

Таблица 1–1. Содержание документации по S7

Название	Содержание
Букварь S7–300 Простой монтаж и программирование	Букварь предлагает очень простое введение в методику сборки и программирования S7–300/400. Он особенно пригоден для новых пользователей контроллеров S7.
Руководство по программированию Разработка программ для S7–300/400	Руководство по программированию "Разработка программ для S7–300/400" знакомит с основными знаниями о структуре операционной системы и прикладной программы CPU S7. Оно может использоваться новым пользователем S7–300/400 для получения обзора по методике программирования и создания затем проекта своей прикладной программы.
Справочное руководство Системные и стандартные функции S7-300/400	CPU S7 содержат в операционной системе встроенные системные функции и организационные блоки, которые можно использовать при программировании. Руководство дает обзор применяемых в S7 системных функций, организационных блоков и загружаемых стандартных функций, а также - как справочную информацию - подробное описание интерфейсов для их использования в прикладной программе.
Руководство пользователя STEP 7	Руководство пользователя "STEP 7" объясняет принципы использования и функции предназначенного для автоматизации программного обеспечения STEP 7. Новичку в использовании STEP 7 и знатоку STEP 5 руководство даст обзор последовательности действий при конфигурировании, программировании и пуске в эксплуатацию S7–300/400. При работе с ПО можно целенаправленно обратиться к оперативной помощи в режиме online, которая обеспечивает детальную поддержку по вопросам использования ПО.
Руководство пользователя Конвертирование программ S5	Руководство пользователя "Конвертирование программ S5" необходимо, если Вы хотите конвертировать имеющиеся программы S5, чтобы затем исполнять их в CPU S7. Руководство дает обзор последовательности действий и использования конвертера; подробные указания по использованию функций конвертера можно получить в online-помощи. Через эту помощь Вы получите также описание интерфейсов доступных конвертированных функций S7.
Руководства по AWL, KOP, SCL¹	Руководства по языковым пакетам AWL, KOP и SCL содержат как указания пользователю, так и описания языков. Для программирования S7–300/400 нужен только один из этих языков, но при необходимости можно смешивать языки внутри одного проекта. Для первичного использования языков рекомендуется с помощью данного руководства познакомиться с методикой разработки программ. При работе с ПО можно использовать оперативную помощь online, которая подробно ответит на все вопросы по использованию соответствующего редактора или компилятора.
Руководства по GRAPH¹, HiGraph¹, CFC¹	Языки GRAPH, HiGraph, CFC предоставляют дополнительные возможности для реализации систем управления исполнением, состоянием или графическими переключениями блоков. Эти руководства содержат как указания пользователю, так и описания языков. Для первоначального использования языка рекомендуется с помощью данного руководства познакомиться с методикой разработки программ. При работе с ПО Вы можете также использовать оперативную помощь в режиме online (за исключением HiGraph), которая подробно ответит Вам на все вопросы по использованию соответствующего редактора или компилятора.
¹ Дополнительные пакеты к системному программному обеспечению для S7–300/400	

Упорядочение
информационного

этой документации.

Для поддержки конфигурирования и программирования контроллеров М7 имеется обширная документация для различных категорий ландшафта М7 пользователей. Следующие пояснения должны облегчить Вам использование

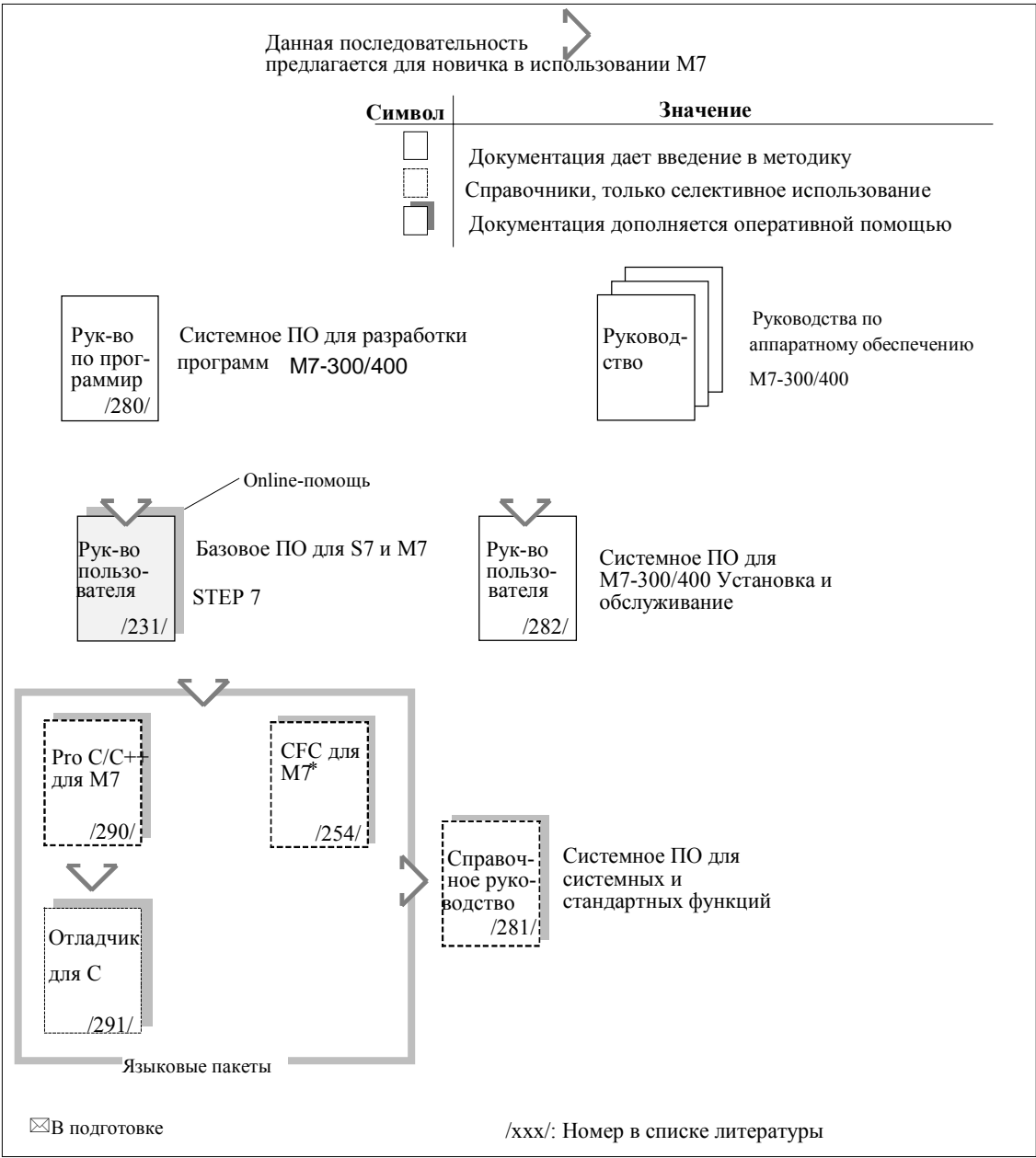


Рис. 1-2. Информационный ландшафт М7

Таблица 1–2. Содержание документации по М7	
Название	Содержание
Руководство по программированию Разработка программ М7–300/400	Руководство по программированию знакомит с основными знаниями о структуре операционной системы и прикладной программы М7-CPU/FM. Оно может использоваться новым пользователем М7–300/400 для получения обзора по методике программирования и создания затем проекта своей прикладной программы.
Справочное руководство Системные и стандартные функции	Руководство дает обзор применяемых в М7 системных и стандартных функций, которые могут быть использованы в программировании.
Руководство пользователя STEP 7	Руководство пользователя STEP 7 объясняет принципы использования и функции программного обеспечения для решения задач автоматизации STEP 7. Новичку в использовании STEP 7 и знатоку STEP 5 руководство даст обзор последовательности действий при конфигурировании М7–300/400. При работе с ПО можно целенаправленно обратиться к оперативной помощи в режиме online, которая обеспечивает детальную поддержку по вопросам использования ПО.
Руководство пользователя Системное программ-ное обеспечение М7–300/400	Руководство пользователя объясняет правила установки системного программного обеспечения М7–300/400, а также пуска в эксплуатацию и обслуживания контроллеров М7.
Руководство ProC/C++ для М7–300/400	Руководство содержит указания для пользователя. Описание языка Вы найдете в оперативной документации к Borland C++. При работе с ПО Вы можете использовать оперативную помощь в режиме online, которая ответит на все частные вопросы по использованию соответствующих инструментов (редактора/компилятора/редактора импорта символов).
Руководство Отладчик для программ на языке C	Руководство описывает управление и правила обращения с отладчиком Organon XDB386 для программ Borland C/C++.
Руководство по CFC¹	С помощью языка CFC (Continuous Function Chart) Вы можете реализовать программы с помощью графического монтажа блоков. Руководство содержит как указания для пользователя, так и описание языка. При использовании языка в первый раз рекомендуется познакомиться с методикой разработки программ. При работе с программным обеспечением Вы можете использовать оперативную помощь, которая ответит на все частные вопросы по использованию редакторов/компиляторов.
1 Дополнительное программное обеспечение к системному ПО для М7–300/400	

Путеводитель

Это руководство разделено на следующие тематические области:

- Часть 1 содержит общую информацию о понятиях и основах управления базового программного обеспечения для S7 и M7, а также о подготовке сеанса программирования. Вы должны прочитать первые три главы, прежде чем работать с программным обеспечением.
- В части 2 описываются конфигурирование и параметрирование аппаратуры.
- В частях 3 и 4 речь идет о программировании контроллеров S7 и M7.
- В части 5 описаны заключительные операции, такие как архивирование прикладных программ.

Первичному пользователю следует использовать руководство следующим образом:

1. Прочитайте первые три главы перед использованием программного обеспечения, чтобы познакомиться с понятиями и принципиальной последовательностью действий.
2. Затем используйте соответствующие главы руководства, если Вы хотите выполнить какой-то определенный шаг (напр., разработку таблицы символов).

Если Вы уже выполнили небольшой проект и благодаря этому набрались некоторого опыта, Вы можете читать отдельные главы руководства независимо друг от друга, чтобы получить информацию по некоторой теме.

Соглашения

Указания на другую документацию даны между косыми чертами /.../ с помощью номеров в списке литературы. Используя эти номера, Вы можете получить точное название документации из списка литературы в конце руководства.

Дальнейшая поддержка

По тем вопросам использования описанного программного обеспечения, ответы на которые Вы не найдете ни в бумажной документации, ни в оперативной online-помощи, обращайтесь, пожалуйста, к представителям фирмы Siemens в соответствующих представительствах и конторах фирмы. Адреса Вы найдете в приложении к /70/ или /100/ или в каталогах и в CompuServe (go autforum). Кроме того, в Вашем распоряжении имеется горячая линия:

Тел. +49(911) 895-7000 (факс 7001)

При наличии вопросов и замечаний к данному руководству заполните, пожалуйста, анкету в конце руководства и пошлите ее по указанному там адресу. Пожалуйста, внесите туда и Вашу личную оценку руководства.

Чтобы облегчить Вам вхождение в систему автоматизации SIMATIC S7, мы предлагаем Вам соответствующие курсы. Обращайтесь, пожалуйста, в Ваш региональный учебный центр или в центральный учебный центр в: D-90327 Nürnberg, тел. 0911 / 895 3154.

Особые указания

Раздел пользователя данного руководства не содержит точных указаний по работе с отдельными последовательностями шагов, а только разъясняется принципиальная последовательность действий. Более точную информацию для отдельных диалогов и их обработки Вы найдете по мере надобности в оперативной помощи.

Содержание

1	Обзор продукта	1–1
2	Инсталляция и деинсталляция	2–1
2.1	Предпосылки инсталляции	2–2
2.2	Авторизация / Права пользования	2–3
2.3	Правила обращения с авторизациями	2–5
2.4	Инсталляция / деинсталляция программного пакета STEP 7	2–7
2.5	Настройка интерфейса PG/PC	2–10
3	Общая философия управления	3–1
3.1	Запуск программного пакета STEP 7	3–2
3.2	Интерфейс пользователя: окна	3–3
3.3	Интерфейс пользователя: диалоговые окна	3–4
3.4	Вызов функций помощи	3–5
3.5	Объектно-ориентированная философия управления	3–6
3.6	Объекты и иерархия объектов	3–7
3.7	Создание и манипулирование объектами	3–10
3.8	Выбор объектов в диалоге	3–12
4	Планирование сеанса программирования	4–1
5	Создание и редактирование проектов	5–1
5.1	Компоненты и структура проекта	5–2
5.2	Компоненты S7/M7- программ	5–3
5.3	Создание проектов	5–4
5.4	Открытие проекта	5–6
5.5	Включение аппаратных компонентов и конфигурирование структуры..	5–7
5.6	Включение S7–и M7–программ	5–8
5.7	Включение компонентов для S7/M7–программ	5–9
5.8	Доступ к контроллерам в пределах проекта	5–11
5.9	Представление проекта в инструментальной системе (offline) / в контроллере (online)	5–12
5.10	Доступ к контроллеру без управления проектом	5–14

6	Определение символов	6–1
6.1	Символы	6–2
6.2	Таблица символов	6–3
6.3	Неполные и многозначные символы	6–4
6.4	Работа с таблицей символов	6–5
6.5	Определение отдельных символов в диалоге	6–6
6.6	Экспорт и импорт таблиц символов	6–7
7	Конфигурирование и параметризация модулей	7–1
7.1	Создание конфигурации - обзор	7–3
7.2	Конфигурирование централизованной структуры	7–5
7.3	Конфигурирование сети SINEC L2–DP	7–9
7.4	Создание мастер-системы	7–11
7.5	Конфигурирование ведомых модулей DP.....	7–13
7.6	Параметризация модулей	7–16
7.7	Параметризация сети SINEC L2–DP	7–17
7.8	Параметризация DP-мастера и сети SINEC L2–DP	7–18
7.9	Параметризация ведомых модулей DP	7–18
7.10	Включение ведомого модуля DP в SYNC–/FREEZE–группы	7–19
7.11	Сохранение, загрузка, чтение, изменение и копирование	7–20
7.12	Дополнительная инсталляция модулей в каталоге модулей	7–24
8	Конфигурирование сетей	8–1
8.1	Конфигурирование сетей - обзор	8–3
8.2	Определение конфигурации сети с помощью базового программного пакета STEP 7	8–4
8.3	Определение конфигурации сети с помощью дополнительного программного пакета NETPRO	8–6
8.4	Символы в изображении сети	8–8
8.5	Особенность конфигурации подсетей MPI в S7–300	8–10
8.6	Загрузка конфигурации через сеть и изменение адреса абонента	8–11
9	Конфигурирование связи через глобальные данные	9–1
9.1	Конфигурирование связи через глобальные данные - обзор	9–2
9.2	Принцип связи через глобальные данные	9–4
9.3	Тестирование GD–связи	9–6
9.4	Примеры конфигураций	9–7

10	Установление коммуникационных соединений	10–1
10.1	Коммуникационные соединения - обзор	10–2
10.2	Создание соединения	10–5
10.3	Свойства однородных соединений	10–7
10.4	Свойства соединений “точка-к-точке”	10–8
10.5	Коммуникационные соединения с ‘другими станциями’	10–10
11	Разработка прикладных программ	11–1
11.1	Программирование S7–CPU	11–2
11.2	Выбор языка программирования и редактора	11–4
11.3	Создание блоков с использованием KOP и AWL	11–5
11.4	Создание файлов с исходными текстами на AWL и S7–SCL	11–7
11.5	Создание блоков с использованием S7–GRAPH	11–8
11.6	Создание исходных текстов с использованием S7–HiGraph	11–9
12	Отображение справочных данных	12–1
12.1	Обзор	12–2
12.2	Принципиальная последовательность действий	12–3
12.3	Отображение структуры программы.....	12–5
12.4	Отображение перекрестных ссылок	12–7
12.5	Отображение занятости	12–9
12.6	Отображение свободных операндов (неиспользуемых символов)	12–10
12.7	Отображение отсутствующих обозначений (отсутствующих символов)	12–11
13	Загрузка прикладных программ	13–1
13.1	Определение и изменение рабочего режима	13–2
13.2	Концепция запоминания и загрузки	13–4
13.3	Общее стирание CPU контроллера	13–6
13.4	Загрузка прикладных программ из PG в контроллер	13–7
13.5	Загрузка блоков из PG в контроллер	13–8
13.6	Стирание блоков в CPU контроллера	13–9
13.7	Дополнительная загрузка блоков из PG в контроллер	13–10
13.8	Редактирование блоков из CPU в PG	13–11
13.9	Сжатие памяти пользователя (RAM)	13–12

14	Наблюдение и управление переменными	14–1
14.1	Введение	14–2
14.2	Создание таблицы переменных	14–3
14.3	Редактирование таблицы переменных	14–4
14.4	Наблюдение и управление переменными	14–6
15	Отображение информации CPU	15–1
15.1	Обзор	15–2
15.2	Отображение сообщений CPU	15–3
15.3	Отображение состояния модулей	15–5
15.4	Система времени и установка времени CPU	15–6
15.5	Отображение данных о CPU	15–7
15.6	Отображение имеющихся в наличии блоков (только S7–CPU)	15–8
15.7	Отображение степени использования памяти пользователя	15–9
15.8	Отображение времен цикла CPU	15–10
15.9	Отображение коммуникационных соединений	15–11
15.10	Отображение содержимого диагностического буфера	15–12
15.11	Отображение содержимого стеков (только S7–CPU)	15–15
16	Введение в системы M7	16–1
16.1	Дополнительный программный пакет M7	16–2
16.2	Операционные системы для M7–300/400	16–5
17	Управление системами M7	17–1
17.1	Подготовка инсталляции	17–2
17.2	Установка M7 RMOS32 на плату памяти	17–7
17.3	Установка M7 RMOS32 на жесткий диск	17–8
17.4	Установка M7 RMOS32 с MS–DOS на жесткий диск	17–9
17.5	Установка M7 RMOS32 с MS–Windows на жесткий диск	17–10
17.6	Установка M7 RMOS32 на OSD	17–11
17.7	Дополнительная инсталляция операционной системы M7	17–13
17.8	Актуализация программ, записанных в ПЗУ	17–14
17.9	Перенос и стирание программ в системе M7	17–16
17.10	Функции опроса и управления для M7–300/400	17–22

18	Сохранение прикладных программ	18–1
18.1	Архивирование и деархивирование проектов и библиотек	18–2
18.2	Сохранение содержимого ОЗУ CPU во встроенном СППЗУ	18–4
18.3	Сохранение блоков и прикладных программ на плате памяти	18–5
19	Печать	19–1
A	Открытие проектов версии 1	A–1
B	Список литературы	A–2

Глоссарий

Предметный указатель

Что такое STEP 7?

STEP 7 является программным пакетом для конфигурирования и программирования контроллеров SIMATIC S7–300/400 и SIMATIC M7–300/400, а также комплектных устройств SIMATIC C7. В отношении программирования и конфигурирования C7 ведет себя как SIMATIC S7–300. STEP 7 состоит из базового программного пакета и дополнительного программного обеспечения, которые исполняются под Windows 95.

Базовый программный пакет

Базовый программный пакет STEP 7 поддерживает Вас во всех фазах процесса разработки решений по автоматизации, таких как, например:

- организация и управление проектами
- конфигурирование и параметризация аппаратных средств и связи
- управление символами
- разработка программ для контроллеров S7 (для разработки программ для контроллеров M7 в Вашем распоряжении имеется дополнительное программное обеспечение)
- загрузка программ в контроллеры
- тестирование установок автоматизации
- диагностика неисправностей установок

Интерфейс пользователя программного пакета STEP 7 оформлен согласно современным эргономическим познаниям и обеспечивает легкое вхождение.

Языки в базовом программном пакете

Языки программирования KOP и AWL для S7–300/400 являются встроенной составной частью базового программного пакета.

- KOP (контактный план) является графическим языком программирования. Синтаксис команд напоминает электрическую цепь. KOP обеспечивает Вам простое прослеживание прохождения сигналов между токовыми шинами через контакты, сложные элементы и обмотки.
- AWL (список команд) является текстовым, близким к машинному коду языком программирования. Если программа программируется в AWL, то отдельные команды соответствуют в дальнейшем рабочим шагам, посредством которых CPU обрабатывает программу. В целях облегчения программирования AWL был дополнен некоторыми конструкциями языка программирования высокого уровня (такими как, например, структурированный доступ к данным и параметры блоков).

Другие языки программирования имеются в продаже как дополнительное программное обеспечение.

Дополнительные языки для SIMATIC S7

Для программирования контроллеров SIMATIC S7–300/400 в распоряжении имеются следующие языки в качестве дополнительного программного обеспечения:

- S7–SCL является текстовым языком программирования высокого уровня, соответствующим DIN EN 61131–3. Он содержит языковые конструкции, которые в похожей форме можно найти также в языках программирования Pascal и C. Поэтому S7–SCL особенно подходит для пользователей, которые уже привыкли работать с языком программирования высокого уровня. В частности, S7–SCL можно применять для программирования часто повторяющихся или очень сложных функций.
- S7–GRAPH является языком программирования для комфортабельного описания управления протеканием процесса (программирования цепочки шагов). При этом протекание процесса разбивается на шаги. Эти шаги содержат, в частности, действия по управлению выходами. Переход от одного шага к следующему управляется условиями последовательного включения.
- S7–HiGraph является языком программирования для комфортабельного описания асинхронных, непоследовательных процессов в форме графов состояний. Для этого управляемая установка расчленяется на функциональные единицы, которые могут принимать разные состояния. Эти функциональные единицы могут синхронизироваться путем обмена сообщениями.
- CFC (на стадии подготовки) является языком программирования для графического соединения [монтажа] имеющихся функций. Эти функции открывают широкую область от простых логических сопряжений до сложного регулирования и управления. Большое количество таких функций имеется в распоряжении в форме блоков в библиотеке. Вы программируете таким образом, что копируете блоки на план и соединяете линиями соединительные стыки блоков.

Дополнительные возможности для M7–300/400

Для программирования контроллеров SIMATIC M7–300/400 в распоряжении имеется следующее дополнительное программное обеспечение:

- M7–SYS содержит операционную систему M7 RMOS 32 и системные программы. Он является предпосылкой для использования пакетов M7–ProC/C++ и CFC для M7 (на стадии подготовки).
- M7–ProC/C++ делает возможной включение среды разработки фирмы Borland для языков программирования C и C++ в среду разработчика STEP 7.
- Borland C++ содержит среду разработки фирмы Borland.
- CFC: смотрите под заголовком “Дополнительные языки для SIMATIC S7”.

Инсталляция и деинсталляция

2

Обзор

Программа Setup дает Вам возможность установки программного пакета STEP 7 под управлением меню. Программа Setup вызывается посредством обычной для Windows 95 процедуры установки программного обеспечения.

В случае вновь поставляемых устройств программирования (PG) пакет STEP 7 уже установлен заранее. Благодаря этому, первая установка сильно упрощается.

Обзор главы

В раз-деле	Вы найдете	на стр.
2.1	Предпосылки инсталляции	2–2
2.2	Авторизация / Права пользования	2–3
2.3	Правила обращения с авторизациями	2–5
2.4	Инсталляция / деинсталляция программного пакета STEP 7	2–7
2.5	Настройка интерфейса PG/PC	2–10

2.1. Предпосылки инсталляции

Операционная система	Microsoft Windows 95.
Базовые аппаратные средства	<p>Устройство программирования или PC с компонентами:</p> <ul style="list-style-type: none">• процессор 80486 (или выше) и• расширенная память, как минимум 16 Мб• VGA–монитор или другой монитор, поддерживаемый Microsoft Windows 95.• клавиатура и, по желанию, но рекомендуемая мышь, поддерживаемая Microsoft Windows 95.
Объем памяти	<p>Требуемый объем памяти на жестком диске:</p> <ul style="list-style-type: none">• 70 -100 Мб занимает базовый пакет. Потребность в памяти зависит от выбранного объема установки базового программного пакета.• Примерно 60 МБ с вычетом расширения основной памяти должен иметь в распоряжении STEP 7 для создания файлов выгрузки (т.е. 44 МБ при расширении основной памяти 16 Мб).• Около 50 Мб Вы должны предусмотреть для Ваших пользовательских данных.• По крайней мере, 1 Мб свободной памяти на дисковом C: для Setup (Setup–файлы стираются по окончании установки). <p>Требуемый объем памяти для дополнительного программного обеспечения:</p> <ul style="list-style-type: none">• Дополнительное программное обеспечение M7 ProC/C++ со средой разработки C нуждается примерно в 100 Мб.• Остальное дополнительное программное обеспечение STEP 7 требует в каждом случае от 10 до 20 Мб.
MPI–интерфейс (по выбору)	<p>MPI–интерфейс между инструментальной системой (устройством программирования или PC) и контроллером необходим только тогда, когда Вы хотите общаться из-под STEP 7 с контроллером посредством MPI:</p> <ul style="list-style-type: none">• либо PC/MPI–кабель, подключенный к коммуникационному порту Вашего устройства, либо• MPI–модуль, установленный в Вашем устройстве. <p>В некоторых устройствах программирования MPI–интерфейс уже встроен.</p>
Внешний программатор ППЗУ	<p>В случае использования PC внешний программатор ППЗУ нужен только тогда, когда Вы хотите прошивать EPROM.</p>

2.2. Авторизация / Права пользования

Введение

Для использования программного пакета STEP 7 необходима специфическая для продукта авторизация (передача права пользования). Защищенное таким образом программное обеспечение можно использовать только тогда, когда на жестком диске соответствующего PG/PC распознается авторизация, требуемая для программы или программного пакета.

В каждом случае требуются разные авторизации, например, для STEP 7, STEP 7-Mini, а также для дополнительного программного обеспечения.

Авторизационная дискета

Для авторизации Вам нужна входящая в комплект поставки, защищенная от копирования авторизационная дискета. Она содержит авторизацию и программу AUTHORS, необходимую для отображения, инсталляции и деинсталляции авторизации.

Количество возможных авторизаций установлено на дискете с использованием счетчика авторизаций. При каждой авторизации счетчик понижается на 1. Если он достиг нулевого значения, то с помощью данной дискеты дальнейшая авторизация больше невозможна.



Осторожно

Примите во внимание указания из файла LIESMICH.TXT на дискете авторизации и правила из раздела 2.3. При их несоблюдении существует опасность того, что авторизация необратимым образом потеряется.

Проведение авторизации при первичной установке

Вы должны проводить авторизацию, если в рамках первичной установки приглашаетесь к этому посредством соответствующего сообщения.

Поступайте следующим образом:

1. Вставьте дискету авторизации, когда будет отображено соответствующее приглашение.
2. Затем квитируйте приглашение.

Право пользования передается на физический дисковод, т.е. Ваш компьютер “запоминает”, что Вы имеете право пользования.

Отложенная авторизация

Если Вы запускаете программный пакет STEP 7 и нет в наличии авторизации, то Вы получаете соответствующее сообщение. Для того, чтобы провести авторизацию задним числом, вызовите программу AUTHORS. С помощью нее можно авторизацию отображать, устанавливать и деинсталлировать. Эта программа управляется меню.



Указание

При установке авторизации для STEP 7 и STEP 7-Mini всегда задавайте дискковод C: в качестве целевого дисквода.

Деинсталляция авторизации

Если необходима повторная авторизация, например, когда Вы хотите форматировать дискковод, на котором находится право пользования, то Вы должны прежде “спасти” право пользования. Для этого Вам нужна оригинальная авторизационная дискета.

Поступайте следующим образом, чтобы перенести авторизацию обратно на авторизационную дискету:

1. Вставьте в дискковод A: оригинальную авторизационную дискету.
2. Вызовите программу AUTHORS.EXE с авторизационной дискеты.
3. Выберите команду меню **Autorisierung** [Авторизация] → **Deinstallieren** [Деинсталлировать].
4. Задайте в появляющемся вслед за этим диалоговом окне дискковод, в котором находится авторизация, и квитируйте диалог. Отобразится список всех авторизаций на соответствующем дискводе.
5. Отметьте авторизацию, которую Вы хотите деинсталлировать, и квитируйте диалог.

Если процесс завершится без ошибок, то Вы получите сообщение

”Autorisierung <Name> erfolgreich von Laufwerk <X:> entfernt.”

[“Авторизация <Имя> успешно удалена с дисквода <X:>.”]

6. Квитируйте это сообщение.

После этого вновь отобразится диалоговое окно со списком оставшихся авторизаций на дискводе. Закройте диалоговое окно, если Вы не хотите деинсталлировать дополнительные авторизации.

Потом Вы можете повторно использовать эту дискету для авторизации.

Если Ваш жесткий диск имеет дефект ...

Если на Вашем жестком диске появляется дефект прежде, чем Вы смогли спасти право пользования, пожалуйста, обратитесь в Ваше компетентное представительство SIEMENS.

2.3. Правила обращения с авторизациями



Осторожно

Примите во внимание указания из данной главы и файла LIESMICH.TXT на дискете авторизации. В случае их несоблюдения существует опасность того, что авторизация необратимым образом потеряется.

Деинсталляция требуется

Перед форматированием, сжатием или восстановлением Вашего жесткого диска или перед установкой Вашей новой операционной системы Вы должны прежде деинсталлировать имеющиеся, возможно, авторизации.

Резервное копирование резервной

Если резервная копия Вашего жесткого диска содержит копии авторизаций, то существует опасность того, что при обратной записи файлов

копии на жесткий диск будут перезаписываться и потому разрушаться установленные и все еще имеющие силу авторизации.

Для того, чтобы предотвратить потерю авторизации вследствие перезаписи авторизованной системы с помощью резервной копии, Вы должны

- либо удалять все авторизации перед созданием резервной копии,
- либо исключить авторизации из процесса сохранения.

Оптимизация жесткого диска

Если Вы используете программу оптимизации, которая предоставляет возможность перемещения фиксированных блоков, то Вам можно применять эту возможность только тогда, когда Вы перед этим переписали авторизации с жесткого диска обратно на авторизационную дискету.

Дефектные сектора

Вместе с авторизацией на целевом дисковом кластере возникает кластер, обозначенный как “дефектный”. Не пытайтесь восстановить этот дефектный кластер.

Защита от записи

Авторизационная дискета не должна быть защищена от записи.

Защита от копирования

Вы не можете создавать резервные копии Вашей дискеты авторизации.

**Допустимые
дисководы**

Невозможно установить авторизацию на следующих дисководах / средах:

- RAM–дисководы,
- дискеты,
- сжатые дисководы (например, DBLSPACE). В случае сжатых дисководов Вы можете делать установку на соответствующем главном дисководе.

Инструмент авторизации предотвращает установку авторизаций на недопустимые дисководы.

Место в памяти

В ходе авторизации заводится каталог защиты с файлами авторизации с атрибутами "System" и "Versteckt" ["системный" и "скрытый"].

- Эти атрибуты нельзя изменять.
- Эти файлы нельзя изменять или стирать.

В противном случае авторизация необратимо теряется.

Каталог защиты 'AX NF ZZ' заводится один раз на дисковод. Он содержит все установленные на дисководе авторизации. Он заводится при установке первой авторизации и стирается при удалении последней авторизации.

На каждую авторизацию в каталоге защиты заводятся два одноименных файла с разными расширениями имени. Эти файлы получают в качестве имени файла имя авторизации.

**Количество
авторизаций
памяти. Не надо опасаться**

На одном дисководе Вы можете установить произвольно много авторизаций, до тех пор, пока в распоряжении имеется требуемое место в взаимных помех этих авторизаций.

**Дефектная
авторизация**

Дефектные авторизации невозможно больше удалять с жесткого диска с помощью инструмента авторизации AUTHORS. Они могут даже блокировать установку новых и действующих авторизаций. В этом случае обращайтесь в Ваше компетентное представительство SIEMENS.

**Инструмент
авторизации**

Используйте актуальную версию V3.x инструмента авторизации AUTHORS, ни в коем случае не используйте устаревшие версии (V1.x, V2.x).

2.4. Инсталляция / деинсталляция программного пакета STEP 7

Обзор

STEP 7 содержит программу Setup, которая автоматически выполняет инсталляцию. Выводимые на экран приглашения ведут Вас шаг за шагом через весь процесс инсталляции.

Существенными фазами инсталляции являются:

- копирование файлов на инструментальную систему,
- установка драйверов для СППЗУ и связи,
- авторизация (если желательна).



Указание

Устройства программирования Siemens (такие как, например, PG 740) поставляются с устанавливаемым программным пакетом STEP 7 на жестком диске.

Подготовка

Прежде чем можно будет начинать инсталляцию, нужно запустить Windows 95.

- Вам не нужны внешние носители данных, если устанавливаемый программный пакет STEP 7 уже находится на жестком диске
- Для того, чтобы установить с дискеты, вставьте дискету 1 в дисковод дискет Вашего устройства программирования/PC (обозначенный в большинстве случаев как дисковод A: или B:)
- Для того, чтобы установить с CD-ROM, вложите CD-ROM в привод CD-ROM Вашего PC.

Запуск программы инсталляции

Поступайте следующим образом:

1. Запустите из-под Windows 95 диалог по инсталляции программного обеспечения посредством двойного щелчка мыши на символе "Software" ["Программное обеспечение"] в "Systemsteuerung" ["Панель управления"].
2. Щелкните мышью на "Installieren".
3. Вложите носитель данных (дискета 1) или CD-ROM и щелкните мышью на "Weiter" ["далее"]. Теперь Windows 95 самостоятельно ищет программу инсталляции setup.exe.
4. Следуйте шаг за шагом указаниям, которые отображает Вам программа инсталляции.

