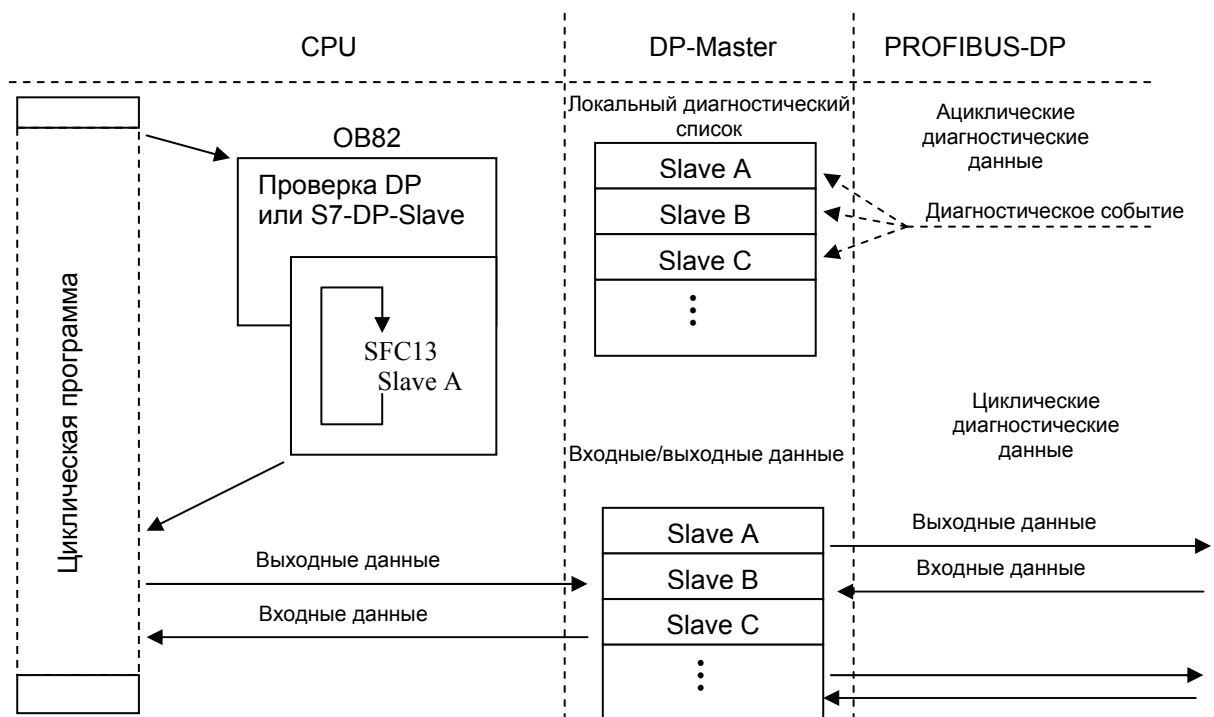


Рис. 7.13 Вызов функции SFC13 *DPNRM_DG* в OB82 в примере

Чтобы тестировать программу-пример, Вы должны создать DB13 с минимальной длиной 132 байта и запрограммировать вызов SFC13 в OB82, как показано на рисунке 7.13. Запустите для этого SIMATIC Manager и откройте проект-пример S7_PROFIBUS_DP из раздела 4.2.5. Проверьте еще раз аппаратную конфигурацию CPU S7-400. К DP-Master'у должны быть подключены только ET200B 16DI/16DO. Проведите сброс CPU, установите вид рабочего режима CPU 416-2DP STOP и загрузите конфигурацию в CPU. Затем с помощью кабеля PROFIBUS создайте связь между DP-интерфейсом CPU и DP-интерфейсом модуля ET200B. Установите переключатель режимов CPU в положение RUN-P.

CPU переходит в режим RUN и все светодиодные индикаторы ошибок гаснут. Откройте теперь в SIMATIC Manager контейнер блоков CPU 416-2DP и выберите через контекстное меню (правая клавиша мыши) *Insert->Organisations Block*. В следующем диалоговом окне задайте "OB82" и нажмите "OK". Благодаря этому в контейнере объектов появляется пустой блок OB82. Откройте OB82 с помощью двойного щелчка. Открывается редактор LAD/FBD/STL, а в нем – блок OB82, Вы можете в нем написать программу (см. рисунок 7.12).

Сохраните OB82 и загрузите его в CPU. Включите функцию STEP 7 "Monitor". Чтобы вызвать неисправность ET200B, отключите напряжение питания модуля. В следствие этого вызывается OB82 и осуществляется оценка диагностики благодаря SFC13. При этом в STL-редакторе показывается состояние обработки. При необходимости можно теперь оценить считанные данные с помощью S7-функции *Monitor/Modify Variables*.

7.3.2 Диагностика с помощью SFC 51 RDSYSST в OB82

S7-DP-Slave'ы, соответственно, модули S7-300 в S7-DP-Slave'е, предлагают расширенные диагностические функции. В модулях S7-DP-Slave, таких, как ET200M с несколькими модулями S7-300, можно точно диагностировать отдельные модули. Эта диагностика может быть проведена с помощью SFC 51 RDSYSST. При неисправности эти компоненты в состоянии возбуждать диагностические сигналы в DP-Master'е, соответственно, в CPU. В CPU благодаря этому вызывается OB82.

