

## 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.

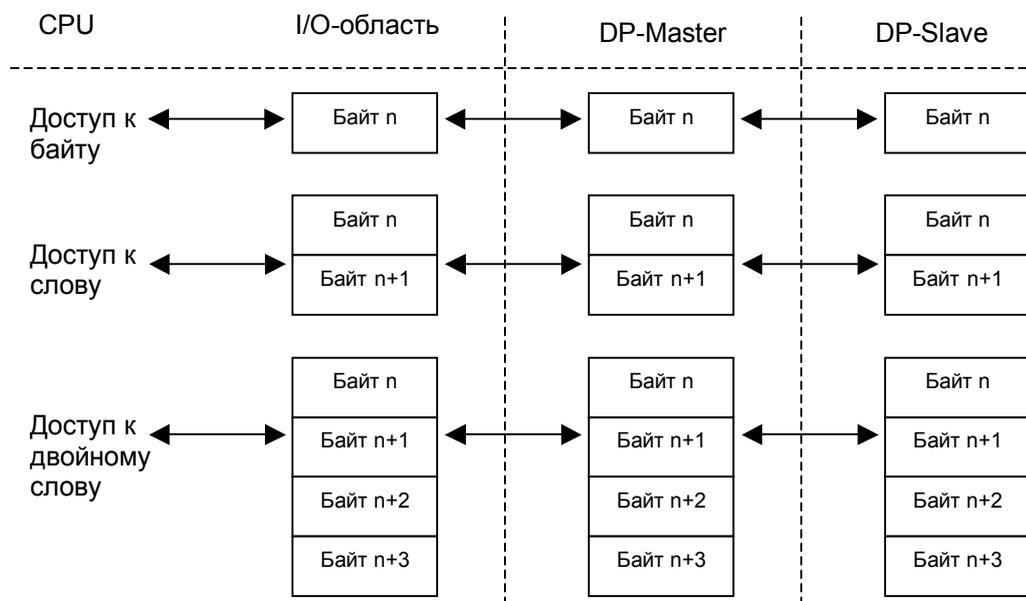
- Для обмена данными со сложными DP-Slave'ами, которые имеют *консистентные* входные/выходные данные, в системе SIMATIC S7 предусмотрены SFC14 DRRD\_DAT и SFC15 DPWR\_DAT.
- Для запуска сигналов от процесса на DP-Master'е из системы S7-300, используемой как I-Slave, предусмотрена функция SFC7 DP\_PRAL.
- 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.

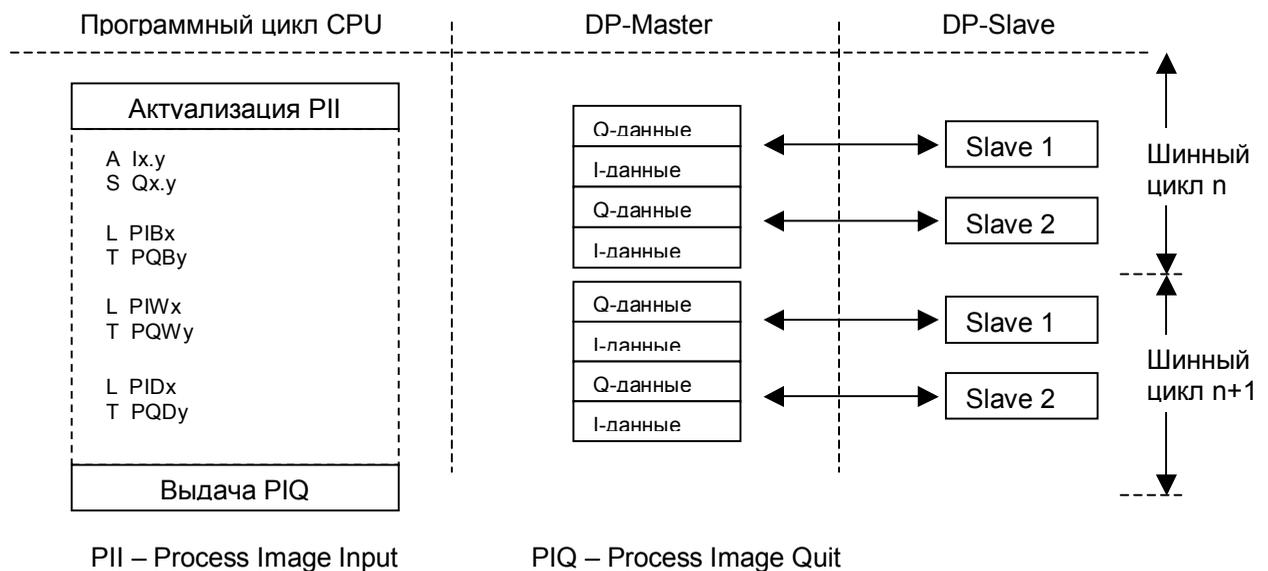
### 6.1 Обмен данными с помощью команд доступа к периферии

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.

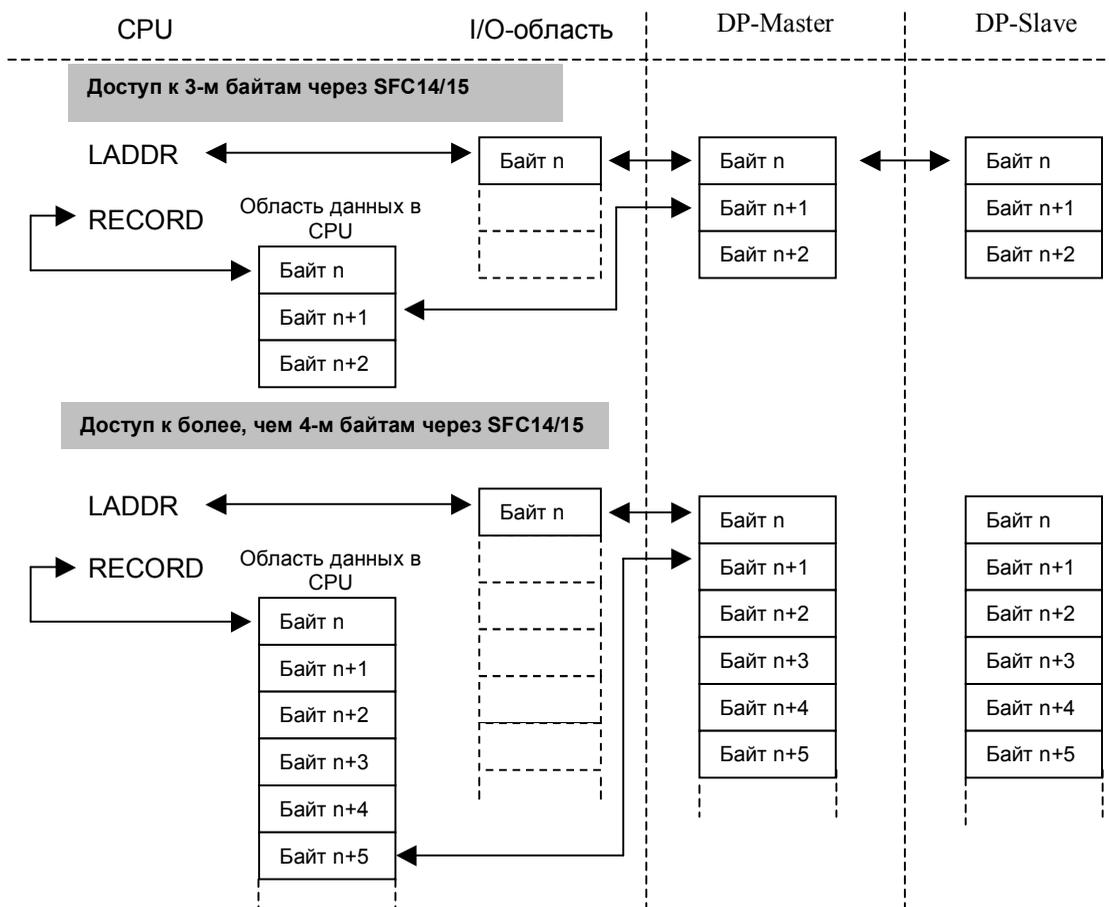


Рис. 6.3 Обмен входными/выходными данными через SFC14 и SFC15

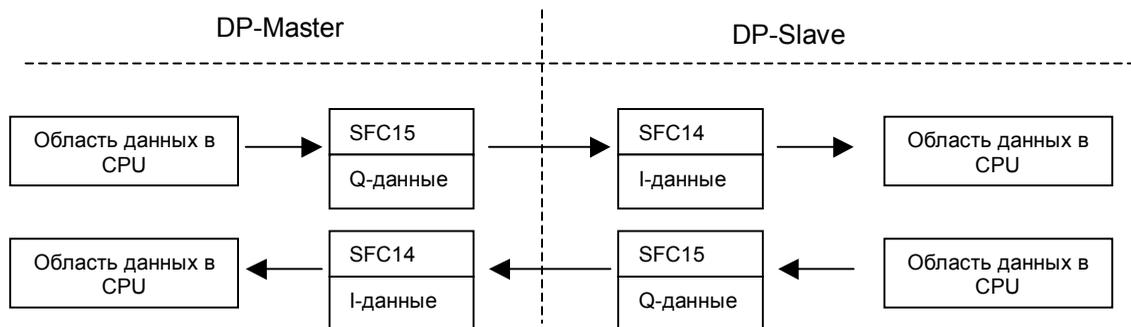
Параметр SFC LADDR служит здесь как указатель на область входных данных для чтения или область выходных данных записи. В этом параметре вызова SFC задается начальный адрес области входных или выходных данных в 16-ичном формате, спроектированный в HW-Config.

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 содержаться 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.



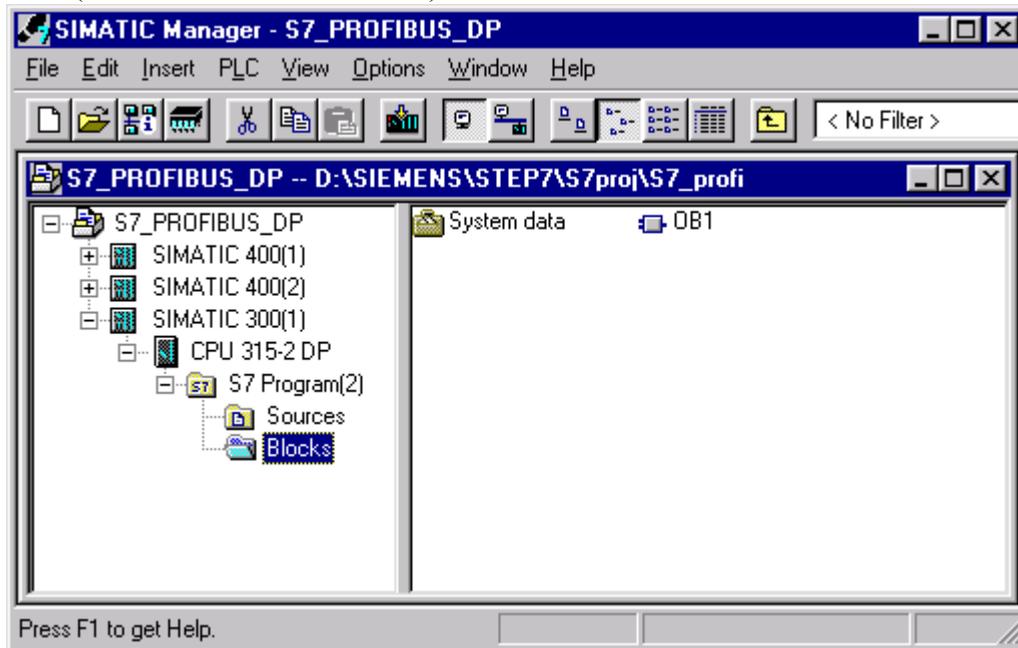
**Рис. 6.4** Обмен входными/выходными данными с I-Slave в проекте-примере через SFC14 и SFC15

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.