Эта программа ведет Вас по шагам через процесс инсталляции. В каждом случае Вы можете переключиться на последующий или предыдущий шаг.

Во время процесса инсталляции в диалоговом окне Вам отображаются вопросы или предлагаются возможности для выбора. Пожалуйста, прочитайте следующие указания для того, чтобы иметь возможность более быстро и легко отвечать в ходе диалога.

**Если одна
версия STEP 7
уже установлена ...**

Если программа инсталляции установит, что одна версия STEP 7 уже находится на инструментальной системе, то отображается соответствующее сообщение, и у Вас есть следующие возможности выбора:

- прервать инсталляцию (чтобы потом деинсталлировать старую версию STEP 7 из-под Windows 95 и после этого повторно запустить инсталляцию) или
- продолжать инсталляцию и тем самым переписать старую версию новой версией.

С точки зрения чистой заботы о программном обеспечении Вы должны перед инсталляцией деинсталлировать содержащиеся, может быть, старые версии. Простое переписывание старой версии имеет, кроме того, тот недостаток, что при последующей деинсталляции еще имеющиеся, возможно, части более старой версии не удаляются.

**Об объеме
инсталляции**

Для установления объема инсталляции у Вас есть три возможности выбора:

- Максимальная конфигурация: все языки интерфейса пользователя, все приложения и все примеры. О необходимом для этого объеме памяти справьтесь, пожалуйста, в актуальной информации о продукте.
- Минимальная конфигурация: только один язык, без примеров. О необходимом для этого объеме памяти справьтесь, пожалуйста, в актуальной информации о продукте.
- Определяемая пользователем: Вы можете выбрать объем инсталляции в отношении программ, базы данных, примеров и связи.

Об авторизации

При инсталляции проверяется, имеется ли на жестком диске авторизация. Если авторизация не обнаруживается, то появляется указание о том, что данное программное обеспечение может использоваться только с авторизацией. Вы можете провести авторизацию сразу, если желаете, или продолжать инсталляцию, а авторизацию наверстать в более поздний момент времени. В первом случае вложите дискету авторизации, когда Вас к этому пригласят (см. разделы 2.2 и 2.3).

**О параметризации
платы памяти**

Во время процесса инсталляции отображается диалог для параметризации карты памяти.

- Если Вы не используете плату памяти, то Вам не нужен драйвер СППЗУ. Выберите "kein EPROM-Treiber" ["не нужен драйвер СППЗУ"].
- В противном случае выберите запись, относящуюся к Вашему PG.
- Если Вы используете РС, то Вы можете выбрать драйвер для внешнего программатора ППЗУ. При этом Вы должны дополнительно задать интерфейс, к которому подключен программатор ППЗУ (например, LPT1).

После инсталляции Вы можете изменять установленные параметры, вызывая программу "Memory Card parametrieren" ["Параметрирование платы памяти"] из программной группы STEP 7.

О системе флэш-файлов

В диалоге по параметризации платы памяти Вы можете указать, должна ли устанавливаться система флэш-файлов.

Система флэш-файлов нужна, в частности, тогда, когда Вы хотите записывать или стирать в плате памяти СППЗУ отдельные блоки, не изменяя остального содержимого платы.

Если Вы используете подходящее устройство программирования (PG 720/740) и хотите использовать эту функциональную возможность, то выберите установку системы флэш-файлов.

Об установке интерфейса PG/PC

Во время процесса установки отображается диалог по настройке интерфейса PG/PC. Прочитайте по этому поводу раздел 2.5 на странице 2-10.

О завершении установки ...

Если установка была успешной, то это объявляется на экране посредством соответствующего сообщения.

Если при установке производились вмешательства в DOS-файлы, то Вы приглашаетесь к новому запуску Windows 95. После нового запуска Windows 95 Вы можете запустить интерфейс входа в STEP 7, Администратор SIMATIC (SIMATIC Manager).

В противном случае Вы можете вызвать запуск администратора SIMATIC сразу же из заключительного диалога.

Ошибки во время установки

К прерыванию установки ведут следующие ошибки:

- Если сразу после запуска Setup появляется ошибка инициализации, то с высокой вероятностью *setup* из-под Windows 95 не запустился.
- Недостаточно места в памяти: независимо от объема установки Вам нужно около 100 Мб свободной памяти на Вашем жестком диске для базового программного обеспечения.
- Дефектная дискета: если Вы установите, что дискета является дефектной, то обратитесь, пожалуйста, в Ваше представительство Siemens.
- Ошибка оператора: начните установку снова и тщательно соблюдайте команды.

Результат

После успешной установки организуется программная группа STEP 7.

Деинсталляция

Для деинсталляции используйте обычный для Windows 95 образ действий:

1. Запустите под Windows 95 диалог по установке программного обеспечения посредством двойного щелчка мышью по символу "Software" ["Программное обеспечение"] в "Systemsteuerung" ["Панель управления"].
2. Отметьте запись STEP 7 в отображаемом списке установленного программного обеспечения. Затем приведите в действие кнопку "Entfernen" ["Удаление"] для удаления программного обеспечения.
3. Если появляются диалоговые окна "Freigegebene Datei entfernen" ["Удалить разблокированный файл"], то в случае сомнения щелкните мышью на кнопке "Nein" ["Нет"].

2.5. Настройка интерфейса PG/PC

Введение	<p>Во время процесса инсталляции отображается диалог по настройке интерфейса PG/PC. Вы можете этот диалог отображать также и после инсталляции, вызывая программу "PG/PC-Schnittstelle einstellen" ["Настройка интерфейса PG/PC"]. Благодаря этому возможно изменять параметры интерфейса также и позже, независимо от инсталляции.</p>
Предварительно определенные профили шины	<p>Для того, чтобы упростить настройку интерфейса PG/PC, Вам в окне диалога предлагаются предварительно определенные наборы основных параметров (профили шины) .</p>
Принципиальная последовательность действий при первичной инсталляции	<p>Поступайте следующим образом (подробное описание в Online–помощи):</p> <ol style="list-style-type: none">1. Выберите профиль шины.2. Отобразите свойства профиля шины.3. Согласуйте специфические для приложения параметры со свойствами профиля шины. <p>Изменения требуются тогда, когда возникают конфликты с другими установками (например, назначения прерываний и адресов).</p>
Проверка назначения прерываний и адресов	<p>Если Вы используете PC с платой MPI, то Вам в любом случае следует проверять, свободны ли предварительно установленное прерывание и предварительно установленная область адресов и при необходимости выбрать свободные прерывание и/или область адресов.</p> <p>Вы можете отобразить действующие назначения, находясь под Windows 95, следующим образом:</p> <ol style="list-style-type: none">1. Откройте "System" в "Systemsteuerung" и выберите вкладку "Geräte-Manager" ["Администратор устройств"] в отображаемом диалоговом окне.2. В отображаемом списке отметьте запись "Computer" и щелкните на кнопке "Eigenschaften" ["Свойства"].3. В следующем диалоговом окне Вы можете отобразить список назначенных прерываний (IRQ) или список назначенных областей адресов (ввод/вывод [E/A]) путем маркирования соответствующего поля выбора.

Обзор

Программное обеспечение для конфигурирования и программирования SIMATIC S7/M7/C7 оформлено в соответствии с современными эргономическими познаниями и поэтому в значительной степени является самообъясняющимся.

Если Вы еще не имеете опыта работы с подобными интерфейсами, то Вы можете справиться в данной главе о важнейших элементах управления и мире понятий.

Если Вы знаете Windows 95, то Вы можете в разделах 3.1, 3.5 и 3.6 познакомиться с основными понятиями STEP 7 и перескочить через остальные разделы данной главы.

Обзор главы

В раз-деле	Вы найдете	на стр.
3.1	Запуск программного пакета STEP 7	3–2
3.2	Интерфейс пользователя: окна	3–3
3.3	Интерфейс пользователя: диалоговые окна	3–5
3.4	Вызов функций помощи	3–7
3.5	Объектно-ориентированная философия управления	3–8
3.6	Объекты и иерархия объектов	3–9
3.7	Создание и манипулирование объектами	3–12
3.8	Выбор объектов в диалоге	3–14

3.1. Запуск программного пакета STEP 7

Запуск

После запуска Windows 95 Вы найдете в интерфейсе Windows символ администратора SIMATIC (SIMATIC Manager), вход в программный пакет STEP 7.

STEP 7 запускается как приложение Windows 95. Быстрее всего Вы можете запустить STEP 7, если Вы расположите указатель мыши на этом символе и дважды щелкнете кнопкой. В ответ на это открывается окно администратора SIMATIC. Отсюда можно обратиться ко всем установленным Вами функциям как базовой системы, так и дополнительного программного пакета.

Вы можете запустить администратор SIMATIC альтернативно также с помощью кнопки "Start" на панели задач в Windows 95: (запись "Simatic/STEP 7 V2").



Указание

Более подробную информацию о стандартном управлении и опциях Windows 95 Вы найдете в Вашем руководстве пользователя Windows или в Online-помощи Windows 95.

Администратор SIMATIC (SIMATIC Manager)

Администратор SIMATIC является входным интерфейсом для конфигурирования и программирования. Вы можете

- конфигурировать и параметризовать аппаратные средства
- программировать модули
- соединять в сеть аппаратные средства
- тестировать и вводить в эксплуатацию Ваши программы.

Доступ к функциям оформлен объектно-ориентированным образом, является интуитивным и легко изучаемым.

С помощью Администратора SIMATIC Вы можете работать

- offline, т.е. без подключенного контроллера или
- online, т.е. с подключенным контроллером.

(При этом обратите внимание на соответствующие указания по безопасности.)

Дальнейший образ действий

Вы разрабатываете решения по автоматизации в форме "проектов". Вы облегчите себе труд, если Вы прежде познакомитесь с некоторыми основополагающими темами:

- интерфейс пользователя,
- Online-помощь,
- некоторые базовые элементы управления.

3.2. Интерфейс пользователя: окна

Обзор

Стандартные компоненты окна представлены на рисунке 3-1

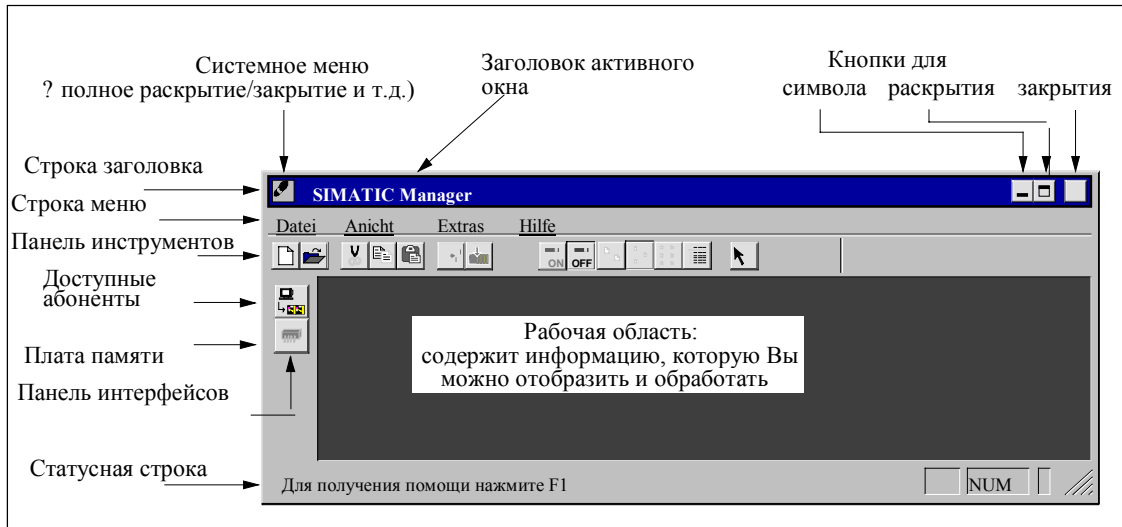


Рис. 3-1. Компоненты окна

Строка заголовка и строка меню

Строка заголовка и строка меню всегда находятся на верхнем краю окна. Строка заголовка содержит заголовок окна и символы для управления окном. Строка меню содержит все меню, которые предоставляются в распоряжение в окне.

Панель инструментов

Панель инструментов содержит символы, через которые Вы посредством щелчка мыши можете получить быстрый доступ к часто применяемым и имеющимся в данный момент в распоряжении командам меню.

Краткая информация о функции передается на фоне, когда Вы на короткое время оставляете курсор расположенным над символом.

Статусная строка

В статусной строке отображаются текущие установки окна, а также дополнительная информация, которая зависит от типа окна.

Панель интерфейсов

Через кнопки "Erreichbare Teilnehmer" ["Доступные абоненты"] и "S7-Memory Card" ["Плата памяти S7"] в каждом случае возможно открыть окно, в котором отображаются доступные абоненты или содержимое платы памяти. Плата памяти для отображения содержимого должна вставляться в разъем Вашего устройства программирования.

Если в действующей конфигурации какой-либо доступ невозможен, то соответствующий символ представляется серым цветом.

3.3. Интерфейс пользователя: диалоговые окна

Ввод в диалоговые окна

Вы можете вводить в диалоговые окна информацию, которая нужна для выполнения определенной задачи. Наиболее часто применяемые компоненты диалоговых окон поясняются на основании примера на рисунке 2-3:

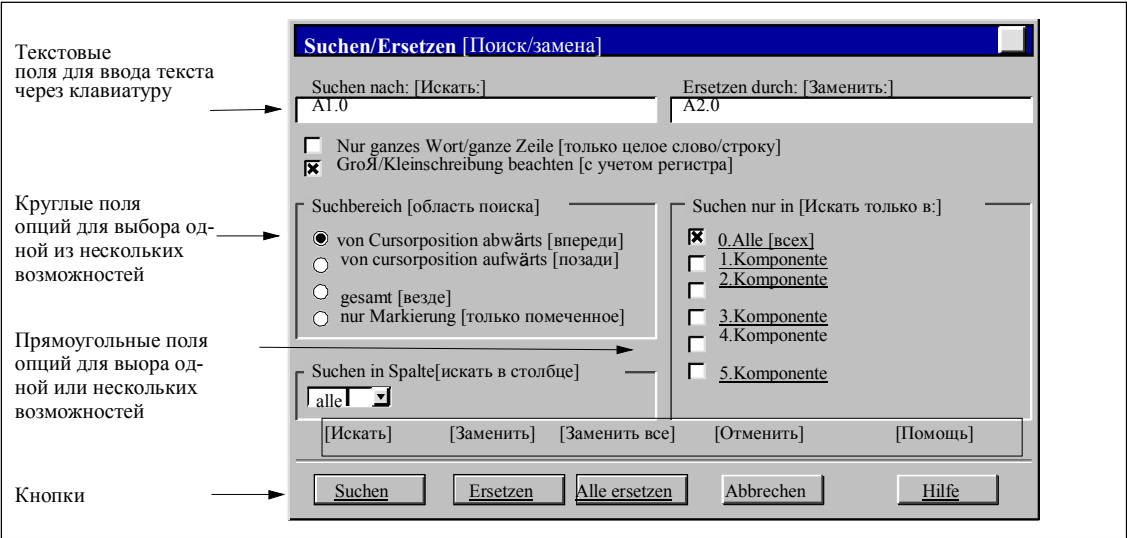


Рис. 3-2. Пример диалогового окна

Списковые поля / комбинационные поля

Иногда текстовые поля снабжены направленной вниз стрелкой. Эта стрелка показывает, что для данного поля в распоряжении имеются дополнительные возможности выбора. Щелкните мышью на стрелке, чтобы раскрыть списковое или комбинационное поле. Если в нем Вы щелкаете мышью на записи, то она автоматически принимается в текстовое поле.

Диалоги на вкладках

Содержимое некоторых диалоговых окон ради лучшей наглядности организовано в виде вкладок. Обозначения отдельных вкладок (регистровые диалоги) находятся на верхнем краю диалогового окна. Для того, чтобы вывести определенную вкладку “на передний план”, просто щелкните мышью на обозначении вкладки.

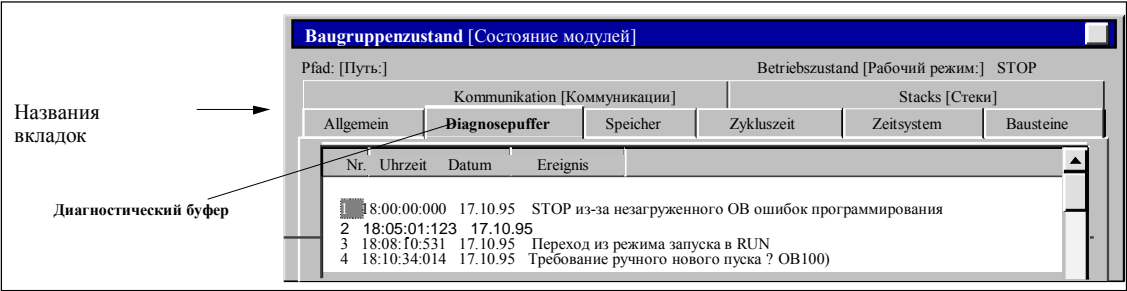


Рис. 3-3. Пример регистрового диалога

3.4. Вызов функций помощи

Online–помощь

Online–помощь предоставляет Вам информацию на том месте, где Вы можете ее быстро понять и реализовать. Так Вы можете быстро и целенаправленно получить информацию без необходимости поиска ее в справочниках. В Online–помощи Вы найдете:

- Темы помощи: предлагает различные пути доступа к отображению вспомогательной информации.
- Введение: дает сжатую справку о применении, существенных признаках и наборе функций приложения.
- Первые шаги: объединяет первые шаги, которые Вы должны выполнить для того, чтобы прийти к первому результату.
- Использование помощи: предоставляет описание возможностей, которые имеются в Вашем распоряжении для того, чтобы найти определенную информацию в данной системе помощи.
- Info: дает информацию о действующей версии приложения.
- Контекстно-зависимая помощь: С помощью клавиши F1 Вы получите информацию по объекту, который Вы перед этим отметили посредством мыши.

Через меню **Hilfe** Вы можете также, исходя из любого окна, получить доступ к темам, которые относятся к текущему состоянию диалога.

Вызов Online–помощи

Вы можете вызывать Online–помощь разными способами:

- Выберите команду меню из меню **Hilfe** в строке меню.
- В диалоговом окне щелкните мышью на кнопке "Hilfe". В ответ на это Вы получите помощь по соответствующему диалоговому окну.
- Расположите указатель мыши в окне на теме, по которой Вы нуждаетесь в помощи, и нажмите клавишу F1.
- Вопросительный знак - курсор Windows 95.

Три последних названных способа Online–помощи называются контекстно-зависимой помощью.

Вызов краткой помощи

Краткая помощь к кнопкам, расположенным на панели инструментов, отображается, когда Вы располагаете курсор на кнопке и оставляете его там на короткое время.

Вызов учебных программ STEP 7

Дополнительно к Online–помощи Вы имеете возможность вызывать учебную программу (Tutorial). Учебная программа объясняет Вам, как использовать базовое программное обеспечение S7. Учебная программа не является составной частью базового программного обеспечения.

3.5. Объектно-ориентированная философия управления

Цель: простое управление

Графическая оболочка пользователя должна обеспечить Вам, по возможности, интуитивное управление. Поэтому Вы находите объекты, которые Вы знаете из мира Вашей повседневной работы, в частности. станции, модули, программы, блоки.

Действия, которые Вы выполняете при работе с пакетом STEP 7, охватывают создание , выбор и манипуляцию такими объектами.

Иерархия объектов

Некоторые предметы реального мира состоят в определенных отношениях друг с другом. Эти отношения из реального мира при отображении объектов на экране проявляются в логической иерархии. Так, например, станция содержит программируемый модуль. Программируемому модулю может сопоставляться программа. Программа содержит символьную информацию и блоки.

Особое положение занимают объекты, которые стоят на вершине такой иерархии. К ним относятся, например, проекты и библиотеки. Они образуют точки входа в диалоговые окна для выбора объектов.

Если Вы копируете объект, то вместе с ним копируется вся лежащая ниже иерархия. Благодаря этому, создаются обширные возможности для повторного использования разработанных один раз компонентов проекта автоматизации.

Отличия от инструментально-ориентированного управления

В случае традиционного инструментально-ориентированного управления нужно было вначале обдумать, какой инструмент требуется для решения определенной задачи, а затем вызывать инструмент.

В случае объектно-ориентированного управления принципиальный образ действий состоит в том, чтобы обдумать, какой объект должен обрабатываться , а затем открыть и обработать этот объект.

В случае объектно-ориентированной философии управления знание специального синтаксиса команд больше не требуется. Объекты представляются на рабочей поверхности графическими символами, которые можно открывать посредством команд меню или щелчками мыши.

При открытии объекта автоматически вызываются подходящие компоненты программного обеспечения для того, чтобы показать или обработать содержимое объекта.

Дальнейшее чтение ...

На следующих страницах представлены важнейшие объекты, а затем описаны некоторые основополагающие действия при разработке объектов. Познакомьтесь с этим уже сейчас, так как позже изложение больше не вдается в подробности этих постоянно повторяющихся базовых команд.

3.6. Объекты и иерархия объектов

Обзор

Объекты служат

- в качестве контейнеров для других объектов,
- в качестве носителей функций, которые оперируют объектом (например, для запуска определенного приложения) или
- в качестве носителей свойств данного объекта.

Таблицы на рисунках 3–4 и 3–5 обзорно показывают важнейшие объекты и их графическое представление на рабочей поверхности:

Контейнер

В контейнере могут находиться другие контейнеры или объекты. Они отображаются, когда Вы открываете контейнер.











Символ	Объект	Описание	находится в контейнере:
	Проект	представляет совокупность всех данных и программ решения задач автоматизации	(на вершине иерархии объектов)
	Библиотека	может содержать S7-программы и служит для хранения блоков	(на вершине иерархии объектов)
	станция SIMATIC 300 станция SIMATIC 400	представляет компоненты аппаратной структуры с одним или несколькими прог. модулями	Проект
	CPUxxx FMyyy	представляет программируемый модуль	Станция
	S7-программа	может содержать таблицу символов, программу пользователя, исходные тексты и планы*	программируемый модуль или проект
	M7-программа	может содержать таблицу символов, C/C++ - программу, Dos/Windows-программу, исходные тексты и планы*	программируемый модуль или проект
	<AP-off> Offline-программа пользов.	может содержать блоки (список см. под блоком)	S7-программа
	<AP-on> Online-программа пользов.	содержит выполнимые файлы, резидентно загруженные в целевую систему	S7-/M7-программа (online)
	SO (исходные тексты)	может содержать исходные программы в текстовой форме ? напр., исход. тексты на AWL)	S7-/M7-программа
	PL (планы*)	может содержать исходную графику на CFC, CFC-план	S7-/M7-программа

Рис. 3-4. Контейнеры-объекты

* Контейнер “Планы”, а также объект “План” нужны при использовании дополнительного программного обеспечения CFC

Объекты как носители функций

Когда Вы открываете объект, появляется окно, в котором Вы можете обрабатывать объект.

Объект является либо контейнером, либо носителем функций. Исключение образуют станции: они являются как контейнерами (для программируемых модулей), так и носителями функций (для конфигурирования аппаратных средств).

- При двойном щелчке мыши на станции отображаются содержащиеся в ней программируемые модули (станция как контейнер)
- После открытия станции с помощью команды меню **Bearbeiten → Objekt öffnen** [Редактировать→ Открыть объект] Вы можете конфигурировать и параметризовать эту станцию (станция как носитель функции).

Символ	Объект	Описание/содержание	находится в контейнере:
	Сеть	для запуска инструмента проектирования сети и для установки свойств сети	Проект
	SC (Конфигурац. станции)	для запуска инструмента конфигурирования аппаратных средств	Станция
	Программируемый модуль	объект представляет данные параметризации программируемого модуля	Станция
	CO (таблица соединений)	для определения соединений в сети	Модуль
	SY (таблица символов)	для сопоставления символов сигналам и другим переменным	S7-/M7-программа
	Блок (offline) Блок (online)	Имя: • Кодовые блоки (OB, FB, FC, SFB, SFC) • Блоки данных (DB) • Определяемые пользователем типы данных (UDT) • Таблицы переменных (VAT)	Программа пользователя
	SDB ? Системные блоки данных)	Объект представляет системные блоки данных. Его нельзя открыть.	Программа пользователя
	Исходный текст (напр., исходный текст на AWL)	Исходная программа в текстовой форме	Исходные тексты (SO)
	C-программа	C-программа, C++-программа, DOS/Windows-программа	M7-программа
	План	Графический исходный текст CFC	Планы (PL)

Рис.3-5. Объекты как носители функций и свойств

В случае приведенных в таблицах объектов (исключая План/Планы) речь идет о объектах базового программного пакета STEP 7. Если Вы установили дополнительное программное обеспечение, то могут появиться другие объекты, которые описаны в документации для соответствующего дополнительного программного обеспечения.

Объекты как носители свойств

Объекты могут нести как функции, так и свойства (например, установки).
После выбора объекта Вы можете

- с помощью команды меню **Bearbeiten → Objekt öffnen**

[Редактировать → Открыть объект] редактировать объект,

- с помощью команды меню **Bearbeiten → Objekteigenschaften**

[Редактировать → Свойства объекта] высветить окно и выполнить в нем специфические для объекта установки.

Носителем свойств может быть также контейнер.



Указание

Если Вы в инструментальной системе изменяете настройки объектов (например, данные параметризации модуля), то они еще не действуют в контроллере, так как для этого в контроллере должны находиться системные блоки данных, в которых запоминаются эти настройки.

Когда Вы загружаете полную прикладную программу, эти системные блоки данных автоматически передаются вместе с ней. Если Вы после загрузки программы производите изменения в настройках, то Вы можете дополнительно загрузить объект "Systemdatenbausteine" ["Системные блоки данных"] для того, чтобы перенести настройки в контроллер.

3.7. Создание и манипулирование объектами

Обзор

Основополагающие действия с объектами одинаковы для всех объектов и не зависят от специальных объектов. Здесь Вы найдете эти принципиальные последовательности действий в итоговом представлении. Это знание предполагается известным в последующих главах данного руководства.

Обычными последовательностями действий при обращении с объектами являются:

- создание объекта,
- выбор объекта,
- выполнение действий с объектом (например, копирование, стирание).

Создание объектов

Посредством команд меню **Datei → Neu → Projekt...** [Файл → Новый → Проект] и **Datei → Neu → Bibliothek...** [Файл → Новый → Библиотека] Вы создаете проекты и библиотеки. Эти объекты являются точками входа в диалог для выбора объектов (раздел 3.8). Они сами могут опять содержать объекты (подобъекты).

Все дополнительные объекты, если они не создаются автоматически, Вы можете создавать в меню **Einfügen** [Вставка]. Исключением являются модули SIMATIC-станции, которые создаются в рамках конфигурирования аппаратных средств.

Открытие объектов

Есть несколько возможностей для открытия уже созданного объекта:

- Двойной щелчок мыши на символе объекта
- Выбор объекта и команда меню **Bearbeiten → Objekt öffnen** [Редактировать → Открыть объект]

После открытия объекта Вы можете создавать или изменять его содержимое. При этом нужно различать

- контейнеры, т.е. объекты, которые сами могут содержать подобъекты (такой как, например, объект "Папка" в Explorer из Windows 95, в котором, в свою очередь, могут располагаться подкаталоги и файлы) и
- объекты, которые не содержат других подобъектов (такой как, например, объект "Файл" в Explorer из Windows 95).

Если Вы открываете объект последнего типа, то его содержимое посредством подходящего компонента программного обеспечения представляется для обработки в новом окне.

Построение иерархии объектов

После открытия контейнера на экране отображаются уже содержащиеся в нем объекты. Теперь Вы можете через меню **Einfügen** [Вставка]

создавать другие подобъекты, например, дополнительные станции в проекте. В этом меню можно вызывать команды для вставки только

таких

объектов, которые в допустимы в текущем контейнере.

Установка свойств объектов

Свойствами объектами являются данные объекта, которые определяют его поведение. Диалог для установки свойств объекта появляется автоматически, когда Вы создаете новый объект и при этом должны устанавливаться свойства. Однако эти свойства можно изменить в последующем:

С помощью команды меню **Bearbeiten → Objekteigenschaften** [Редактировать → Свойства объекта] вызывается диалоговое окно, в котором можно читать или устанавливать свойства выбранного объекта.

Вырезание, копирование,

Большинство объектов Вы можете вырезать так же, как обычно вырезают в Windows 95. Соответствующие команды меню находятся в меню **вставка Bearbeiten** [Редактировать].

Вы можете также передвигать и копировать объекты посредством буксировки (Drag&Drop). Если Вы укажете на недопустимый объект, то в качестве курсора отобразится знак запрета.

Печать

Вначале откройте объект, чтобы отобразить его содержимое. Команда для печати находится в первом меню окна (например, в меню “Файл”). Эта команда высвечивает диалоговое окно, в котором Вы можете установить принтер, печатаемую область и количество подлежащих печати копий.

Некоторые диалоговые окна предоставляют возможность распечатывать части его содержимого. В таких окнах находится кнопка ”Drucken” [“Печать”]. В таком случае щелкните мышью по этой кнопке для того, чтобы запустить печать.

Переименование объектов

Вы можете изменить имя объекта непосредственно или через свойства объекта.

- Непосредственно :

Если дважды медленно щелкнете мышью по имени маркированного объекта, то вокруг текста появится рамка. Вы можете теперь изменить имя через клавиатуру.

- Через свойства объекта:

для этого выберите желаемый объект и введите команду меню **Bearbeiten → Objekteigenschaften** [Редактировать → Свойства объекта]. Измените имя в диалоговом окне. После завершения диалога по свойствам объект переименовывается и отображается с новым именем.

Следующие контейнеры / объекты невозможно переименовать:

- таблица символов (SY)
- прикладная программа offline/online (<AP-off>, <AP-on>).

3.8. Выбор объектов в диалоге

Обзор

Выбор объектов в диалоговом окне (броузер) является действием, которое Вы должны снова и снова выполнять при различных рабочих шагах.

Вызов броузера

Диалоговое окно (броузер) вызывается, например, при конфигурировании аппаратных средств посредством такой команды меню, как Station→ Neu.../Öffnen.../Speichern unter... [Станция→Новая.../Открытие.../Сохранить как...] (Исключение образует окно входа "SIMATIC Manager").

Структура диалогового окна

В диалоговом окне Вы имеете возможности выбора, показанные на рисунке 3–6.

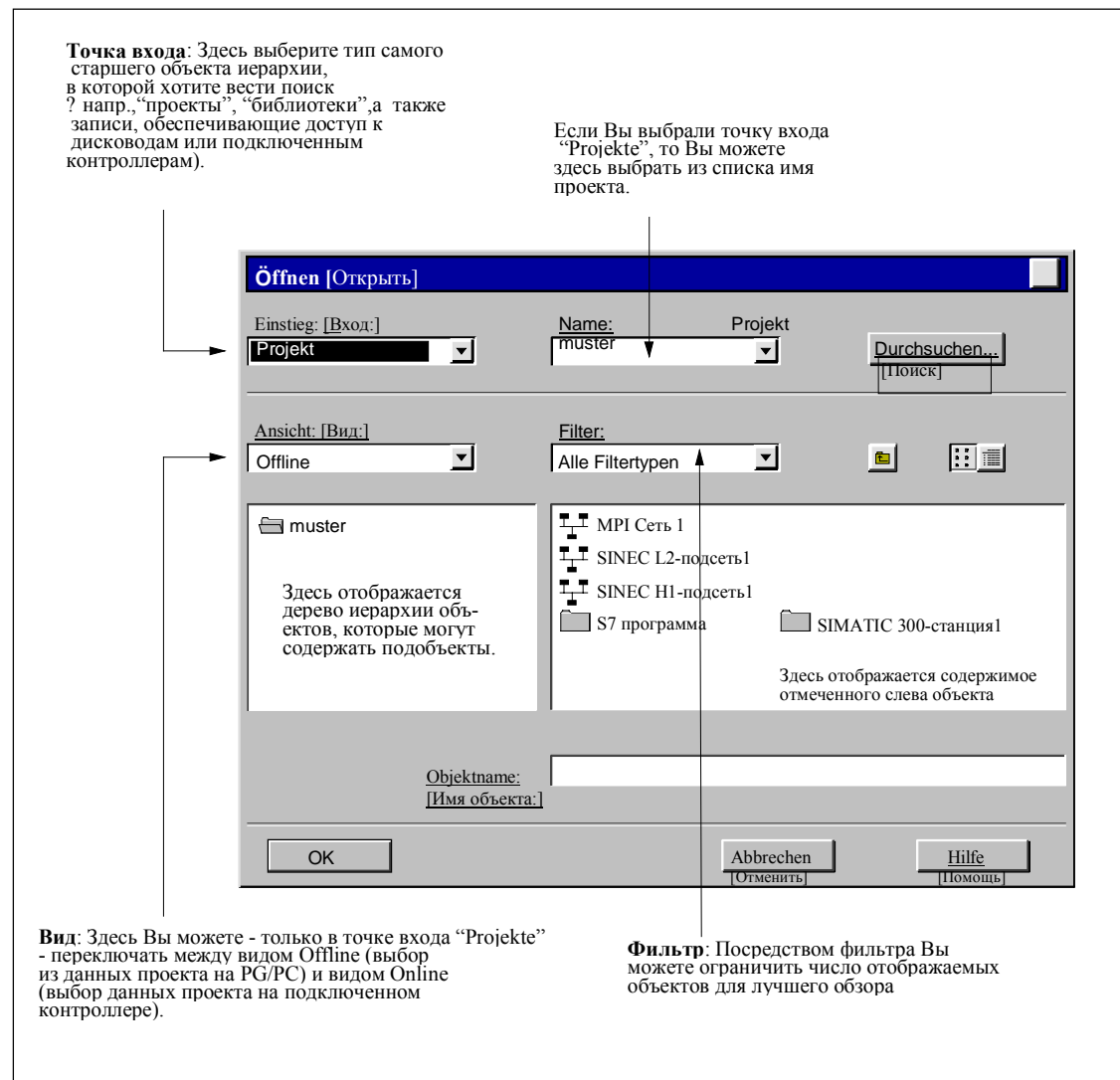


Рис. 3-6. Блок диалога для выбора объектов

Обзор

Различные возможности выбора для разработки Вашей программы описаны в таблице 4-1. В случае отдельных возможностей должны выполняться определенные предпосылки. Другие решения Вы можете принимать по собственному усмотрению, например, то, на каком языке программирования Вы хотите создавать Вашу прикладную программу.