SFC 51 – асинхронно работающая системная функция. Это означает, что она использует несколько вызовов, пока данные считываются и заносятся в целевую область, указанную в параметре DR. *Только если SFC 51 вызывается в OB82 и считывает из модуля набор данных "0" или "1", вызов выполняется немедленно, то есть синхронно.* Так же в процессе чтения гарантируется, что будут считаны только диагностические данные, которые послужили причиной неисправности.

Благодаря вызову SFC 51 можно S7-DP-Slave'ы, соответственно, модули S7-300 точно диагностировать благодаря тому, что считывается набор данных "0" (4 байта), соответственно, набор данных "1" (16 байт). Считанные наборы данных соответствуют по структуре и содержанию диагностическим данным центральных модулей. Благодаря этому можно проводить прямую диагностику центральных и распределенных модулей.

Благодаря использованию локальных данных OB82 можно гибко программировать вызов SFC 51. Это означает, что можно отдельный вызов SFC

51 программировать не для каждого S7-DP-Slave'a, соответственно, модуля S7-300.

В приведенном ниже листинге читается набор данных "1" неисправного модуля, который возбудил диагностический сигнал. Осуществляется отличие между "приходящей неисправностью" и "уходящей неисправностью". Точная оценка прочитанных диагностических данных может проводиться затем в OB82 или в циклической программе (OB1).

SFC 51 программируется в примере с помощью локальных данных OB82. Локальная переменная OB82_EV_CLASS (класс события и идентификатор) имеет следующие значения:

- Уходящее событие B#16#38;
- Приходящее событие B#16#39.

Локальная переменная OB82_IO_FLAG (тип модуля) снабжается следующими значениями:

- Входной модуль B#16#54;
- Выходной модуль B#16#55.

Для вызова SFC 51 в OB82 используйте представленную в таблице 7.13 структуру переменной "SZL_HEADER". Дополните локальные данные OB82 переменной "SZL_HEADER".

Табл. 7.13 Структура переменной "SZL_HEADER"

Имя	Тип
SZL_HEADER	STRUCT
LENGTH_DR	WORD
NUMBER_DR	WORD
END_STRUCT	

Параметр INDEX должен для вызова SFC 51 быть обеспечен следующими данными:

в OB82_MDL_ADDR 15-й бит должен быть установлен, если диагностический сигнал затребован от *выходного* канала. OB82 должен быть далее запрограммирован, как показано на рисунке 7.14.

Листинг можно создать и тестировать тем же способом, как уже описанный при вызове SFC13. Измените аппаратную конфигурацию станции S7-400: удалите ET200B 16DI/16DO от DP-Master'a и спроектируйте ET200M/IM153-2, как описано в разделе 4.2.5. Перенесите измененную аппаратную конфигурацию и OB82 с текстом программы. Установите связь между ET200M и DP-интерфейсом CPU 416-2DP. Чтобы вызвать диагностический сигнал, выключите напряжения питания в ET200M. ET200M генерирует диагностический сигнал, который будет прочитан в OB82.

В работающей программе можно после вызова SFC 51 провести оценку прочитанных данных и, таким образом, осуществить целевую реакцию.

```

L   #OB82_IO_FLAG //Опрос типа блока
L   B#16#54        //Загрузить идентификатор входного блока
==I //Входной блок ?
JC  GO             //Бит 15 остается неизменным при входном блоке

//.
L   #OB82_MDL_ADDR //Предоставляемый адрес из локальных данных
L   W#16#8000      //Загрузить 8000h
OW  //Побитовое ИЛИ для слова – установить 15-й бит
T   #OB82_MDL_ADDR //Сохранить в локальных данных

//.
//*****
//.
//Определить, произошло приходящее или уходящее событие
GO: L   #OB82_EV_CLASS //Класс события и идентификатор
    L   B#16#39        //Идентификатор для приходящего события
    ==I //Событие приходящее
    JC  come           //Переход для чтения
//*****
//Чтение и сохранение диагностической информации
CALL SFC 51 //Уходящее событие
REQ  :=TRUE          //Всегда TRUE
SZL_ID :=W#16#B3      //Идентификатор набора данных 1
INDEX :=#OB82_MDL_ADDR //Образованный адрес
RET_VAL :=MW100       // RET_VAL в MW100
BUSY  :=M102.0       // Флаг BUSY в M102.0
SZL_HEADER:=#SZL_HEADER //Локальная структура данных
DR    :=P#M 10.0 BYTE 16 //Прочитанные данные с меркерного слова 10
BEU
come: CALL SFC 51 //Приходящее событие
REQ  :=TRUE          //Всегда TRUE
SZL_ID :=W#16#B3      //Идентификатор набора данных 1
INDEX :=#OB82_MDL_ADDR //Образованный адрес
RET_VAL :=MW104       // RET_VAL в MW104
BUSY  :=M102.7       // Флаг BUSY в M102.7
SZL_HEADER:=#SZL_HEADER //Локальная структура данных
DR    :=P#M 20.0 BYTE 16 //Прочитанные данные с меркерного слова 20

```

Рис 7.14 Вызов SFC51 в OB82

7.3.3 Диагностика с помощью SFB54 RALRM

DP-Slave'ы, соответственно, модули в DP-Slave'ах могут в зависимости от своей функциональности посылать различные прерывания. При этом часть диагностических данных посылается в локальный стек вызываемого OB. Полная набор диагностической информации может быть прочитан с помощью вызова SFB54 RALRM в соответствующем OB.