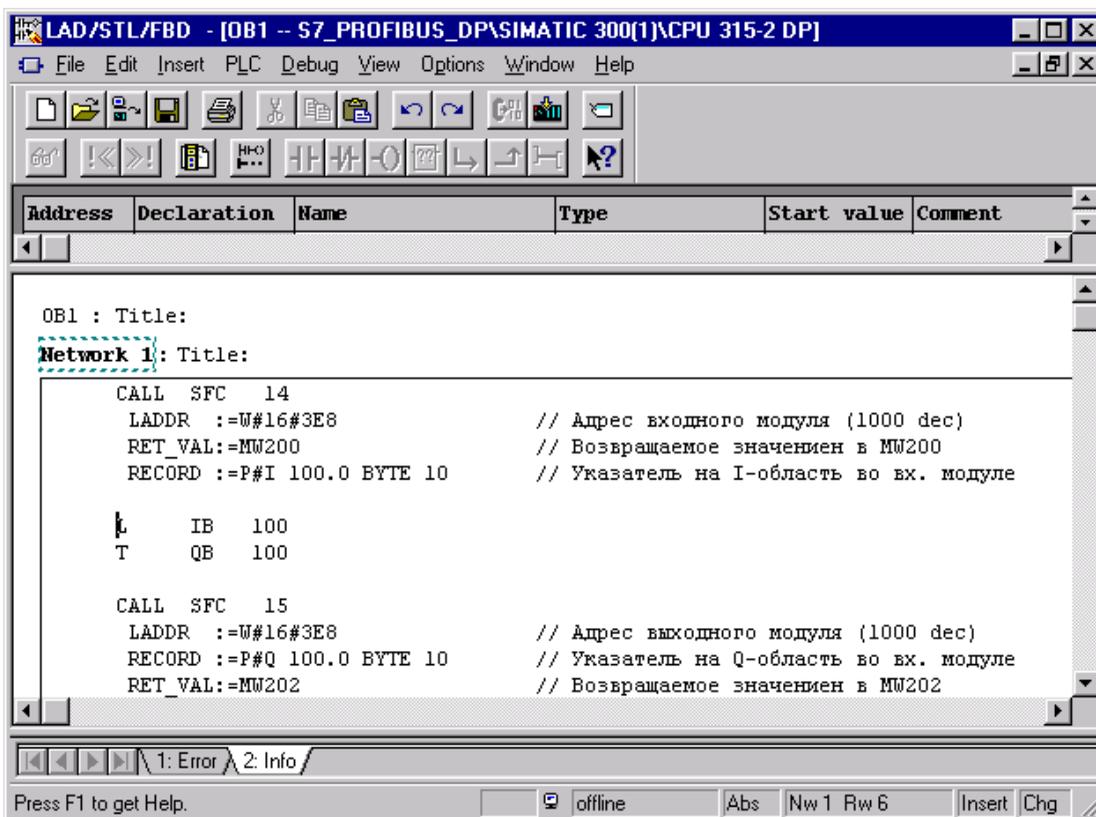
Для создания необходимой пользовательской программы Вы должны действовать, как указано ниже:

- 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 остановки 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 ОБ сигнала от процесса при выходе процесса за заданные границы.

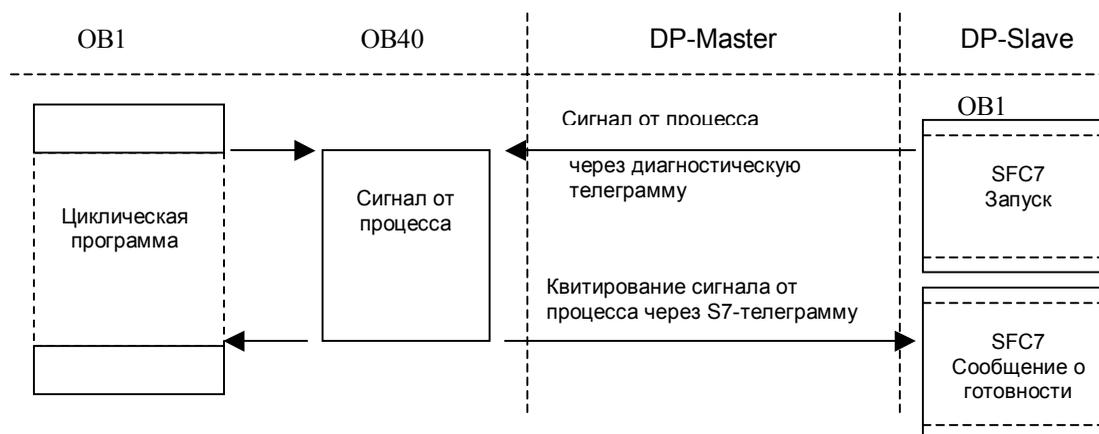
Следующий пример описывает, как станция S7, которая используется, как I-Slave на шине PROFIBUS-DP, порождает сигнал от процесса. После этого описывается, как сигнал от процесса распознается и обрабатывается в DP-Master'е (S7-400).

### 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 процесса 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, возбуждивший 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"
BES
=   M   100.0      // Вызов нового сигнала от процесса

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'e

Для тестирования обработки сигнала от процесса в DP-Master'e 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'e с помощью функции *Debug->Monitor*.

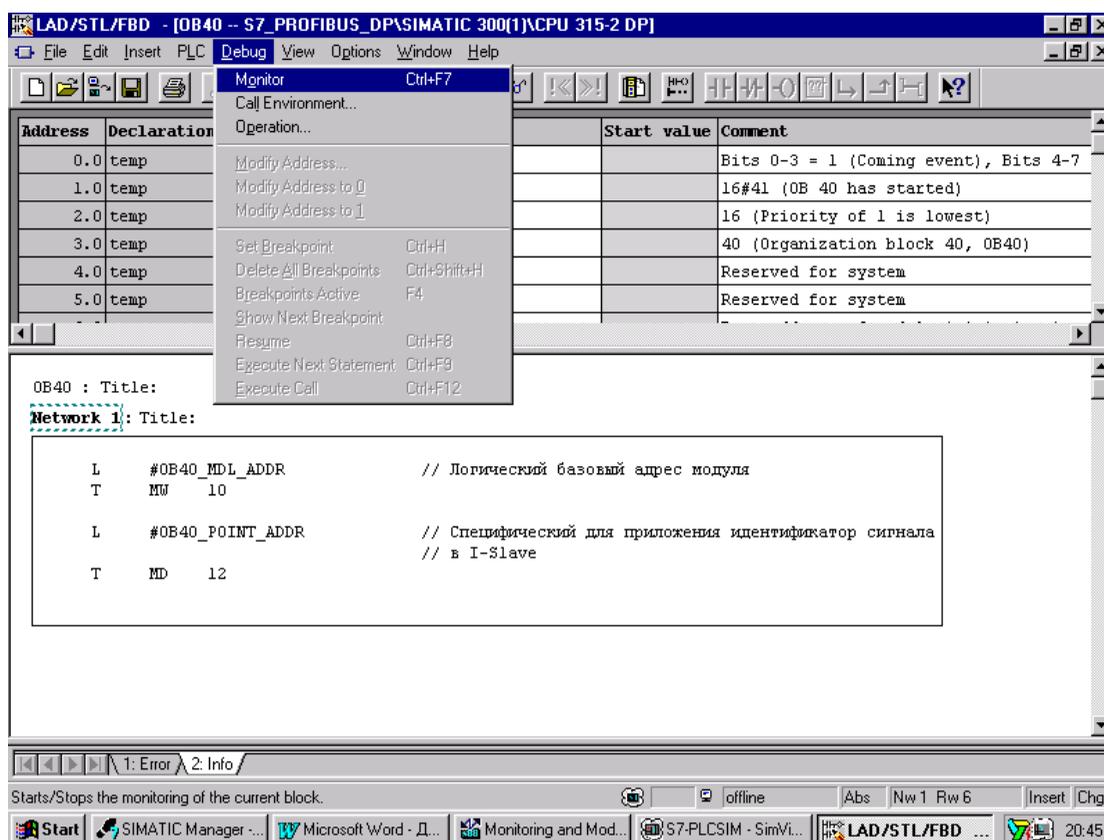


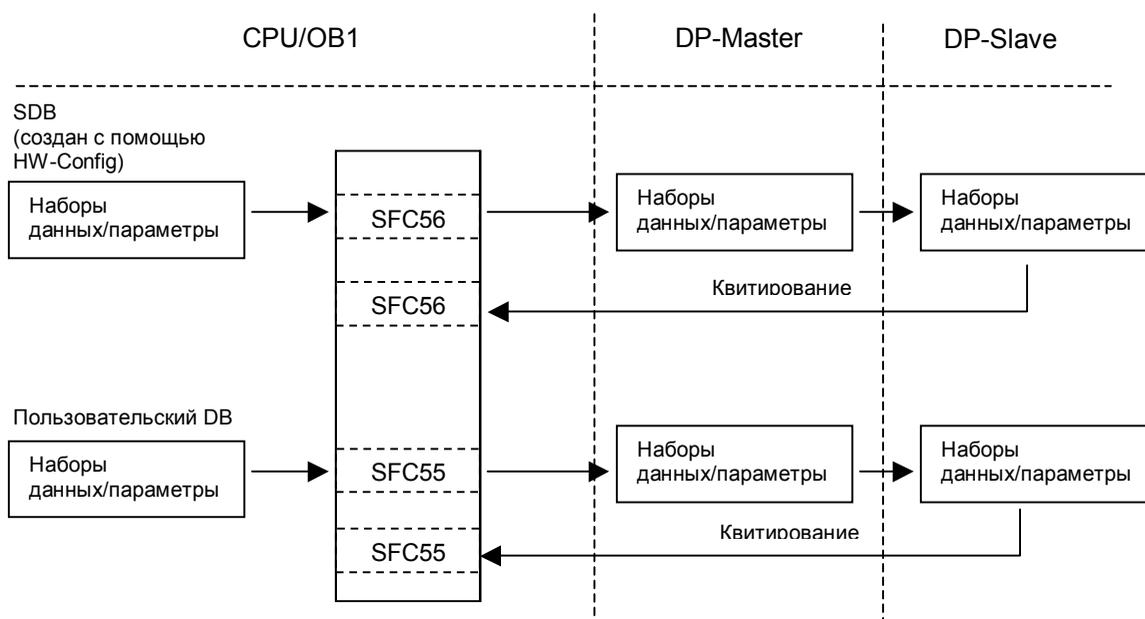
Рис.6.13 Функция состояния для OB40 в программе-примере

## 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 перепараметрировать спроектированный в разделе 4.2.5 аналоговый модуль в станции ET200M: первоначально заданный диапазон измерений +/-10V с помощью SFC55 изменить на +/-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.

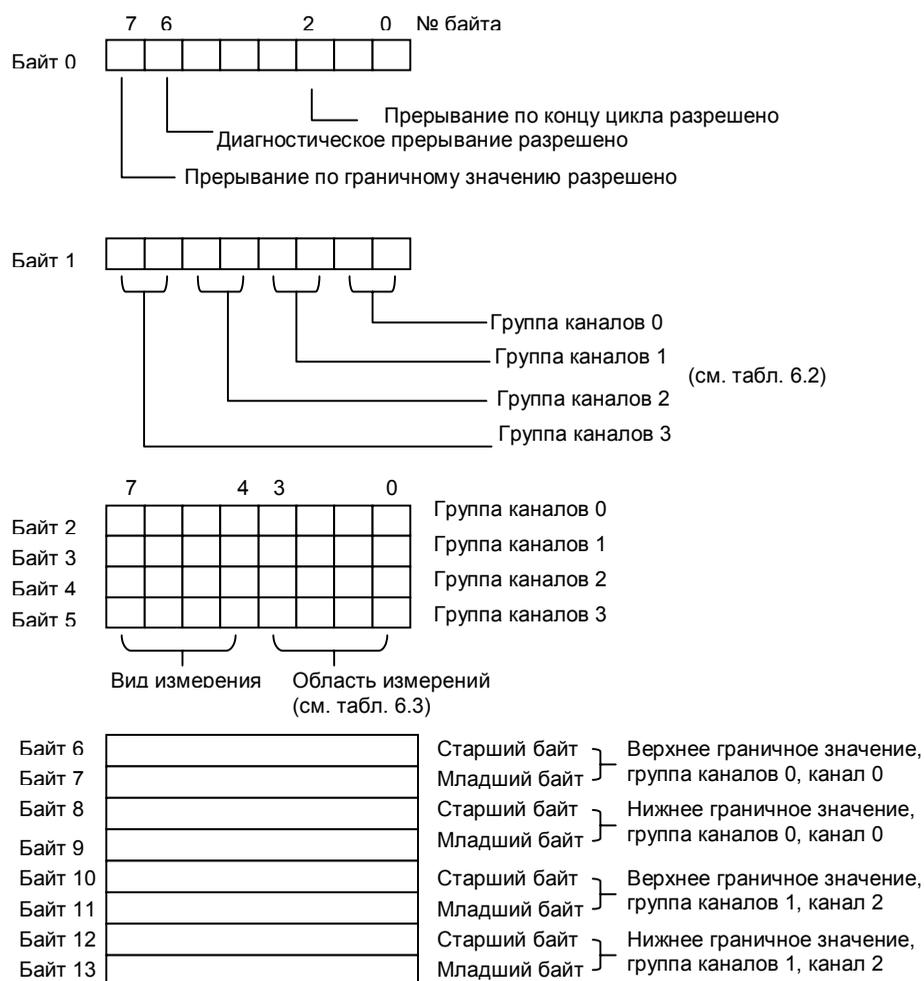
#### 6.4.1 Структура набора данных (DS1) для аналоговых входных модулей SIMATIC S7-300

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	± 80 mV	2#0001
		± 250 mV	2#0010
		± 500 mV	2#0011
		± 1V	2#0100
		± 2,5 V	2#0101
		± 5 V	2#0110
		от 1 до 5 V	2#0111
		± 10 V	2#1001
		± 25 V	2#1010
		± 50 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 V#16#01. Активизируйте показ значений. Значение для MB30 пока V#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     := V#16#54        // Идент. периф. модуля входов  
  LADDR    := W#16#200       // Адрес входного модуля (512 dec)  
  RECNUM   := V#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.