Возможности выбора при разработке программ для S7

Возможности выбора при разработке программ для S7 сведены в таблицу 4-1.

Таблица 4–1. Возможности выбора при разработке Вашей S7–программы

Возможность выбора	Описание
Выберите структуру проекта: <ul style="list-style-type: none"> Использование поставляемого в комплекте примерного проекта и его программная структура Создание нового проекта с новой структурой программы 	STEP 7 предоставляет Вам шаблонный проект и шаблонную программу. Однако Вы можете создавать также новые проекты и новые программы. Вы можете давать этим проектам и программам легко узнаваемые имена.
Выберите язык программирования: <ul style="list-style-type: none"> Контактный план (KOP) Список команд (AWL) Другие языки программирования, имеющиеся в распоряжении в качестве дополнительного программного обеспечения 	Выберите язык программирования, который лучше всего удовлетворяет требованиям Вашего проекта. Более подробную информацию по этому вопросу возьмите из справочников по отдельным языкам программирования (смотрите список литературы в приложении).
Пример: Если Вы программируете на AWL, то выберите способ ввода: <ul style="list-style-type: none"> Создание блока в инкрементном режиме Создание блока в виде текстового файла (исходный текст на AWL) 	Вы можете вводить команды AWL прямо в блок, причем после ввода каждой команды проверяется синтаксис. Однако Вы можете вводить команды AWL также и в текстовый файл. Тогда синтаксис команд проверяется при компиляции этого файла.
Выберите способ адресации: <ul style="list-style-type: none"> Абсолютная адресация Символическая адресация 	Есть два вида символов: <ul style="list-style-type: none"> Глобальные символы, которые используются всеми блоками Вашей программы. Эти символы организуются в таблицу символов. Символы, которые используются только внутри определенного блока (“локальные для блока” символы). Эти символы создаются в описании переменных того блока, в котором они используются.

Таблица 4–1. Возможности выбора при разработке Вашей S7–программы (продолжение)

Возможность выбора	Описание
Выберите вид параметризации: <ul style="list-style-type: none"> • Прием предварительно установленных параметров • Задание параметров, обусловленных процессом 	Более подробную информацию по конфигурированию аппаратных средств и параметризации модулей Вы получите в главе 7. О том, какие параметры Вы можете задавать, справьтесь в Online-помощи или в руководствах по соответствующим аппаратным средствам.
Определите требования к связи для проекта: <ul style="list-style-type: none"> • Нет связи 	В проектных решениях с одним CPU тема связи не требует внимания.
<ul style="list-style-type: none"> • Связь через глобальные данные (несколько CPU) 	Если Вы хотите организовать обмен данными между CPU Вашего проекта, то Вы должны завести таблицу глобальных данных. В этой таблице глобальным данным, которые должны передаваться или приниматься CPU, должны сопоставляться адреса. Для того, чтобы CPU могли обмениваться глобальными данными, все программы должны размещаться в одном проекте.
<ul style="list-style-type: none"> • Связь через функциональные блоки (несколько CPU) 	В сети соединения можно удобно определять и параметризовать и использовать при программировании с коммуникационными функциональными блоками.

Путеводитель по S7 При программировании не требуется определенная последовательность. Однако есть несколько основополагающих задач, которые выполняются в большинстве проектов. Таблица 4-2 приводит эти общие задачи для разработки S7– и M7–программ и при этом указывает соответствующие главы.

Таблица 4–2. Принципиальный образ действий и путеводитель

Деятельность	смотрите для этого
Составление проектов (Использование проектов, которые разрабатывались с помощью STEP 7–версия 1)	Глава 5 (Приложение А)
Определение символов	Глава 6
Конфигурирование структуры аппаратных средств и параметризация модулей	Глава 7
Конфигурирование связи	Главы 8 – 10
Ввод программы	Глава 11 Руководства по языковым пакетам
Создание и использование справочных данных	Глава 12
Загрузка программ в контроллер	Глава 13
Тестирование программы	Глава 14, Специфические для языков тесты описаны в руководствах по языковым пакетам.
Контроль работы	Глава 15

Возможности выбора при разработке программ для М7

Возможности выбора при разработке программ для М7 сведены в таблицу 4-3.

Таблица 4_3 Возможности выбора при разработке Вашей М7–программы

Возможность выбора	Описание
Выберите структуру проекта: <ul style="list-style-type: none"> Использование поставляемого в комплекте примерного проекта и его программной структуры Разработка нового проекта с новой структурой программы 	STEP 7 предоставляет Вам шаблонный проект и шаблонную программу. Однако Вы можете создавать также новые проекты и новые программы. Вы можете давать этим проектам и программам легко узнаваемые имена.
Выберите язык программирования: <ul style="list-style-type: none"> C/C++ CFC (Continuous Function Chart) Эти языки программирования имеются в распоряжении как дополнительное программное обеспечение	Выберите язык программирования, который лучше всего удовлетворяет требованиям Вашего проекта. Более подробную информацию по этому вопросу возьмите из руководств по языковым пакетам.
Выберите способ адресации: <ul style="list-style-type: none"> Абсолютная адресация Символическая адресация 	Для символической адресации Вы получаете поддержку как в STEP 7, так и в дополнительном программном обеспечении ProC/C++.
Выберите вид параметризации: <ul style="list-style-type: none"> Прием предварительно установленных параметров Задание параметров, обусловленных процессом 	Более подробную информацию по конфигурированию аппаратных средств и параметризации модулей Вы получите в главе 7. О том, какие Вы можете задавать параметры, а также предварительные установки, справьтесь в Online-помощи или в руководствах по соответствующим аппаратным средствам.

Путеводитель по М7 При программировании не требуется определенная последовательность. Однако есть несколько основополагающих задач, которые выполняются в большинстве проектов. Таблица 4-4 приводит эти общие задачи и при этом указывает соответствующие главы.

Таблица 4–4. Принципиальный образ действий и путеводитель

Деятельность	смотрите для этого
Составление проектов	Глава 5
Определение символов	Глава 6
Конфигурирование аппаратных средств и параметризация модулей	Глава 7
Конфигурирование связи	Главы 8, 9, 10
Разработка и тестирование программы	Руководства по языковым пакетам
Выбор и перенос операционной системы, загрузка программ	Главы 16, 17
Контроль работы	Глава 15

Обзор

Проекты представляют собой совокупность всех данных и программ решения задачи автоматизации. Они служат для того, чтобы упорядоченно хранить данные и программы, появляющиеся при разработке решения задачи автоматизации. Сосредоточенные в проекте данные охватывают в частности

- конфигурационные данные о структуре аппаратных средств и данные параметризации для модулей,
- данные проектирования для связи через сети,
- программы для программируемых модулей.

Основными задачами при разработке проекта являются, таким образом, подготовка этих данных, а также разработка программы. STEP 7 не предписывает Вам строгой последовательности разработки. Вы можете начинать с любой частной задачи.

Данная глава описывает первые шаги, которые Вы должны выполнить для того, чтобы иметь возможность составлять проекты и после этого вводить программы.

Обзор главы

В раз-деле	Вы найдете	на стр.
5.1	Компоненты и структура проекта	5–2
5.2	Компоненты программ S7/M7	5–3
5.3	Создание проектов	5–4
5.4	Открытие проекта	5–6
5.5	Включение аппаратных компонентов и конфигурирование структуры	5–7
5.6	Включение S7–и M7–программ	5–8
5.7	Включение компонентов для S7/M7–программ	5–9
5.8	Доступ к контроллерам в пределах проекта	5–11
5.9	Представление проекта в инструментальной системе (offline) / в контроллере (online)	5–12
5.10	Доступ к контроллеру без управления проектом	5–14

5.1. Компоненты и структура проекта

Компоненты

Структура проекта представляется в соответствующем окне проекта. Рисунок 5-1 показывает компоненты, которые могут встретиться в проекте. Окно разделено на две части. В левой половине отображается иерархия объектов, из которых состоит проект. В правой половине отображаются те объекты, которые содержатся в объекте, открытом слева.

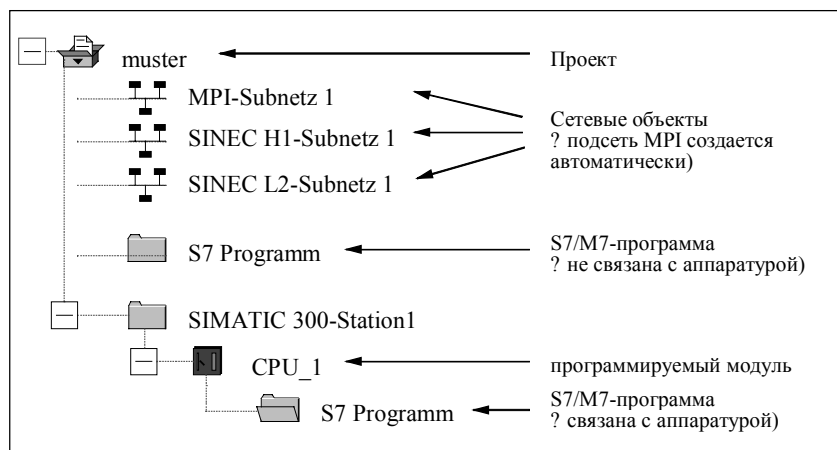


Рис. 5-1. Возможные компоненты проекта

Проект

На первом уровне находится символ проекта, который может использоваться для копирования (с помощью Speichern unter ... [Сохранить как...]), стирания или отображения свойств проекта.

Станция

Вновь созданные станции (например, станция S7–300, станция S7–400) содержат только объект "SC" (конфигурация станции). Он служит для запуска конфигуратора аппаратных средств. После конфигурирования, кроме того, содержатся созданные при этом программируемые модули.

S7-/M7-программы

Объекты представляют собой контейнеры для программных средств и служат для входа в разработку программного обеспечения.

Сеть

Символы для сети представляют информацию о сети указанного типа и служат для входа в проектирование сети. Ненужные символы Вы можете стереть и в случае потребности снова создать с помощью команды меню **Einfügen → Subnetz** [Вставка → Подсеть].

5.2. Компоненты программ S7/M7

Обзор

S7-/M7-программа содержит компоненты, которые создаются автоматически, и компоненты, которые Вы сами должны добавлять в соответствии с используемым Вами языком программирования (рисунки 5-2 и 5-3). В частности, S7-блоки нужны Вам для программирования на языке AWL или KOP. Контейнер для источников (SO) нужен Вам для программирования на языках программирования, которые имеются в распоряжении в качестве дополнительного программного обеспечения, а также для файлов с AWL-текстами. Контейнер “Планы” (PL) с объектами для CFC-планов (Continuous Function Chart) необходим в случае дополнительного программного обеспечения CFC.

S7-программа

Рисунок 5-2 показывает возможную структуру S7-программы.

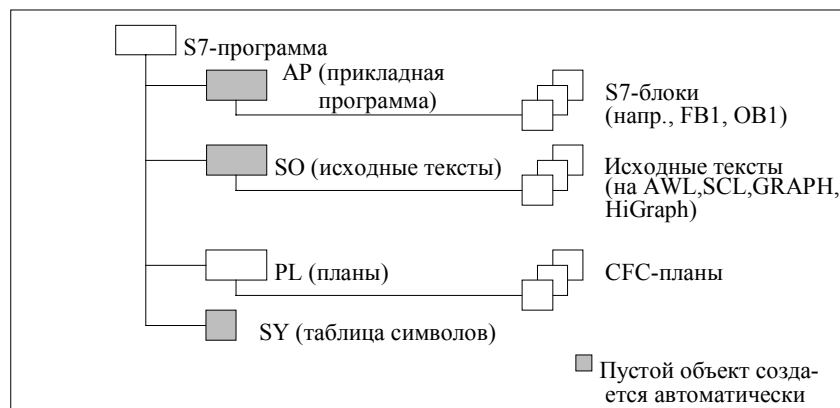


Рис. 5-2. Возможные компоненты S7-программы

Контейнеры “Anwenderprogramm” [“Прикладная программа”] (для блоков), “Quellen” [“Исходные тексты”] и “Pläne” [“Планы”] могут встречаться в S7-программе только один раз. Ненужные контейнеры Вы можете стирать и в случае потребности вновь добавлять.

M7-программа

Рисунок 5-3 показывает возможную структуру M7-программы.

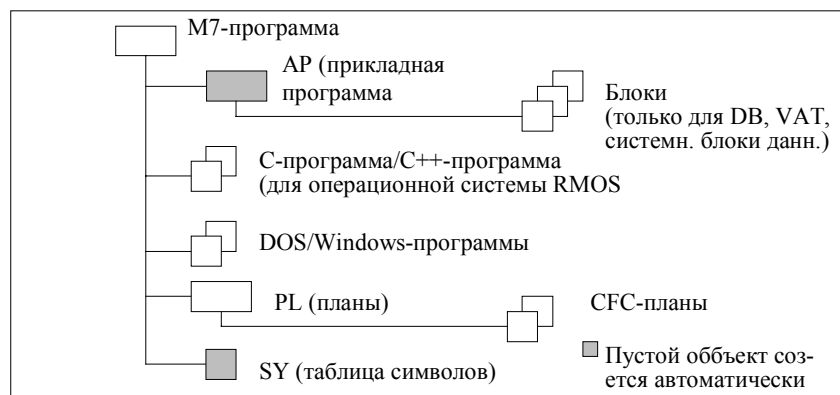


Рис. 5-3. Возможные компоненты M7-программы

5.3. Создание проектов

Шаблонный проект

Вы можете упростить разработку нового проекта, подгоняя содержащийся в комплекте поставки шаблонный проект к Вашим требованиям.

Для этого поступайте следующим образом:

1. Откройте шаблонный проект (см. раздел 5.4).
2. Сохраните в памяти шаблонный проект под другим именем (команда меню **Datei → Speichern unter...** [Файл → Сохранить как...]). После этого шаблонный проект автоматически закрывается и открывается копия.
3. Назовите подобъекты в копии так, чтобы имена Вам о чем-то говорили.
4. Откройте объекты копии и подгоните их содержимое.

Это очень удобная возможность для того, чтобы составлять структуру проекта. Однако и в этом случае Вам следует прочитать остаток данной главы, чтобы познакомиться с отдельными компонентами проекта, так чтобы Вы могли подгонять компоненты шаблонного проекта.

Новый проект

Если шаблонный проект не годится для Вас как исходный пункт, то для того, чтобы построить структуру проекта, поступите следующим образом:

1. Создайте в SIMATIC Manager [Администратор SIMATIC] новый проект и откройте его (команда меню **Datei → Neu → Projekt...** [Файл → Новый → Проект]).
2. Вставьте в проект станцию и откройте ее (например, с помощью **Einfügen → Hardware → SIMATIC 300-Station** [Вставка → Аппаратные средства → станция SIMATIC 300]).
3. Щелкните мышью в окне проекта на знаке “+” перед символом проекта, если под ним еще не отображается станция.
4. Установите структуру станции. Окно для конфигурирования появляется, когда Вы отмечаете станцию и применяете команду меню **Bearbeiten → Objekt öffnen** [Редактировать → Открыть объект]. В частности, Вы должны создать CPU. Дополнительную информацию по этому поводу Вы найдете в части 2 данного руководства.
5. Щелкните мышью в окне проекта на знаке “+” перед символом станции, если под ним еще не отображается символ CPU.
6. Щелкните мышью в окне проекта на знаке “+” перед символом CPU, если под ним еще не отображается S7/M7-программа.

S7/M7-программа служит в качестве контейнера для программного обеспечения и автоматически создается для каждого CPU, который предусматривается при конфигурировании структуры (шаг 4).



Указание

С помощью команды меню **Einfügen → Programm → S7-Programm** [Вставка → Программа → S7-программа] Вы можете без предварительного конфигурирования аппаратных средств создать непосредственно под проектом S7-программу (аналогично для M7-программы). Тогда позже Вы можете эту программу поставить в соответствие CPU.

7. Откройте S7- или M7-программу.
8. Откройте таблицу символов в S7- или M7-программе и определите символы. (Этот шаг можно выполнять также в более поздний момент времени). Дополнительную информацию по этому поводу Вы найдете в главе 6.
9. Выберите контейнер "<AP-off>" (прикладная программа), если Вы хотите разрабатывать блоки, или контейнер "SO" (исходные тексты), если Вы хотите разрабатывать исходный текст программы.
10. Вставьте блок или исходный текст. Команды меню для этого:
 - **Einfügen → S7-Baustein...** [Вставить → S7-блок...],
 - **Einfügen → S7-Software...** [Вставить → программное обеспечение S7],
 - **Einfügen → M7-Software...** [Вставить → программное обеспечение M7].
11. Откройте блок или исходный текст и введите программу. Информацию по этому поводу Вы найдете в руководствах по программированию.
12. Откройте таблицу соединений CPU и установите соединения. (Этот шаг можно выполнять также в более поздний момент времени). Дополнительную информацию по этому поводу Вы найдете в части 2 данного руководства.
13. Документируйте проект (удобный пакет для документирования находится на стадии подготовки). (Команда меню **Einfügen → Projektdokumentation** [Вставить → Проектная документация]).

В зависимости от приложения, не все пункты требуются при разработке.

Указание



С помощью команды меню **Extras → Einstellungen** [Дополнительные → Установки] Вы можете вызвать регистровый диалог. На вкладке "SIMATIC Manager" ["Администратор SIMATIC"] Вы можете установить то, что при открытии проекта должна создаваться резервная копия. Для этого отметьте запись "archivieren bei Projekt öffnen" ["Архивация при открытии проекта"]..

5.4. Открытие проекта

Открытие проекта, отображение содержимого

Для того, чтобы выбрать и открыть существующий проект, задайте вначале команду меню **Datei → Öffnen → Projekt...** [Файл → Открыть → Проект]. Потом выберите проект в ходе последующего диалога. После этого открывается окно проекта.

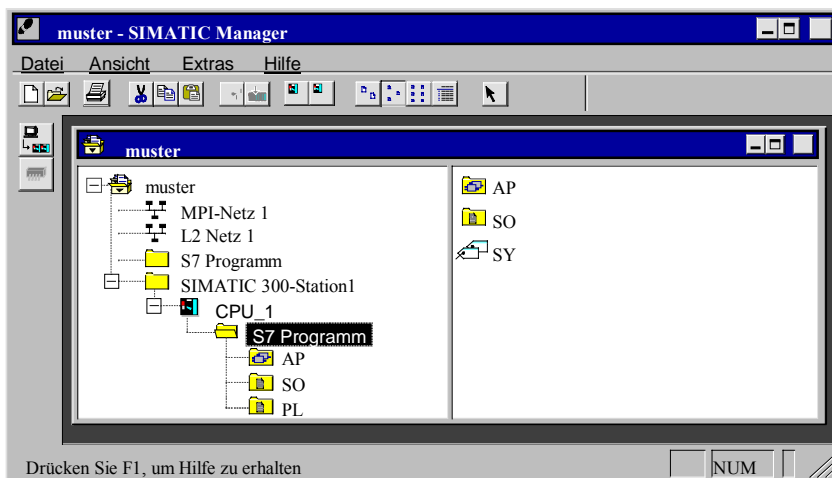


Рис. 5-4. Окно проекта (пример)

Альтернативы для дальнейшей работы

При выборе последовательности разработки проекта Вы имеете максимально возможную гибкость. После создания проекта Вы имеете возможность вести дальнейшую работу с ориентацией на аппаратные средства или с ориентацией на программное обеспечение.

- Продолжение работы с ориентацией на аппаратные средства:

Вначале Вы конфигурируете Ваши аппаратные средства и после этого разрабатываете программное обеспечение для программируемых модулей. Для этого в Вашем проекте Вам нужен компонент аппаратных средств, например, станция SIMATIC 300 или станция SIMATIC 400 (раздел 5.5). Для каждого конфигурируемого CPU станции автоматически создается контейнер для программного обеспечения (S7–программа или M7–программа).

- Продолжение работы с ориентацией на программное обеспечение:

Если Вы хотите вначале разрабатывать программное обеспечение, то Вы должны сами вставить в Ваш проект контейнер для программного обеспечения (S7/M7–программы). Потом Вы можете разрабатывать программное обеспечение также без предшествующего конфигурирования аппаратных средств и выполнить конфигурирование в более поздний момент времени. Для ввода программ нет нужды в том, чтобы структура аппаратных средств была твердо установленной. Вы решаете лишь, должна ли идти речь о программах для аппаратных средств S7 или для аппаратных средств M7.

После того, как Вы конфигурировали аппаратные средства, Вы можете M7–или S7–программу сопоставить CPU (например, посредством Drag&Drop). В разделе 5.9 описано то, как Вы можете тестировать программы, также не используя конфигурирование аппаратных средств.

Проекты версии 1

Информацию по данному вопросу Вы найдете в приложении.

5.5. Включение аппаратных компонентов и конфигурирование структуры

Что такое компонент аппаратных средств?

В проекте объекты для компонентов аппаратных средств представляют структуру аппаратных средств и соответствующие данные по конфигурированию и параметризации отдельных модулей.

Вставка компонента аппаратных средств Для того, чтобы в проекте создать новый компонент аппаратных средств, откройте проект так, чтобы отображалось окно проекта (если еще не появилось). Потом создайте объект для желаемого аппаратного средства посредством команды меню Einfügen → Hardware [Вставка → Аппаратные средства]. В последующем меню Вы можете выбрать:

- станция SIMATIC 300
- станция SIMATIC 400
- чужие станции, т.е. не SIMATIC S7/M7 (функция на стадии подготовки).

Символ для аппаратного средства и соответствующее имя отображаются в окне проекта.

Выполнение конфигурирования

Когда Вы отмечаете станцию и применяете команду меню **Bearbeiten → Objekt öffnen** [Редактировать → Открыть объект], появляется окно "Hardware konfigurieren" ["Конфигурирование аппаратных средств"]. В нем Вы можете планировать структуру аппаратного средства. Для этого в Вашем распоряжении имеется каталог модулей, который Вы можете высветить с помощью команды меню **Ansicht → Katalog** [Вид → Каталог].

Вначале выберите из каталога модулей носитель модулей (Rack). После этого Вы можете подбирать модули и помещать их в слоты носителя.

Конфигурирование структуры аппаратных средств описано в части 2 данного руководства.

Результат конфигурирования

Для каждого программируемого модуля, который был создан при конфигурировании, автоматически создается S7– или M7–программа и таблица соединений.

Если эти объекты еще не видны в окне проекта, то щелкните мышью на знаке "+" перед символом станции для того, чтобы отобразить модуль, и на квадратике перед модулем для того, чтобы отобразить S7/M7–программу. Таблица соединений становится видимой, когда Вы отмечаете модуль.

Создание таблицы соединений

Таблица соединений (пустая) автоматически создается для каждого программируемого модуля. Она не может копироваться, стираться или переименовываться. После открытия высвечивается окно и в нем таблица для определения соединений (определение соединений смотрите в части 2 данного руководства).

5.6. Включение S7–и M7–программ

Обзор

Объекты "S7–программа" и "M7–программа" служат в качестве контейнеров для программного обеспечения программируемых модулей из SIMATIC S7– или SIMATIC M7–спектра.

Возможные способы действий

- использовать, если они автоматически созданы для уже сконфигурированных аппаратных средств, или
- разрабатывать, независимо от специальной конфигурации аппаратных средств, и позже после завершения конфигурирования аппаратных средств ставить в соответствие специальному компоненту аппаратных средств. Вы должны задавать только лишь тип аппаратных средств (S7 или M7).

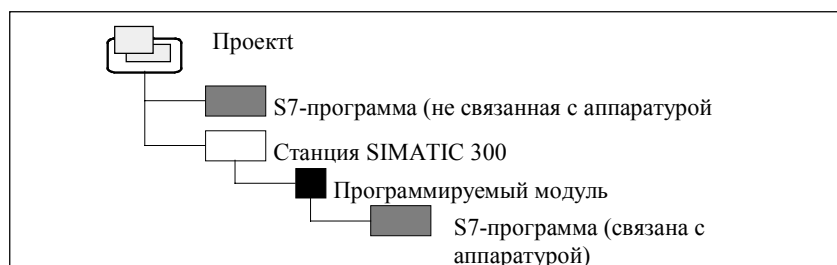


Рис. 5-5. S7-программа, связанная и не связанная с аппаратурой в окне проекта

Использование программы для программируемого модуля

Для каждого создаваемого при конфигурировании программируемого модуля автоматически создается S7– или M7–программа. Щелкните мышью в окне проекта на знаке "+" перед символом программируемого модуля, если под ним еще не отображается S7/M7–программа.

Включение программы независимо от аппаратных средств

Проект должен быть создан и окно проекта должно быть открыто.

1. Отметьте в окне проекта символ проекта.
2. Вызовите команду меню **Einfügen → S7–Programm** [Включить → S7–программа] или **Einfügen → M7–Programm** [Включить → M7–программа].

S7– или M7–программа создается **внутри проекта**, следовательно, не соответствует конкретному аппаратному средству. На это место Вы можете перемещать также программы, которые Вы хотите сохранить перед стиранием станции.

Сопоставление программы программируемому модулю

Вырежьте S7- или M7-программу, не поставленную в соответствие никаким аппаратным средствам, или скопируйте ее. Потом отметьте программируемый модуль, которому Вы хотите сопоставить программу, и вставьте ее. При этом уже имеющаяся в нем, может быть, программа переписывается после квитирования сообщения. Альтернативно Вы можете также отметить несопоставленную программу, отбуксировать ее на программируемый модуль и там отпустить.

5.7. Включение компонентов для S7/M7–программ

Уже созданные компоненты

В S7–программе уже созданы:

- таблица символов (SY),
- прикладная программа <AP–off> в качестве контейнера для блоков вместе с первым блоком,
- исходные тексты (SO) в качестве контейнера для исходных текстов программ

В M7–программе уже созданы:

- таблица символов (SY),
- прикладная программа <AP–off>.

S7–блоки

Вы хотите создавать AWL– или KOP–программы. Для этого отметьте уже созданный объект ”Anwenderprogramm” [“Прикладная программа”] и потом щелкните мышью по команде меню **Einfügen → S7–Baustein** [Вставка → S7–блок]. В последующем меню Вы можете выбрать тип блока (например, функциональный блок, функция, организационный блок, блок данных, тип данных, определенный пользователем (UDT), таблица переменных (VAT)).

После открытия (пустого) блока Вы можете вводить AWL– или KOP–программу. Дополнительную информацию по этому поводу Вы найдете в руководствах по AWL /232/ и KOP /233/.



Указание

Объект ”Systemdatenbausteine” [“Системные блоки данных”], который Вы, возможно, найдете в прикладной программе, создается системой и не может открываться или переименовываться. Он служит для того, чтобы

после загрузки программы дать возможность проводить изменения конфигурации и дополнительно загружать их в контроллер.

Использование блоков из стандартных библиотек

Для разработки прикладных программ Вы можете использовать также блоки из содержащихся в комплекте поставки стандартных библиотек. Доступ к библиотекам Вы получите через команду меню **Datei → Öffnen** → Bibliothek... [Файл → Открыть → Библиотека]. Дополнительные указания по использованию стандартных библиотек, а также по созданию собственных библиотек Вы найдете в Online–помощи.

Создание исходных текстов/СFC–планов Вы хотите создавать исходные тексты на определенном языке программирования или СFC–план. Для этого отметьте в S7–программе объект ”SO” (исходные тексты) или ”PL” (планы) и после этого щелкните мышью на команде меню **Einfügen → S7–Software** [Вставить → Программное обеспечение S7]. Выберите в последующем меню исходный текст, подходящий к языку программирования. После открытия пустого исходного текста Вы можете вводить программу.

Разработка программ для M7

Вы хотите разрабатывать программы на языке C или C++ для операционной системы RMOS или программы для DOS/Windows программируемого модуля из M7-спектра. Для этого отметьте M7-программу и после этого щелкните мышью по команде меню **Einfügen → M7-Software** [Вставить → Программное обеспечение M7]. Выберите в последующем меню язык программирования или объект, подходящий для операционной системы. После открытия созданного объекта Вы попадаете в соответствующую среду разработки.

Таблица символов

(Пустая) таблица символов (SY) создается автоматически при разработке S7/M7-программы. После открытия высвечивается окно "Symbole definieren" ["Определение символов"] и в нем отображается таблица символов (смотрите главу 6).

Переименование объектов в S7/M7-программах

В отличие от контейнерного объекта SO ("Исходные тексты"), в S7/M7-программе не могут переименовываться следующие объекты:

- таблица символов (SY)
- контейнеры "Прикладная программа offline/online" (<AP-off>, <AP-on>)
- "Системные блоки данных" (SDB)

Вставка внешних исходных текстов

Вы можете создавать и разрабатывать файлы с исходными текстами с помощью любых ASCII-редакторов. После этого Вы можете эти файлы импортировать в проект и преобразовать в отдельные блоки. Для этого выберите контейнер "Исходные тексты", в который должен импортироваться файл с исходным текстом. Затем выберите команду меню **Einfügen → Externe Objekt** [Вставка → Внешний объект] и задайте файл с исходным текстом в появившемся диалоговом окне.

Возникающие при компиляции импортированного источника блоки сохраняются в контейнере "Прикладная программа".

5.8. Доступ к контроллерам в пределах проекта

Обзор	В пределах проекта Вы можете путем выбора вида (Offline/Online) переключаться между проектными данными в Вашей инструментальной системе и проектными данными в контроллере.
Действия в режиме Offline	Режим Offline используется для составления структуры проекта, а также для разработки и выбора объектов для всех видов проектных данных.
Действия в режиме Online	Режим Online используется для доступа к контроллеру. В частности, Вы можете выполнять функции, имеющиеся в распоряжении в меню "Zielsystem" ["Контроллер"] (например, Urlauben..., Betriebszustand..., Uhrzeit stellen..., Baugruppenzustand... [Полное стирание..., Рабочий режим..., Установка часов..., Состояние модулей...]) и получить справку о программном обеспечении, загруженном в контроллер.
Переключение	Окно проекта с режимом Online можно высветить с помощью команды меню Ansicht → Online [Вид → Online].
Вид окна проекта в режиме Offline	Это настройка для первичной разработки проекта. В окне проекта отображаются набор данных и программы проекта в инструментальной системе (PG/PC) (рис. 5–6).
Вид окна проекта в режиме Online	Окна проекта, которые показывают вид Online, помечаются в заголовке добавкой "<Online>". В режиме Online в окне проекта Вы видите программное обеспечение, которое резидентно загружено в CPU подключенного контроллера (рисунок 5-7). Содержимое объекта "S7–программа" или "M7–программа" устанавливается в контроллере в зависимости от загруженного программного обеспечения.

5.9. Представление проекта в инструментальной системе (offline)/ в контроллере (online)

Режим offline

S7– или M7–программы в режиме Offline могут содержать все компоненты, описанные в разделе 5.2.

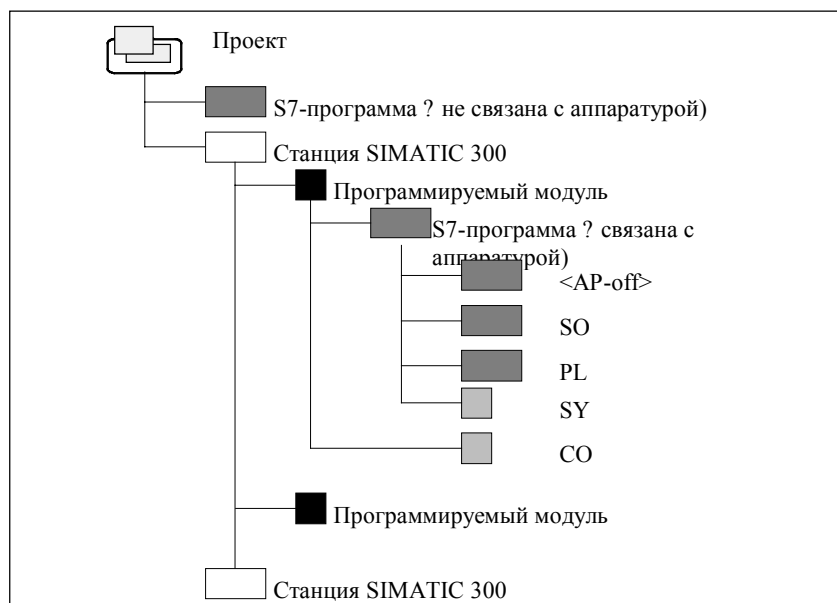


Рис. 5-6 Структура, представленная в окне проекта offline

Обзор online

В S7–программе видна только одна прикладная программа, в которой находятся загруженные блоки, в M7–программе Вы видите загруженные блоки. Контейнеры для источников и планов, таблица символов и таблица соединений в режиме Online не отображаются, так как они не находятся в контроллере.