Если SFB54 вызывается в некотором OB, стартовое событие которого не является сигналом из периферии, SFB предоставляет на своих выходах соответственно меньше информации (см. также гл. 5.4.7, табл. 5.37). Кроме того, при каждом вызове SFB54 в различных OB должны использоваться новые (различные) экземпляры DB. Если данные, получаемые с помощью вызова SFB54, должны использоваться вне вызывающего OB, должен использоваться свой экземпляр DB даже для каждого стартового события.

SFB54 может вызываться в различных режимах. Задание режима осуществляется через соответствующий входной параметр SFB54 – MODE:

- В режиме “0” в параметре ID будет выдаваться DP-Slave, соответственно, блок, пославший сигнал, а параметр NEW будет содержать “TRUE”. Все остальные параметры не изменяются.
- В режиме “1” напротив будут переписываться все выходные параметры соответствующими диагностическими данными, независимо от компонента, пославшего сигнал.
- В режиме “2” SFB54 проверяет, послал ли компонент, заданный входным параметром F_ID, сигнал. Если да, то выходной параметр NEW получит значение “TRUE” и все другие выходные параметры получают соответствующие данные. В случае, если F_ID и компонент, пославший сигнал не совпадают, параметр NEW содержит значение “FALSE”,

В следующем примере программы (рис. 7.15) с помощью SFB54 оцениваются диагностические данные в OB82. При этом целевая область для диагностических данных должна быть достаточна, чтобы поместились

- стартовая диагностика (6 байт),
- диагностика, специфицированная идентификатором (3 байта для 12 слотов),
- диагностика, специфичная для прибора (еще 7 байт для состояния модуля).

Для дальнейшей расширенной оценки (диагностики, специфичной для канала) должны быть зарезервированы дополнительные байты, если DP-Slave поддерживает эту функциональность.

Листинг можно проверить тем же самым способом, как описано в главе 7.3.2:

Откройте OB82 и сотрите предыдущую программу. Затем введите соответствующий листинг и загрузите OB82 через MPI-интерфейс в CPU.

При поступлении диагностического сигнала будут прочитаны диагностические данные с помощью SFB54.

В работающей программе установки можно после вызова SFB54 провести оценку прочитанных данных и, таким образом, осуществить необходимую реакцию пользовательской программы.

```

//...
//*****
//Вызов SFB54. В качестве экземпляра DB выбран DB54
//*****
CALL "RALRM" , DB54
MODE :=1           // Все выходные параметры будут установлены
F_ID :=            // Адр. слота, от которого должна быть получена диагностич.
                  //информация
MLEN :=20          // Максимальная длина диагностических данных в байтах
NEW :=M80.0
STATUS:=MD90       // Результирующее значение, сообщение об ошибке
ID :=MD94          // Адр. слота, от которого принимается Alarm
LEN :=MW82         // Длина дополнительной информации (4 байта заголовков +
16 байт           // диагн. данные)
TINFO :=P#M 100.0 BYTE 28 // Указатель на стартовую информацию OB +
                  // управляющую информацию: 28 байт
AINFO :=P#M 130.0 BYTE 20 // Указатель на целевую область - диагностические
                  // данные
//*****
// Структура диагностических данных
// MB130 - MB133 информация заголовка (длина, идентификатор, слот)
// MB134 - MB139 стандартная диагностика (6 байт)
// MB140 - MB142 диагностика, определяемая спецификатором
// MB143 - MB149 состояние модуля (7 байт)
//...
//*****
// Примерная оценка диагностических данных
//*****
A M 141.0          // Слот 1 имеет ошибку ?
JC stp1
BE
//*****
// Оценка ошибки в слоте 1
//*****
stp1: L MB 147 // Состояние модулей в слотах 1-4
AW W#16#3 // Выделяем слот 1
L W#16#2 // 2-х битовое состояние "wrong module"- установлен неправильный
// модуль

==|
S Q 0.1 // Реакция на неправильный модуль
L MB 147 // Состояние модулей в слотах 1-4
AW W#16#3 // Выделяем слот 1
L W#16#1 // 2-х битовое состояние "invalid data" - незаконные пользовательские
// данные

==|
S Q 0.2 // Реакция на недействительные данные
//*****
// Выполнение окончания
//*****
//..