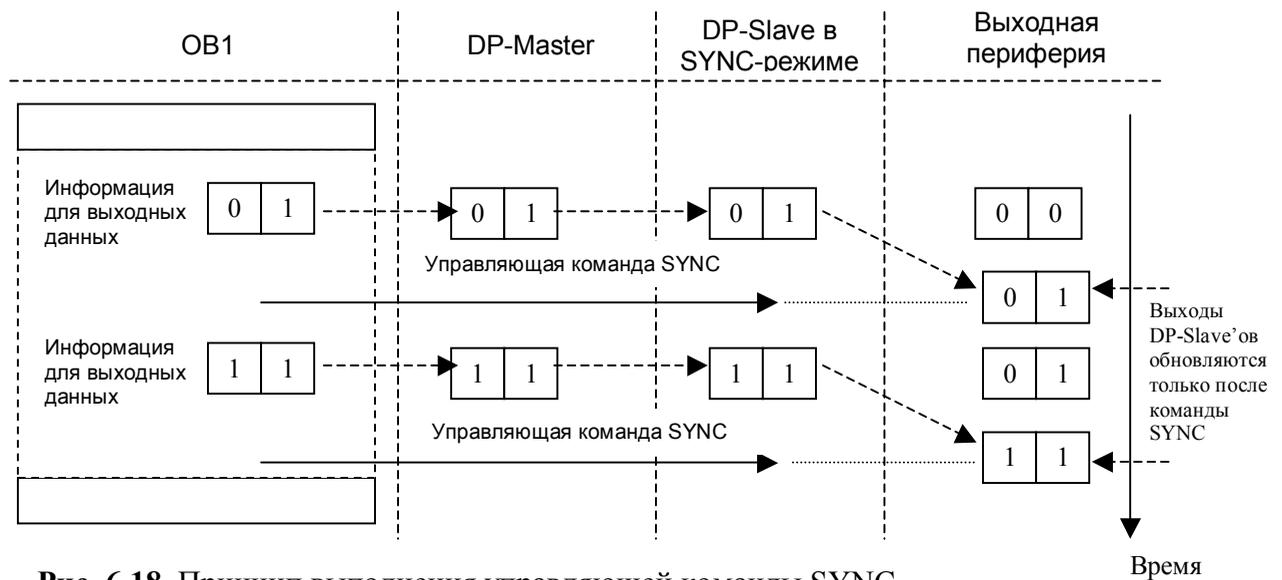
Задайте для этого в таблице переменных обе переменные MB40 (M40.0 = REQ и M40.1 = BUSY) и MW42 (RET\_VAL). Для MB40 задайте значение B#16#01. Активизируйте наблюдение переменных. Значение для MB40 должно пока быть равно B#16#00, а значение для MW42 (RET\_VAL) – B#16#7000. Активизируйте теперь заданные 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 входов, 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.

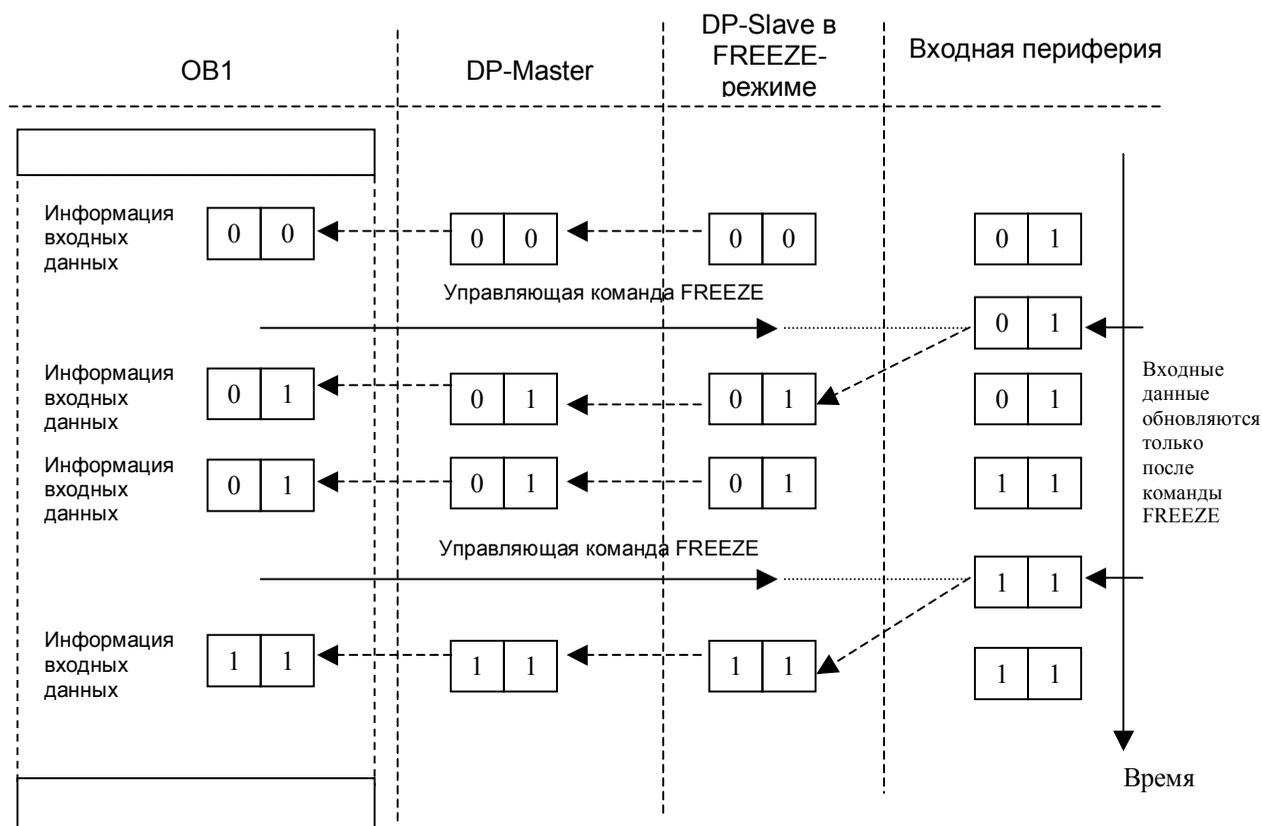


Рис. 6.19 Принцип выполнения управляющей команды FREEZE

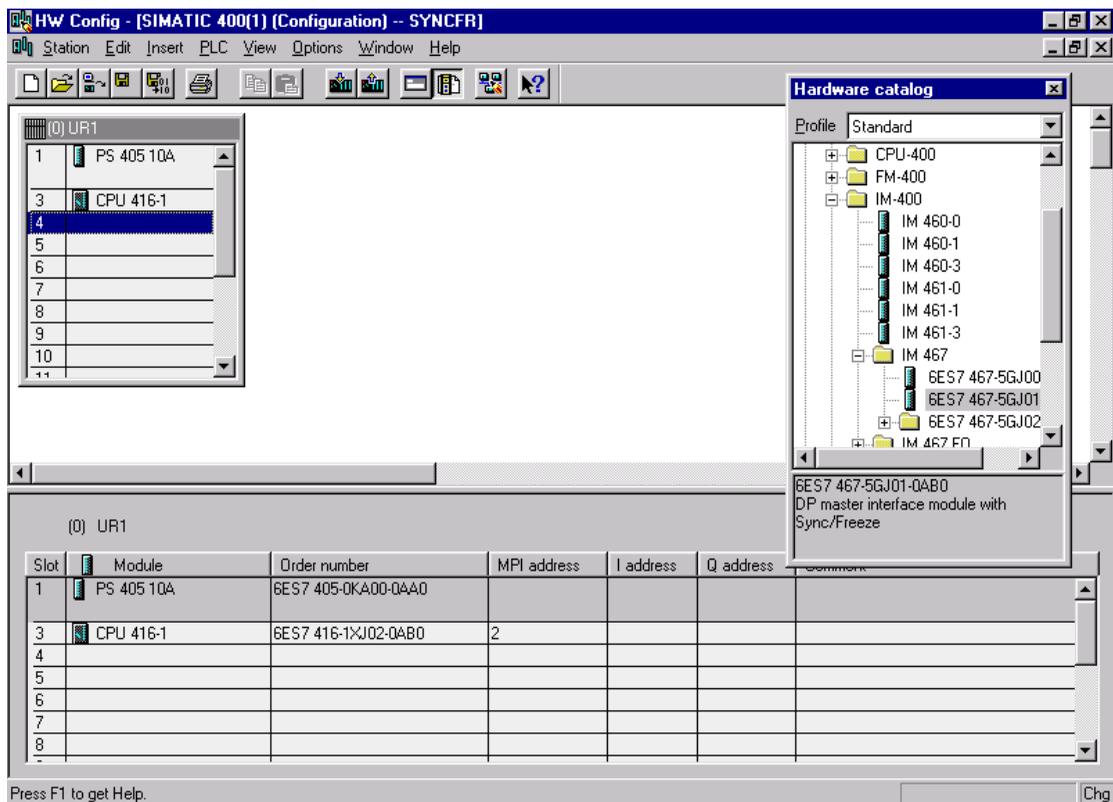
### 6.5.1 Пример применения команд SYNC/FREEZE с DP-Master'ом IM467

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 ОК. 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'a (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).



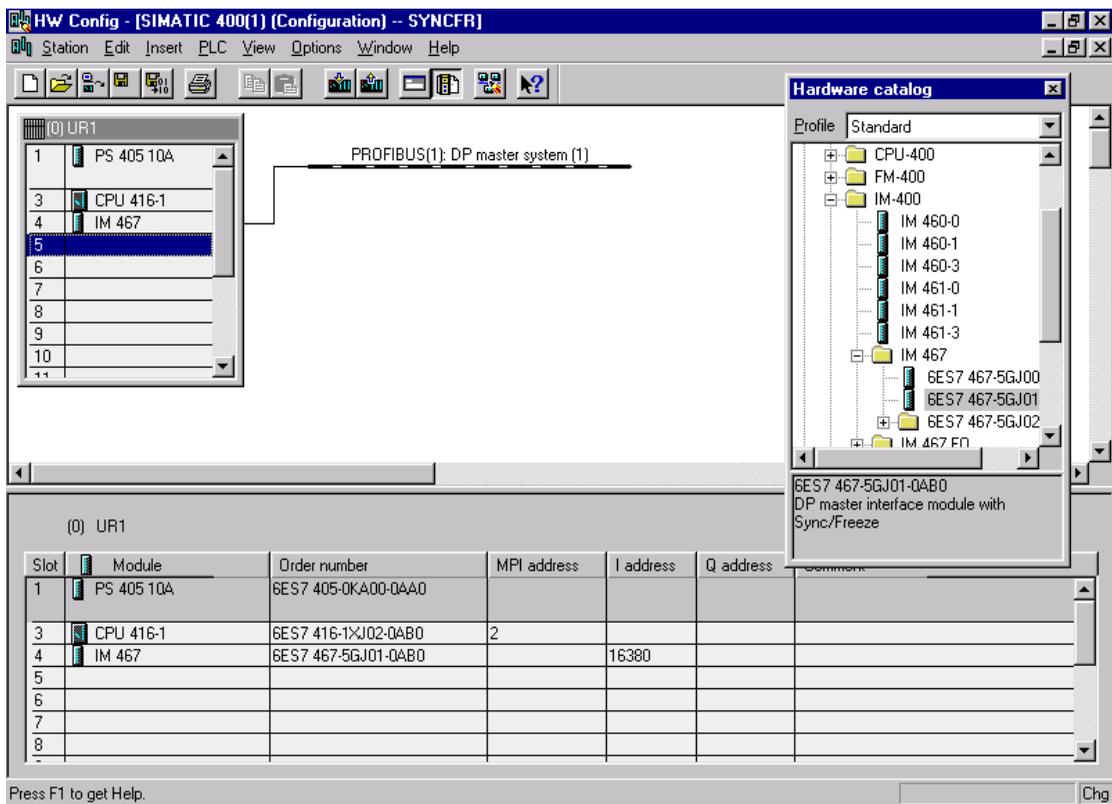
**Рис. 6.20** Выбор IM467 из Hardware catalog

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.