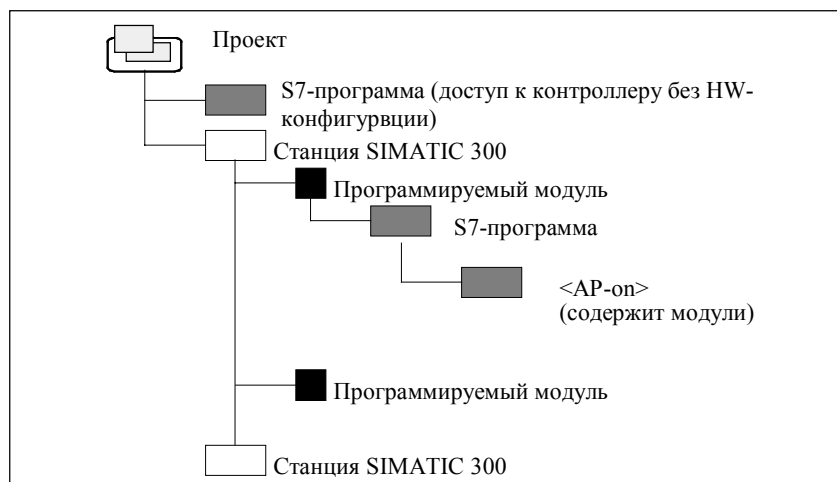


Рис.5-7. Структура, представленная в окне проекта online

Доступ без конфигурирования аппаратных средств

Если Вы в режиме offline создаете S7/M7-программу, которая не сопоставлена аппаратным средствам (т.е. она находится непосредственно под проектом), то online также создается соответствующий объект. Вы можете использовать этот объект в режиме Online для того, чтобы загружать программное обеспечение в контроллер, не имея необходимости конфигурировать перед этим аппаратные средства.

Исходя из Offline-окна проекта, поступайте следующим образом:

1. С помощью команды меню **Fenster → Neues Fenster** [Окно → Новое окно] выделите второе окно проекта.
2. С помощью команды меню **Ansicht → Online** [Вид → Online] переключите новое окно в Online.
3. Отметьте S7/M7-программу в Online и с помощью команды меню **Bearbeiten → Objekteigenschaften** [Редактировать → Свойства объекта] вызовите соответствующий диалог.
4. Установите на вкладке "Speziell" ["Специальный"] диалогового окна "Eigenschaften" ["Свойства"] адрес абонента контроллера, по которому должна загружаться программа.

Объекты, которые Вы копируете в Online-программу, загружаются в контроллер.

Нестираемые объекты

Следующие объекты в контроллере представляются в режиме Online, но не могут в нем стираться.

- системные функции (SFC)
- системные функциональные блоки (SFB)
- системные блоки данных.

5.10. Доступ к контроллеру без управления проектом

Обзор

В пределах проекта Вы можете в Online-окне проекта получить доступ к подключенным контроллерам. STEP 7 предоставляет Вам возможность работать непосредственно в подключенном контроллере без управления проектом. Такая возможность предусмотрена для целей ввода в эксплуатацию и технического обслуживания.

Предпосылка

Между PG и контроллером должна быть создана связь.

Отображение подключенных контроллеров

Щелкните мышью по верхней кнопке панели интерфейсов Администратора SIMATIC [SIMATIC Manager] для того, чтобы открыть окно "Erreichbare Teilnehmer" ["Доступные абоненты"] (сравн. рисунок 5- 8). Если панель интерфейсов не видна, то Вы можете высветить ее с помощью команды меню **Ansicht** → **Schnittstellenleiste** [Вид → Панель интерфейсов]. В окне видны все абоненты, которые STEP 7 может выявить в сети.

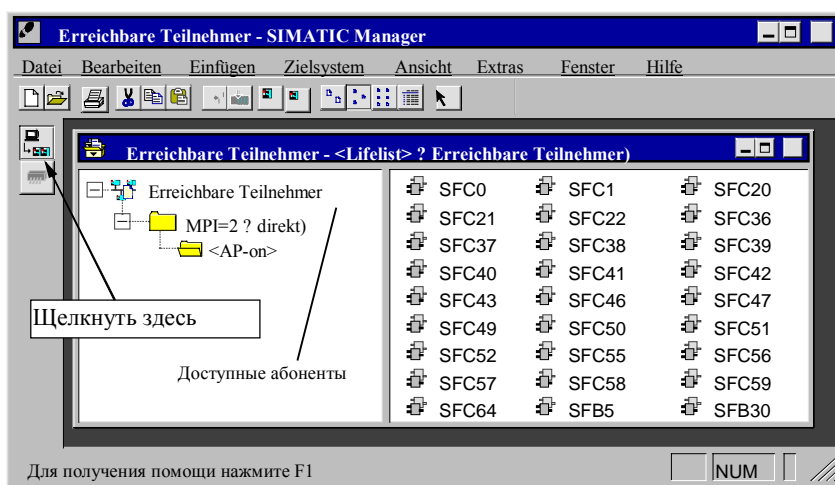


Рис. 5-8. Прямой доступ к целевой системе без управления проектом

Вызываемые функции

Если Вы отметили абонента, то Вы можете выполнить для этого абонента функции, имеющиеся в распоряжении в меню **Zielsystem** [Контроллер] (например, Umlöschen..., Betriebszustand..., Uhrzeit stellen..., Baugruppenzustand... [Общее стирание..., Рабочий режим..., Установка часов..., Состояние модулей...]).

Обработка загруженных

Если Вы дважды щелкнете мышью на абоненте, то отобразится объект "Anwenderprogramm" ["Прикладная программа"]. В прикладной блоках находятся все блоки, которые находятся в контроллере. Вы можете открывать и редактировать эти блоки.

Вы можете сохранять измененные блоки в окне блоков с помощью команды меню **Datei** → **Speichern unter...** [Файл → Сохранить как...] или с помощью **Zielsystem** → **Laden** [Контроллер → Загрузить] снова загружать в контроллер.

Обзор

В программе на языке STEP 7 Вы работаете с такими операндами, как сигналы входов/выходов, меркерами, счетчиками, таймерами, блоками данных и функциональными блоками. Вы можете использовать для этих операндов абсолютные адреса (напр., E 1.1, M 2.0, FB 210, однако читаемость программ значительно возрастает, если Вы для этого будете использовать символы (напр., Motor_A_Ein [включить_двигатель_A] или обозначения в соответствии с принятой в Вашей отрасли системой условных обозначений). Потом в программе к этому операнду можно будет обращаться через этот символ.

Обзор главы

В раз-деле	Вы найдете	на стр.
6.1	Символы	6–2
6.2	Таблица символов	6–3
6.3	Неполные и многозначные символы	6–5
6.4	Работа с таблицей символов	6–6
6.5	Определение отдельных символов в диалоге	6–7
6.6	Экспорт и импорт таблиц символов	6–8

Указание чтения

Вслед за разделами, знакомящими Вас с основными знаниями о **для** символах, познакомьтесь с возможностями определения глобальных символов

1. Символы и соответствующие им адреса можно непосредственно вносить в таблицу символов (раздел 6.4). Этот способ рекомендуется для ввода многих символов и для создания таблицы символов, так как при этом на экран выводятся уже определенные символы, что облегчает сохранение общего представления о них.
2. Можно открыть в окне для ввода программы диалоговое окно и определить в нем новый символ (раздел 6.5). Этот способ пригоден для определения отдельных символов, если, например, Вы в процессе программирования установили, что какого-то символа не хватает или он должен быть скорректирован. Тем самым Вы избавляете себя от вывода на экран таблицы символов.

6.1. Символы

Обзор	Символы позволяют работать вместо адресов с более выразительными обозначениями. Различают локальные и глобальные символы.
Область действия	<p>Глобальный символ известен во всей прикладной программе, т.е. его могут использовать все блоки программы. Он должен иметь одно и то же значение во всей программе.</p> <p>Локальный символ известен только в том блоке, в котором он был определен. В разных блоках для разных целей можно использовать одно и то же обозначение.</p>
Использование глобальных символов	<p>Глобальные символы можно определить для входов, выходов, счетчиков, меркеров и блоков. Допустимы следующие адреса:</p> <ul style="list-style-type: none">• Сигналы входов/выходов (отображение процесса) E, A• Периферийные входы/выходы PE, PA• Меркеры M• Таймеры, счетчики T, Z• Кодовые блоки FB, FC, SFB, SFC, OB• Блоки данных DB• Типы данных, определенные пользователем UDT• Таблица переменных VAT
Использование локальных символов	Локальные символы можно использовать для параметров блока (входных, выходных и проходных), для статических или временных данных блока.
Где определяются символы?	<p>Глобальные символы определяются в таблице символов.</p> <p>Локальные символы определяются при вводе программы при описании переменных блока.</p>
Указание для чтения	Основное внимание в главе уделяется определению глобальных символов. Создание локальных символов описывается в связи с соответствующим языком программирования в руководствах по языковым пакетам.

6.2. Таблица символов

Структура

(Пустая) таблица символов (SY) генерируется автоматически при создании S7– или M7– программы. Ее структура видна из рис. 6–1.

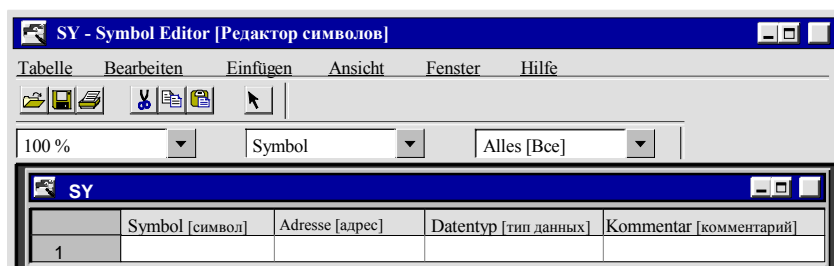


Рис. 6-1. Структура таблицы символов

Символ

Длина наименования символа не может превышать **24** знаков.

Адрес

Адрес - это условное обозначение определенного операнда.

Пример: вход E 12.1

Синтаксис адреса проверяется при вводе. Кроме того, проверяется, может ли адрес соответствовать указанному типу данных.

Тип данных

Имеется возможность выбирать среди различных типов данных, предоставляемых в распоряжение пакетом STEP 7. Это поле предварительно заполнено типом данных по умолчанию, однако его можно изменить. Если это изменение не годится для адреса или неверен синтаксис, то при выходе из поля появляется сообщение об ошибке.

Комментарий

Всем символам можно поставить в соответствие комментарии. Комбинируя краткие символы и подробные комментарии, можно выполнить требования как к хорошей программной документации, так и к эффективной разработке программы. Комментарий может иметь длину до 80 знаков.

Область действия

Таблица символов действительна для того модуля, в котором размещена программа. Если Вы хотите использовать в разных CPU одинаковые символы, то Вы сами должны позаботиться о том, чтобы соответствующие записи в соответствующих таблицах символов совпадали (напр., путем копирования).

Преобразование в переменные языка C

Имеется возможность выбирать символы из таблицы символов M7–программы и преобразовывать их в связи с дополнительным программным обеспечением Pro C/C++ в соответствующие переменные языка

С. Дальнейшую информацию можно получить в соответствующем руководстве пользователя /290/.

6.3. Неполные и многозначные символы

Цель	<p>Благодаря возможности сохранять неполные символы можно, например, сначала определить только имя символа, а адрес внести позднее. Однако, чтобы использовать символ при разработке программного продукта (без сообщения об ошибке), необходимо внести обозначение символа, адрес и тип данных.</p> <p>Благодаря возможности хранить в таблице символов многозначные символы, в этой таблице можно копировать записи.</p> <p>В частности, работу с таблицей символов можно прервать в любой момент времени и сохранить промежуточное состояние.</p>
Возникновение многозначных символов	<p>Многозначные символы возникают, когда в таблицу вводится символ,</p> <ul style="list-style-type: none">• имя (Symbol) и/или• адрес <p>которого уже имеются в описании другого символа. При этом как новый, так и ранее существовавший символы становятся многозначными.</p> <p>Это происходит, например, тогда, когда символ копируется и вносится в таблицу, чтобы потом его можно было легко модифицировать.</p>
Выделение многозначных	<p>В таблице символов многозначные символы выделяются графически (цветом, видом шрифта). Изменение отображения указывает на СИМВОЛОВ необходимость последующей доработки. На экран можно вывести все символы или отфильтровать отображение так, чтобы выводились только однозначные или только многозначные символы.</p>
Устранение многозначности	<p>Многозначный символ становится однозначным при изменении компонента (имени и/или адреса), обусловившего многозначность. Если многозначны два символа, и Вы один из них делаете однозначным, то второй тоже становится однозначным.</p>

6.4. Работа с таблицей символов

Предварительное Данные для таблицы символов можно создавать и с помощью **указание** предпочитаемого Вами табличного редактора (напр., Microsoft Excel), а затем импортировать их в таблицу символов (раздел 6.6).

Открытие таблицы символов Имеется несколько возможностей для открытия таблицы символов (SY):

- двойной щелчок на таблице символов в окне проекта или
- выделение таблицы символов в окне проекта и команда меню **Bearbeiten** [Редактировать] → **Objekt öffnen** [Открыть объект].

Таблица символов для текущей программы отображается в собственном окне. Теперь имеется возможность создавать или изменять символы. При первом открытии после создания таблицы она еще пуста.

Ввод символов Для ввода новых символов в таблицу перейдите в первую пустую строку и заполните ее поля. Новую пустую строку можно вставить перед текущей командой меню **Einfügen** [Вставка] → **Symbol** [Символ]. Имеющиеся записи можно копировать, а затем изменять командой меню **Bearbeiten** [Редактировать]. Затем таблицу следует сохранить и закрыть. Можно также сохранить еще не полностью определенные символы (см. раздел 6.3).

Сортировка символов Записи в таблице символов можно отсортировать в алфавитном порядке по именам символов или адресам.

Изменение сортировки производится выбором критерия сортировки в окне таблицы символов или в диалоговом окне, которое можно вызвать командой меню **Ansicht** [Вид] → **Sortieren...** [Сортировать...]

Фильтрация Из всех записей таблицы с помощью фильтра можно выбрать их **символов** некоторое подмножество.

В окне таблицы символов можно установить, должны ли отображаться

- одно- и многозначные символы,
- только однозначные символы или
- только многозначные символы.

Дополнительны возможности для фильтрации предоставляет диалоговое окно, которое можно вызвать командой меню **Ansicht** [Вид] → **Filtern...** [Фильтровать...]. Здесь можно определить критерии для столбцов "Symbol" и "Adresse", которым должны удовлетворять записи при фильтрации. Отдельные критерии объединяются с помощью UND [И]. Отфильтрованные записи начинаются указанными последовательностями символов.

6.5. Определение отдельных символов в диалоге

Обзор	<p>Описанная здесь последовательность действий показывает, как можно изменять или заново определять символы через диалоговое окно при программировании блоков, не выводя при этом на экран таблицу символов.</p> <p>Эта последовательность действий полезна, когда Вы хотите обработать отдельный символ. Чтобы изменить несколько символов, нужно открыть таблицу символов и работать непосредственно в таблице.</p>
Отображение	<p>При открытом блоке можно в его окне включить или выключить символов отображение символов командой меню Ansicht [Вид] → Symbolische Darstellung [Символическое представление].</p>
Определение символов	<p>Для определения отдельных символов во время программирования действуйте следующим образом:</p> <ol style="list-style-type: none">1. Обеспечьте отображение символов в окне блока (команда меню Ansicht [Вид] → Symbolische Darstellung [Символическое представление]).2. Выделите в разделе команд своей программы абсолютные адреса, которым Вы хотите назначить символическое имя.3. Выберите команду меню Einfügen [Вставить] → Symbol...[Символ...]. Результат: на экран выводится диалоговое окно, уже содержащее выбранные адреса..4. Заполните диалоговое окно и завершите диалог. Результат: определенный символ вносится в таблицу символов. Ввод данных, который привел бы к появлению многозначных символов, отклоняется с сообщением об ошибке.
Редактирование в таблице символов	<p>С помощью кнопки "Symboltabelle" ["Таблица символов"] - исходя из описанного выше диалога - можно открыть таблицу символов для правки. Это имеет преимущество, если необходимо заново внести или исправить более одного символа.</p>

6.6. Экспорт и импорт таблиц символов

Применение

Выведенную на экран таблицу символов можно экспортировать в текстовый файл, для того чтобы, например, в дальнейшем обрабатывать ее любым текстовым редактором.

Таблицы, созданные с помощью других инструментов, можно импортировать в свою таблицу символов для дальнейшей обработки. Функцию импортирования можно, например, использовать, чтобы после конвертирования внести в таблицу символов список соответствия, созданный в STEP5/ST.

Выбирать можно из следующих форматов файлов: *.SDF, *.ASC, *.DIF и *.SEQ.

Экспорт

Имеется возможность экспортировать всю таблицу символов, ее подмножество, ограниченное фильтрованием, или строки, выделенные в представлении таблицы. Для экспорт выведенной на экран таблицы символов в файл формата *.SDF, *.ASC, *.DIF или *.SEQ (список соответствия), действуйте следующим образом:

1. Откройте таблицу символов.
2. Выберите с помощью подходящего фильтра символы, которые Вы хотели бы экспортировать.
3. Выберите пункт меню **Tabelle** [Таблица] → **Exportieren...** [Экспортировать...].
4. Укажите в диалоговом окне "Exportieren" ["Экспортировать"] желаемый формат файла и внесите имя файла, в который Вы хотите экспортировать таблицу символов.
5. Подтвердите ввод щелчком на "Speichern" ["Сохранить"].

Обращение с символами при экспорте

При экспорте принимаются во внимание именно те символы, **многозначными** которые были выбраны с помощью фильтра (напр., все символы или только однозначные или только многозначные символы).

Импорт

Для импорта таблицы символов, представленной в файле формата *.SDF, *.ASC, *.DIF или *.SEQ, действуйте следующим образом:

1. Откройте таблицу символов, в которую Вы хотите импортировать данные.
2. Выберите команду меню **Tabelle** [Таблица] → **Importieren...** [Импортировать...].
3. Укажите в диалоговом окне "Importieren" ["Импортировать"] формат и имя файла, который Вы хотите импортировать.
4. Подтвердите ввод щелчком на "Öffnen" ["Открыть"].

Обращение с многозначными символами при

Если при импорте встречаются многозначные символы, то дальнейший образ действий можно определить в диалоговом окне. Это окно содержит вопрос о том, нужно ли импортировать многозначные **импорте** символы.

- Если щелкнуть на кнопке "Alle" ["Все"], то все многозначные символы будут импортированы без дальнейших запросов.
- Если щелкнуть на кнопке "Ja" ["Да"], то будет импортирован многозначный символ.
- Если щелкнуть на кнопке "Nein" ["Нет"], то многозначный символ при импортировании не будет приниматься во внимание.

В двух последних случаях при появлении следующего многозначного символа вновь выводится на экран такое же диалоговое окно.

Пример: импорт файла EXCEL

Используйте для импорта и экспорта данных из приложения Microsoft EXCEL формат DIF.

При импортировании действуйте следующим образом:

1. Создайте в Excel таблицу с четырьмя столбцами "Symbol" ["Символ"], "Adresse" ["Адрес"], "Datentyp" ["Тип данных"] и "Kommentar" ["Комментарий"] и заполните эту таблицу.
2. Вызовите командой меню **Datei** [Файл] → **Speichern unter...** [Сохранить как...] соответствующее диалоговое окно.
3. Выберите в диалоговом окне тип данных ".dif" (data interchange format [формат обмена данными]).
4. Выберите каталог и имя файла и завершите диалог.
5. Откройте таблицу символов.
6. Вызовите командой меню **Tabelle** [Таблица] → **Importieren** [Импортировать] соответствующее диалоговое окно.
7. Выберите в диалоговом окне заранее созданный *.dif-файл и закройте диалог.

Указания по работе файлами ACCESS

Используйте для импорта и экспорта данных приложения Microsoft **с** ACCESS формат SDF.

- Выберите в ACCESS в качестве формата файла "Text (mit Trennzeichen)" ["Текст (с разделительными знаками)"].
- Используйте в качестве разделительного знака кавычки(" ").
- Используйте в качестве разделительного знака запятую (,).

Конфигурирование и параметризация модулей

7

Обзор

При конфигурировании с помощью STEP 7 Вы определяете, какие модули будут использоваться в Вашей установке, независимо от того, существует эта установка или нет. Последовательность действий при конфигурировании централизованной и децентрализованной периферии одинакова.

Конфигурацию можно как угодно часто копировать в другие проекты STEP 7, при известных условиях модифицировать и загружать в одну или несколько существующих установок. При запуске микроконтроллера CPU сравнивает созданную с помощью STEP 7 конфигурацию с фактической действительной конфигурацией установки. Благодаря этому возможные ошибки немедленно распознаются и сообщаются.

При параметризации устанавливаются свойства модулей. Для этого нет необходимости задействовать какие-либо переключатели на модулях. Параметры просто вводятся с помощью STEP 7. Они загружаются в CPU, а из CPU передаются в соответствующие модули.

Модули можно очень просто заменять, так как параметры, установленные с помощью STEP 7, при запуске автоматически загружаются в новый модуль.

При адресации можно изменять адреса, предоставленные STEP 7. Т.е. Вы определяете, через какие адреса происходит обращение к модулям из прикладной программы.

Обзор главы

В раз-деле	Вы найдете	на стр.
7.1	Создание конфигурации - обзор	7–3
7.2	Конфигурирование централизованной структуры	7–5
7.3	Конфигурирование сети SINEC L2–DP	7–9
7.4	Создание мастер-системы	7–11
7.5	Конфигурирование ведомых модулей DP	7–13
7.6	Параметризация модулей	7–16
7.7	Параметризация сети SINEC L2–DP	7–17
7.8	Параметризация DP-мастера и сети SINEC L2–DP	7–18
7.9	Параметризация ведомых модулей DP	7–18
7.10	Включение ведомого модуля DP в SYNC–/FREEZE–группы	7–19
7.11	Сохранение, загрузка, чтение, изменение и копирование	7–20
7.12	Дополнительная инсталляция модулей в каталоге модулей	7–24

**Какую главу Вам
прочитать?**

В следующей таблице приведены темы глав в рекомендуемой последовательности чтения:

- если Вы хотите конфигурировать централизованную структуру.
- если Вы дополнительно конфигурировать SINEC L2-DP.

Таблица 7-1. Содержание глав		
В этой главе Вы узнаете, как...	Для централизованной структуры читайте...	Для сети SINEC L2-DP читайте ...
конфигурировать с помощью STEP 7 (обзор)	главу 7.1	
конфигурировать модули в соответствии с запланированной структурой**	главу 7.2	главы с 7.3 до 7.5
установить параметры для модулей (напр., у CPU режим запуска и циклической работы, реманентность областей операндов и т.д.)	главу 7.6*	-
установить параметры для сети SINEC L2-DP (параметры для DP-мастера, для ведомого DP, для модулей/идентификаторов и параметры шины)	-	главы с 7.7 до 7.10*
сохранить, загрузить, прочесть, изменить и скопировать конфигурацию	главу 7.11	главу 7.11
затем установить модули в каталоге модулей	главу 7.12	главу 7.12
* Параметризация модуля/сети SINEC L2-DP требуется только тогда, когда необходимо изменение предустановленных параметров.		
** Конфигурирование настоятельно необходимо у станций S7-400 с несколькими CPU или дополнительными носителями модулей. У станций S7-300 конфигурирование необязательно, так как CPU создает отсутствующую заданную конфигурацию из фактической.		

**Первичный
пользователь
SINEC L2-DP**

AS-I (Система автоматизации S7/M7. Децентрализация с

Если Вы хотите конфигурировать сеть SINEC L2-DP и у Вас еще нет опыта в построении децентрализованной периферии, то мы рекомендуем Вам для быстрого вхождения в тему брошюру /21/ *Automatisierungssystem S7/M7, Dezentralisieren mit PROFIBUS-DP und* помощь PROFIBUS-DP и AS-I).

7.1. Создание конфигурации - обзор

Введение

Далее следует обзор того, как сконфигурировать запланированную структуру и параметризовать модули внутри нее.

С помощью STEP 7 можно конфигурировать и параметризовать как модули централизованной структуры, так и модули децентрализованной периферии. Последовательность действий одинакова.

Конфигурирование

В дальнейшем под “конфигурированием” мы понимаем размещение носителей модулей, модулей и интерфейсных модулей в конфигурационной таблице (см. рис. 7–2).

В конфигурационной таблице STEP 7 автоматически присваивает каждому модулю адрес. Адреса модулей станции можно изменять, если CPU станции свободно адресуем (адрес для каждого канала модуля выделяется свободно, независимо от места установки).

Конфигурирование настоятельно необходимо у станций S7–400 с несколькими CPU или дополнительными носителями модулей. У станций S7–300 конфигурирование необязательно, так как CPU создает отсутствующую заданную конфигурацию из фактической.

Параметризация

Под “параметризацией” мы далее понимаем:

- установку параметров у параметризуемых модулей для централизованной структуры и для сети. Пример: CPU - это параметризуемый модуль. Время контроля цикла - это параметр, который можно установить
- установку параметров шины, параметров DP–мастера и ведомых DP (DP–Slaves) для сети SINEC L2–DP.

Параметризация модуля/сети SINEC L2–DP требуется только тогда, когда необходимо изменить предустановленные параметры.

Последовательность действий Для конфигурирования и параметризации структуры действуйте следующим образом:

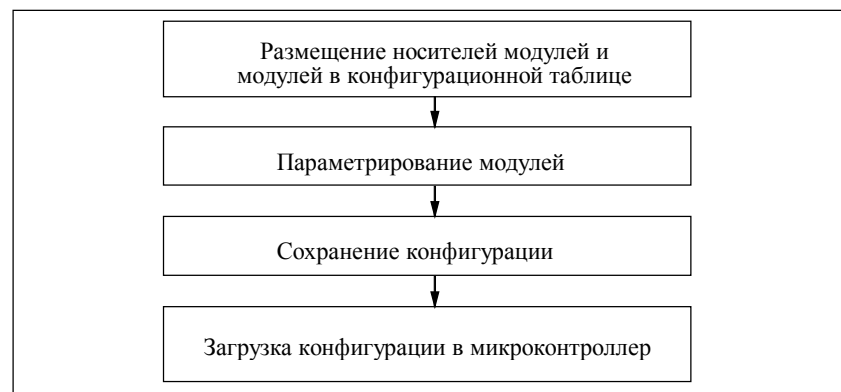


Рис. 7-1. Последовательность действий при конфигурировании и параметрировании

Предпосылки для ввода новой конфигурации

Перед вводом новой конфигурации необходимо:

- создать проект,
- в проекте сгенерировать подлежащий конфигурированию объект, т.е. станцию, и
- эту станцию выделить.

Последовательность действий описана в гл. 5.3 (Создание проекта).

Вызов конфигурационной таблицы

Для вызова конфигурационной таблицы действуйте следующим образом:

1. В окне проекта выделите объект "Station" [станция].
2. Вызовите команду меню **Bearbeiten**[Редактировать] → **Objekt öffnen**[Открыть объект].

Результат: на экране появляются конфигурационная таблица и каталог модулей.

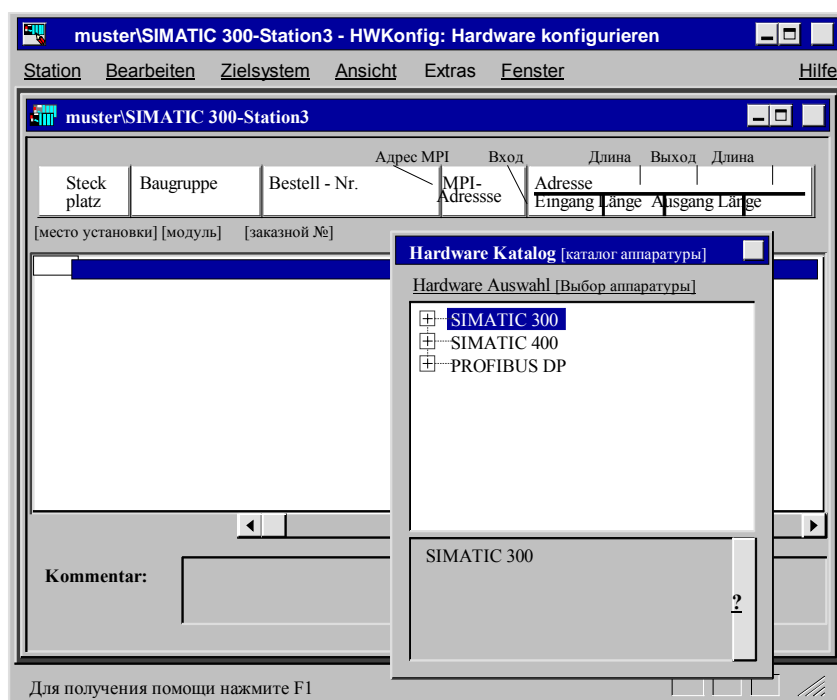


Рис. 7-2. STEP 7 с конфигурационной таблицей и каталогом модулей

7.2. Конфигурирование централизованной структуры

Конфигурирование централизованной структуры Для централизованной структуры модули размещаются рядом с CPU на носителе модулей и продолжаютс на других носителях. Количество **структуры** оснащаемых носителей модулей определяется применяемым CPU.

Последовательность действий Смонтируйте модули с помощью STEP 7 на носителе модулей точно так же, как на реальной установке, т.е. разместите их на носителе модулей в конфигурационной таблице. Когда все модули для Вашей структуры размещены, сохраните конфигурационную таблицу.

Перенос в конфигурационную таблицу На следующем рисунке показан пример переноса реальной структуры в конфигурационную таблицу.

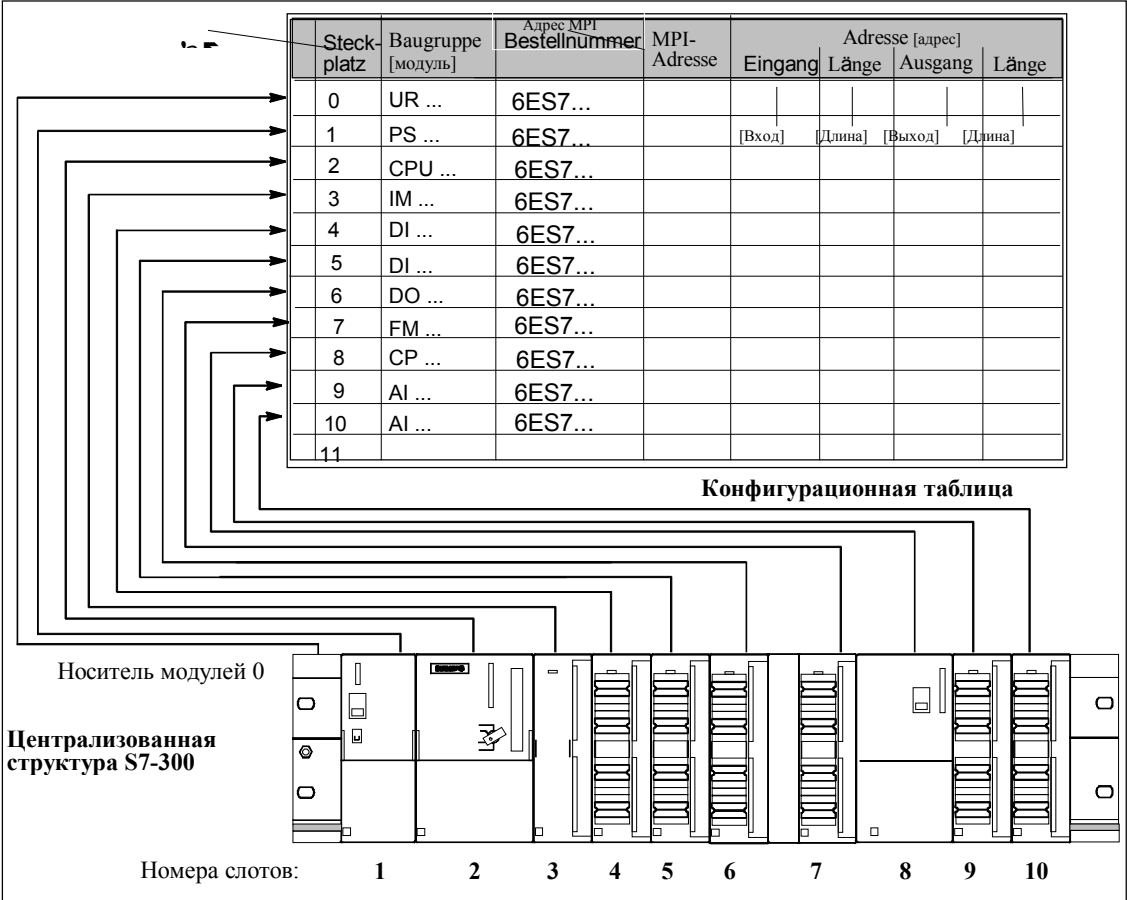


Рис. 7-3. От централизованной структуры к конфигурационной таблице

Предпосылка для конфигурирования

Вызвана конфигурационная таблица, как описано в гл. 7.1.