```

Рис. 7.15 Вызов SFB54 в OB82

7.4 Диагностика с помощью диагностического блока SIMATIC S7 FB125

Сам блок и информация о нем находится на Internet-сервере Siemens A&D CS по адресу <http://www4.ad.siemens.de> -> Finden -> Suchbegriff: FB125

7.4.1 Блок диагностики FB125

FB125 предлагает возможность детальной диагностики DP-Master-системы. Блок определяет DP-Slave, которые вышли из строя или имеют сбой. Для “сбойного” Slave’a показывается детальная диагностическая информация о причине отказа (место установки, соответственно, номер модуля, состояние модуля, номер канала, ошибку канала). Благодаря отдельной диагностике (индивидуальной диагностике) можно читать диагностическую информацию произвольного DP-Slave’a и интерпретировать ее. Обзорная диагностика дает информацию о спектрированных, имеющихся, вышедших из строя и сбойных DP-Slave’ах.

Во время обработки FB задерживаются все прерывания.

7.4.2 Области применения FB125

FB125 применим для следующих интегрированных и внешних DP-интерфейсов:

CPU 31x-2DP (с 6ES7 315-2AF01-0AB0)

C7-626DP (с 6ES7 626-2AG01-0AE3)

C7-633DP и C7-637DP

SINUMERIK 840D

CPU 41x-2DP

CP 443-5

IM 467 и IM467FO

WIN AC

WIN LC

Не для CP 342-5

7.4.3 Вызов блока диагностики DP FB125 в S7-пользовательской программе

Блок диагностики DP вызывается в пользовательской программе с помощью следующей инструкции:

CALL FB125, DB ху (ху – любой номер экземпляра блока данных).

Вслед за этим автоматически появляется список параметров с формальными операндами, которые снабжаются фактическими операндами. Замечание: при вызове FB не обязательно все параметры снабжать фактическими значениями, так как фактические операнды хранятся в экземпляре DB.

Этот вызов (включая одинаковые номера экземпляров DB и одинаковые фактические операнды) может происходить в трех уровнях обработки OB1, OB82 и OB86. Возможен вызов FB125 в этих трех программных уровнях из каких-либо FB и FC (например: OB82 ->FC120->FB125).

Состав и содержание 20-и байт временных локальных данных организационных блоков OB1, OB82 и OB86 могут не изменяться, но могут и быть расширены. Внутри FB125 применяются следующие SFC: SFC13 и SFC51 с SZL 0C91 (при внутреннем DP-интерфейсе в Master-CPU) и, соответственно, SZL 4C91 (при внешнем DP-интерфейсе в CP/IM). Невозможно в OB1 наряду с вызовом FB125 дополнительно вызывать для Slave'а со сбоями SFC13 и/или SFC51 с SZL 0C91/4C91.

Оценку информации из выходных параметров блока целесообразно проводить в циклической части программы (OB1).

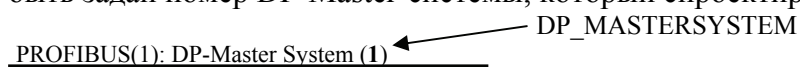
При использовании нескольких DP-Master-систем (например, интегрированный DP-интерфейс (CPU) и внешний DP-интерфейс (CP/IM)) блок диагностики должен вызываться отдельно для каждой DP-Master-системы. При каждом новом вызове FB125 должен назначаться также новый экземпляр DB. Вызов блока для внутреннего интерфейса должен быть сделан перед вызовом его для внешнего интерфейса.

Например, CALL FB125, DB 125 (Master system 1, внутренний DP-интерфейс)
CALL FB125, DB 126 (Master system 2, внешний DP-интерфейс)

7.4.4 Параметры блока диагностики DP FB125 и их значение

Входные параметры:

- DP_MASTERSYSTEM (Int) Идентификатор DP-Master System. Здесь должен быть задан номер DP-Master-системы, который спроектирован в STEP 7.

DP_MASTERSYSTEM

PROFIBUS(1): DP-Master System (1)

- EXTERNAL_DP_INTERFACE (Bool) Сообщается, идет ли речь о встроенном (Master-CPU=0) или внешнем (CP/IM=1) DP-интерфейсе мастера.
- MANUAL_MODE (Bool) Ручной режим работы: при этом режиме работы возможна индивидуальная диагностика.
- SINGLE_STEP_SLAVE (Bool) Переход к следующему DP-Slave'у, который неисправен или имеет сбой.
- SINGLE_STEP_ERROR (Bool) Переход к следующей ошибке в указанном DP-Slave'е.
- RESET (Bool) Обновление оценки. Оценка DP инициализируется и снова стартует. Общая DP-Master-система снова регистрируется, то есть все спроектированные, существующие, неисправные и сбойные DP-Slave'ы автоматически определяются в подпрограмме инициализации. *Следует запускать при каждом новом чтении.*
- SINGLE_DIAG (Bool) Ручной режим: чтение индивидуальной диагностики DP-Slave'а. Номер Slave'а задается пользовательской программой в параметре "SINGLE_DIAG_ADR".
- SINGLE_DIAG_ADR (Byte) Ручной режим: номер Slave'а для индивидуальной диагностики.