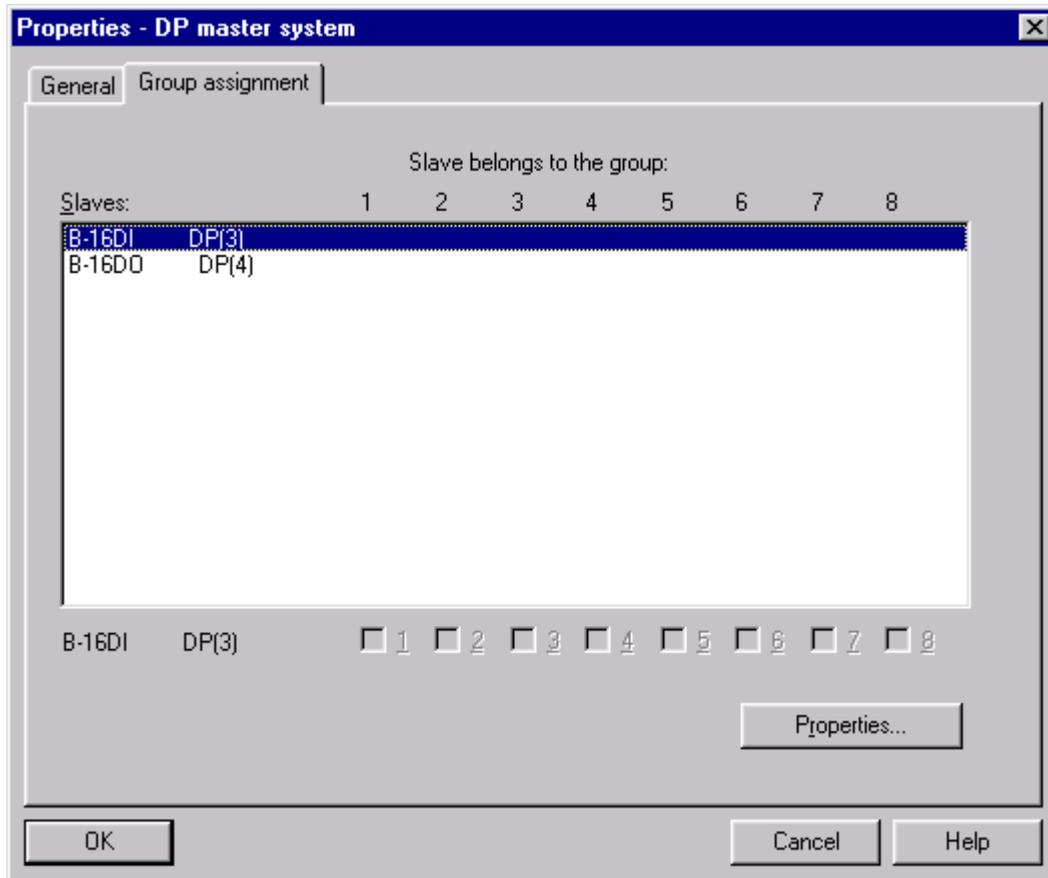


Рис. 6.22 Назначение групп в *HW Config*

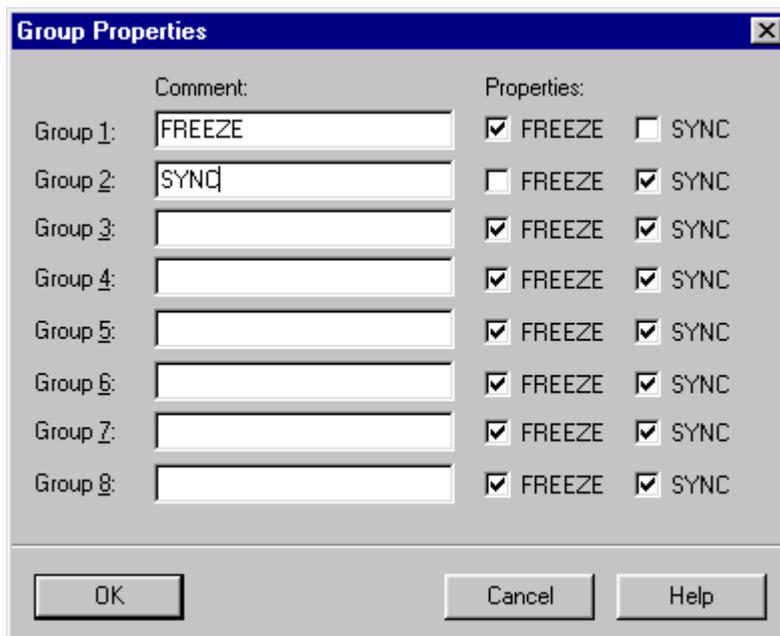
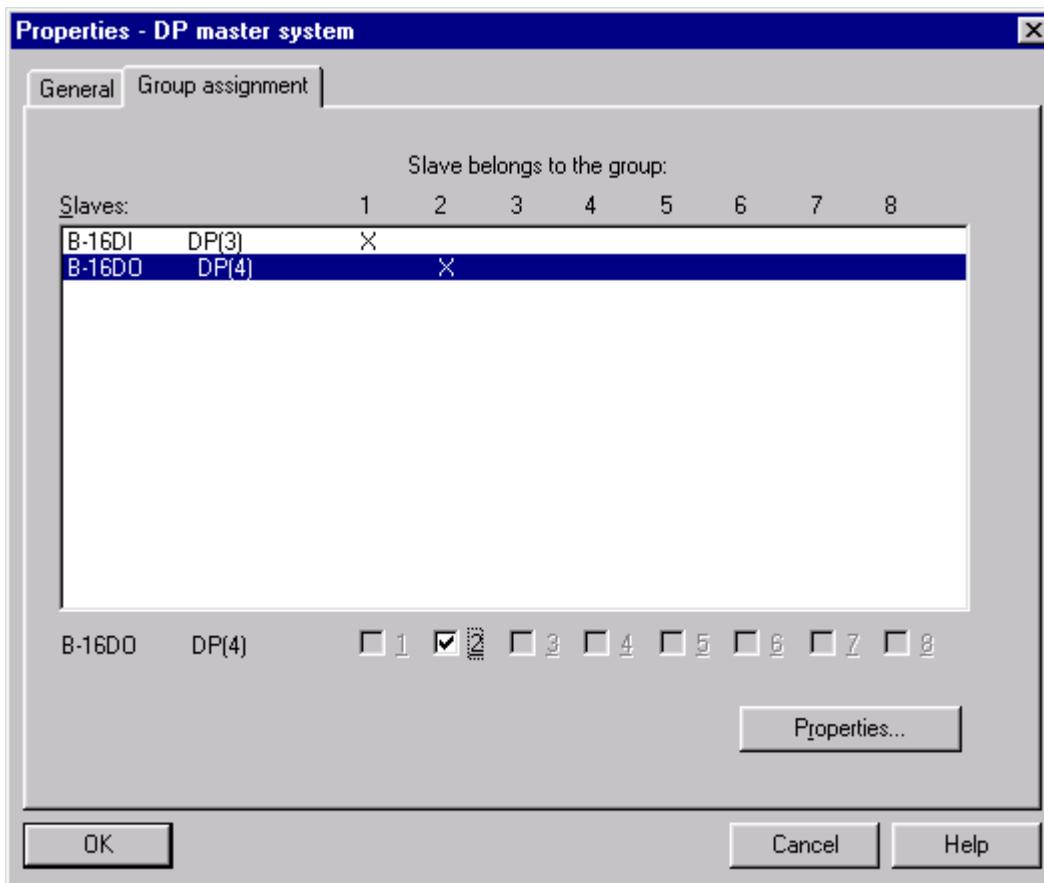


Рис.6.23 Свойства групп в *HW Config*



**Рис.6.24** Проектирование групп для модулей ET 200B

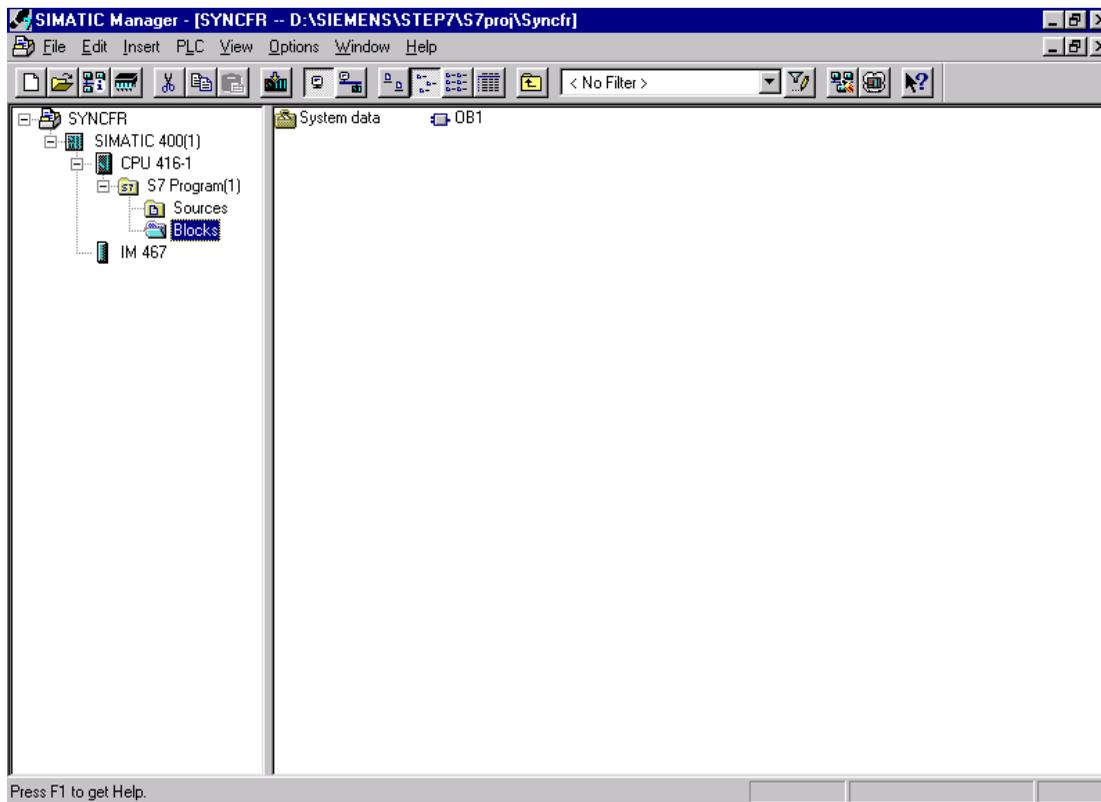
Выберите теперь *STATION*->*Save and Compile*. Переведите проектируемую станцию в STOP и загрузите аппаратную конфигурацию в CPU S7-400.

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.