Каталог модулей

Каталог модулей содержит все доступные носители модулей, модули и интерфейсные модули.

Каталог модулей открывается автоматически, когда конфигурационная таблица создается в первый раз.

Выбор и размещение носителя модулей

Действуйте следующим образом:

1. Выберите из каталога модулей подходящий для Вашей структуры носитель (“Rack”). В SIMATIC 300 - профильную шину.
2. Отбуксируйте этот носитель в первую строку конфигурационной таблицы.

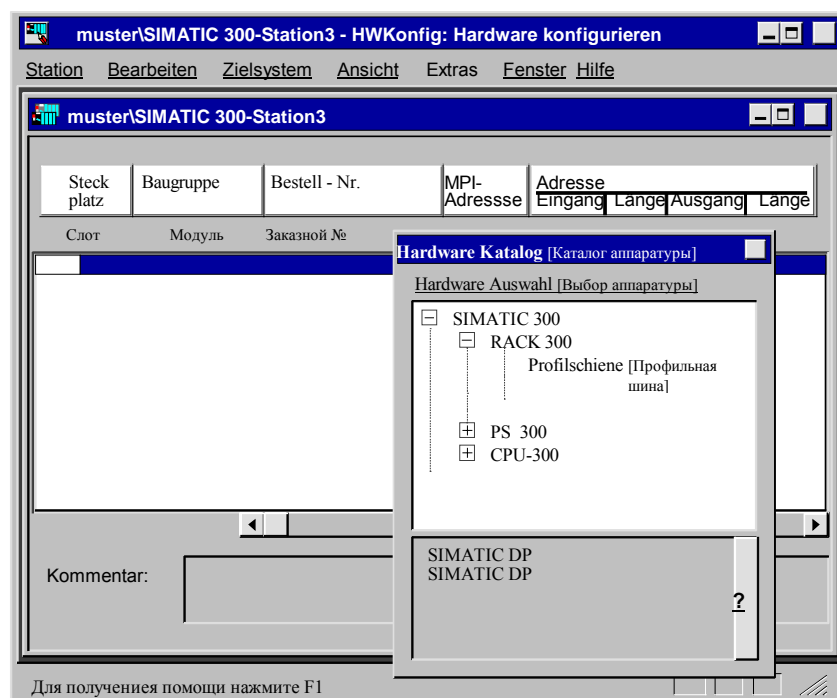


Рис. 7-4. Размещение носителя модулей в конфигурационной таблице

Выбор и размещение модулей

Действуйте следующим образом:

1. Откройте носитель модулей, щелкнув на знаке “+” в конфигурационной таблице (слева от слота носителя модулей).
Результат: отображаются слоты носителя модулей.
2. Выберите модуль из каталога модулей.
3. Отбуксируйте модуль в соответствующую строку конфигурационной таблицы.

В качестве альтернативы шагам 2 и 3 можно пометить слот в конфигурационной таблице, а затем дважды щелкнуть на модуле в каталоге аппаратуры.

**Применение
интерфейсных**

- Интерфейсный модуль можно:
- непосредственно вставить в CPU, если в нем есть гнезда для **модулей** модулей, или
 - в контроллерах M7 вставить в модуль расширения (EXM), предназначенный CPU или FM.
- Если Вы разместили CPU/EXM в конфигурационной таблице, то слева от слота CPU/EXM появляется знак “+”. При щелчке на этом знаке “+” отображаются строки для размещения интерфейсных модулей.
- Некоторые примеры показаны на следующем рисунке.


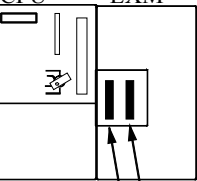
Конфигурация	Пример	Перенос примера в конфигурационную таблицу																												
CPU с 3 гнездами для модулей может принять до 3 интерфейсных модулей	<div>CPU 488-4</div> <div></div> <div>Интерфейсный модуль</div> <div>M7-400</div>	<div>[Слот Модуль Заказной №]</div> <table><tr><th></th><th>Steckplatz</th><th>Baugruppe</th><th>Bestellnumm</th></tr><tr><td>-</td><td>0</td><td>UR ..</td><td>6ES7...</td></tr><tr><td></td><td>1</td><td>PS ..</td><td>6ES7...</td></tr><tr><td>-</td><td>2</td><td>CPU 488-4</td><td>6ES7...</td></tr><tr><td></td><td>2.1</td><td>IF ...</td><td>6ES7...</td></tr><tr><td></td><td>2.2</td><td></td><td></td></tr><tr><td></td><td>2.3</td><td></td><td></td></tr></table>		Steckplatz	Baugruppe	Bestellnumm	-	0	UR ..	6ES7...		1	PS ..	6ES7...	-	2	CPU 488-4	6ES7...		2.1	IF ...	6ES7...		2.2				2.3		
	Steckplatz	Baugruppe	Bestellnumm																											
-	0	UR ..	6ES7...																											
	1	PS ..	6ES7...																											
-	2	CPU 488-4	6ES7...																											
	2.1	IF ...	6ES7...																											
	2.2																													
	2.3																													
CPU или FM с модулем расширения, который может принять до 3 интерфейсных модулей	<div><div>CPU EXM</div><div></div><div>M7-300</div><div>Интерфейсные модули</div><div>M7-400</div></div>	<table><tr><th></th><th>Steckplatz</th><th>Baugruppe</th><th>Bestellnu</th></tr><tr><td>-</td><td>0</td><td>UR ..</td><td>6ES7...</td></tr><tr><td></td><td>1</td><td>PS ..</td><td>6ES7...</td></tr><tr><td>+</td><td>2</td><td>CPU ...</td><td>6ES7...</td></tr><tr><td>-</td><td>3</td><td>EXM ...</td><td>6ES7...</td></tr><tr><td></td><td>3.1</td><td>IF ...</td><td>6ES7...</td></tr><tr><td></td><td>3.2</td><td></td><td></td></tr></table>		Steckplatz	Baugruppe	Bestellnu	-	0	UR ..	6ES7...		1	PS ..	6ES7...	+	2	CPU ...	6ES7...	-	3	EXM ...	6ES7...		3.1	IF ...	6ES7...		3.2		
	Steckplatz	Baugruppe	Bestellnu																											
-	0	UR ..	6ES7...																											
	1	PS ..	6ES7...																											
+	2	CPU ...	6ES7...																											
-	3	EXM ...	6ES7...																											
	3.1	IF ...	6ES7...																											
	3.2																													

Рис. 7-5. Связь интерфейсного модуля и конфигурационной таблицы

**Расширение
конфигурации**

- Если Вы хотите расширить конфигурацию добавлением других носителей модулей, действуйте следующим образом:
1. Выберите носитель модулей из каталога модулей.
 2. Отбуксируйте этот носитель из каталога модулей на любую серую строку.
- Результат:** на этом месте вставляется носитель модулей.
3. Откройте носитель модулей и поставьте ему в соответствие модули, как описано в подразделе “Выбор и размещение модулей”.

7.3. Конфигурирование сети SINEC L2–DP

Сеть SINEC L2–DP Сеть SINEC L2–DP состоит из ведущего модуля DP (DP-мастера) и ведомых модулей DP (DP-Slaves), которые соединены через шинный кабель и связываются друг с другом через протокол DP (децентрализованная периферия).

Ведущий модуль DP В качестве ведущего модуля DP можно использовать:
(DP-мастер)

- CPU с жестко встроенным или вставляемым интерфейсом ведущего модуля DP (напр., жестко встроен CPU 315–2 DP)
- интерфейсный модуль, размещенный в CPU/FM (напр., IF 964–DP в CPU 488–4)
- CP в соединении с CPU (напр., CP 342–5: он параметрируется специальным, вызываемым в STEP 7 программным обеспечением).

Ведомые модули DP
(DP-Slaves)

В качестве ведомых модулей DP можно использовать:

- модули с встроенными цифровыми/аналоговыми каналами ввода и вывода (компактные DP–Slaves, напр., ET 200B)
- модули подключения с соответствующими S5– или S7–модулями (модульные DP–Slaves, напр., ET 200M)

Последовательность действий В принципе сеть SINEC L2–DP конфигурируется точно так же, как централизованная структура. Действуйте следующим образом:

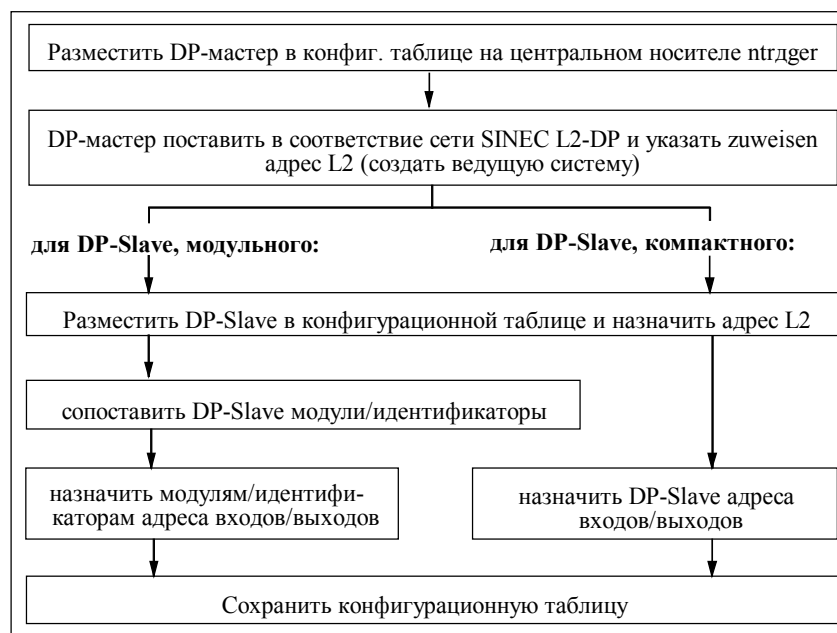


Рис. 7-6. Последовательность действий при конфигурировании сети SINEC L2-DP

Перенос в конфигурационную таблицу

Разместите DP-мастер централизованно на носителе модулей.
Разместите ведомые модули DP в собственной конфигурационной таблице.Переход в эту таблицу происходит только после конфигурирования DP-мастера.

Следующий рисунок показывает на примере перенос реальной сети SINEC L2 в конфигурационную таблицу для ведомых DP. Точно так же, как и при открытии носителей модулей следующие строки с устройствами ввода/вывода или модулями ведомых DP появляются при щелчке на поле “+”.

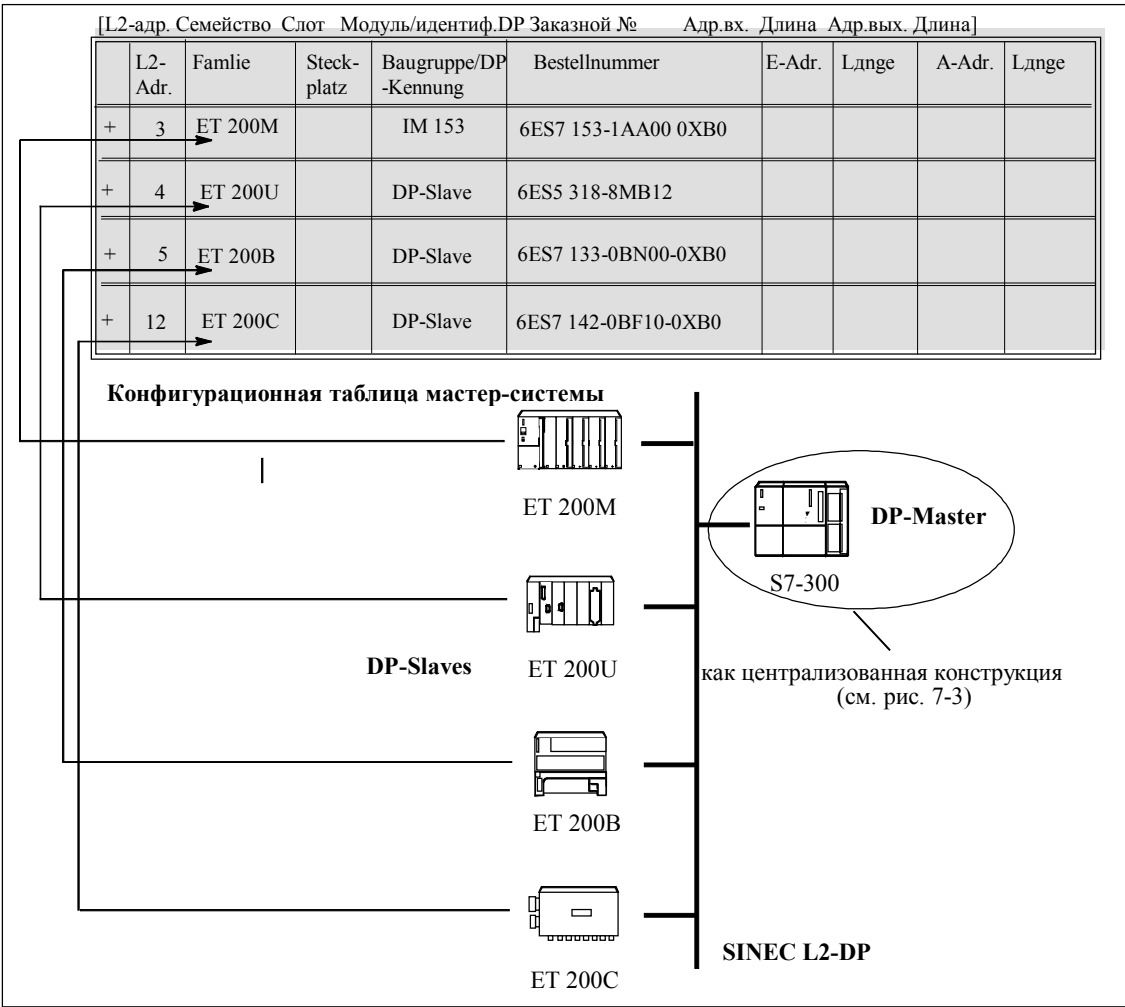


Рис. 7-7. От реальной сети SINEC L2-DP к “Конфигурационной таблице мастер-системы”

7.4. Создание ведущей системы

- Предпосылка

Вы вызвали конфигурационную таблицу (см. гл. 7.1).
- Мастер-система

Все ведомые DP, поставленные в соответствие DP-мастеру, образуют вместе с DP-мастером мастер-систему. Мастер-система - это часть сети SINEC L2-DP (см. гл. 8).
- Создание мастер-системы

После того как DP-мастер размещен в конфигурационной таблице, можно создавать соответствующую мастер-систему.

Создать мастер-систему – это значит подчинить DP-мастеру сеть SINEC L2–DP и назначить ему L2-адрес.
- Выбор и DP–мастера

DP-мастер выбирается из каталога модулей и размещается в **размещение** конфигурационной таблице (как модули централизованной структуры, см. гл. 7.2).

Результат: Если DP-мастер является встроенным интерфейсом CPU, то слева от CPU находится знак “+”. Если на нем щелкнуть, появляется вторая строка с DP-мастером.
- Два примера

На следующем рисунке показаны два примера для DP-мастера и их размещения в конфигурационной таблице.

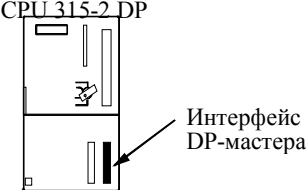
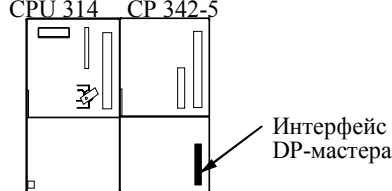
Подсоединение AS к SINEC L2-DP	Пример для модуля с функциями DP-мастера	Перенос примера в конфигурационную таблицу																		
S7-300 CPU с интерфейсом DP-мастера		<table><tr><th>Слот</th><th>Модуль</th><th>Заказной №</th></tr><tr><th>Steckplatz</th><th>Baugruppe</th><th>Bestellnum</th></tr><tr><td></td><td>UR ..</td><td>6ES7...</td></tr><tr><td>1</td><td>PS</td><td>6ES7...</td></tr><tr><td>2</td><td>CPU 315-2 DP</td><td>6ES7...</td></tr><tr><td>2.1</td><td>DP-Master</td><td></td></tr></table>	Слот	Модуль	Заказной №	Steckplatz	Baugruppe	Bestellnum		UR ..	6ES7...	1	PS	6ES7...	2	CPU 315-2 DP	6ES7...	2.1	DP-Master	
Слот	Модуль	Заказной №																		
Steckplatz	Baugruppe	Bestellnum																		
	UR ..	6ES7...																		
1	PS	6ES7...																		
2	CPU 315-2 DP	6ES7...																		
2.1	DP-Master																			
S7-300 Коммуникационный процессор как DP-мастер		<table><tr><th>Слот</th><th>Модуль</th><th>Заказной №</th></tr><tr><th>Steckplatz</th><th>Baugruppe</th><th>Bestellnum</th></tr><tr><td></td><td>UR ..</td><td>6ES7...</td></tr><tr><td>1</td><td>PS</td><td>6ES7...</td></tr><tr><td>2</td><td>CPU ...</td><td>6ES7...</td></tr><tr><td>3</td><td>CP 342-5</td><td>6ES7...</td></tr></table>	Слот	Модуль	Заказной №	Steckplatz	Baugruppe	Bestellnum		UR ..	6ES7...	1	PS	6ES7...	2	CPU ...	6ES7...	3	CP 342-5	6ES7...
Слот	Модуль	Заказной №																		
Steckplatz	Baugruppe	Bestellnum																		
	UR ..	6ES7...																		
1	PS	6ES7...																		
2	CPU ...	6ES7...																		
3	CP 342-5	6ES7...																		

Рис. 7-8. Взаимосвязь DP-мастера и конфигурационной таблицы

Дальнейшие действия

После размещения DP-мастера в конфигурационной таблице Вы можете:

- продолжить размещение “централизованных” модулей в конфигурационной таблице (см. гл. 7.2)
- или
- далее конфигурировать сеть SINEC L2–DP, поставив DP-мастер в соответствие сети SINEC L2–DP и назначив ему L2–адрес.

L2–адрес

Каждому DP-мастеру и каждому ведомому модулю DP мастер-системы необходимо для однозначной идентификации в сети SINEC L2–DP назначить L2-адрес.

Подчинение сети и назначение L2-адреса

Для подчинения сети и назначения L2-адреса действуйте следующим образом:

1. Отметьте строку, в которой находится DP-мастер.
2. Вызовите команду меню **Bearbeiten → Mastersystem → Öffnen** [Редактировать → Мастер-система → Открыть].
Появляется диалоговое окно для подчинения сети и назначения L2-адреса.
3. Поставьте DP–мастер в соответствие сети SINEC L2–DP (предлагается сеть по умолчанию).
4. Назначьте L2-адрес для DP-мастера (в качестве значения по умолчанию предлагается самый младший свободный L2-адрес).

Результат: после подчинения сети SINEC L2–DP и назначения L2-адреса отображается “Конфигурационная таблица мастер-системы”.

Конфигурационная таблица мастер-системы

В конфигурационной таблице мастер-системы разместите все ведомые модули DP мастер-системы и параметрируйте их.

7.5. Конфигурирование ведомых модулей DP

Виды ведомых DP (DP-Slaves)

В таблице на следующем рисунке Вы найдете различные виды DP-slaves с их свойствами. DP-Slaves перечислены в каталоге модулей в папке “SIMATIC DP” по их семействам.

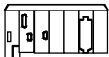




Вид DP-Slave	Семейство DP-Slave (примеры)	Свойства
DP-Slave, модульный	ET 200U 	состоит из модуля подключения и модулей из спектра S5
	ET 200M 	состоит из модуля подключения и модулей из спектра S7-300
DP-Slave, компактный	ET 200B 	модуль с встроенными каналами ввода/вывода
Интеллекту- альный DP- Slave	CPU CP 342-5 	коммуникационный процессор спектра S7-300, соединенный с DP-Slave S7-300-CPU параметрируется специальным ПО, которое вызывается в STEP 7
	S5-95U 	Устройство автоматизации с функциями DP-Slave

Рис. 7-9. Виды ведомых DP

Предпосылки для конфигурирования

Выбор и размещение DP-Slave

Вы создали мастер-систему и находитесь в конфигурационной таблице “Конфигурация мастер-системы”.

Чтобы разместить DP-Slave в конфигурационной таблице, действуйте следующим образом:

1. Выберите из каталога модулей компактный DP-Slave (напр., ET 200B) или модуль подключения для модульного DP-Slave (напр., IM 153 для ET 200M).

Обратите внимание: Если DP-Slave не появляется в каталоге модулей, Вы должны считать файл типов ведомых DP. Последовательность действий объяснена в гл. 7.12.

2. Разместите DP-Slave/модуль подключения в конфигурационной таблице.

STEP 7 автоматически предоставляет в Ваше распоряжение список свободных L2-адресов. В качестве значения по умолчанию предлагается ближайший свободный L2-адрес.

3. Задайте L2-адрес.

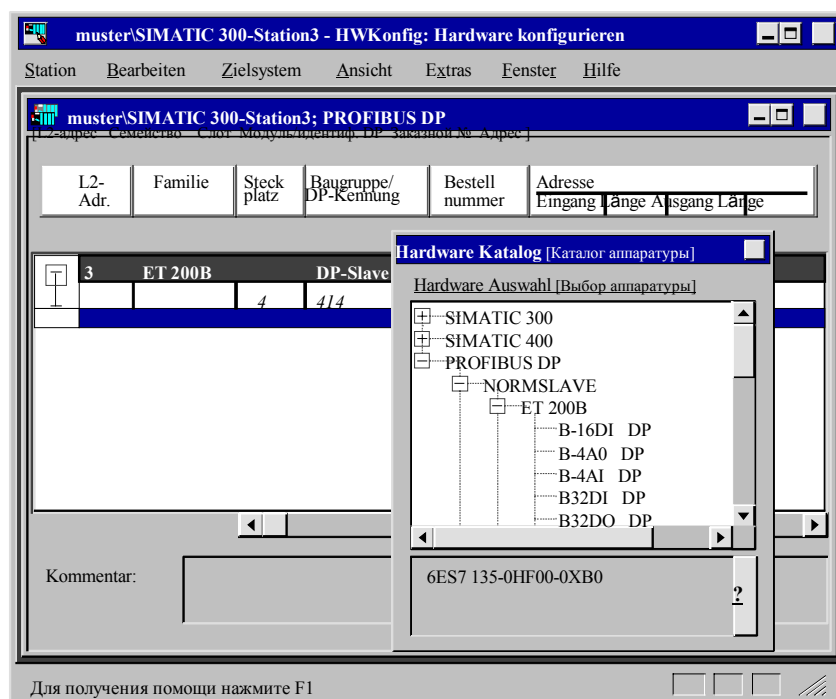


Рис. 7-10. Размещение DP-Slave в "Конфигурационной таблице мастер-системы"

Результат

DP-Slave появляется в выбранной строке “Конфигурационной таблицы мастер-системы”. Назначенный L2-адрес находится в соответствующем столбце “L2-Adr.”.

**Подчинение
модулей
модульному
DP-Slave
“+”).**

Для подчинения модулей модульному DP-Slave действуйте так же, как при размещении модулей в централизованной структуре (см. гл. 7.2). Обратите, пожалуйста, внимание на то, что DP-Slave должен быть открыт, прежде чем Вы сможете подчинить ему модули (щелкнуть на знаке

**Присвоение
идентификаторов
модульному
DP-Slave**

Модульным DP-Slaves (напр., ET 200M) присваиваются идентификаторы DP. В идентификаторе DP закодированы такие свойства, как адресная область и непротиворечивость данных.

Идентификатор DP появляется автоматически, когда модуль вставляется в конфигурационную таблицу. Идентификатор DP можно редактировать, дважды щелкнув на строке, где он находится.

**Назначение
адресов
входов/выходов**

Каждому входу/выходу ведомого модуля DP ставится в соответствие ровно один адрес, через который к этому входу/выходу можно обратиться. Поэтому Вы должны каждому модулю/идентификатору DP назначить начальный адрес.

STEP 7 автоматически назначает адреса по умолчанию, как только Вы включаете модуль/DP-Slave в конфигурационную таблицу. Адреса по умолчанию и их длины вносятся в столбцы “E-Adr.” [“Адрес входа”] или “A-Adr.” [“Адрес выхода”] и “Länge” [“Длина”].

**Конфигурационная
таблица**

На следующем рисунке показана “Конфигурационная таблица” для полностью сконфигурированной мастер-системы.

	L2-Adr.	Familie	Steckplatz	Baugruppe/DP-Kennung	Bestellnummer	E-Adr.	Länge	A-Adr.	Länge
-	3	ET 200M	1	IM 153	6ES7 153-1AA00 0XB0				
			4	DI 16	6ES7....	105	2		
			5	DI 16	6ES7....	107	2		
			6	DO 32	6ES7....			101	4
			7	D18/D08 ...	6ES7....	109	1	105	1
			8	AI 2	6ES7....	150	2		
-	4	ET 200U	2	DP-Slave	6ES5 318-8MB12				
			4	8DE	6ES5....	110	1		
			5	8DA	6ES5....			106	1
			6	2AX	6ES5....	111	2	107	2
-	5	ET 200B	3	DP-Slave	6ES7 133-0BN00-0XB0				
			4	24DE		130	3		
			5	8DA				100	1
+	12	ET 200C	4	DP-Slave	6ES7 142-0BF10-0XB0				

Рис. 7-11. "Конфигурационная таблица мастер-системы"

7.6. Параметризация модулей

Параметризация	Вы можете устанавливать параметры параметризуемых модулей. Какие параметры могут быть установлены, зависит от модуля.
Предпосылка	Перед параметризацией модуля его необходимо разместить в конфигурационной таблице.
Параметризация модулей	<p>Для параметризации модулей действуйте следующим образом:</p> <ol style="list-style-type: none">1. Дважды щелкните в конфигурационной таблице на заказном номере модуля, параметры которого Вы хотите установить, или отметьте соответствующий модуль и выберите команду меню Bearbeiten → Objekteigenschaften [Редактировать → Свойства объекта]. <p>Результат: Появляется диалоговое окно с одной или несколькими вкладками с данными о модуле и устанавливаемыми параметрами модуля.</p>
Другие возможности параметризации в S7-300/400	<p>Для контроллеров S7-300/400 Вы имеете возможность у некоторых модулей устанавливать параметры в прикладной программе (напр., у аналоговых модулей). Для этого вызывайте в прикладной программе системные функции (SFC) WR_PARM, WR_DPARM и PARM_MOD.</p> <p>Точную информацию о системных функциях Вы найдете в Справочном руководстве <i>System- und Standardfunktionen</i> [Системные и стандартные функции] /235/.</p>
Другие возможности параметризации в M7-300/400	<p>Для контроллеров M7-300/400 Вы имеете возможность устанавливать параметры сигнальных модулей в программе на языке C.</p> <p>Для этого вызовите в C-программе функцию прикладного программного интерфейса M7 (M7-API-функцию) “M7StoreRecord”. Эта функция передает параметры сигнальному модулю.</p> <p>Точную информацию о функциях прикладного программного интерфейса M7 Вы найдете в руководствах по системному программному обеспечению для M7-300/400 /280/, /281/, /282/.</p>

7.7. Параметризация сети SINEC L2-DP

Предпосылка Перед параметризацией ведущего и ведомых модулей DP эти модули должны быть размещены в конфигурационной таблице.

Последовательность действий Для параметризации сети SINEC L2-DP следует применять следующую последовательность действий:

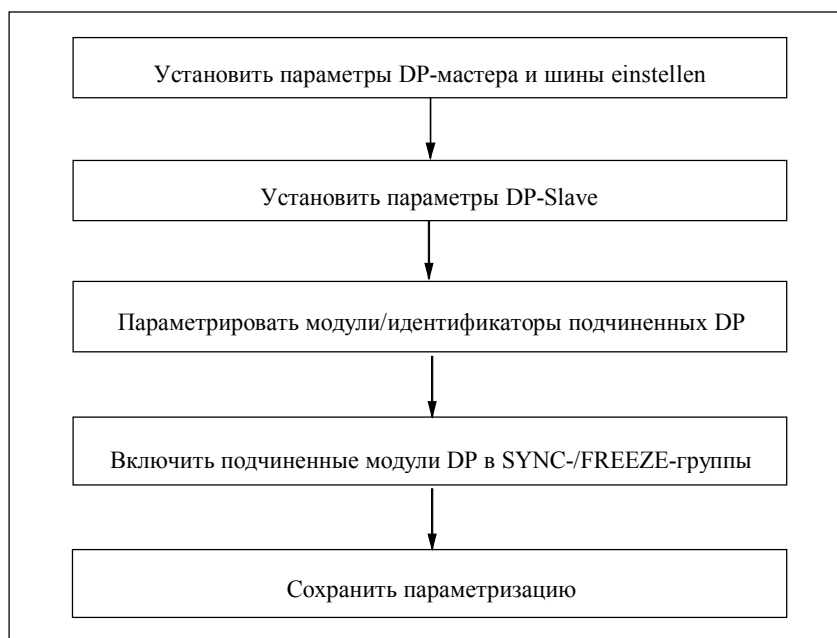


Рис. 7-12. Последовательность действий при параметризации сети SINEC L2-DP

7.8. Параметризация DP-мастера и сети SINEC L2-DP

Параметры DP-мастера	С помощью параметров DP-мастера Вы получаете - так же как и для любых других модулей - информацию об этом модуле. Кроме того, отображается L2-адрес DP-мастера, который также можно изменять.
Параметры шины	С помощью параметров шины определяются свойства, действительные для всей сети - здесь для сети SINEC L2-DP (напр., профиль и скорость передачи).
Предпосылка для чтения /изменения	Прежде чем Вы сможете читать/изменять параметры DP-мастера и шины, Вы должны создать мастер-систему и находиться в параметров конфигурационной таблице.
Чтение/изменение параметров	<p>Для чтения/изменения параметров действуйте следующим образом:</p> <ol style="list-style-type: none">1. Дважды щелкните в конфигурационной таблице на строке, в которой находится DP-мастер, или отметьте эту строку и выберите команду меню Bearbeiten → Objekteigenschaften [Редактировать → Свойства объекта]. <p>Результат: Появляется несколько вкладок с параметрами.</p>

7.9. Параметризация ведомых модулей DP

Параметры DP-Slave	Вы можете для любого DP-Slave вывести на экран его параметры и отчасти их установить/изменить. Параметрами DP-Slave являются, например, его тип, заказной номер и диагностический адрес.
Установка параметров DP-Slave	<p>Для установки параметров DP-Slave действуйте следующим образом:</p> <ol style="list-style-type: none">1. Дважды щелкните в конфигурационной таблице для мастер-системы на строке, в которой находится DP-Slave. <p>Результат: появляется несколько вкладок с параметрами DP-Slave.</p>
Параметры для модулей/идентификаторов	Параметризация модулей и идентификаторов для DP-Slave производится так же, как и модулей централизованной структуры.

7.10. Включение ведомого модуля DP в SYNC-/FREEZE-группы

Команды управления SYNC/FREEZE

DP-мастер с соответствующим набором функций может посылать одновременно группе ведомых модулей DP команды управления SYNC и/или FREEZE для синхронизации ведомых DP. Для этого ведомые модули необходимо разделить на SYNC - и FREEZE - группы.

Предпосылка

Прежде чем иметь возможность поставить в соответствие модулю DP-Slave SYNC-/FREEZE-группу, Вы должны создать мастер-систему и находиться в “Конфигурационной таблице мастер-системы”.

Сопоставление SYNC-/FREEZE-групп

Каждый DP-Slave можно поставить в соответствие максимум 8 SYNC-и/или FREEZE-группам. Для этого действуйте следующим образом:

1. Отметьте в “Конфигурационной таблице мастер-системы” строку, в которой находится DP-Slave, который Вы хотите включить в группу.
2. Выберите команду меню **Bearbeiten → Mastersystem → Eigenschaften** [Редактировать → Мастер-система → Свойства] .

Результат: появляется диалоговое окно с таблицей, в которую вносится принадлежность к группам.

Выдача команд управления

Команды управления SYNC и FREEZE ведомым модулям DP подаются в прикладной программе.

- Используйте SFC DP_SYNC и DP_FREEZE (см. Справочное руководство *System- und Standardfunktionen (Системные и стандартные функции) /235/*) для контроллеров S7-300/400.
- Используйте специальные функции прикладного программного интерфейса M7 (см. руководства по системному программному обеспечению для M7-300/400 /280/, /281/, /282/) для контроллеров M7-300/400 .

7.11. Сохранение, загрузка, чтение, изменение и копирование

Введение

В дальнейшем конфигурация включает в себя централизованную структуру, соответствующие мастер-системы и все параметризации.

Вы узнаете, как сохранить созданную конфигурацию, загрузить ее в контроллер и как прочитать существующую конфигурацию и изменить ее. Централизованная структура и сеть SINEC L2-DP рассматриваются совместно, так как последовательность действий для них одинакова.

Сохранение конфигурационной таблицы

Для сохранения конфигурационной таблицы используйте команду меню **Station → Speichern** [Станция → Сохранить]. После этого PG сохраняет всю конфигурацию (конфигурационную таблицу, включая, возможно, выполненные параметризации) в текущем проекте (на жестком диске).

Предпосылки для загрузки

Созданная конфигурация должна соответствовать фактической структуре, а конфигурационная таблица выведена на экран.