Выходные параметры

- ALL_DP_SLAVE_OK (Bool) Общая индикация, что DP-система работает без сбоев (TRUE – все Slave'ы OK)
- SUM_SLAVES_DIAG (Byte) Число соответствующих Slave'ов (неисправных и со сбоями). Число определяется уже при полном старте или рестарте.
- SLAVE_ADR (Byte) DP-Slave, который неисправен или имеет сбой (от 1 до 126). При каждом импульсе в SINGLE_STEP_SLAVE показывается следующий неисправный, соответственно, сбойный Slave. Упомянутые Slave'ы показываются в порядке возрастания их адресов.
- SLAVE_STATE (Byte) Информация для SLAVE_ADR: состояние показываемого Slave'а:
 - 0: DP-Slave OK
 - 1: DP-Slave неисправен (отсутствует)
 - 2: DP-Slave имеет сбой
 - 3: DP-Slave не спроектирован, соответственно, DP-Slave не имеет диагностики.
- SLAVE_IDENT_NO (Word) Информация для SLAVE_ADR: идентификационный PROFIBUS-номер показываемого DP-Slave'а по PNO.
- ERROR_NO (Byte) Информация для SLAVE_ADR: номер актуальной ошибки, которая показывается для соответствующего Slave'а. Таким образом, можно все ошибки отличать друг от друга благодаря однозначной нумерации.
- ERROR_TYPE (Byte) Информация для SLAVE_ADR:
 - 1: Диагностика стойки (места установки) (общее сообщение для вышедших из строя стойки/модуля).
 - 2: Состояние модуля (улучшение диагностики носителя для состояния носителя/модуля).
 - 3: Диагностика канала (локализация № модуля, № канала, типа ошибки канала по нормам DP).
 - 4: S7-диагностика (локализация № модуля, № канала, типа ошибки канала). Справедливо только для S7-Slave'ов, соответственно, S7-модулей фирмы Siemens. Диагностическая информация берется из S7-диагностических наборов данных DS0 и DS1.
- MODULE_NO (Byte) Информация для SLAVE_ADR: номер вышедшего из строя модуля Slave'а (№ слота, соответственно, № модуля).
- MODULE_STATE (Byte) Информация для SLAVE_ADR: состояние модуля (только для состояний модулей, указанных в ERROR_TYP)
 - 0: Модуль OK, пользовательские данные имеют силу (законны).
 - 1: Ошибка модуля, пользовательские данные не имеют силы.
 - 2: Неверный модуль, пользовательские данные не имеют силы.
 - 3: Нет модуля, пользовательские данные не имеют силы.
- CHANNEL_NO (Byte) Информация к SLAVE_ADR: номер вышедшего из строя канала модуля.
- CHANNEL_ERROR_INFO (Dword) Информация к SLAVE_ADR: двоично кодированная информация об ошибке для вышедшего из строя канала. Типам ошибок соответствует различная информация об ошибке. Типу ошибок (3) диагностика канала соответствует информация об ошибках канала согласно нормам DP.

Бит	Ошибка канала согласно нормам DP
0	Зарезервировано
1	Короткое замыкание (например, провода датчика или привода)
2	Пониженное напряжение питания
3	Повышенное напряжения питания
4	Перегрузка
5	Перегрев
6	Обрыв провода
7	Превышение верхней границы
8	Выход за пределы нижней границы
9	Ошибка (например, источника питания датчика, источника питания выхода)
10-15	Зарезервировано
16	Ошибка параметрирования
17	Отсутствует напряжение питания или напряжение в датчике
18	Дефект защиты
19	Зарезервировано
20	Ошибка заземления
21	Ошибка опорного канала
22	Потеря сигнала от процесса (Process Alarm)
23	Предупреждение от привода
24	Привод выключен
25	Защита отключена
26	Внешняя ошибка (например, ошибка датчика, привода)
27	Неизвестная ошибка
28-31	Зарезервировано

Типу ошибок (4) S7-диагностика соответствует информация об ошибках канала, которая содержится в диагностических наборах данных DS1 S7-Slave'ов, соответственно, S7-модулей.

Бит	Дискретные входы	Дискретные выходы
0	Ошибка проектирования/параметрирования	Ошибка проектирования/параметрирования
1	Ошибка заземления	Ошибка заземления
2	+ -короткое замыкание (датчика)	+ -короткое замыкание
3	- -короткое замыкание	- -короткое замыкание
4	Обрыв провода	Обрыв провода
5	Ошибка источника питания датчика	Ошибка защиты
6	Зарезервировано	Ошибка рабочего напряжения
7	Зарезервировано	Перегрев
8-15	Зарезервировано	Зарезервировано

Бит	Аналоговые входы	Аналоговые выходы
0	Ошибка проектирования/параметрирования	Ошибка проектирования/параметрирования
1	Ошибка синфазности (синхронизации)	Ошибка синфазности (синхронизации)
2	+ -короткое замыкание (датчика)	+ -короткое замыкание
3	- -короткое замыкание	- -короткое замыкание
4	Обрыв провода	Обрыв провода
5	Ошибка опорного канала	Зарезервировано
6	Выход за нижн. границу области измерения	Внешняя ошибка напряжения нагрузки
7	Выход за верхн. границу области измерения	Зарезервировано
8	Зарезервировано	Присоединение не скоммутировано
9	Зарезервировано	Открытый провод на +
10	Зарезервировано	Открытый провод на -
11	Зарезервировано	Ошибка калибровки
12	Зарезервировано	Ошибка области (выход за нижн. или верхн. границы области)
13	Зарезервировано	Зарезервировано