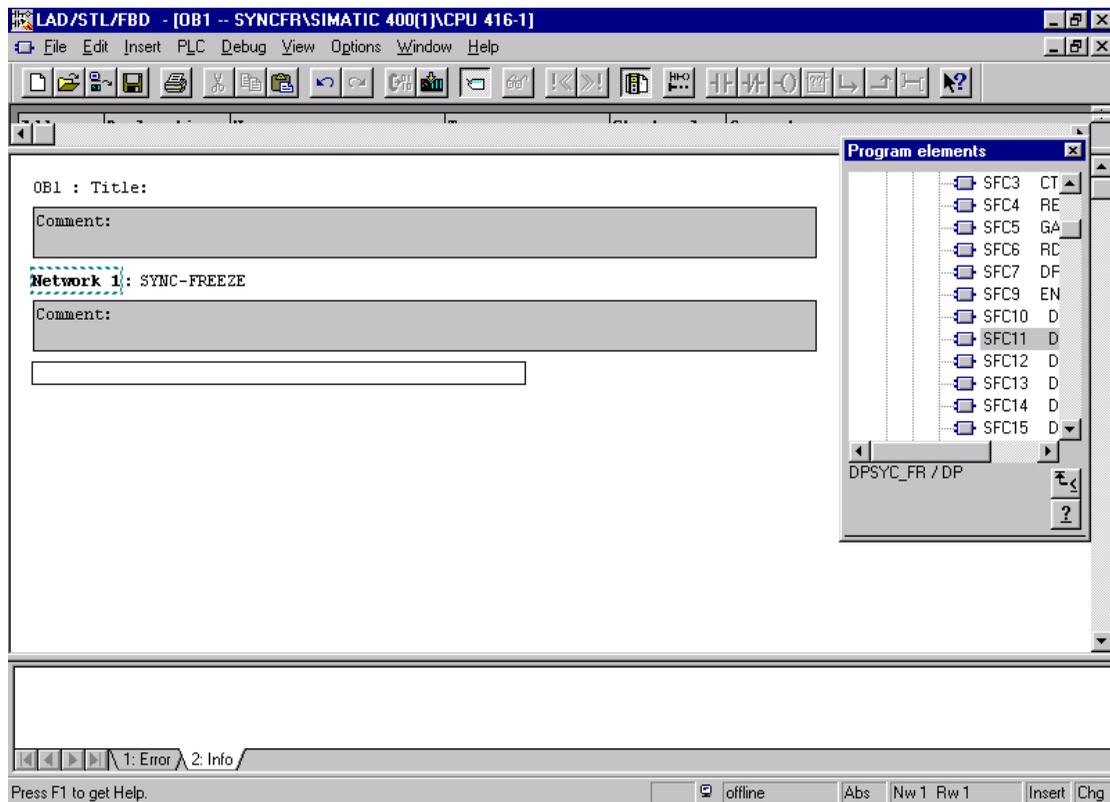


Рис.6.26 LAD/STL/FBD-редактор с каталогом программных элементов

```

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)

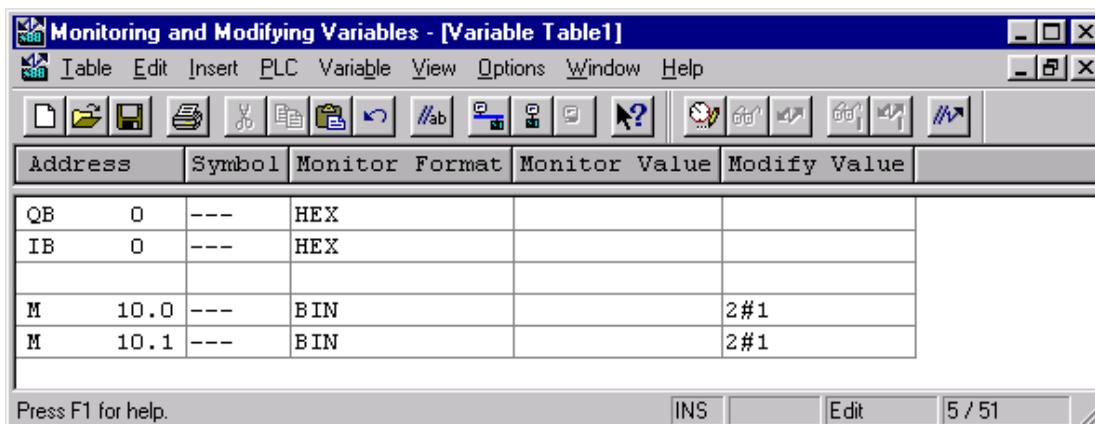
GO2: CALL "DPSYC_FR"
    REQ :=M10.6 //Маркер запуска
    LADDR :=W#16#200 //Входной адрес IM467 (512dez)
    GROUP :=B#16#2 //Выбрана группа 2
    MODE :=B#16#20 //Выбран режим SYNC
    RET_VAL:=MW14 //RET_VAL в MW14
    BUSY :=M10.7 //Флаг BUSY в M10.7

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.



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

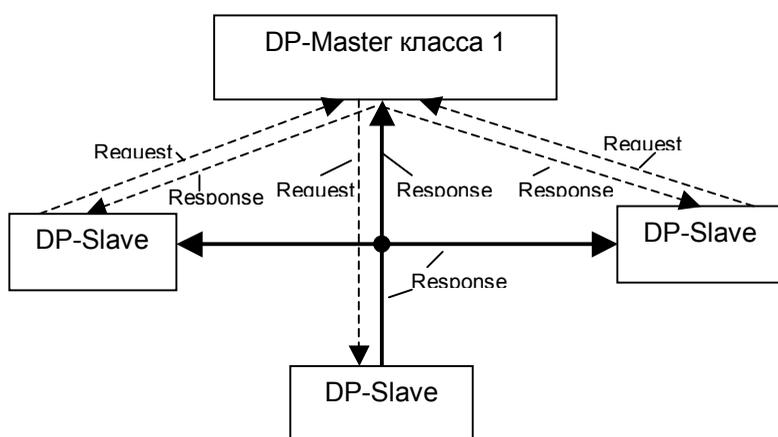
После запуска шинной системы DP все Slave’ы находятся в циклическом обмене данными. С установкой меркеров 10.0 и 10.1 в состоянии “1” запускаются управляющие команды SYNC и FREEZE.

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).

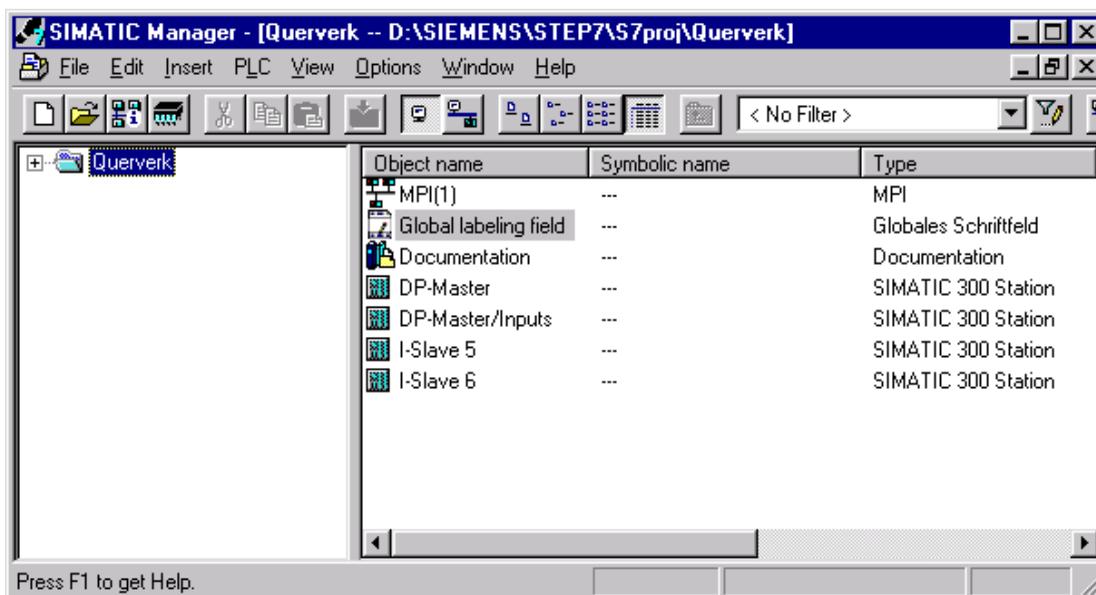