Советы

Вы можете перевести конфигурацию станции в загружаемые системные блоки данных (SDB) командой меню

Station → Übersetzen.... [Станция → Перевести...].

SDB сохраняются в текущем проекте и могут быть перенесены, например, на платы памяти.

Командой меню:

Station → Konsistenz prüfen... [Станция → Проверить непротиворечивость...]

Вы можете проверить, можно ли создать SDB из текущей конфигурации станции.

Загрузка конфигурации в контроллер

Для загрузки конфигурации в контроллер действуйте следующим образом:

1. Переведите CPU в состояние STOP.
2. Выберите команду меню **Zielsystem → Laden...** [Контроллер → Загрузить...]. STEP 7 приведет Вас к результату через диалоговое окно.

Результат: конфигурация для всего контроллера загружается в CPU. Параметры CPU становятся действительными немедленно, параметры для остальных модулей переносятся в эти модули при запуске.



конфигурацию.

Указание

Частичные конфигурации, например, конфигурации отдельных носителей модулей, не могут быть загружены в контроллер. Из соображений непротиворечивости STEP 7 всегда загружает в контроллер только полную

Сброс параметров на значения по умолчанию



Параметры хотя бы для одного модуля на их значения по умолчанию, Вы должны перед переносом конфигурации произвести полное стирание CPU!

Если Вы хотите сбросить параметры для отдельных модулей на их значения по умолчанию, примите во внимание следующие важные указания:

Предупреждение

Параметры уже параметрированного модуля могут быть сброшены на значения по умолчанию только путем полного стирания CPU.

Если Вы уже создали и перенесли конфигурацию и затем сбрасываете

Если Вы не стираете CPU, то первоначальная параметризация для модуля сохраняется, т.е. **не** параметризован по умолчанию.

Чтение конфигурации

загрузите конфигурацию из контроллера в PG.

Чтение существующей конфигурации из контроллера выполняют, например, для того, чтобы на ее основе создать подобную конфигурацию или чтобы изменить установки параметров.

из AS

Командой меню **Zielsystem → Laden in PG** [Контроллер → Загрузить в PG]

Эту конфигурацию можно сохранить в существующем проекте (команда меню **System → Speichern unter** [Система → Сохранить как]. Появляющиеся не полностью заказные номера дополняйте командой меню **Extras → Baugruppen spezifizieren** [Дополнительные функции → Специфицировать модули].



Указание

Мастер-системы нельзя считать из контроллера. Если Вы хотите изменить существующую мастер-систему, то Вы должны сконфигурировать в STEP 7 новую мастер-систему и загрузить ее в контроллер. Мастер-система в контроллере перезаписывается путем загрузки.

Изменение конфигурации

В следующей таблице Вы найдете важнейшие возможности изменений, имеющиеся в Вашем распоряжении в мастер-системе для модулей централизованной структуры или модульных ведомых DP.

Возможности изменения ? центр./сеть SINEC L2-DP)	Вы должны...	Команда меню
стереть модуль из конфигурационной таблицы	пометить строку, в которой находится модуль	Bearbeiten→Löschen [Обработать→Стереть]
заменить в конфигурационной таблице один модуль другим	пометить строку, в которой находится модуль выбрать модуль из каталога модулей и поместить его в конфигурационной таблице на пустое место	Bearbeiten►Löschen -
поместить дополнительный модуль в конфигурационную таблицу	выбрать модуль из каталога модулей и поместить его в конфигурационной таблице на желаемое место	-

Рис. 7-13. Изменение конфигурации - в центральной конструкции и в SINEC L2-DP

Изменение конфигурации SINEC L2-DP

В следующей таблице Вы найдете важнейшие возможности изменения, имеющиеся в Вашем распоряжении для мастер-системы.

Возможности изменения (сеть SINEC L2-DP)	Вы должны...	Команда меню
стереть существующую мастер-систему	находиться в “Конфигурационной таблице мастер-системы”	Bearbeiten→ Löschen
стереть DP-мастер существующей мастер-системы	отметить строку, в которой находится DP-мастер	[Обработать→ Стереть]
назначить дополнительный модуль модульному DP-Slave	выбрать модуль из каталога модулей и поместить в конфигурационной таблице под соответствующим модулем подключения DP-Slave	-
изменить в STEP 7 (offline) запрограммированный L2-адрес для DP-Master/DP-Slave	дважды щелкнуть на DP-Master /DP-Slave и изменить L2-адрес в блоке диалога	-
изменить L2-адрес для DP-Slave в целевой системе (online)	находиться в окне проекта или в конфигурационной таблице мастер-системы	Zielsystem→ L2-Adresse vergeben
изменить параметры шины	см. главу 7.8	-

Рис. 7-14. Изменение конфигурации - SINEC L2-DP

Копирование конфигурации в главе 3.7.

Как копировать конфигурацию в Администраторе SIMATIC (SIMATIC Manager), например, для серии аналоговых систем управления, описано

7.12. Дополнительная инсталляция модулей в каталоге модулей

DP–Slaves

STEP 7 нуждается для каждого ведомого модуля DP (DP-Slave) в типовом файле, для того чтобы можно было выбрать DP-Slave из каталога модулей. В типовом файле хранятся все свойства ведомого модуля DP.

Для ведомых модулей DP фирмы Siemens имеется типовый файл для каждого типа DP-Slave, который Вы можете прочитать, но не изменить. Для ведомых модулей DP других изготовителей типовый файл поставляется изготовителем.

Считывание типового файла

Если DP-Slave не появляется в каталоге модулей, Вы должны после запуска STEP 7 установить соответствующий типовый файл в каталог "TYPDATEI" и затем его считать. Применяйте для этого команду меню

Extras → DP-Typdateien aktualisieren [Дополнительные функции → Актуализировать типовые файлы DP].

Каталог типовых файлов

Типовой файл должен быть установлен в каталоге "STEP7_V2\DATA\TYPDATEI". Типовые файлы, не зависящие от языка, имеют расширение имени файла "*.200", немецкие типовые файлы имеют расширение "d.200".

Обзор

Аналогично конфигурации аппаратных средств, в которой Вы охватываете взглядом отдельную станцию, при конфигурировании сети Вы рассматриваете взаимодействие всех абонентов, участвующих в коммуникации, и необходимые для этого установки.

Безразлично, хотите ли Вы осуществить связь с помощью глобальных данных или с помощью коммуникационных блоков в прикладной программе: основой для коммуникации всегда является предварительно сконфигурированная сеть.

При конфигурировании сети все установки всегда проверяются на достоверность и непротиворечивость. Дважды заданные адреса абонентов или недействительные установки распознаются уже при вводе. Таким образом, ошибок удастся избежать уже перед первым включением, благодаря чему экономятся время и деньги.

В этой главе Вы прочтете, как создать конфигурацию сети и какие параметры сети и станции должны быть установлены.

Обзор главы

В раз-деле	Вы найдете	на стр.
8.1	Конфигурация сети - обзор	8–3
8.2	Определение конфигурации сети с помощью базового программного пакета STEP 7	8–4
8.3	Определение конфигурации сети с помощью дополнительного программного пакета NETPRO	8–6
8.4	Символы в изображении сети	8–8
8.5	Особенности конфигурации подсетей MPI в S7–300	8–10
8.6	Загрузка конфигурации через сеть и изменение адреса абонента	8–11

Сеть и подсети

Сеть установки состоит из одной или нескольких подсетей с различными типами шин (SINEC L2, SINEC H1, MPI, связь “точка-к-точке”). К этим сетям подключаются отдельные станции.

**Пример
для отдельной
установки**

Рис. 8–1 показывает пример установки, включенной в сеть. Сеть состоит из двух подсетей MPI и одной подсети SINEC L2.

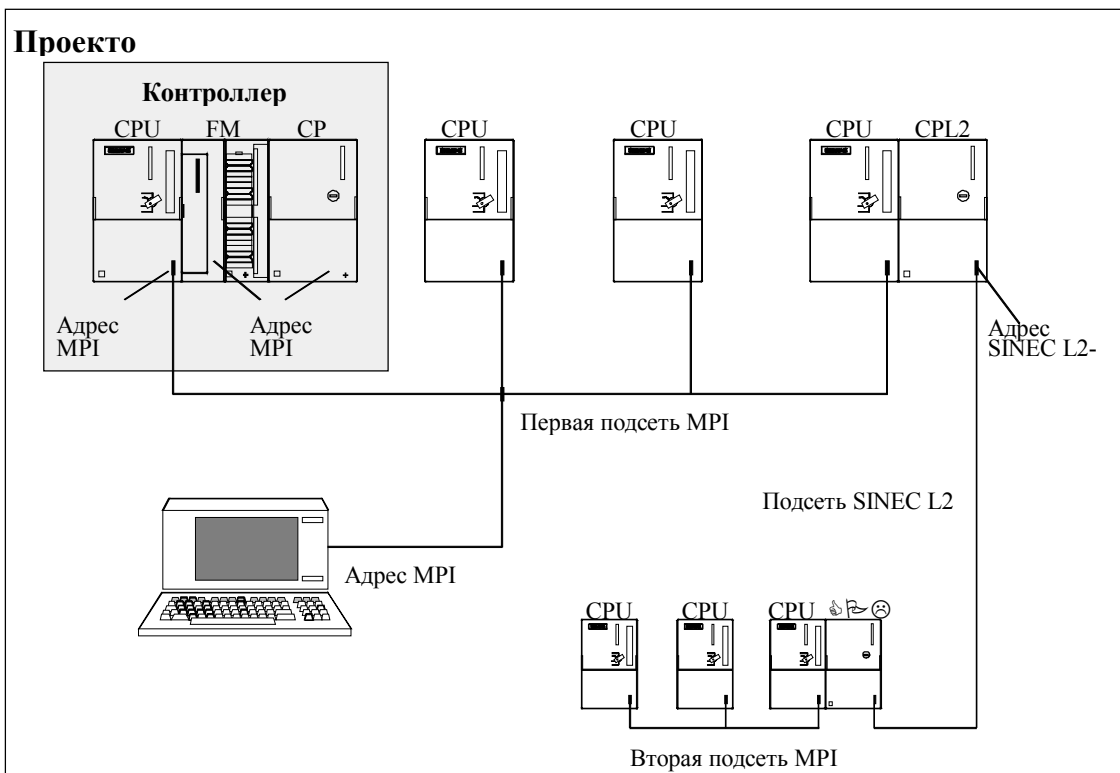


Рис. 8-1. Пример конфигурации установки



Указание

Следует обращать внимание на то, чтобы адреса абонентов (напр., адреса MPI) не задавались дважды!

8.1. Конфигурация сети - обзор

Последовательность действий Чтобы создать конфигурацию сети для проекта, действуйте следующим образом:

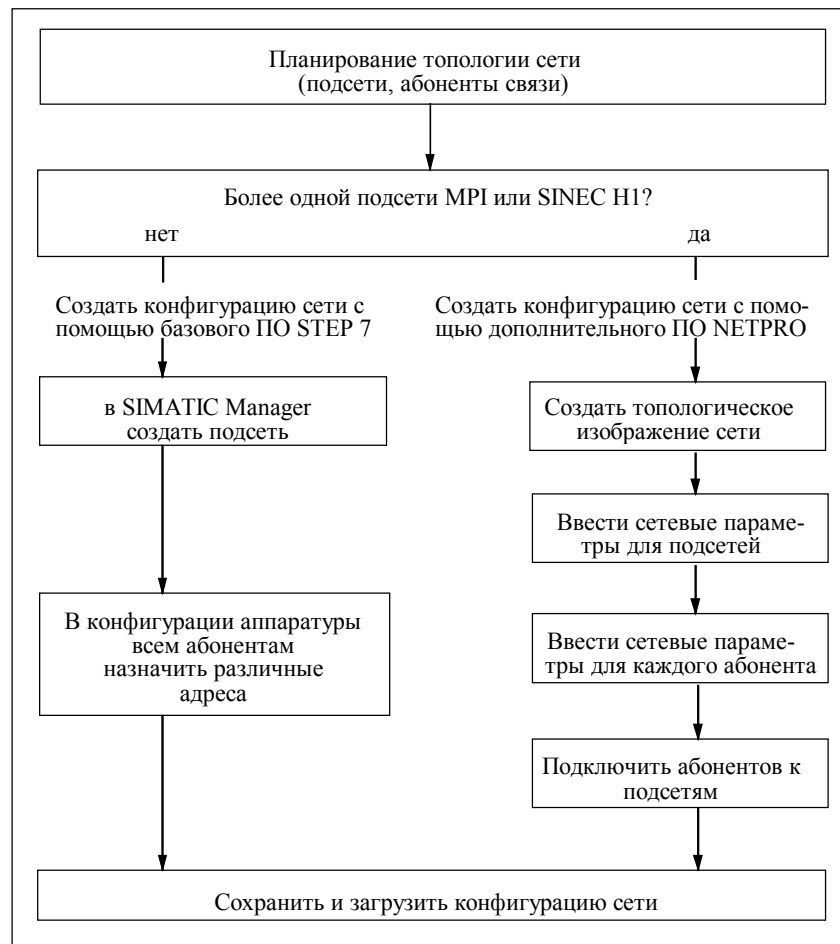


Рис. 8-2 . Последовательность действий при конфигурировании сети

8.2. Определение конфигурации сети с помощью базового программного пакета STEP 7

Введение	Если у Вас нет дополнительного программного пакета NETPRO и Ваш сетевой проект состоит только из одной подсети MPI, то конфигурацию сети можно определить через подсеть, создаваемую по умолчанию.
Данные для конфигурирования сети	<p>Для того чтобы полностью конфигурировать сеть Вы должны:</p> <ul style="list-style-type: none">• создать необходимые подсети• для каждой подсети установить ее свойства/параметры (например, наименование подсети, применяемую скорость передачи, самый старший адрес в сети).• для каждого включенного в сеть модуля установить свойства подключения к сети(например, адрес абонента, наименование подсети, к которой абонент подключен).
Создание подсети	<p>С помощью базового пакета STEP 7 Вы можете создать одну подсеть MPI, одну подсеть SINEC H1 и любое количество подсетей SINEC L2 и “точка-к-точке”. Для создания подсети действуйте следующим образом:</p> <ol style="list-style-type: none">1. Откройте проект в SIMATIC Manager.2. Выберите команду меню Einfügen → Subnetz [Вставить → Подсеть]. <p>Результат: STEP 7 включает в проект подсеть.</p>
Установка свойств подсети	<p>Для установки свойств подсети действуйте следующим образом:</p> <ol style="list-style-type: none">1. Выберите в своем проекте подсеть, для которой Вы хотите задать свойства (напр., подсеть MPI).2. Выберите команду меню Bearbeiten → Objekteigenschaften [Редактировать → Свойства объекта]. <p>Результат: на экран выводятся две вкладки, в которых можно установить имя сети, наивысший адрес MPI и скорость передачи.</p> <ol style="list-style-type: none">3. Установите параметры подсети.. <hr/> <p>Указание</p> <p>Для подсетей SINEC L2 можно выбрать или стандартный профиль шины, или параметризовать профиль шины, определенный пользователем. Информацию для определения профиля пользователем Вы найдете в описании используемого Вами коммуникационного процессора.</p> <hr/>



используемого Вами коммуникационного процессора.

Определение свойств абонента и подключения к сети

Предпосылка: Вы находитесь в конфигурационной таблице аппаратного обеспечения и уже разместили в ней программируемый модуль. Чтобы установить свойства абонента и подключение к сети, действуйте следующим образом:

1. Отметьте программируемый модуль.
2. Дважды щелкните на строке, в которой находится программируемый модуль, или выберите команду меню **Bearbeiten → Objekteigenschaften**[Редактировать → Свойства объекта].

Результат: открываются вкладки диалогового окна модуля.

3. Щелкните на кнопке для установки свойств абонента.

Результат: открывается окно для установки свойств подключения к сети.

4. Установите параметры для подключения к сети. Для этого выделите управляющее поле "vernetzt" ["включено в сеть"] и выберите адрес абонента и подсеть.

Если поле "vernetzt" не выделено, то модуль не включен ни в какую подсеть.

8.3. Определение конфигурации сети с помощью дополнительного программного пакета NETPRO

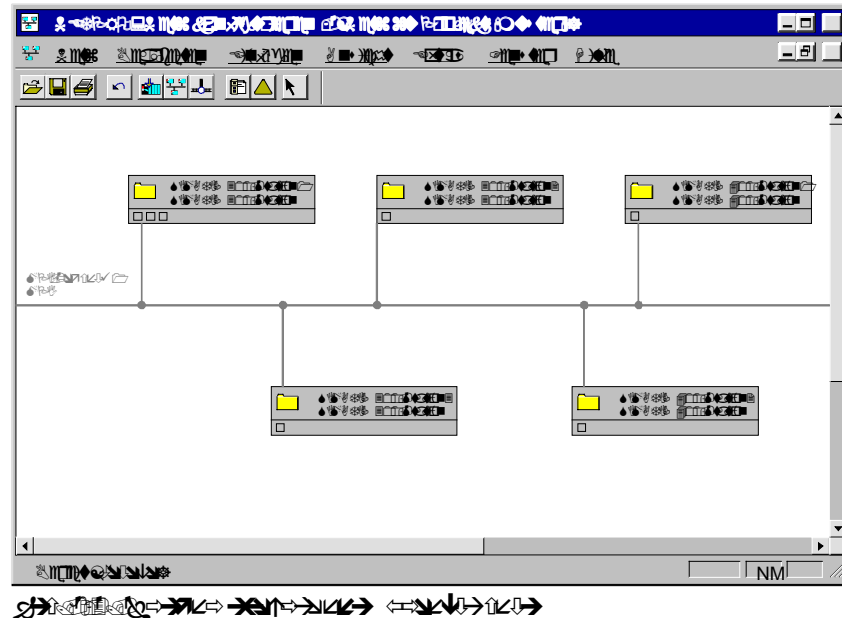
Введение	NETPRO обеспечивает особенно удобный и обозримый ввод конфигурации сети, так как с помощью NETPRO создается графическое изображение сети, в котором можно устанавливать все свойства для подсетей и абонентов сети.
Когда нужен NETPRO?	В принципе NETPRO можно использовать для любого сетевого проекта. Его необходимо использовать, если Ваша сеть состоит из нескольких подсетей MPI или SINEC H1.
Что делает NETPRO?	NETPRO показывает графическое изображение Вашего сетевого проекта, которое удобно изменять и расширять с помощью мыши. Разумеется, NETPRO проверяет все вводимые параметры на допустимость.
Проверка параметров сети, также возможна.	непротиворечивости, которой подвергаются все без исключения
	Для документирования Вы можете распечатать графическое изображение сети и табличный обзор всех сетевых параметров и абонентов.
Предпосылка работы с NETPRO	Перед тем как работать с NETPRO, Вы должны выполнить для конфигурирование программируемых модулей. Что для этого нужно сделать, описано в гл. 7 “Конфигурирование и параметризация модулей”.
Работа с NETPRO	Конфигурация сети создается в режиме offline. Для обработки всегда нужен только проект.

Вызов NETPRO

Для вызова NETPRO действуйте следующим образом:

1. Откройте в Администраторе SIMATIC (SIMATIC Manager) свой проект.
2. Отметьте подсеть (напр., подсеть MPI).
3. Запустите NETPRO, дважды щелкнув на подсети, или выберите команду меню **Bearbeiten** → **Objekt öffnen** [Редактировать → Открыть объект].

Результат: открывается окно с изображением проекта сети.



Создание отображения сети

Создание графического изображения сети в NETPRO настолько просто, насколько это возможно. Вы выбираете символы для станций и подсетей каталога и размещаете их буксировкой на экране по своему желанию. Затем Вы соединяете станции с соответствующей подсетью.

Установка параметров сети

Двойным щелчком на элементе (станции, сетевой или соединительной линии) или выбором элемента из меню Вы можете ввести или изменить параметры для выбранного элемента.

Могут быть введены/изменены

- параметры подсети
- аппаратная конфигурация станции
- параметры подключения к сети

Подробное описание установки параметров сети Вы найдете в online-помощи.

8.4. Символы в изображении сети

Введение

Каждое изображение сети состоит в сущности из четырех символов, которые далее будут объяснены более подробно. На рис. 8–4 показан фрагмент изображения сети, в котором использованы различные символы.

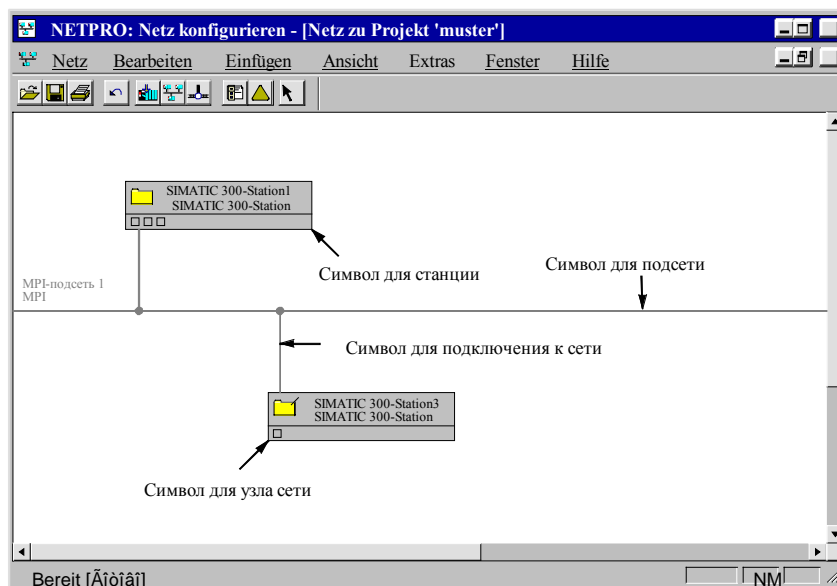


Рис. 8-4. Изображение проекта сети

Символ для подсети

Горизонтальные линии всегда обозначают подсеть. Двойным щелчком на этой линии Вы попадаете в диалог "Netzeigenschaften" ["Свойства сети"], в котором Вы устанавливаете все параметры, важные для подсети, такие как имя подсети, применяемая скорость передачи и наивысший адрес абонента в подсети.

Символ для станции

Большие прямоугольники обозначают станцию. Двойным щелчком на этой станции Вы попадаете в диалог для конфигурирования аппаратного обеспечения, в котором Вы можете, среди прочего, параметры используемых программируемых модулей, как, например, имя и адрес абонента.

Символ для узла сети

Маленькие квадраты обозначают узлы сети. Каждый сконфигурированный Вами программируемый модуль, получает собственный символ узла сети.

Символ для подключения к сети

Вертикальные линии обозначают подключение к сети. Двойным щелчком на этой линии Вы попадаете в диалог "Netzanschluß-Eigenschaften" ["Свойства подключения к сети"], в котором Вы можете определить, к каким модулям к какой подсети подключены.

Создать подключение к сети с помощью NETPRO особенно легко:

- Щелкните на символе “Узел сети” и удерживайте кнопку мыши в нажатом состоянии.
- Переведите курсор мыши на подсеть, к которой Вы хотите подключить узел.

Результат: NETPRO вставляет символ (вертикальную линию) в изображение сети.

8.5. Особенность конфигурации подсетей MPI в S7-300

Особенности для CP и FM в S7-300

CP и FM с собственным адресом MPI имеют одну особенность: Ваш адрес MPI устанавливается CPU автоматически по следующему образцу:

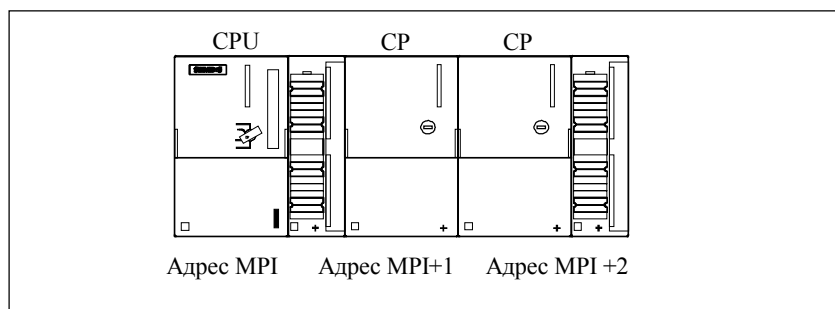


Рис. 8-5. Автоматическое задание адресов MPI для программируемых модулей

STEP 7 учитывает этот автоматизм при установлении адресов MPI.

Правило

При планировании адресов MPI для процессоров Вы должны, на основе вышеописанного, предусмотреть “пропуски адресов MPI” для FM и CP, чтобы не произошло двойное распределение адресов.

8.6. Загрузка конфигурации через сеть и изменение адреса абонента

Введение

Для безупречной работы сети все абоненты одной подсети должны иметь различные адреса.

Центральным модулям по умолчанию присваивается адрес 2. Так как этот адрес можно использовать только один раз, Вы должны изменить предустановленные адреса во всех остальных центральных модулях.

Изменение адреса абонента

Если в Вашей фактической структуре имеется несколько программируемых модулей с одинаковым адресом абонента, то измените адрес абонента программируемого модуля следующим образом:

1. Переключите модуль в состояние STOP и соедините свой программатор вставным кабелем с интерфейсом программируемого модуля (соединение точка-к-точке).
2. Откройте в SIMATIC Manager свой проект.
3. Откройте конфигурационную таблицу для желаемой станции.
Результат: появляется конфигурационная таблица станции.
4. Дважды щелкните в строке, в которой находится программируемый модуль.
Результат: в диалоговом окне появляются вкладки для параметрирования модуля.
5. Щелкните во вкладке "Allgemein" ["Общее"] на кнопке "CPU".
Результат: появляется диалог "Netzanschluß-Eigenschaften" ["Свойства подключения к сети"].
6. Выберите в этом диалоге новый адрес абонента, подтвердите ввод и закройте конфигурационную таблицу.
7. Загрузите конфигурацию командой меню **Zielsystem → Laden** [Контроллер → Загрузить].



Указание

Перед переносом конфигурации в подключенный модуль Вы должны задать адрес абонента для контроллера. Обратите внимание, что здесь Вы должны задать старый адрес абонента (так как в этот момент времени этот адрес еще действителен).

Загрузка конфигурации через сеть

Только после того, как все без исключения модули сети получили различные адреса, и Ваша фактическая структура совпала с созданной **через** конфигурацией, Вы можете выполнять все установки через инсталлированную сеть.

Загрузка конфигурации в контроллер S7/M7/C7

Для загрузки конфигурации в контроллер S7/M7/C7 действуйте следующим образом:

1. Подключите свой программатор к программируемому модулю.
2. Переключите контроллер, в который Вы хотите загрузить конфигурацию, в состояние STOP.
3. Выберите команду меню **Zielsystem** → **Laden in** [Контроллер → Загрузить в] STEP 7 приведет Вас затем через диалоговые окна к результату.

Результат: конфигурация для всей системы автоматизации загружена в CPU. Параметры CPU становятся действительными немедленно, параметры для других модулей переносятся в эти модули при пуске.



Указание

Исходя из требований непротиворечивости STEP 7 всегда загружает в контроллер полную конфигурацию (аппаратную и сетевую).

Загрузка конфигурации в другие контроллеры

Если Вы используете абонентов, не являющихся компонентами S7/M7/C7, например, устройства автоматизации из SIMATIC S5, то Вы должны сохранить сетевую информацию и у этих абонентов. Для этого прочитайте документацию используемых абонентов.

Конфигурирование связи с помощью глобальных данных

9

Обзор

Центральные модули SIMATIC предлагают коммуникационные возможности как будто “из дома”. Ниже объясняется, как путем согласования “глобальных данных” исчезают границы между пространственно разделенными системами автоматизации без необходимости писать хотя бы одну строчку в прикладной программе.

А именно, связь через глобальные данные не программируется, а конфигурируется. Конфигурирование обмена данными через глобальные данные очень просто, потому что необходимо только заполнить таблицу.

Что такое “связь через глобальные данные”?

Глобальные данные в том виде, как они используются в способе “связи через глобальные данные”, - это входы, выходы, меркеры, таймеры, счетчики и области в блоках данных, которые “делятся” двумя или более связанными друг с другом сетью системами автоматизации.

Связь через глобальные данные имеет место не более, чем 5 абонентами MPI. Она рассчитана на небольшие объемы данных, которые, как правило, передаются циклически. В некоторых центральных модулях из спектра SIMATIC S7-400 возможна также управляемая событиями передача с помощью SFC.

В этой главе описано, как производится конфигурирование связи через глобальные данные.

Обзор главы

В раз-деле	Вы найдете	на стр.
9.1	Связь через глобальные данные -обзор	9-2
9.2	Принцип связи через глобальные данные	9-4
9.3	Тестирование GD-связи	9-6
9.4	Примеры конфигураций	9-8

9.1. Связь через глобальные данные - обзор

Конфигурирование связи

Связь через глобальные данные (GD-связь) не программируется, а конфигурируется.

GD–

С помощью *STEP 7* создается таблица глобальных данных (GD-таблица), определяющая данные конфигурации для обмена данными. В эту таблицу вносится информация о том:

- какими данными CPU производится обмен
- каковы адресные области данных, подлежащих обмену

Дополнительно можно указать

- коэффициент редукции, определяющий через сколько циклов CPU данные должны посылаться/приниматься и
- адресную область (двойное слово) для информации о статусе

Создание соединений, как это требуется при обмене данными через коммуникационные блоки, для GD-связи отпадает.

Последовательность действий

Чтобы иметь возможность использовать глобальные данные в прикладной программе для передачи данных, действуйте следующим образом:

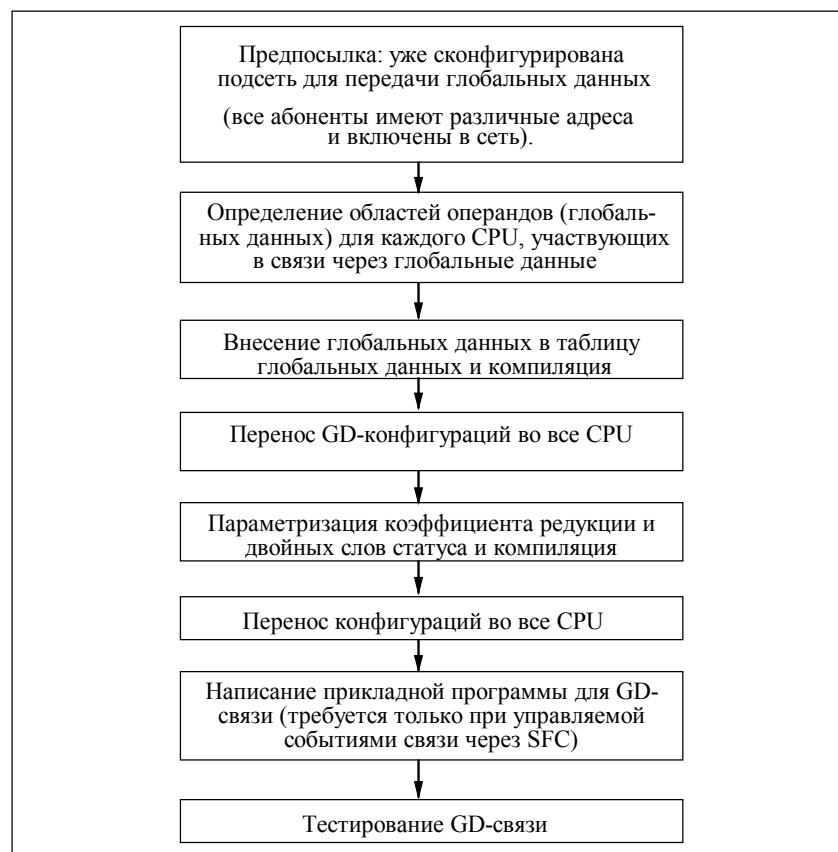


Рис. 9-1. Последовательность действий для связи через глобальные данные

Вызов GD-таблицы

Предпосылка: Ваш проект открыт и подсеть MPI помечена.

Вызов GD-таблицы осуществляется выбором в SIMATIC Manager команды меню **Extras → Globaldaten definieren** [Дополнительные функции → Определение глобальных данных].

Результат: на экране отображается таблица глобальных данных.

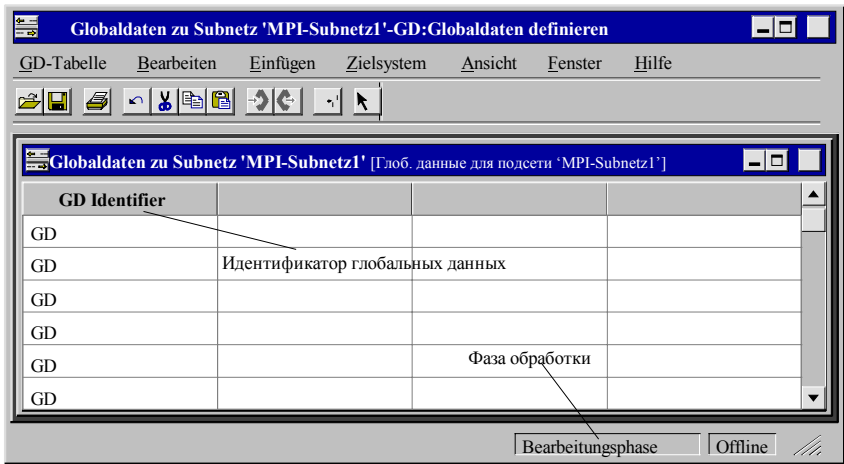


Рис. 9-2. Вид новой таблицы глобальных данных

Дальнейшая информация

Вы найдете в online-помощи для GD-связи.