14	Зарезервировано	Открытый провод источника питания
15	Зарезервировано	Пользовательская калибровка не соответствует параметрированию

S7-функциональные модули (FM)

Бит	FM350	FM350-2
0	Сигнал А ошибочный	Зарезервировано
1	Сигнал В ошибочный	Зарезервировано
2	Сигнал Ношибочный	Зарезервировано
3	Зарезервировано	Зарезервировано
4	Снабжение датчика ошибочно	Снабжение датчика ошибочно
5	Зарезервировано	Зарезервировано
6	Зарезервировано	Провод датчика накоротко замкнут/оборван
7-15	Зарезервировано	Зарезервировано

Бит	FM351	FM352
0	Обрыв провода датчика	Зарезервировано
1	Ошибка датчика абсолютного значения	Ошибка датчика абсолютного значения
2	Ошибочный импульс инкрементирования	Ошибочный импульс инкрементирования соотв. ошибочная нулевая отметка
3	Зарезервировано	Зарезервировано
4	Зарезервировано	Зарезервировано
5	Зарезервировано	Зарезервировано
6	Зарезервировано	Зарезервировано
7	Ошибка функционирования	Ошибка функционирования
8	Ошибочные данные машины	Ошибочные данные машины
9	Ошибочный список параметров шага	Ошибочные данные кулачка
10-15	Зарезервировано	Зарезервировано

Бит	FM353	FM354
0	Зарезервировано	Разрыв канала датчика инкрементирования
1	Зарезервировано	Ошибка датчика абсолютного значения
2	Ошибка нулевой отметки	Ошибочный импульс датчика инкрементирования соотв. ошибочная нулевая отметка
3	Зарезервировано	Зарезервировано
4	Зарезервировано	Зарезервировано
5	Зарезервировано	Зарезервировано
6	Зарезервировано	Зарезервировано
7	Ошибка функционирования	Ошибка функционирования
8-15	Зарезервировано	Зарезервировано

Бит	FM355
0	Аппаратная ошибка аналогового канала
1	Зарезервировано
2	Обрыв провода в аналоговом канале (только 4-20 mA)
3	Зарезервировано
4	Выход аналогового входного сигнала за нижнюю границу
5	Выход аналогового входного сигнала за верхнюю границу
6	Обрыв провода в аналоговом выходном канале
7	Короткое замыкание в аналоговом выходе
8-15	Зарезервировано

- SPECIAL_ERROR_INFO (Dword) Информация для SLAVE_ADR: Всем S7-Slave'ам, соответственно, S7-модулям при типе ошибки (4) S7-диагностика назначается дополнительно специальная информация об ошибках, которая соответствует содержанию набора данных DS0:

Бит	S7-диагностические данные	S7-диагностические данные специально у ASI-Link/CP 342-2
0	Неисправность S7-модуля	Суммарная ошибка
1	Внутренняя ошибка	Внутренняя ошибка (например, дефект EEPROM)
2	Внешняя ошибка	Внешняя ошибка (например, Slave вышел из строя или APF)
3	Имеется ошибка канала	По крайней мере один ASI-Slave отличается от заданного
4	Ошибка внешнего вспомогательного напряжения	Напряжение в ASI-интерфейсе мало (APF)
5	Ошибка фронтштекера	0
6	Ошибка параметрирования модуля	0
7	Ошибочный параметр в модуле	0
8-11	Класс модуля: 0000: зарезервировано 0001: зарезервировано 0010: S7-спец. модуль, напр., капсула-адаптер 0011: Стандартный DP-Slave 0100: S7-интерфейсный модуль (IM) 0101: S7-аналоговый модуль 0110: зарезервировано 0111: зарезервировано 1000: S7-функциональный модуль (FM) 1001: зарезервировано 1010: зарезервировано 1011: S7-подключение к DP, например, I-Slave 1100: S7-коммуникационный процессор (CP) 1101: зарезервировано 1110: зарезервировано 1111: Цифровой модуль	Класс модуля: 1100: S7-коммуникационный модуль (CP)
12	Информация канала имеется	1
13	Пользовательская информация имеется	0
14	Диагностический сигнал от исполнительного устройства	0
15	Зарезервировано	0
16	Пользовательский модуль неправильный или имеет ошибку	По крайней мере один ASI-Slave отклоняется от заданного
17	Коммуникационные ошибки в модуле	0
18	Рабочее состояние (0:RUN, 1:STOP), например, CPU 31x-2DP/BM147 как DP-Slave.	0:ASI-Link находится в нормальном состоянии 1:ASI-Link находится в состоянии offline
19	Внутренний контроль времени (Watchdog)	Аппаратная ошибка (внутренний Watchdog)
20	Выход из строя напряжения питания внутри модуля	0
21	Батарея разряжена	0
22	Выход из строя общей буферизации	0
23	Зарезервировано	0
24	Выход из строя устройства расширения	0
25	Выход из строя процессора	0
26	Ошибка EPROM	EPROM имеет дефект
27	Ошибка RAM	0
28	ADU/DAU - ошибка	0
29	Выход из строя защиты, все защиты канала вышли из строя	0
30	Потерян сигнал от процесса	0
31	Зарезервировано	0

- DIAG_OVERFLOW (Bool) Число одновременно прибывших диагнозов > 32, это означает, что запущено больше диагнозов Slave'ов, чем может

обработать FB125. Этот случай может встретиться, когда CP или IM применяется в качестве DP-Master'a. В этом случае целесообразно провести RESET.