**Рис. 6.29** Response-телеграмма DP-Slave при перекрестной связи

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).

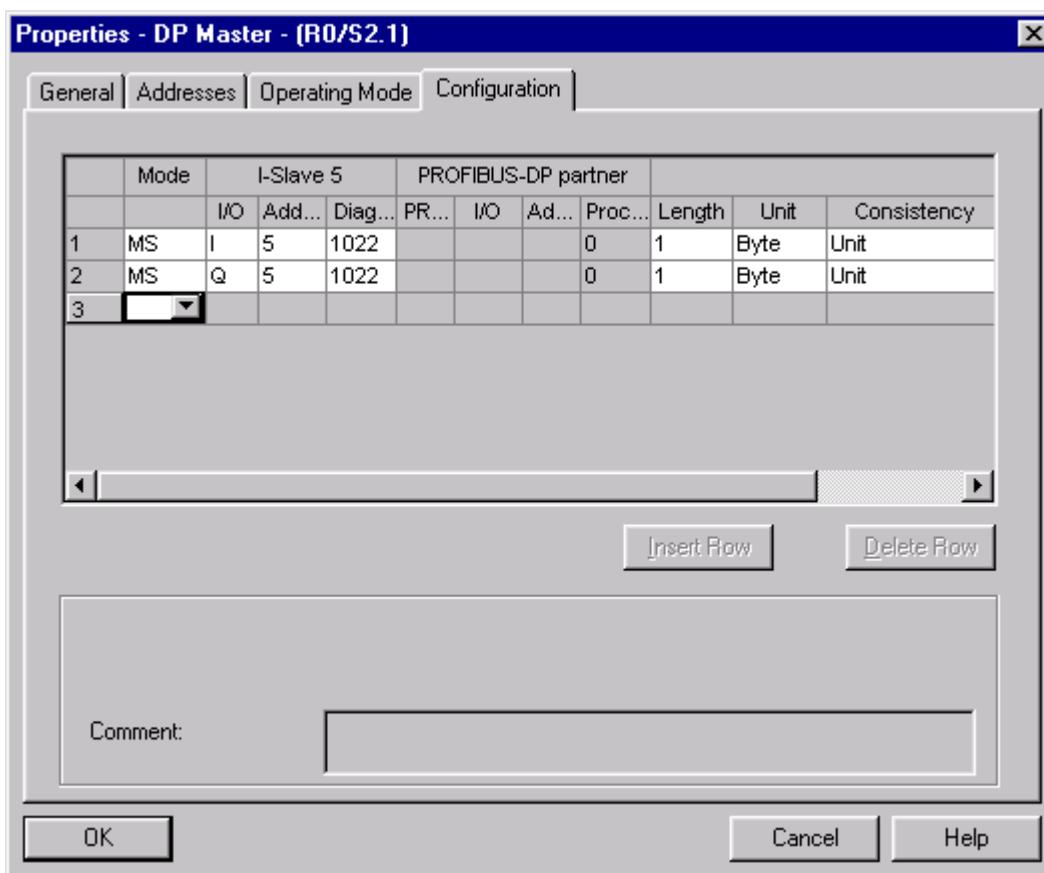


**Рис. 6.30** Проект "Querverk" с принадлежащими ему станциями SIMATIC 300

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.

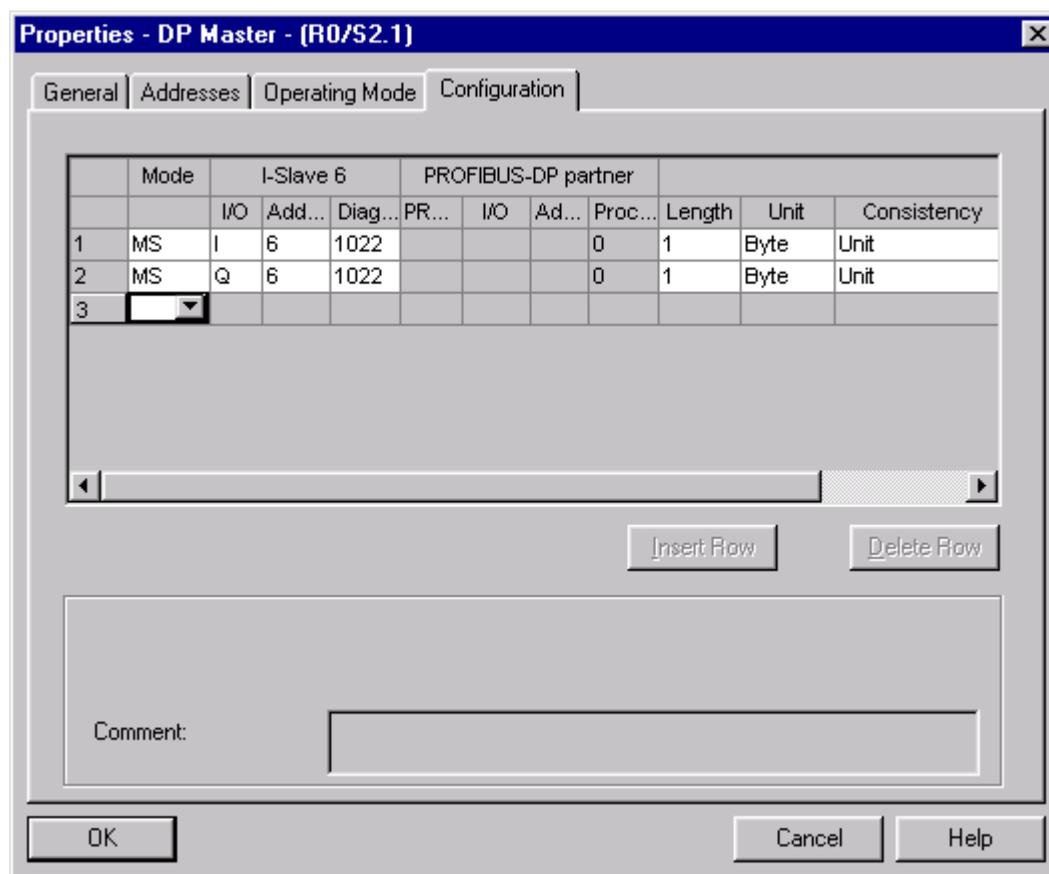
Вставьте теперь из Hardware Catalog SIMATIC 300 Rack-300. На установочном месте 1 разместите источник питания “PS 307-2A”. При выборе CPU необходимо обратить внимание, чтобы он поддерживал перекрестную связь. Поэтому выберите CPU 315-2DP с заказным номером 6ES7315-2AF03-0AB0 и разместите его на установочном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

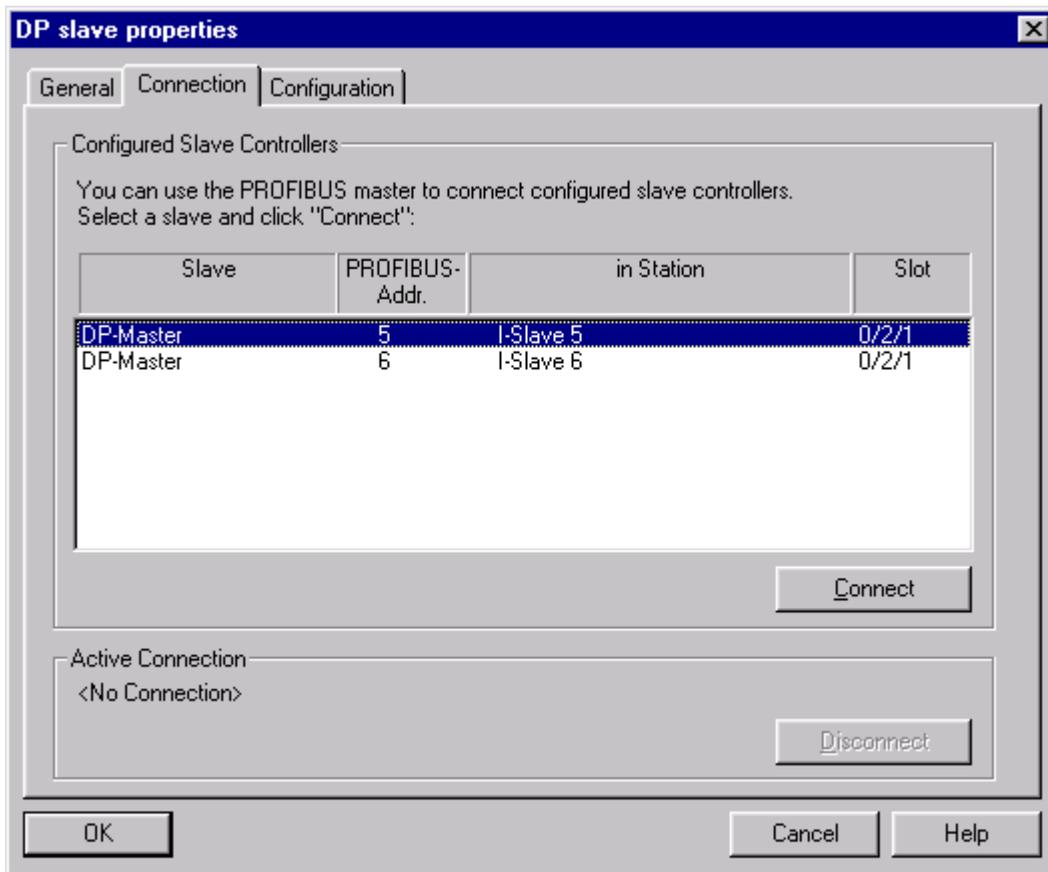


Рис. 6.33 Присоединение I-Slave 5 к подсети PROFIBUS

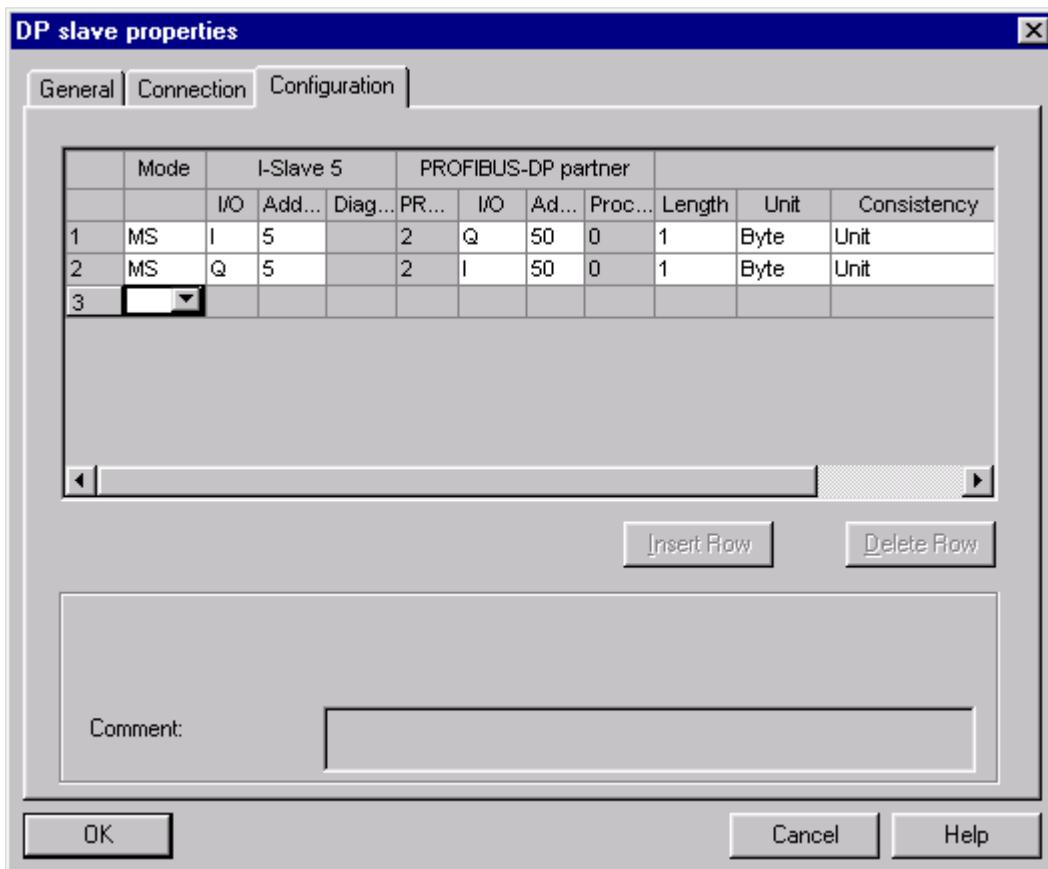
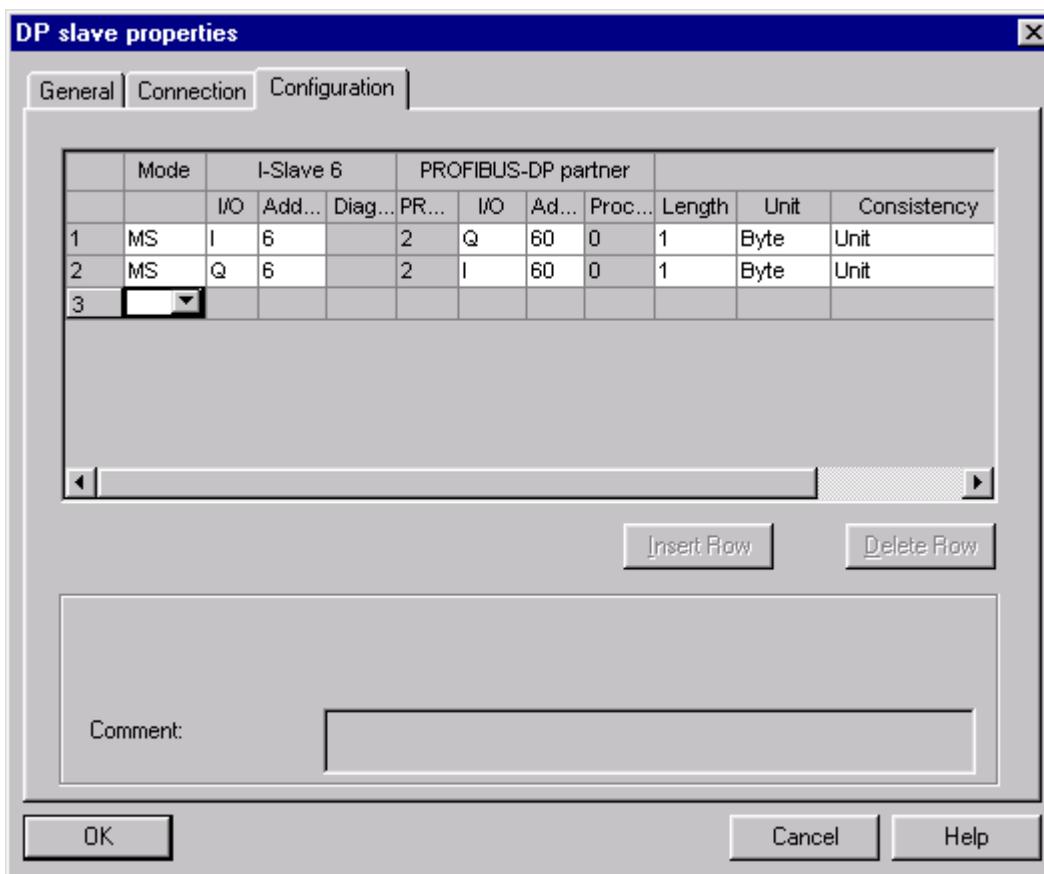