Подробную информацию о структуре GD-таблицы и последовательности действий при ее заполнении, сохранении, загрузке печати

9.2. Принцип связи через глобальные данные

Принцип

Рис. 9–3 наглядно поясняет принцип GD-связи: отправляемые или принимаемые глобальные данные должны быть внесены как GD-элементы в GD-таблицу. Затем эта таблица компилируется и загружается в участвующие в обмене CPU.

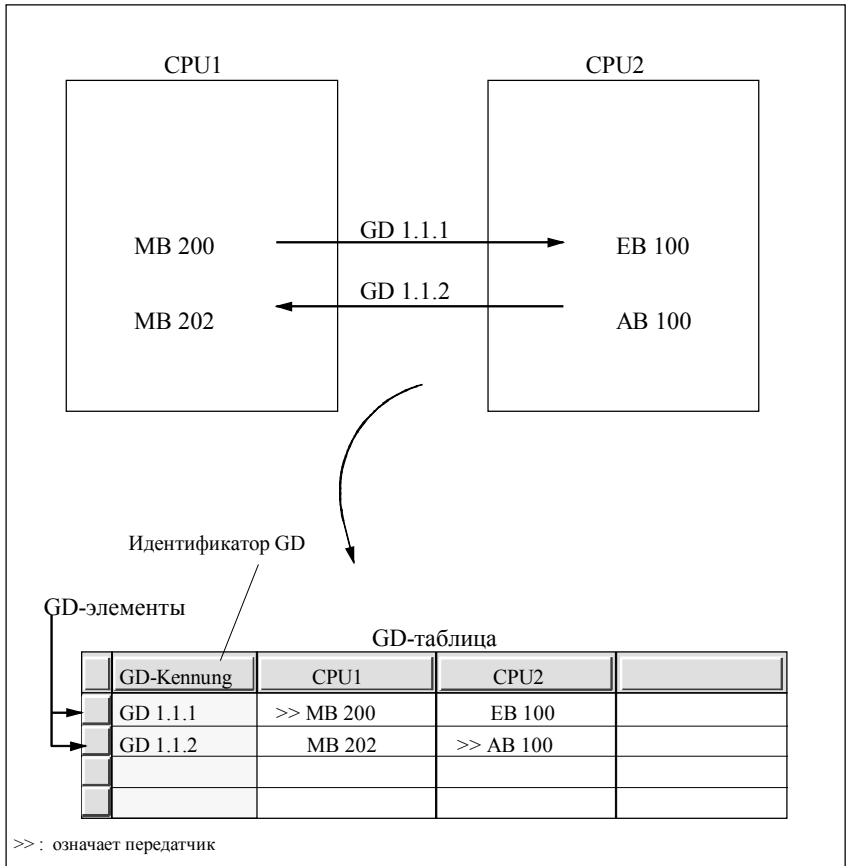


Рис. 9-3. Принцип GD-связи

Дополнительные функции GD-таблицы

GD-таблица берет на себя следующие задачи:

- Она проверяет длину GD-элементов (записей в GD-таблицу).
- Она объединяет GD-элементы в один GD-пакет, который при передаче посылается как один “пакет данных”.
- Она проверяет записи в GD-таблице на соблюдение допустимых “коммуникационных ресурсов” (величин, независимых от CPU).

Циклическая передача

Передающий CPU посылает глобальные данные в конце цикла, а принимающий CPU читает эти данные в начале цикла. С помощью **глобальных** фактора редукции, указываемого в таблице глобальных данных, можно **данных** определить, должна происходить передача или прием данных.

Передача глобальных

С помощью системных функций SFC 60 GD_SND и SFC 61 GD_RCV можно посылать или принимать GD-пакеты дополнительно или **данных**, альтернативно циклической передаче в любом месте прикладной **управляемая** программы. Предпосылкой для этого, однако, является **событиями** конфигурирование обмена данными, т.е. создание таблицы глобальных **данных**.

В качестве параметров SFC задаются номера GD-контуров и GD-пакетов, создаваемых при конфигурировании таблицы глобальных данных.

Если в таблице глобальных данных коэффициент редукции указан равным 0, то глобальные данные передаются только при вызове соответствующих SFC.

Коммуникационные ресурсы CPU

Коммуникационным ресурсом CPU называется “максимальное количество GD-контуров, которым может принадлежать CPU”. Какими “коммуникационными ресурсами” обладает Ваш CPU, можно узнать из его технических данных.

9.3. Тестирование GD-связи

Индикация состояния Для каждого GD-пакета можно определить для “соответствующих” CPU двойное слово состояния. Эти слова имеют в таблице идентификатор “GDS”.

Оценка состояния Состояние можно оценить в прикладной программе или с помощью команды PG, назначив двойное слово состояния (GDS) операнду CPU такого же формата (напр., MD 120).

Структура двойного состояния Состояние GD (GD-статус) имеет битовую структуру. Значение каждого слова установленного бита представлено на рис. 9–4. Бит остается установленным до тех пор, пока, он не будет сброшен прикладной программой или командой PG.

Биты, не приведенные на рисунке, зарезервированы и в настоящее время не имеют значения.

GD-статус занимает двойное слово; для облегчения понимания на рисунке использовано двойное слово MD 120.

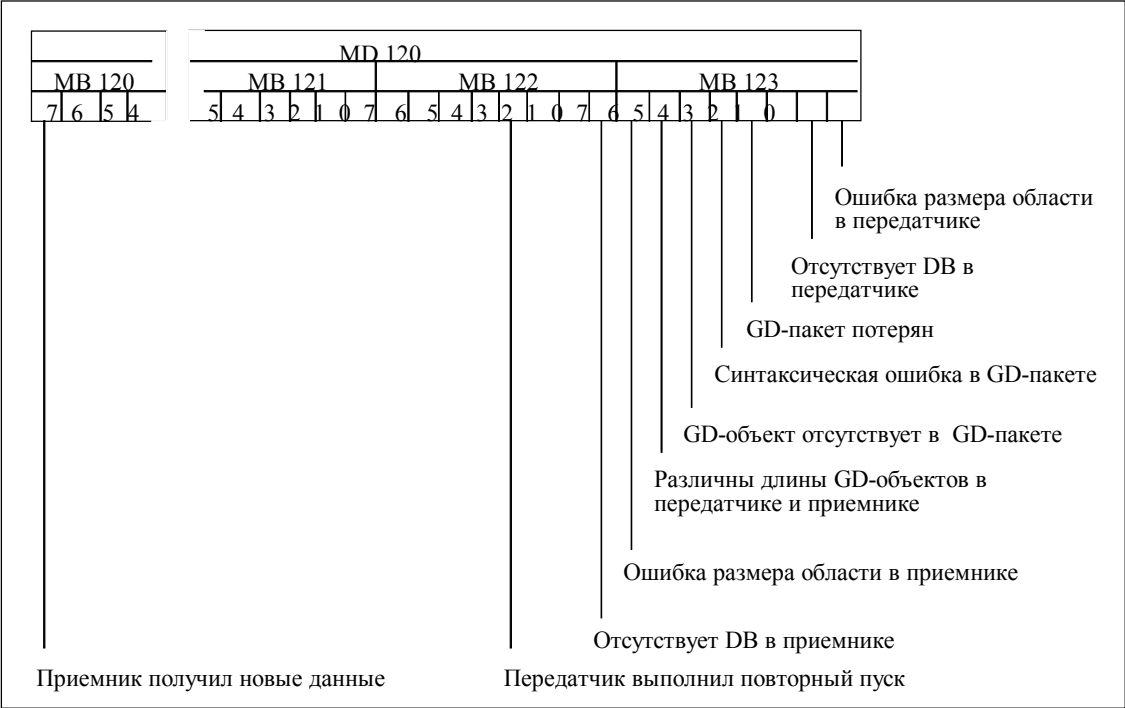


Рис. 9-4. Структура слова состояния GD

Общее состояние STEP 7 создает общий статус (GST) для всех GD-пакетов. Общий статус, тоже двойное слово со структурой, идентичной двойному слову состояния (GDS), получается с помощью логической операции ИЛИ над всеми двойными словами состояния.

9.4. Примеры конфигураций

Пример 1: Один
передает
данные
нескольким CPU

В первом примере конфигурации CPU с программным именем CPU Station1_CPU1 передает массив из 22 байт нескольким выбранным CPU в сети (см. рис. 9–5). Эта конфигурация указывает, что CPU1 станции Station 1 посылает данные от MB50 до MB71. Другие CPU принимают эти данные в такие же или другие операнды.

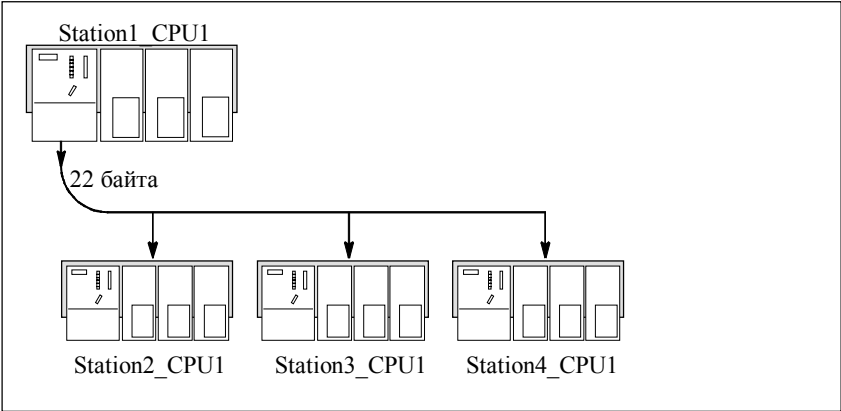


Рис.. 9-5. Пример: один CPU посылает данные нескольким CPU

Рис. 9–6 показывает таблицу глобальных данных для этой конфигурации. Таблица представляет GD-контур. Для CPU 314 можно, например, определить четыре таких конфигурации.

GD-Identifier [Идентификатор GD]	Station1_CPU1	Station2_CPU1	Station3_CPU1	Station4_CPU1
GD 1.1.1	»MB50:22	MB50:22	MB110:22	MB110:22

Рис. 9-6. Один CPU посылает данные нескольким CPU

**Пример 2: Два CPU
обмениваются
данными**

В этом примере конфигурации CPU с именем Station1_CPU1 посылает массив из 10 байт, начиная с MB80, в CPU с именем Station2_CPU1, который принимает данные в MB20 до MB29.

CPU1 станции Station2 тоже определен как передатчик и посылает 20 байт из DB10, начиная с адреса 0 в CPU1 станции Station1, который принимает данные в такие же операнды (см. рис. 9–7).

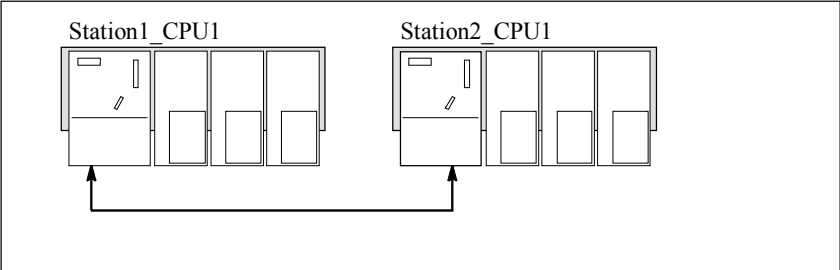


Рис. 9-7. Связь через глобальные данные

Рис. 9–8 показывает таблицу глобальных данных для этой конфигурации. Таблица представляет GD-контур. Для CPU 314, например, можно определить четыре таких конфигурации.

GD-Identifier	Station1_CPU1	Station2_CPU1
GD 1.1.1	»MB80:10	MB20:10
GD 1.2.1	DB10.DBB0:20	»DB10.DBB0:20

Рис. 9-8. Два CPU обмениваются данными

Установление коммуникационных соединений

10

Обзор

Коммуникационные соединения или, кратко, соединения требуются во всех тех случаях, когда Вы хотите осуществить в прикладной программе обмен данными с помощью коммуникационных блоков.

Соединение определяет отношение связи для двух абонентов. Благодаря определению соединения значительно упрощается программирование коммуникации. Это определение действительно для всех вызываемых коммуникационных блоков, и его не требуется выполнять каждый раз заново.

В этой главе Вы прочтете, как с помощью STEP 7 определить необходимые соединения, на какие особенности Вы должны обратить внимание и какое значение имеют соединения для дальнейшего программирования, параметризации коммуникационных блоков.

Обзор главы

В раз-деле	Вы найдете	на стр.
10.1	Коммуникационные соединения - обзор	10–2
10.2	Создание соединения	10–5
10.3	Свойства гомогенных соединений	10–7
10.4	Свойства соединений “точка-к-точке”	10–8
10.5	Коммуникационные соединения с ‘другими станциями’	10–10

10.1. Коммуникационные соединения - обзор

Введение

Установление необходимых коммуникационных соединений является условием для управляемого события обмена данными посредством программируемой коммуникации.

Коммуникационное соединение - это логическое соединение, которое характеризует

- участвующих абонентов связи
- тип соединения (напр., в S7-гомогенное, точка-к-точке, FDL или транспортное соединение ISO)
- специальные свойства (напр., остается ли соединение постоянным или динамически создается и отменяется в прикладной программе).

Зачем

устанавливаются соединения?

Соединение определяет свойства связи между двумя абонентами. STEP 7 сохраняет все свойства соединения и назначает для каждого соединения каждому партнеру по коммуникации однозначный идентификатор, так называемый ID соединения. И при параметризации коммуникационных блоков нужен только этот ID соединения; это значительно облегчает программирование.

Последовательность действий

Для создания коммуникационного соединения действуйте следующим образом.

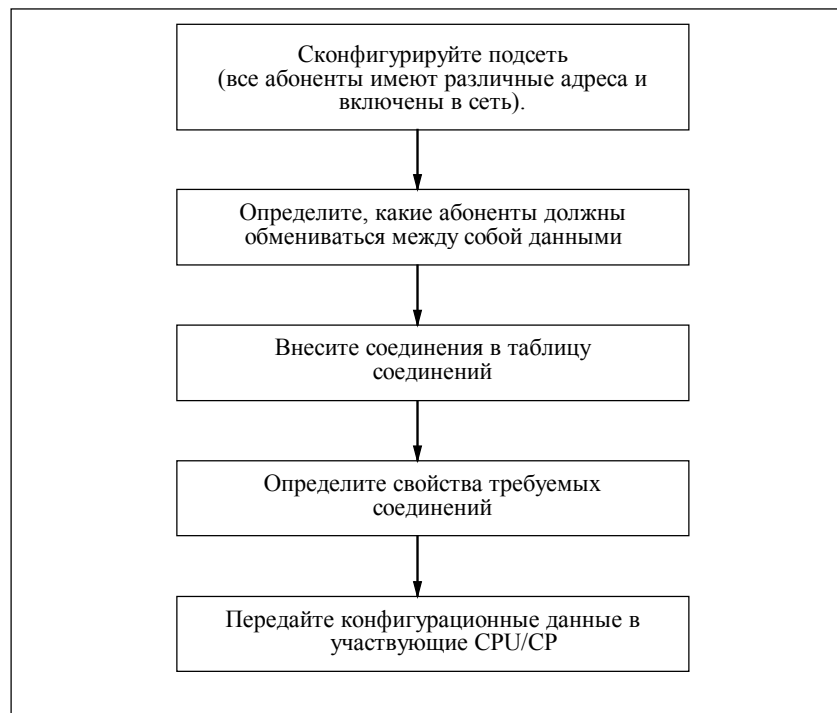


Рис. 10-1. Последовательность действий для коммуникационного соединения

Вызов таблицы

Предпосылка: Вы находитесь в своем проекте и открыли станцию, в которой находится модуль, для которого Вы хотите создать таблицу соединений.

Для вызова таблицы соединений, действуйте следующим образом:

1. Выделите объект "Verbindungstabelle (CO)" ("Таблица соединений").
2. Щелкните дважды на объекте или выберите команду меню **Bearbeiten** → **Objekt öffnen** (Редактировать → Открыть объект).

Результат: открывается окно с таблицей для проектирования соединений..

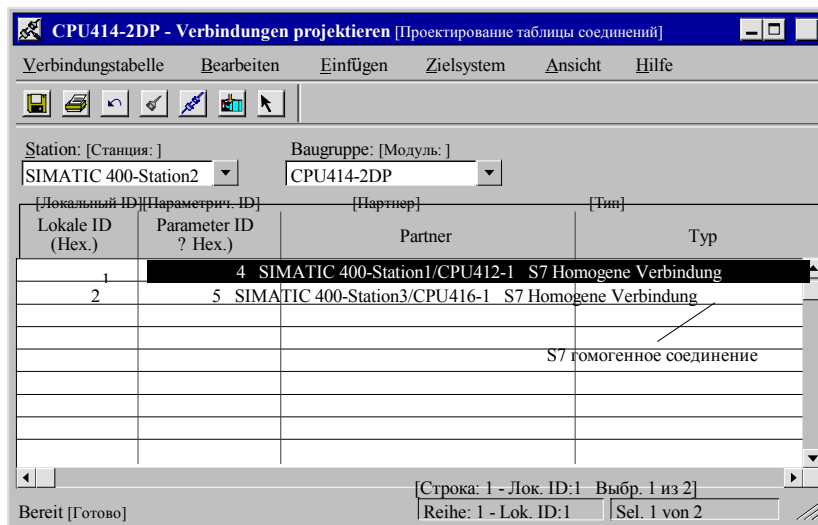


Рис. 10-2. Вид таблицы соединений

Локальный абонент

Для каждого программируемого модуля, который может быть абонентом соединения, существует собственная таблица соединений.

Выберите в полях в верхней части таблицы соединений станцию и модуль, для которых Вы хотите создать таблицу соединений (этот модуль называется также "локальным абонентом").

Одно- и двусторонние соединения

Максимальное количество программируемых соединений зависит от ресурсов используемого CPU. Необходимо различать односторонне и двусторонне проектируемые соединения.

Двусторонне проектируемое соединение использует ресурсы обоих абонентов и требуется во всех тех случаях, когда используется "парочка блоков", например, у одного абонента SFB 12 BSEND, а у другого - SFB 13 BRCV.

Напротив, односторонне проектируемое соединение использует ресурсы только одного абонента, от которого исходит соединение. Однако в этом случае можно использовать только блоки, пригодные для односторонней связи, например, SFB 14 GET.

Коммуникационные блоки Следующая таблица дает обзор используемых коммуникационных блоков и показывает, какое соединение должно быть Вами спроектировано.

Таблица 10–1. SFB и SFC для обмена данными

SFB/SFC		Краткое описание	проектируемое соединение
Функции передачи			
SFB 8 SFB 9	USEND URCV	Некоординируемый обмен данными через один передающий и один принимающий SFB	двустороннее
SFB 12 SFB 13	BSEND BRCV	Обмен блоками данных переменной длины между одним передающим и одним принимающим SFB	двустороннее
SFB 14	GET	Чтение данных из удаленного устройства	одностороннее
SFB 15	PUT	Запись данных в удаленное устройство	одностороннее
Функции управления			
SFB 19	START	Выполнение нового пуска в удаленном устройстве	одностороннее
SFB 20	STOP	Перевод удаленного устройства в состояние STOP	одностороннее
SFB 21	RESUME	Выполнение повторного пуска в удаленном устройстве	одностороннее
Функции контроля			
SFB 22	STATUS	Целенаправленный опрос состояния удаленного устройства	одностороннее
SFB 23	USTATUS	Прием сообщений о состоянии удаленного устройства	двустороннее

10.2. Создание соединения

Введение

В таблице соединений отображаются все соединения, исходящие из модуля. Сама таблица не редактируется. Ниже показано, как создать новое соединение. Подобным же способом можно изменить уже созданное соединение.

Создание нового соединения

Предпосылка: таблица соединений открыта.

Для создания нового соединения действуйте следующим образом:

- дважды щелкните в таблице соединений на пустой строке
или
- выберите команду меню **Einfügen→ Neue Verbindung** (Вставить → Новое соединение)

Результат: на экране появляется следующий диалог.

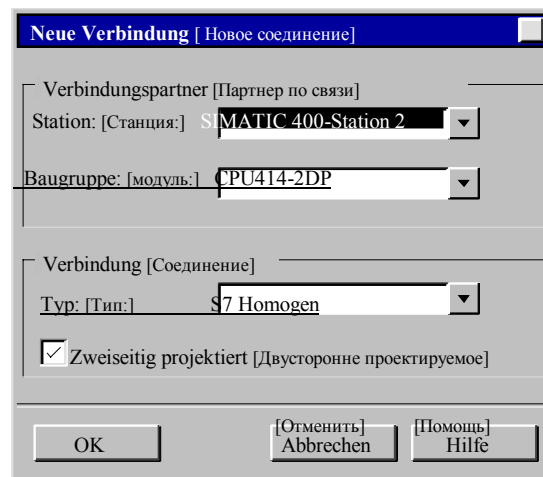


Рис. 10-3. Диалог для создания соединения

Партнеры по соединению

Выберите программируемый модуль, для которого Вы хотите создать соединение.

Если Вы хотите только зарезервировать соединение и не вносить пока партнеров, то выберите в поле "Station" ("Станция") запись "unspezifiziert" ("не определена"). Обратите внимание, что такое резервирование в настоящее время возможно не для всех соединений.

Тип соединения

Выберите в поле ввода “Тур” тип соединения, который Вы хотите применить. Возможные типы соединений показаны в следующей таблице.

Таблица 10–2. Типы соединений

Тип соединения	Пояснение
Гомогенное S7–соединение	Применяйте этот тип, если Вы хотите соединить друг с другом два модуля SIMATIC S7/M7.
Соединение “точка-к-точке”	Применяйте этот тип, если Вы хотите связать друг с другом (через соединение “точка-к-точке”) два коммуникационных модуля и применить протокол RK512 или 3964(R).
Соединение FDL	Применяйте это соединение, если Вы хотите связать модуль из SIMATIC S7/M7 с модулем S5 или модулем иной фирмы, и оптимизированный способ
Транспортное соединение ISO	передачи SIMATIC S7/M7 (S7–протокол) не может быть использован. В этих случаях для передачи данных используйте стандартный протокол, например, транспортный протокол ISO.

Локальный ID и ID партнера

После подтверждения диалога “Neue Verbindung” (“Новое соединение”) в таблице соединений появляется указанный партнер и тип соединения. Если соединение было спроектировано двусторонним, в таблицу соединений автоматически вносятся оба партнера.

Одновременно STEP 7 предоставляет для созданного соединения два идентификатора. Благодаря этим идентификаторам соединение идентифицируется однозначно.

Модуль, из которого рассматривается соединение (локальный партнер), содержит локальный ID (Lokal-ID); модуль, к которому ведет соединение, получает удаленный ID (Remote-ID).

Локальный и удаленный ID нужны при программировании коммуникационных блоков.

10.3. Свойства гомогенных соединений

Введение

Наряду с вводом в таблицу соединений, для каждого проектируемого гомогенного соединения необходимо установить специальные свойства.

Вызов диалога

Чтобы вызвать диалог для специальных свойств соединения, действуйте следующим образом:

1. Выберите таблицу соединений для желаемого соединения.
2. Выберите в меню **Bearbeiten** → **Objekteigenschaften** (Редактировать → Свойства объекта).

Результат: появляется диалог "Objekteigenschaften" ("Свойства объекта").

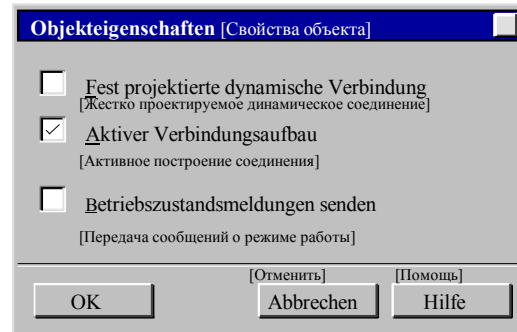


Рис. 10-4. Свойства объекта

Жестко проектируемое соединение

Активизируйте это управляющее окошко, если Вы хотите создавать и отменять соединение в прикладной программе. Это имеет то **динамическое** преимущество, что сеть загружена только тогда, когда соединение Вам необходимо.

Если это окошко не выбрано, соединение автоматически создается при запуске и сохраняется до отключения.

Активное построение соединения

Активизируйте это окошко, если соединение должен создавать локальный абонент. Построение соединения ведет к небольшому продлению времени цикла.

Если это управляющее окошко не активизировано, то построение соединения берет на себя партнер.

Передача сообщений о режиме работы USTATUS.

Активизируйте это управляющее окошко, если локальный абонент должен посылать сообщения о своем режиме работы партнеру. Сообщения о режиме работы могут быть приняты партнером через

SFB23

10.4. Свойства соединения “точка-к-точке”

Введение

Наряду с вводом в таблицу соединений для каждого проектируемого соединения “точка-к-точке” необходимо установить специальные свойства.

Вызов диалога

Чтобы вызвать диалог для специальных свойств соединения, действуйте следующим образом:

1. Выделите в таблице соединений желаемое соединение “точка-к-точке”.
2. Выберите в меню **Bearbeiten** → **Objekteigenschaften** (Редактировать → Свойства объекта).

Результат: появляется диалог ”Objekteigenschaften” (“Свойства объекта”).

Рис. 10-5. Свойства объекта для соединения “точка-к-точке”

Соединение

Наряду с описанными в гл. 10.3 управляющими окошками, Вы здесь устанавливаете и другие свойства соединения.

Канал: коммуникационные модули из SIMATIC S7 имеют несколько каналов, через которые могут быть построены соединения “точка-к-точке”. Выберите нужный канал, который будет использоваться для проектируемого соединения.

Протокол: через соединение “точка-к-точке” данные могут обмениваться с использованием протокола RK512 или протокола 3964(R). Протокол устанавливается при параметризации модуля.

Направление связи: определите, в каком направлении должна осуществляться связь, активизируя соответствующее управляющее окошко.

Локальный абонент (при RK 512) Устройства автоматизации, способные к многопроцессорной работе (напр., S5–135U/155U), могут содержать несколько CPU. Вследствие этого Вы должны указать номер CPU, через который партнер может обращаться к соединению.

Если в качестве направления связи выбрано **Partner -> Lokal** или **Lokal <-> Partner**, то введите здесь номер CPU (от 1 до 4), через который партнер будет обращаться к соединению.

Партнер (при RK512) Устройства автоматизации, способные к многопроцессорной работе (напр., S5–135U/155U), могут содержать несколько CPU. Вследствие этого Вы должны указать номер CPU, к которому должно идти соединение.

Если в качестве направления связи выбрано **Lokal -> Partner** или **Lokal <-> Partner**, то введите здесь номер CPU (от 1 до 4), к которому пойдет соединение.

10.5. Коммуникационные соединения для ‘других станций’

Введение Обмен данными через коммуникационные блоки возможен также между станциями S7 и ‘другими станциями’ (SIMATIC S5, устройство производства другой фирмы).

Построение соединения Действуйте так:

1. Создайте в проекте ‘другую станцию’ командой меню **Einfügen → Hardware → Andere Station** (Вставить → Аппаратура → Другая станция).
2. Определите свойства ‘другой станции’: станция, например, имеет один узел РТР (“точка-к-точке”) и один узел Н1.
3. Подключите другую станцию к соответствующей подсети.
4. Теперь Вы можете определить коммуникационные соединения типа “точка-к-точке”, FDL или транспортное ISO между станцией S7 и ‘другой станцией’.

Обзор

Для создания S7-программы Вам нужен подходящий язык программирования из пакета STEP 7. При этом имеется выбор между различными языками и редакторами.

При этом некоторые языки (KOP, AWL, S7-SCL) дают возможность создавать все необходимые для прикладной программы виды блоков, с помощью других (S7-GGRAPH, S7-HiGraph) можно создавать только определенные виды блоков. Независимо от того, какой (какие) редактор(ы) Вы используете, целью программирования является исполняемая прикладная программа из блоков.

Эта глава дает обзор принципиальных возможностей разработки программ. Управление отдельными редакторами, специфические для языков тестовые функции, а также синтаксис и набор команд различных языков описаны в руководствах по отдельным языкам программирования.

Обзор главы

В раз-деле	Вы найдете	на стр.
11.1	Программирование S7-CPU	11-2
11.2	Выбор языка программирования и редактора	11-4
11.3	Создание блоков с использованием KOP и AWL	11-5
11.4	Создание файлов с исходными текстами на AWL и S7-SCL	11-7
11.5	Создание блоков с использованием S7-GGRAPH	11-8
11.6	Создание исходных текстов с использованием S7-HiGraph	11-9

11.1. Программирование S7–CPU

Предпосылки

Чтобы получить возможность создания новой S7-программы, создайте в SIMATIC Manager **проект**. После этого имеется две возможности создать S7-программу, которая среди прочего содержит и прикладную программу:

- Вы создаете программу **независимо от модулей**. В этом случае вставьте объект для Вашей S7-программы непосредственно под проектом. Разработанную таким образом программу Вы лишь потом поставите в соответствие программируемому модулю станции.
- Вы редактируете S7-программу, **сопоставленную модулю** (ср. рис. 11–1). В этом случае в проекте должна быть по крайней мере одна станция SIMATIC 300 (400), которую Вы должны сконфигурировать так, чтобы она содержала программируемый модуль (S7-CPU). Вместе с CPU одновременно вставляется S7-программа.

Если Вы хотите работать с глобальными символами, прежде всего отредактируйте **таблицу символов** своей S7-программы, содержащую соответствия между абсолютными адресами и символами (именами). В этой таблице могут быть определены также символы для блоков.

Для того чтобы программа позднее могла исполняться в CPU, Вы должны, кроме того, полностью **skonфигурировать и параметризовать аппаратное обеспечение**. Это может быть сделано до или после собственно программирования.

Прикладная программа, исходные тексты и планы

S7-программу можно хранить как прикладную программу (блоки), в виде исходного текста или плана. Правда, исходные тексты и планы служат при программировании в S7 только основой для генерации блоков.

Создаете ли Вы исходный текст, блок в прикладной программе или план, зависит от выбранного языка программирования и используемого редактора.

На рис. 11–1 показано, с помощью каких редакторов какие объекты программы можно создавать.

Прикладная программа для S7–CPU

В CPU можно загружать только прикладную программу с ее блоками. В зависимости от требований сюда относятся организационные блоки (OB), функциональные блоки (FB), функции (FC) и блоки данных (DB). Определяемые пользователем типы данных (UDT) упрощают программирование, однако, они не загружаются в CPU.

Наряду с этим необходимы системные блоки данных, содержащие системную информацию (конфигурацию и параметры системы). Однако эти системные блоки данных (SDB) не нужно создавать пользователю. Они создаются и снабжаются данными с помощью STEP 7 при конфигурировании и параметризации. Все SDB в SIMATIC Manager представляются как один объект в прикладной программе.

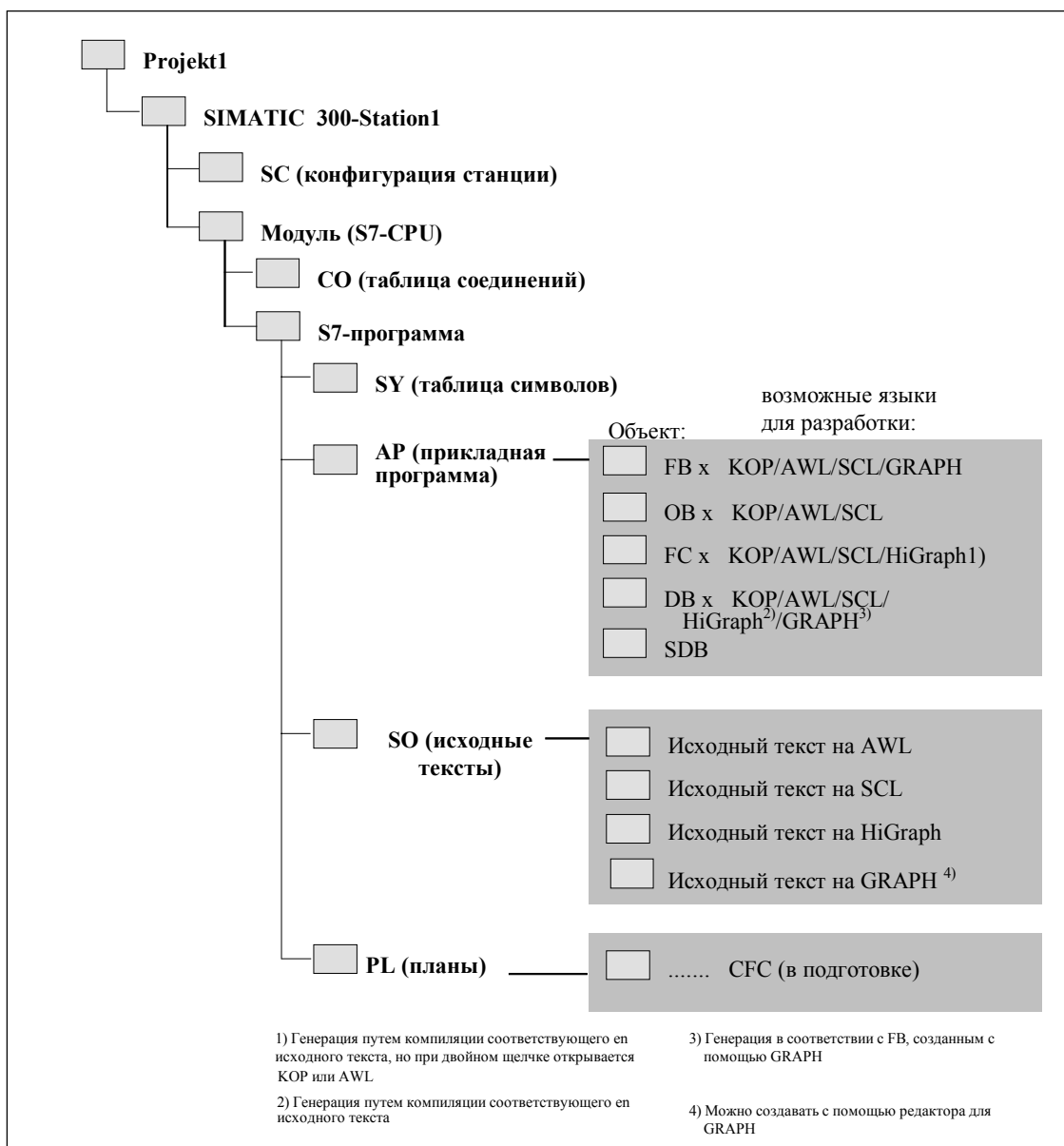


Рис. 11-1. S7-программа в структуре проекта с сопоставлением с языковыми редакторами

Дальнейшая информация

языковым пакетам.