- BUSY (Bool) Этот параметр показывает, что оценка DP-системы благодаря FB в настоящее время идет. Следующая обработка показанной информации имеет смысл только после окончания оценки.

7.4.5 Описание экземпляра блока данных

Экземпляр блока данных назначается диагностическому FB. Номер блоку данных ни в коем случае нельзя изменять с помощью доступа на запись (прямой записью).

Следующие данные могут быть считаны для пользователя из экземпляра DB и оценены.

Байты с 928 по 1171: Стандартные диагностические данные актуального DP-Slave.

Байты с 1172 по 1187: Запроектированные Slave'ы, как битовый список.

Байты с 1188 по 1203: Имеющиеся Slave'ы, как битовый список.

Байты с 1204 по 1219: Вышедшие из строя Slave'ы, как битовый список.

Байты с 1220 по 1235: Slave'ы, имеющие ошибки, как битовый список.

Байты с 1236 по 1251: Вышедшие из строя или имеющие ошибки Slave'ы, как битовый список.

Байты с 1252 по 1267: Соответствующие Slave'ы сохранены, как битовый список. Это означает, что приходящая диагностика Slave'ов (имеющих ошибки или вышедших из строя) будет так долго храниться в этом битовом списке, пока благодаря новому старту CPU или сбросу (RESET) блока этот список не сотрется.

Во всех битовых списках каждому DP-участнику назначается один бит.

7.4.6 Технические данные FB 125

Время выполнения без стоящих в очереди диагностических сообщений:
зависит от DP-Master'a,
например, CPU 315-2DP, как
DP-Master: около 4 ms.

Время выполнения со стоящими в очереди диагностическими сообщениями:
зависит от DP-Master'a,
например, CPU 315-2DP, как
DP-Master: около 11 ms.

Пользовательская память в CPU: 5,4 KByte.

7.4.7 Применение FB 125

Общее число Slave'ов (вышедших из строя и имеющих ошибки) может быть взято из выходного параметра "SUM_SLAVES_DIAG". Если нет соответствующих Slave'ов, бит "ALL_DP_SLAVES_OK", как суммарная информация, установлен.

Благодаря “SINGLE_STEP_SLAVES” будет совершаться переход от одного вышедшего из строя или имеющего ошибки Slave’а к другому. При показе Slave’а можно видеть ошибку через “SINGLE_STEP_ERROR”. Различие ошибок происходит благодаря номеру ошибок, которые при переходе от одной ошибки к другой инкрементируются. Если во время показа ошибок с более высоким номером, возникают обновленные ошибки, возникает опять ошибки с номером 1.

Если для соответствующего Slave, информация которого показывается, поступает новая диагностика (например, о 3-х наступивших ошибках), будет оценка ошибок Slave начата снова, это означает, что будет показана ошибка с номером 1.

Все найденные ошибки будут показаны. Таким образом может также резервироваться информация, которая находится уже в диагностической телеграмме Slave и в выходных параметрах FB125 показываться. Пользователь должен сам различить, какой тип ошибки для него представляет интерес. Он может через выходной параметр “ERROR_TYP” различить и упорядочить тип ошибки.

В ручной режиме (MANUAL_MODE) можно проводить с помощью задания номера DP-Slave’а в параметре “SINGLE_DIAG_ADR” индивидуальную диагностику любого Slave’а. Благодаря “SINGLE_STEP_ERROR” можно затем также просмотреть ошибки Slave’а. Индивидуальная диагностика представляет собой “моментальный снимок” диагностики Slave’а. Во всех битовых списках каждому DP-участнику назначается один бит.

7.5 Диагностика с помощью NCM

Коммуникационные процессоры CP342-5DP и CP443-5 Extended, используемые как Master, можно диагностировать с помощью дополнительного программного пакета NCM (Network Communication Management). Пакет NCM поставляется вместе с базовым пакетом STEP 7. Master-модули могут при этом диагностироваться без проекта STEP 7. Пакет NCM предлагает следующие функции:

- Чтение шинных параметров
- Чтение диагностического буфера
- Список функционирования (Lifelist – нем.) всех подключенных PROFIBUS-устройств
- Чтение статистики PROFIBUS
- Обзор диагностики DP-Master’ов
- Отдельная диагностика DP-Slave’ов, которые работают с этим DP-Master’ом