Рис. 6.34 I/O-конфигурация I-Slave 5

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 ОК.

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.

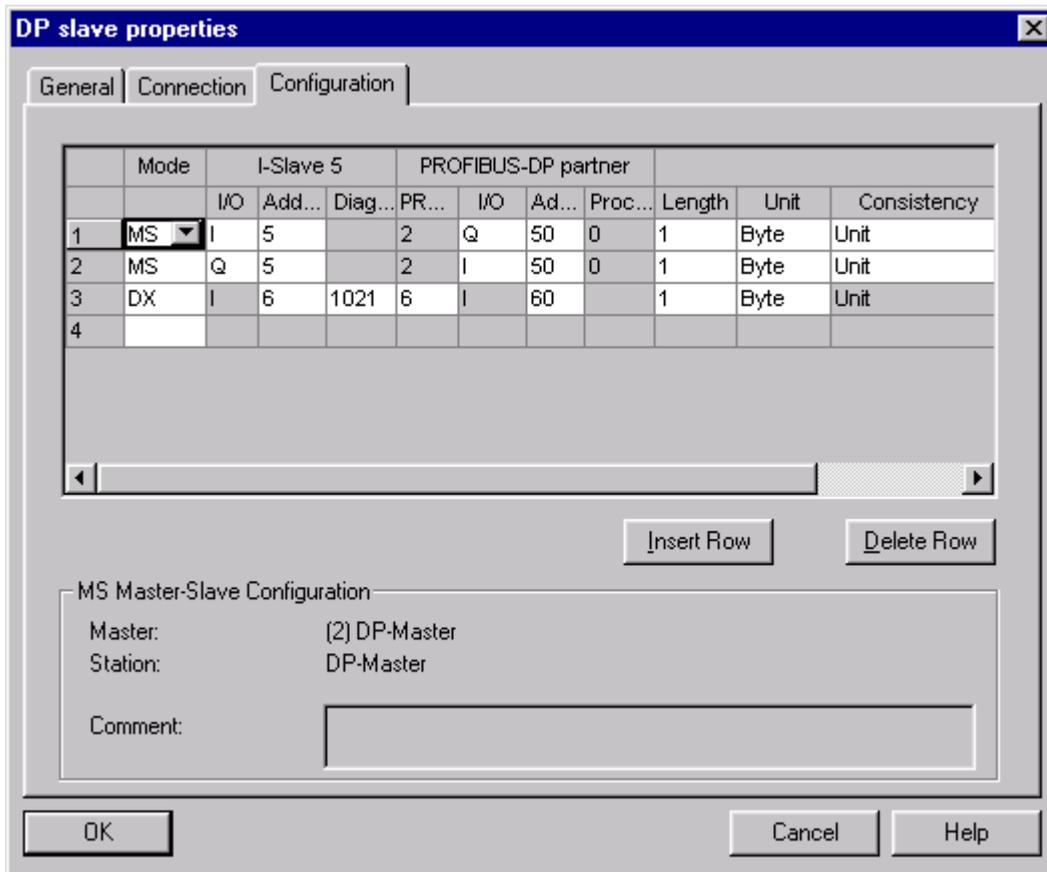


Рис. 6.36 Перекрестная связь I-Slave 5 с I-Slave 6

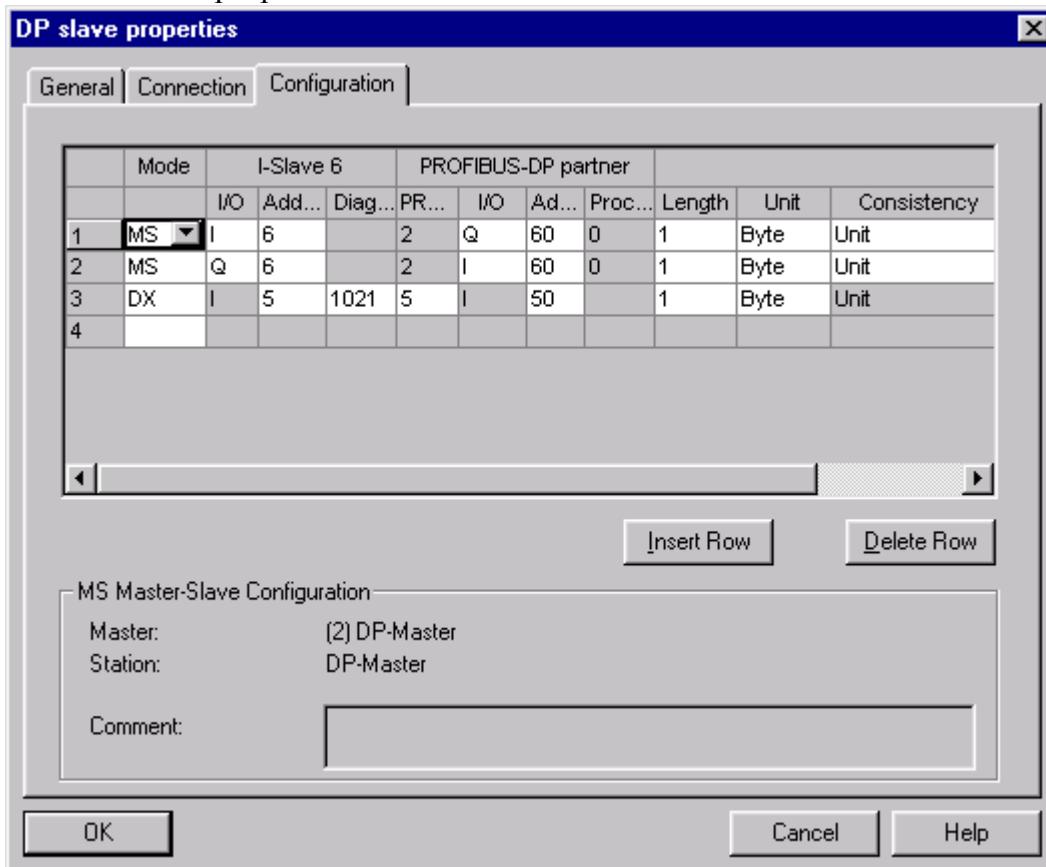
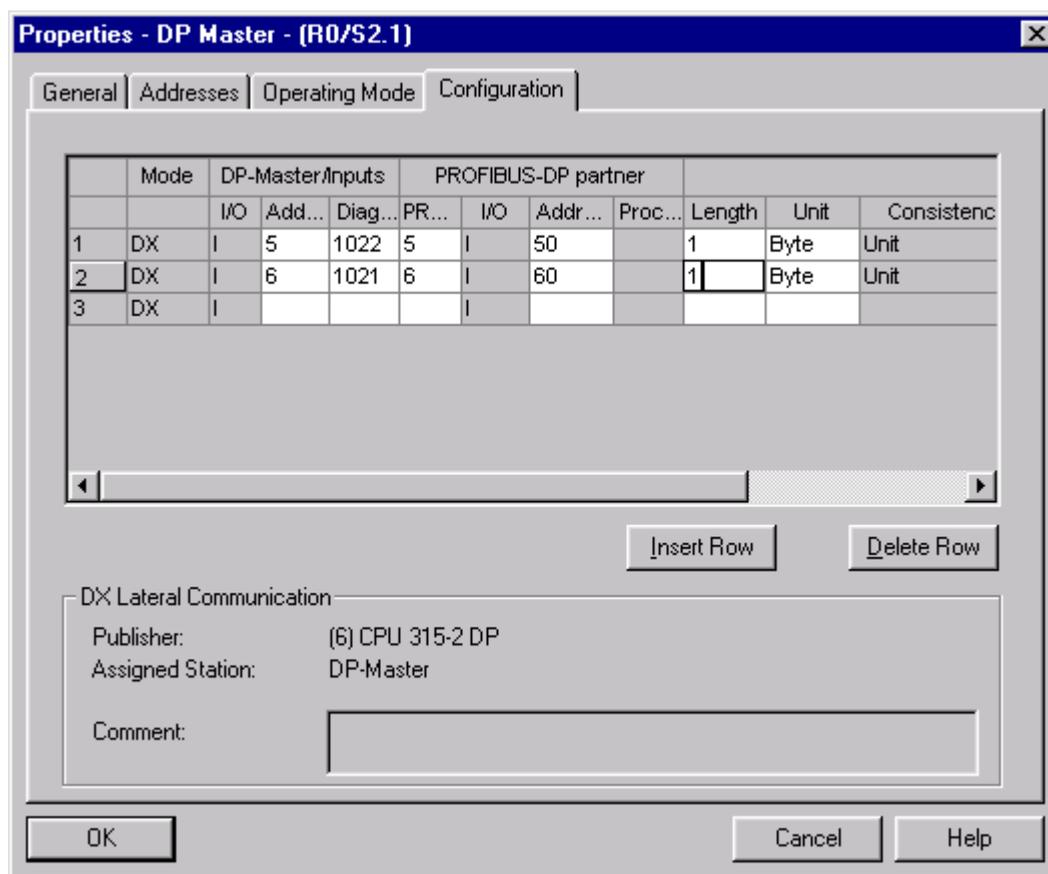


Рис. 6.37 Перекрестная связь I-Slave 6 с I-Slave 5

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).

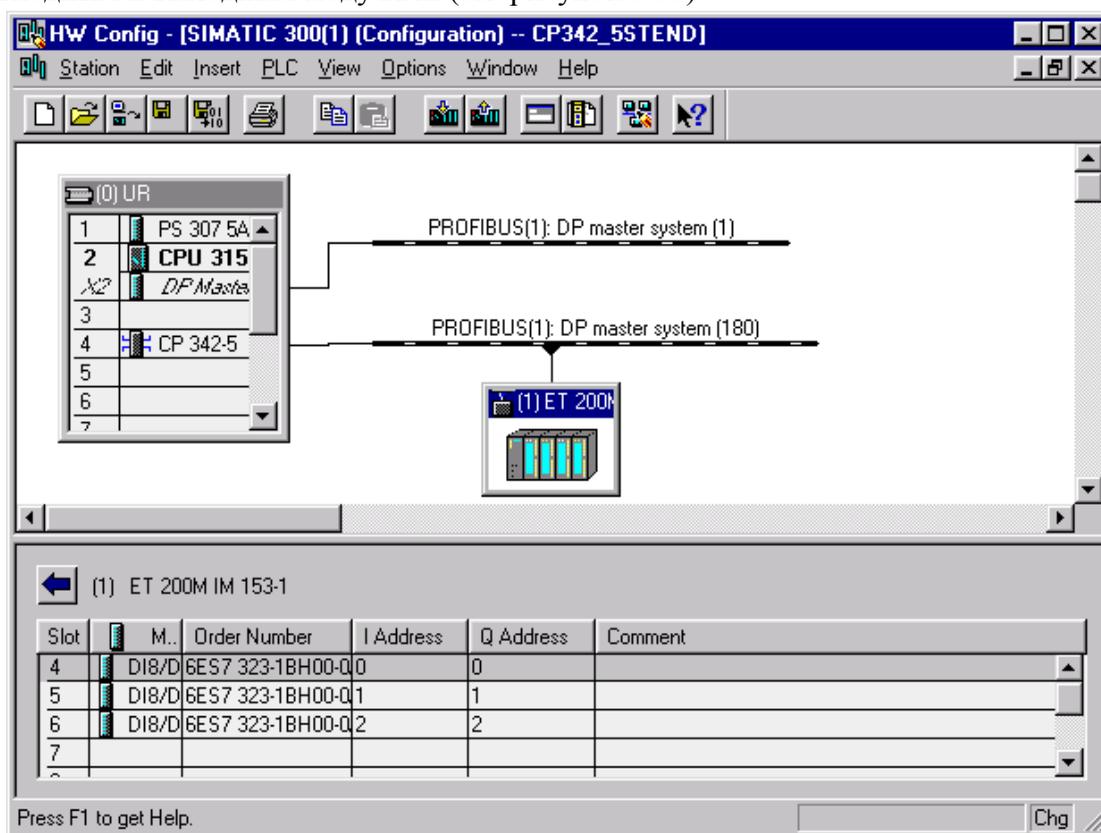
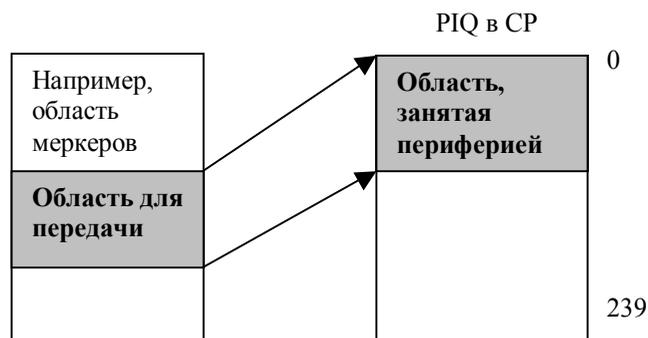


Рис. 6.39 Утилита HW-Config для PLC с CP342-5 - Master



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 PI CP 342-5 произвольно, то есть не обязательно читать всю область PI, 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.

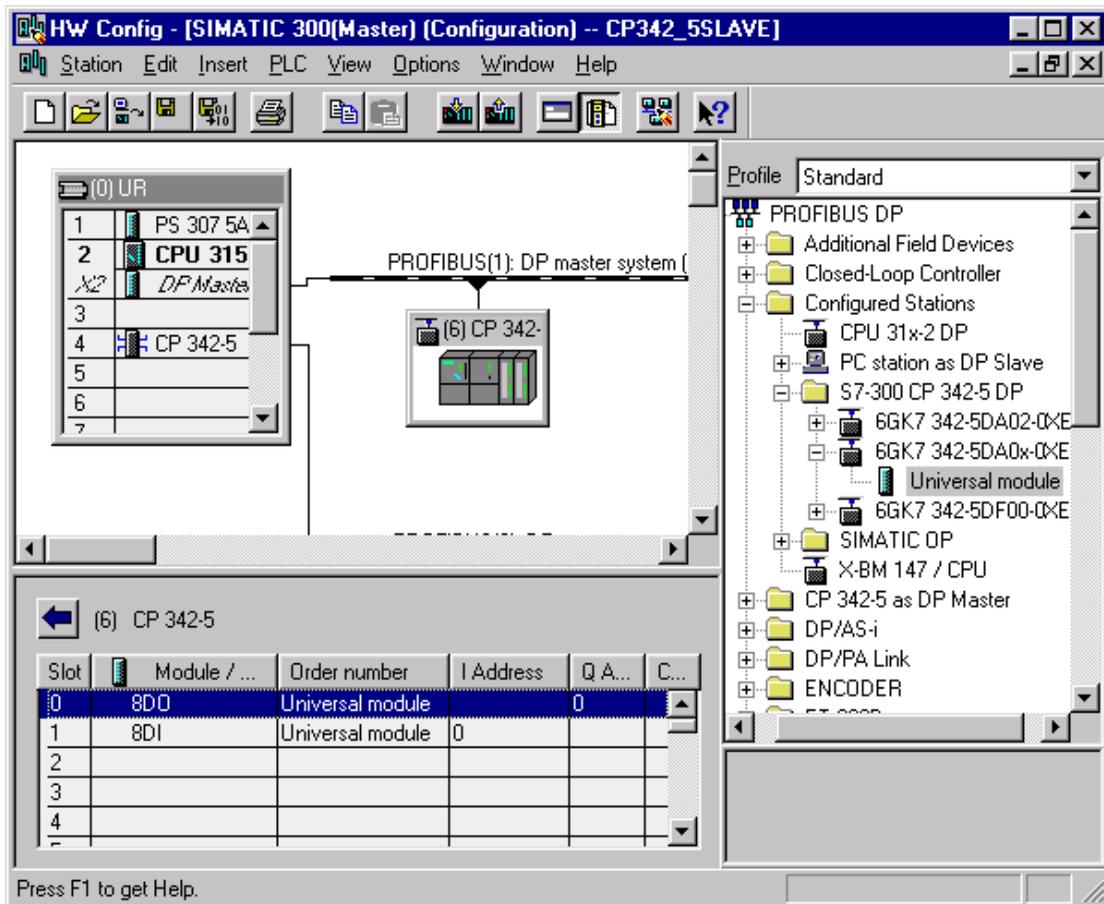


Рис. 6.42 CP342-5 как DP-Slave