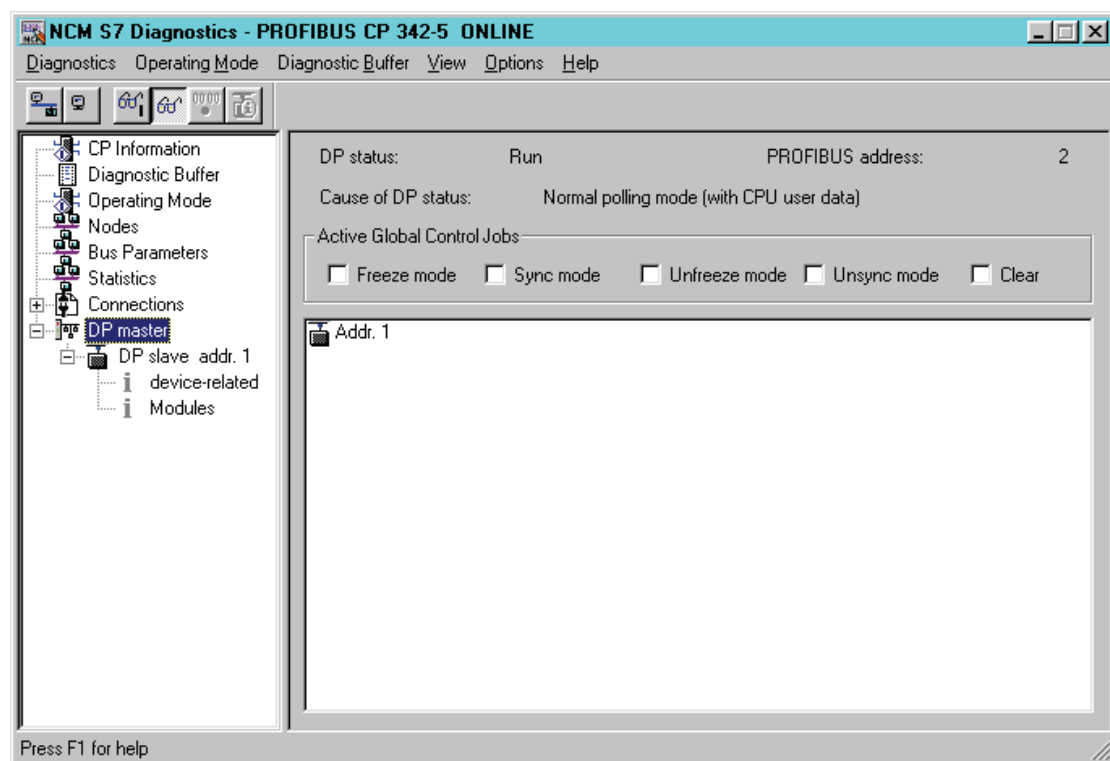


Рис. 7.16 Обзорная NCM-диагностика мастеров

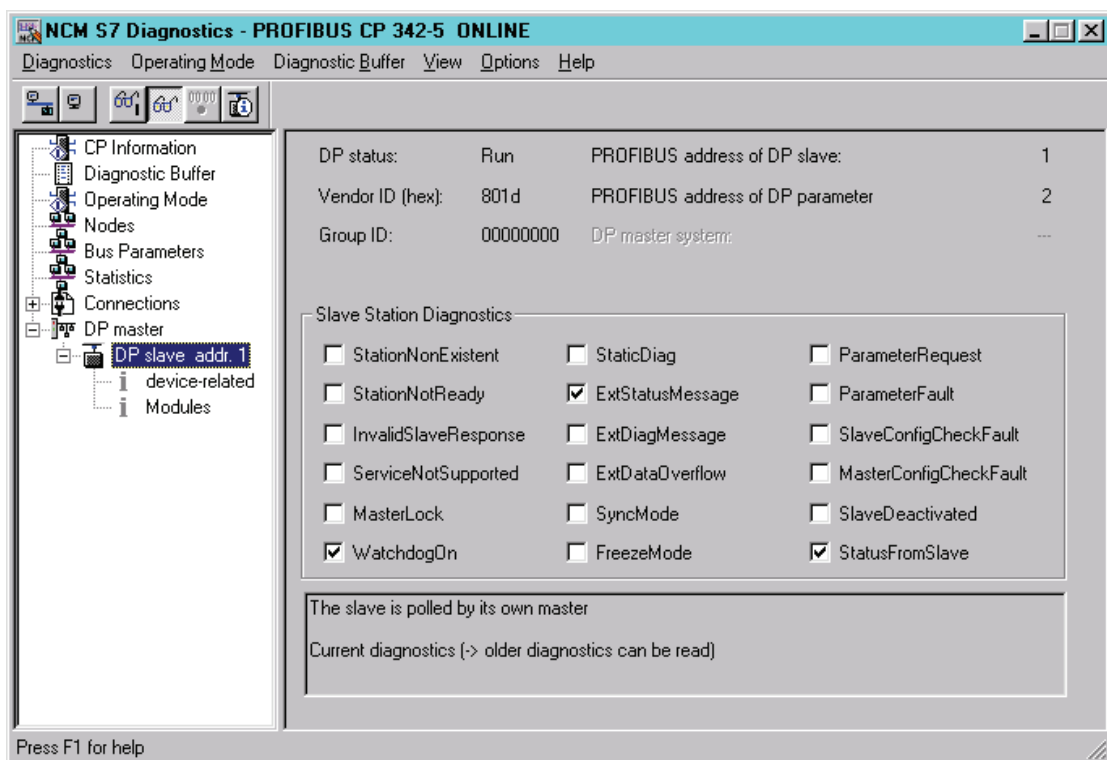


Рис. 7.17 NCM-диагностика для DP-Slave