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 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 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).

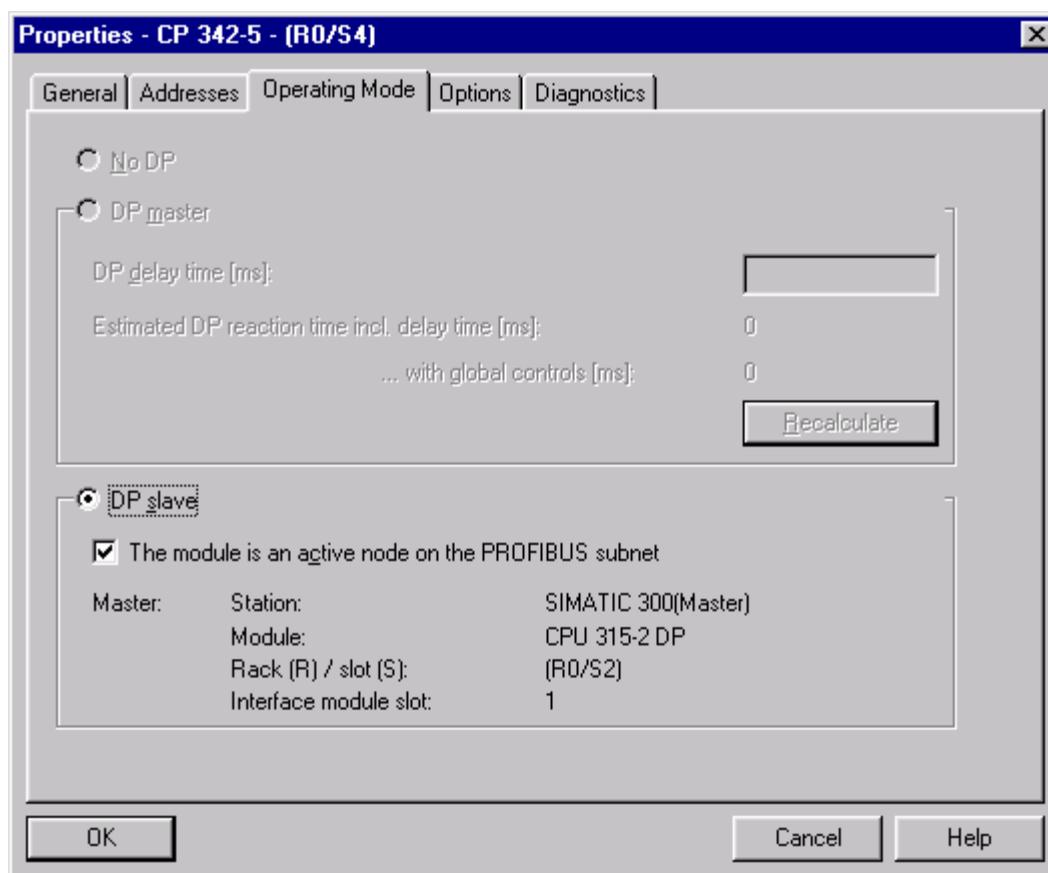


Рис. 6.44 CP342-5: активный DP-Slave

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.

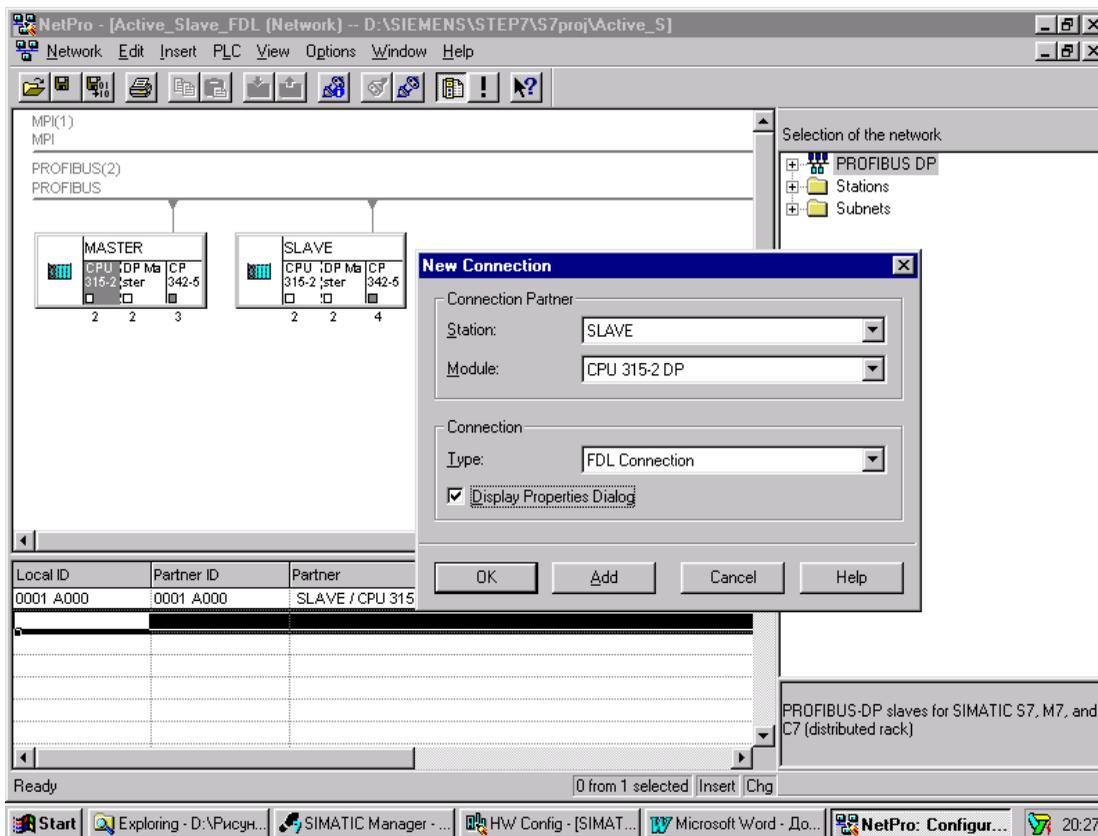


Рис. 6.46 FDL-соединение: определение соединения

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).

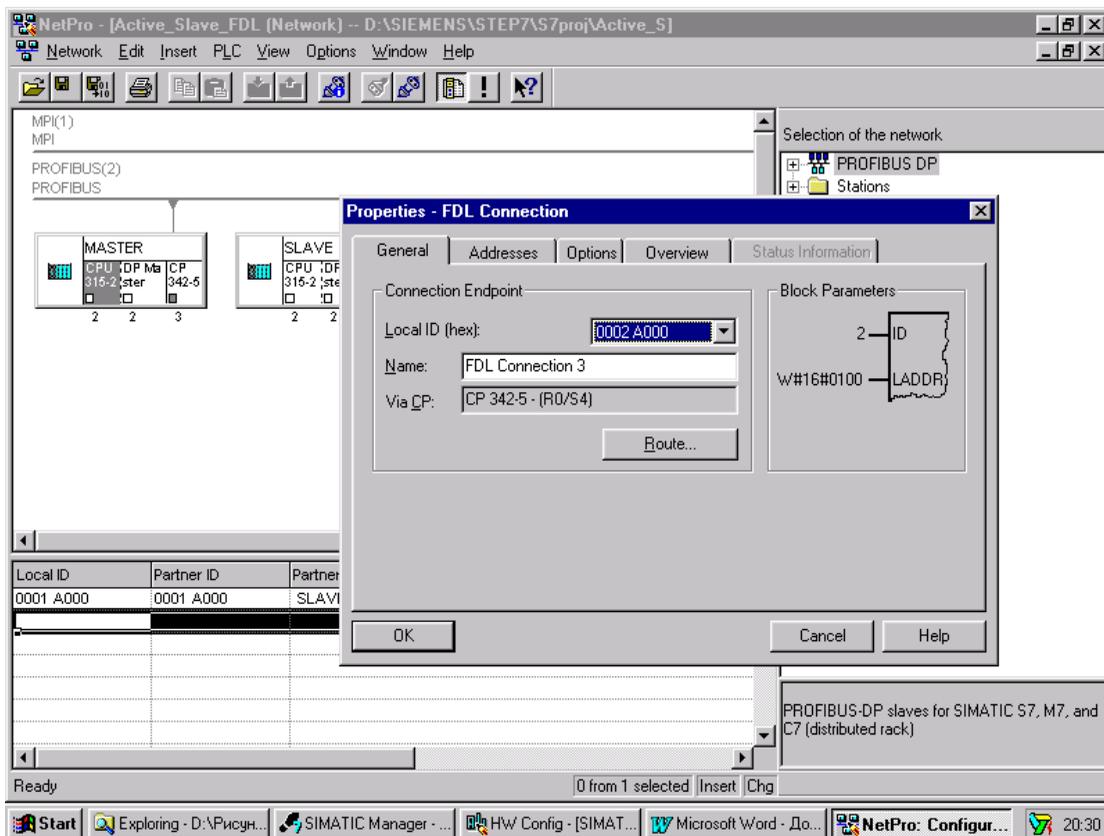


Рис. 6.47 FDL-соединение: параметры функций

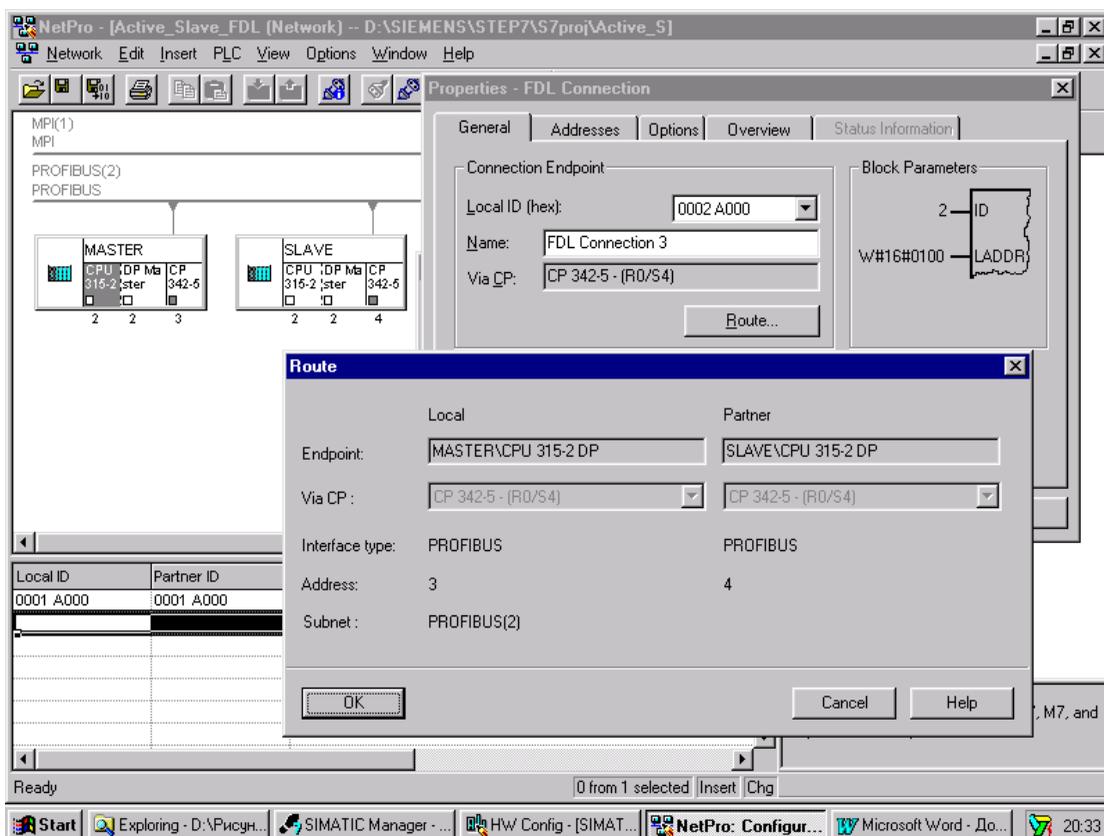
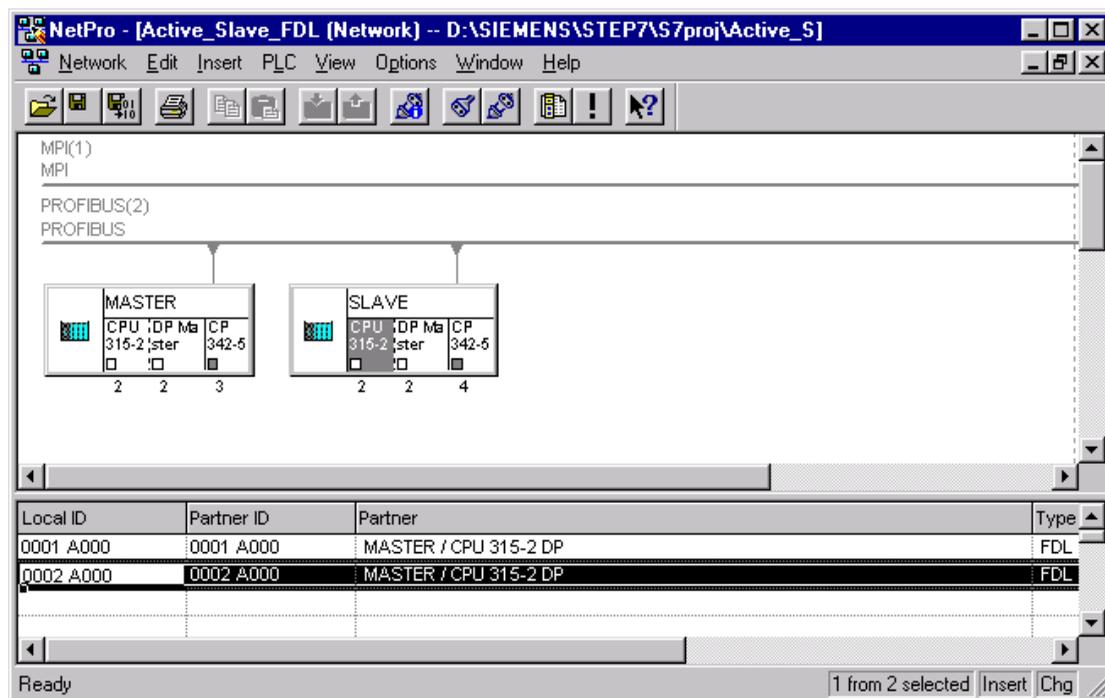


Рис. 6.48 FDL-соединение: маршрут

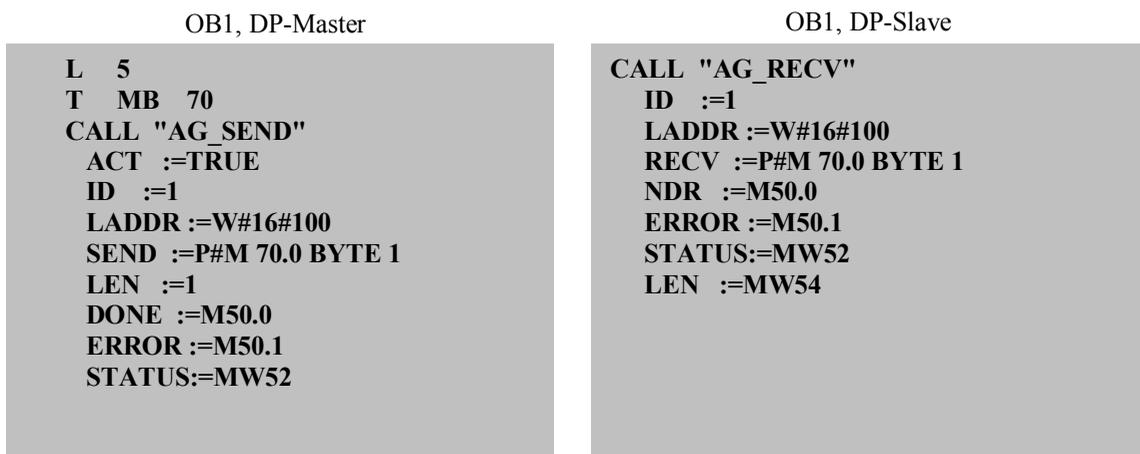
Закройте все окна с помощью “OK”. Окно в NetPro будет выглядеть, как показано на рисунке 6.49.



**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.



**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.