Дальнейшую информацию о разработке программ Вы можете получить из руководства по программированию /234/ или вводных частей руководств по

Организация проектов и обращение с объектами в SIMATIC Manager описаны в главах 3, 4 и 5 данного руководства.

11.2. Выбор языка программирования и редактора

Языки программирования

Для создания S7-программы имеются в распоряжении несколько языков программирования:

- KOP (контактный план)
- AWL (список команд)
- S7-SCL (Structured Control Language - структурированный язык управления)
- S7-GRAH (управление выполнением)
- S7-HiGraph (граф состояний)

Таким образом, у Вас есть выбор между различными философиями программирования (список команд, контактный план, язык высокого уровня, управление выполнением или граф состояний) и между программированием в текстовой или графической форме.

Языки программирования KOP и AWL содержатся в базовом пакете STEP 7. Остальные языки программирования Вы можете приобрести как дополнительное программное обеспечение.

Инкрементные и текстовые редакторы

В зависимости от языка программирования в Вашем распоряжении имеются инкрементные и текстовые (ориентированные на исходный текст) редакторы:

- **Инкрементные редакторы для KOP, AWL или S7-GRAH:** в инкрементных редакторах для KOP, AWL и S7-GRAH создаются блоки, которые хранятся в прикладной программе. “Инкрементный” означает, что каждый ввод немедленно проверяется. При этом адреса и символы проверяются так же, как и специальный синтаксис языка программирования. Благодаря этому в KOP и AWL можно сохранить только блоки, свободные от ошибок, в S7-GRAH можно хранить блоки с ошибками и неполные блоки в виде графа-источника.
- **Редакторы, ориентированные на исходный текст (текстовые) для AWL, S7-SCL или S7-HiGraph:** в редакторах, ориентированных на исходный текст, создаются исходные тексты, которые затем компилируются в блоки. Для компиляции важно, чтобы Вы принимали во внимание предписанный синтаксис языка программирования. Проверка синтаксиса выполняется только по команде о проверке непротиворечивости или при компиляции блоков. Поэтому здесь можно сохранять и блоки, содержащие ошибки.

Определение языка программирования/ редактора

На каком языке и в каком редакторе Вы хотите создавать блок или файл с исходным текстом, Вы определяете при создании блока или исходного текста в свойствах объекта. В соответствии с этой записью запускается соответствующий редактор при открытии блока/файла с исходным текстом.

Запуск языкового редактора

Соответствующий текстовый редактор Вы запускаете в SIMATIC Manager двойным щелчком на соответствующем объекте (блоке, файле с исходным текстом и т.д.), командой меню **Bearbeiten → Objekt öffnen** (Редактировать → Открыть объект) или с помощью соответствующей пиктограммы на панели инструментов.

11.3. Создание блоков с использованием КОР и AWL

На чем основан КОР?

Представление на графическом языке программирования КОР напоминает коммутационные схемы. Элементы коммутационной схемы, например, размыкающие и замыкающие контакты, объединяются в сети. Одна или несколько сетей составляют раздел операторов кодового блока.

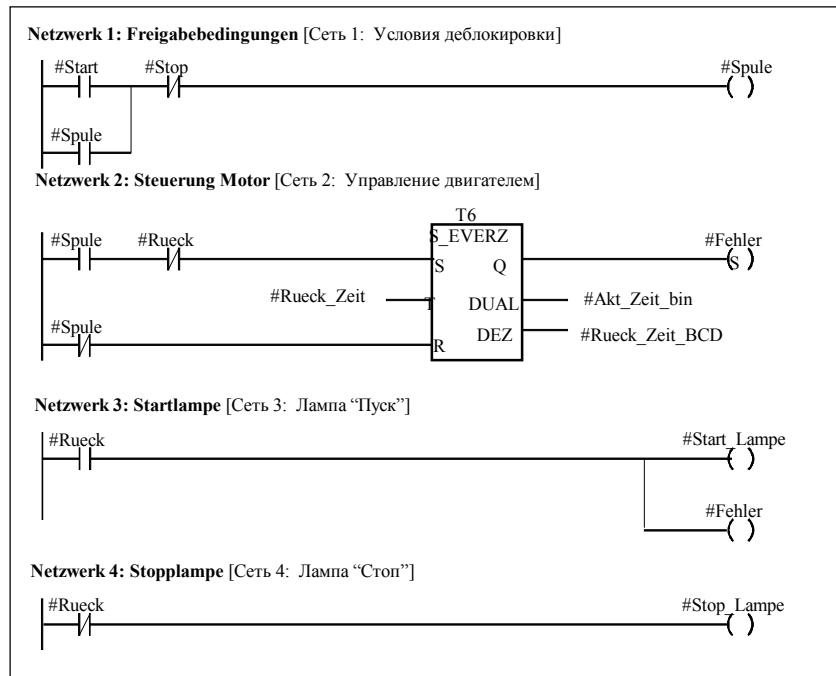


Рис. 11-2. Пример сетей в КОР

На чем основан

Язык программирования AWL (Anweisungsliste - список команд) - это **AWL?** близкий к машинному текстовый язык, команды которого выполняют простые операции. Несколько команд могут быть объединены в сети.

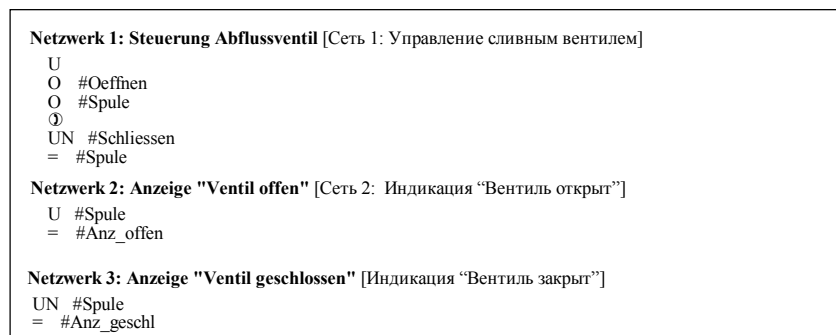


Рис. 11-3. Пример сетей в AWL

Создаваемые блоки	<p>С помощью инкрементных редакторов для KOP и AWL программируются блоки, которые хранятся в контейнере для прикладной программы. Вы можете создавать кодовые блоки (OB, FB, FC), блоки данных (глобальные DB или экземпляры DB) или типы данных, определенные пользователем (UDT). Так как блоки данных и UDT не имеют раздела операторов, язык программирования играет роль только для кодовых блоков.</p>
Связь с другими редакторами	<p>У блоков, не содержащих ошибок, можно менять представление блока на языках программирования KOP и AWL. Переход из KOP в AWL возможен всегда, из AWL в KOP - только при соблюдении определенных правил. Блоки на языке AWL можно создавать из файлов с исходными текстами и вновь переводить их в исходный текст.</p>
Доступ к символическим	<p>Если Вы не хотите программировать в абсолютных адресах, Вы можете обратиться...</p> <ul style="list-style-type: none"> • к сигналам или блокам через символы таблицы символов или • к локальным переменным и параметрам, которые Вы определяете для кодовых блоков в таблице описания переменных.

11.4. Создание файлов с исходными текстами на AWL и S7-SCL

На чем основан SCL?

Язык программирования SCL (Structured Control Language - структурированный язык управления) - это текстовый язык высокого уровня, в основном соответствующий по своему определению IEC 1131-3. Этот паскалеобразный язык благодаря командам высокого уровня упрощает по сравнению с AWL, например, программирование циклов и условных ветвлений. Поэтому SCL пригоден в частности для расчетов по формулам, для сложных оптимизационных алгоритмов или управления большими объемами данных.

```
FUNCTION_BLOCK FB20
VAR_INPUT
    ENDWERT : INT;
END_VAR
VAR_IN_OUT
    IQ1 : REAL;
END_VAR
VAR_OUTPUT
    CONTROL : Bool;
END_VAR
VAR
    INDEX : INT;
END_VAR
BEGIN
    CONTROL:=FALSE;
    FOR INDEX:= 1 TO ENDWERT DO
        IQ1:= IQ1 * 2;
        IF IQ1 >10000 THEN
            CONTROL := TRUE
        END_IF
    END FOR
END_FUNCTION_BLOCK
```

Рис. 11-4. Исходный файл на SCL (функциональный блок с циклом и условием)

На чем основан AWL?

Язык программирования AWL (список команд) - это близкий к машинному текстовый язык, команды которого выполняют простые операции. Несколько команд могут быть объединены в сети (ср. гл. 11.3).

Создаваемые исходные тексты и блоки

Текстовые редакторы для AWL или SCL дают возможность ввода программы в файл с исходным текстом. Исходные тексты хранятся в контейнере исходных текстов S7-программы как **AWL-Quelle** (Исходный текст на AWL) или **SCL-Quelle** (Исходный текст на SCL). Файл с исходным текстом может содержать код для одного или нескольких блоков. С помощью такого редактора можно создать код для **OB, FB, FC** и **UDT**

и, таким образом, для прикладной программы в целом. Соответствующие блоки создаются и сохраняются в прикладной программе только путем компиляции файла с исходным текстом.

Доступ к символическим

Если Вы не хотите работать с абсолютными адресами, Вы можете получить доступ к символам для сигналов или блоков из таблицы символов. Доступ к локальным переменным или параметрам в SCL происходит только символически, в AWL - по выбору: абсолютно или символически.

11.5. Создание блоков с использованием S7–GRAPH

На чем основан S7–GRAPH? S7–GRAPH дает возможность графического программирования систем управления исполнением в виде последовательности шагов. Сюда относится создание функциональных блоков для последовательностей шагов, определение содержания шагов и условий дальнейшего переключения (переходов). Содержание шага определяется на специальном языке (подобном AWL), переходы вводятся в виде контактного плана (ограниченный объем KOP).

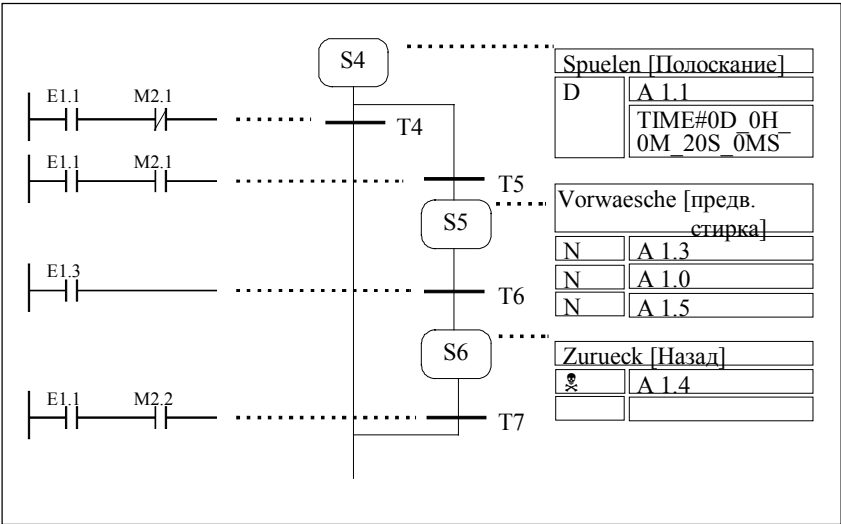


Рис. 11-5. Пример системы управления исполнением на языке Graph

Создаваемые блоки С помощью редактора для S7-GRAPH программируется функциональный блок, содержащий последовательность шагов. Данные для последовательности шагов, например, параметры FB, условия для шагов и переходов, содержит соответствующий экземпляр блока данных. В редакторе S7-GRAPH обеспечивается автоматическое создание этого экземпляра DB.

Так как созданный таким образом функциональный блок с его экземпляром DB сам по себе не является исполняемым, то Вы должны его вызвать из другого кодового блока своей прикладной программы, созданном на другом языке программирования (напр., KOP/AWL).

Доступ к символике Если Вы не хотите работать с абсолютными адресами, обращайтесь к записям в таблице символов S7-программы.

Файл с исходным текстом Из FB, созданного с помощью S7-GRAPH, может быть сгенерирован файл с исходным текстом (Graph-Quelle), который может быть интерпретирован с панели оператора или текстового дисплея для отображения управления исполнением.

11.6. Создание исходных текстов с помощью S7-HiGraph

На чем основан S7-HiGraph?

Графический редактор для S7-HiGraph дает возможность некоторых блоков программы в виде **графа состояний**. При этом установка разбивается на самостоятельные функциональные единицы, которые смотря по обстоятельствам могут принимать различные состояния. Для смены состояний определяются переходы. Операции, соответствующие этим состояниям, и условия переходов между состояниями описываются на близком к AWL языке.

Для каждой функциональной единицы создается граф, описывающий поведение этой единицы. Графы установки объединяются в группы графов. Между графами может передаваться информация. Благодаря этому может быть достигнута синхронизация функциональных единиц.

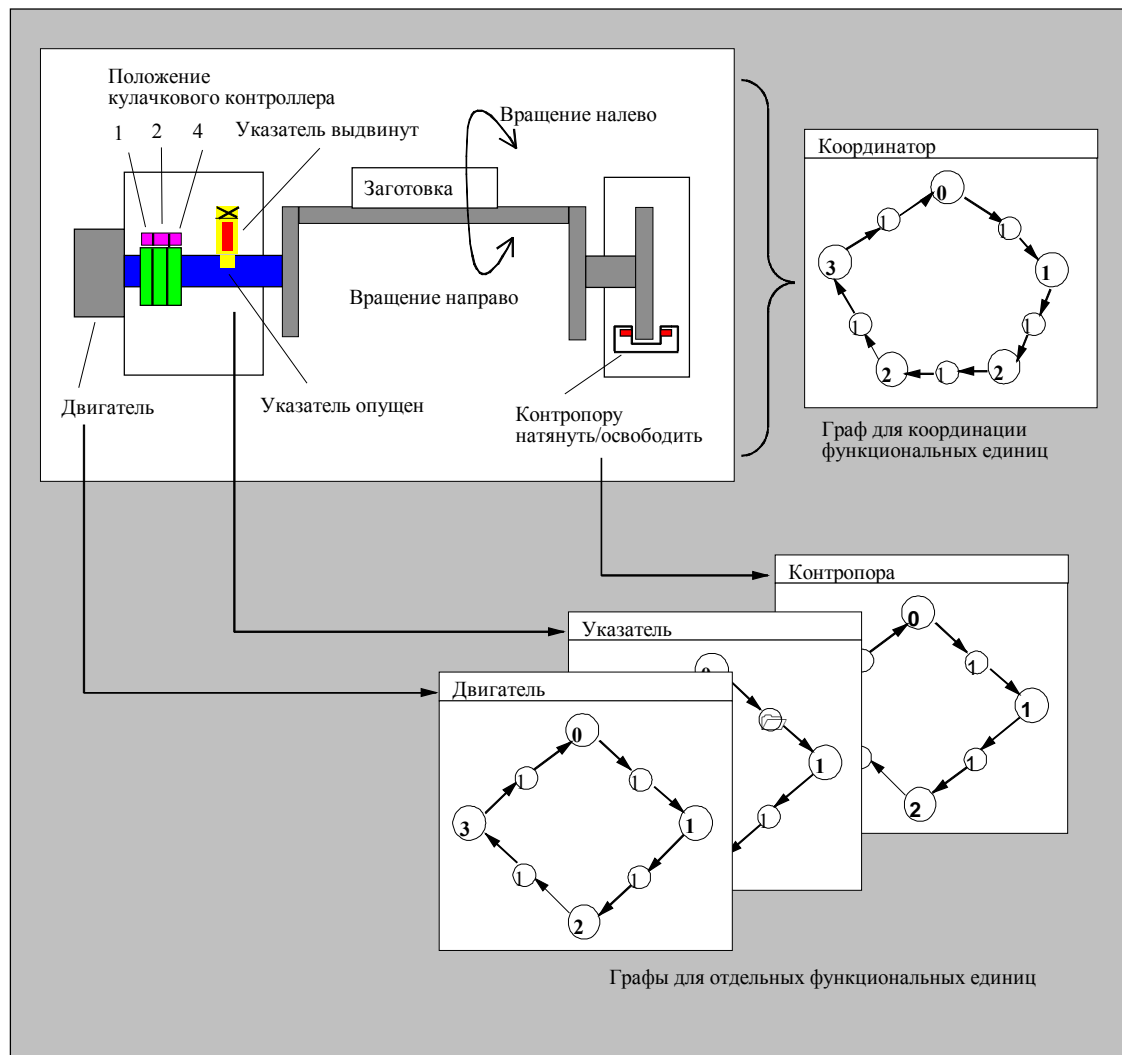


Рис. 11-6. Создание графов для функциональных единиц (пример)

Создаваемые исходные тексты и блоки

Созданные в редакторе группы графов сохраняются в контейнере исходных текстов S7-программы как **HiGraph-Quellen**. Эти группы графов затем компилируются в блоки. При этом в поставляемой в настоящее время версии на каждую группу графов создается по одной функции (FC) и одному глобальному блоку данных (DB), которые хранятся в прикладной программе. Предпосылкой для этого является внесение в таблицу символов для каждой группы графов символа для FC и DB. При этом символическое имя для FC должно соответствовать имени группы графов, а из него образуется символическое имя DB.

Синтаксис и формальные параметры проверяются только после последнего ввода для графа (с закрытием рабочего окна). Адреса и символы проверяются только при компиляции.

Так как созданная таким образом функция с блоком данных одна не может исполняться, ее нужно вызывать из другого кодового блока прикладной программы, написанного на альтернативном языке программирования (напр., KOP или AWL).

Доступ к символическим

При символической адресации обращайтесь к записям в таблице символов S7-программы.

Обзор

В этой главе Вы узнаете,

- какие справочные данные прикладной программы Вы можете отобразить и
- как приспособить к своим требованиям выведенные списки соответствующими установками фильтра.

Справочные данные применяются как обзор всей прикладной программы и как основа для изменений и тестирования. Кроме того, они могут служить для дополнения программной документации.

Обзор главы

В раз-деле	Вы найдете	на стр.
12.1	Обзор	12–2
12.2	Принципиальная последовательность действий	12–3
12.3	Отображение структуры программы	12–5
12.4	Отображение перекрестных ссылок	12–7
12.5	Отображение занятости	12–9
12.6	Отображение свободных операндов (неиспользуемых символов)	12–10
12.7	Отображение отсутствующих обозначений (отсутствующих символов)	12–11

12.1. Обзор

Какие имеются справочные данные?

Справочные данные выбранной прикладной программы включают в себя:

- структуру программы
- перекрестные ссылки
- занятость
- свободные операнды (неиспользуемые символы)
- отсутствующие обозначения (отсутствующие символы)

Можно создать и отобразить несколько списков для одной программы и/или дополнительно для другой прикладной программы.

Применение


Справочные данные применяются как обзор всей прикладной программы и как основа для изменений и тестирования. Кроме того, они могут служить для дополнения программной документации.

Следующая таблица показывает, какую информацию Вы можете получить из отдельных списков.

Таблица 12–1. Применение справочных данных

Список	Применение
Структура программы	Иерархия вызовов блоков внутри прикладной программы и обзор применяемых блоков и их зависимостей.
Перекрестные ссылки	Обзор применения операндов областей памяти E, A, M, P, T, Z и DB внутри прикладной программы.
Занятость	Обзор того, какие биты операндов областей памяти E, A и M внутри прикладной программы уже заняты, является важной основой для поиска ошибок и изменений в прикладной программе.
Свободные операнды	Обзор всех символов, которые определены в таблице символов, но внутри прикладной программы не применяются.
Отсутствующие обозначения	Обзор всех абсолютных адресов (абсолютных операндов), примененных в прикладной программе, для которых не определены символы в таблице символов.

12.2. Принципиальная последовательность действий

Отображение данных (стандартная установка).	<p>Вы запускаете приложение, открыв соответствующим языковым справочных редактором блок в прикладной программе offline. Командой меню Extras → Referenzdaten (Дополнительные функции → Справочные данные) откройте рабочее окно, в котором отобразится список перекрестных ссылки</p> <p>Эту команду можно вызвать и в SIMATIC Manager, если в представлении offline проекта выделен контейнер "Anwenderprogramm" ("Прикладная программа").</p>
Преимущество	<p>Справочные данные для выбранного Вами блока Вы можете отобразить непосредственно из языкового редактора и создать тем самым текущий обзор своей прикладной программы (offline).</p>
Генерирование данных	<p>Если Вы хотите отобразить для прикладной программы справочные базы данные в первый раз, то автоматически создается база данных.</p> <p>Если база данных уже существует, то перед отображением рабочего окна поступает запрос, нужно ли базу для справочных данных генерировать заново.</p> <p>При отказе от генерирования поступает сообщение, что данные возможно больше не совпадают с текущим состоянием прикладной программы, так как, например, после предыдущего отображения справочных данных в прикладной программе были произведены изменения.</p> <p>Для прикладной программы создается только одна база данных, даже если Вы отображали разные справочные данные из нескольких рабочих окон.</p>
Поиск записей	<p>Командой меню Bearbeiten → Suchen (Редактировать → Искать) Вы можете искать в активном рабочем окне заданные тексты. Поиск текста может производиться от позиции курсора вверх, вниз или повсюду (см. рис. 12–3).</p> <p>Обратите, пожалуйста, внимание, что в этой функции поиска речь идет о чисто текстовом поиске, т.е. искомый текст Вы должны задать точно.</p> <hr/>
	<p>Указание</p> <p>В отображенных справочных данных можно проводить дальнейший поиск, если окно поиска не активизировано, щелкнув на кнопке "Suchen" ("Поиск").</p> <hr/>
Сортировка записей в списках	<p>Записи в списках можно сортировать, щелкнув на заголовке столбца: столбцы, состоящие из букв (напр., операндов), упорядочиваются по алфавиту, цифровые столбцы - в порядке возрастания.</p>

**Представление
справочных данных**

Представление справочных данных может быть специфически изменено для каждого рабочего окна, т.е. Вы можете адаптировать содержимое списка соответственно текущим требованиям к содержанию информации.

В статусной строке рабочего окна отображается краткое описание текущих команд меню или сообщения о происходящих операциях.

Пример

В качестве примера установки свойств на рис. 12–1 представлена одна из вкладок списка перекрестных ссылок:

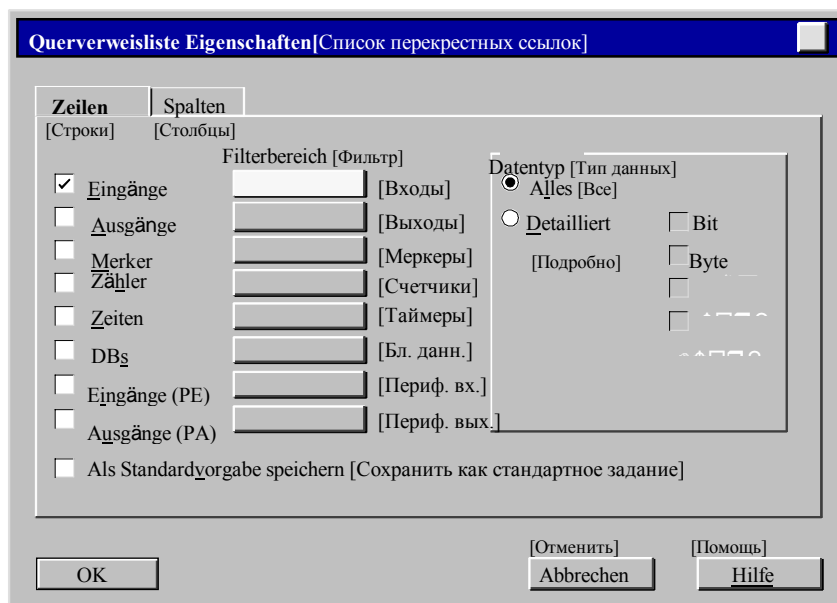


Рис. 12-1. Свойства строк для списка перекрестных ссылок

**Сохранение
установок**

Установки можно сохранить для каждого рабочего окна, отметив в диалоговом окне "Als Standardvorgabe speichern" ("Сохранить как стандартное задание"). Эти установки сохраняются до следующего выбора опции.

12.3. Отображение структуры программы

Отображение структуры программы

После Выбора прикладной программы, структуру которой Вы хотели бы отобразить, активизируйте структуру программы командой меню **Ansicht** → **Programmstruktur** (Вид → Структура программы) или задействуйте соответствующий символ на панели инструментов.

Использование

Структура программы - это графическая форма ее отображения. Дается представление иерархии вызовов блоков внутри прикладной программы и обзор примененных блоков, их зависимостей и потребности в локальных данных.

Построение структуры программы

В каждой структуре программы имеется ровно один блок, являющийся корнем структуры, так называемый стартовый блок. Этот блок определяется с помощью установок фильтра.

В структуре программы принимаются во внимание только прямые вызовы блоков. Косвенные вызовы не отображаются.

Значение применяемых в структуре программы изображений (напр., невызываемых блоков) объясняется в online-помощи.

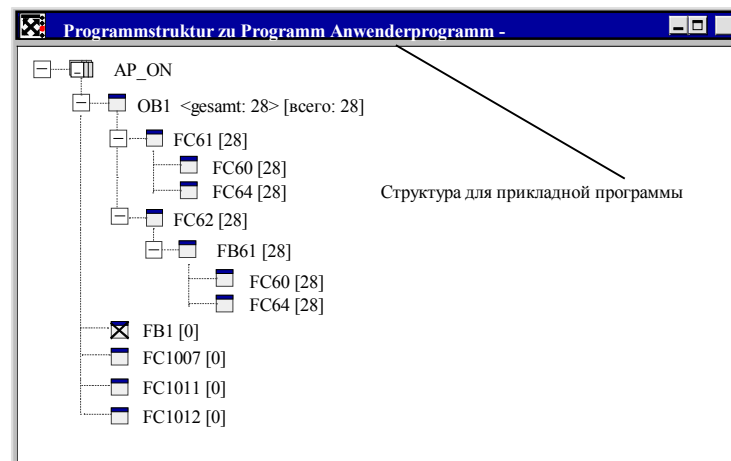


Рис. 12-2. Пример структуры программы - отображение в виде дерева

Максимальная потребность в локальных данных

Чтобы дать быстрый обзор потребности в локальных данных показанной прикладной программы, на самом верху вслед за OB1 указывается максимальная потребность в локальных данных для этой прикладной программы.

Это действительно также и для альтернативного отображения в табличной форме ("Tabellenform"). Здесь максимальная потребность в локальных данных отображается в угловых скобках вслед за соответствующим блоком.

Представление

При отображении структуры программы Вы можете выбирать между двумя формами представления:

- древовидная структура
- табличная форма

Древовидная структура

При использовании древовидной структуры отображается **полная иерархия**, т.е. все вложения вызовов блоков.

Если щелкнуть на знаке “+”, то отображаются дальнейшие ответвления, при щелчке на знаке “-” отображение минимизируется, и ответвления исчезают с экрана.

Рекурсии в вызове распознаются и обозначаются на древовидной структуре звездочкой (*) в конце цепочки вызовов блоков, содержащей рекурсию.

Табличная форма

При этой форме представления отображаются только **два уровня**:

Каждому вызову блока отводится одна строка. Вслед за вызванным блоком можно отобразить сеть, к которой принадлежит это блок, а также потребность блока в локальных данных или иерархию блоков (у ОВ1).

12.4. Отображение списка перекрестных ссылок

Отображение списка перекрестных ссылок После того как выбрана прикладная программа, для которой Вы хотели бы отобразить список перекрестных ссылок, активизируйте список перекрестных ссылок командой меню **Ansicht → Querverweise** (Вид → Список перекрестных ссылок) или задействуйте соответствующий символ на панели инструментов.

Использование Вы получаете обзор применения операндов областей памяти E, A, M, P, T, Z и DB внутри прикладной программы. Функция поиска облегчает целенаправленное нахождение определенных операндов и символов.

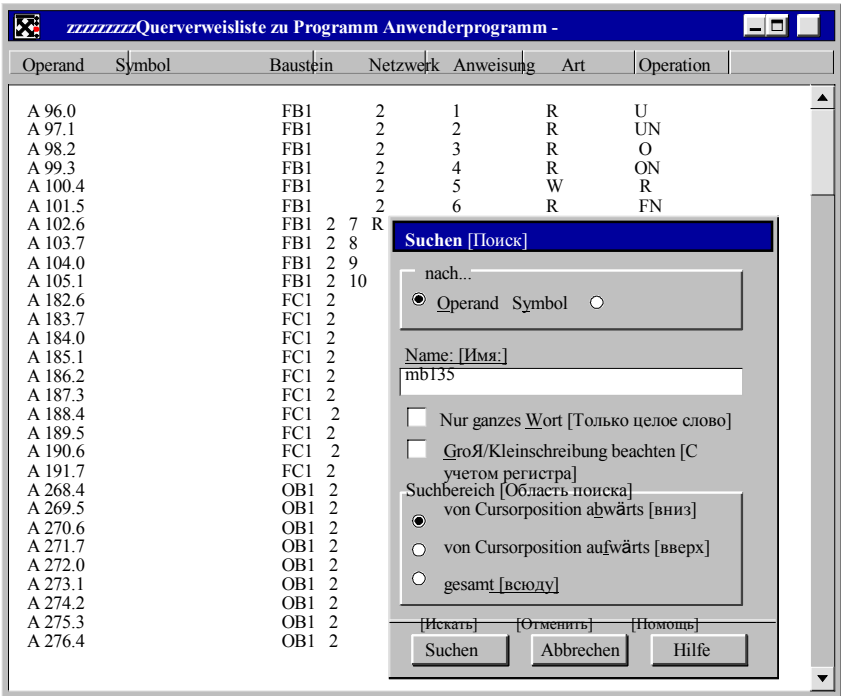


Рис. 12-3. Пример перекрестных ссылок

Структура списков перекрестных ссылок Каждая строка, выведенная в окне, соответствует записи в списке перекрестных ссылок. Он состоит из столбцов Operand (операнд), Symbol(символ), Baustein (блок), Netzwerk (сеть), Anweisung (команда), Art (вид) и Operation (операция).

Значение записей для отдельных столбцов объяснено в табл. 12–2:

Таблица 12–2. Столбцы в списке перекрестных ссылок

Столбец	Содержание/значение
OPERAND	Абсолютный адрес используемого операнда
SYMBOL	Имя используемого операнда
BAUSTEIN	Блок, в котором используется операнд
NETZWERK	Сеть блока, в которой используется операнд
ANWEISUNG	Номер команды относительно начала сети
ART	Указание на вид доступа: для чтения (<u>R</u> ead) или для записи (<u>W</u> rite)
OPERATION	Операция, в которой используется операнд

Сортировка

Предварительной установкой в списке перекрестных ссылок является сортировка по областям памяти. Щелкнув мышью на заголовке столбца, Вы произведете упорядочение по записям этого столбца.

Установки

Командой меню Ansicht → Filtern (Вид → Фильтровать) откройте регистровый диалог, с помощью которого можно установить свойства строк и столбцов в соответствии со своими требованиями. На вкладке "Spalte" ("Столбец") можно установить, какие столбцы Вы хотели бы отобразить. В соответствии с предварительной установкой отображаются все столбцы.

Ширину столбцов Вы можете приспособить к своим требованиям в списке перекрестных ссылок, выведенном на экран, с помощью мыши.

12.5. Отображение занятости

Занятость

После выбора прикладной программы, для которой Вы хотели бы отобразить занятость, активизируйте занятость командой меню **Ansicht** → **Belegung** (Вид → Занятость) или задействуйте соответствующий символ на панели инструментов.

Использование

Вы получаете обзор того, какие биты операндов областей памяти E, A и M внутри прикладной программы уже заняты. Это отображение является важной основой для поиска ошибок и изменений в прикладной программе.

Построение списка занятости

Этот список отображает только области памяти входов (E), выходов (A) и меркеров (M).

Каждая строка содержит один байт области памяти, в котором восемь битов обозначены в соответствии с видом доступа. Дополнительно еще указывается, происходит ли доступ через байт, слово или двойное слово. Смысл обозначений вида доступа объясняется в табл. 12–3.

Таблица 12–3. Обозначения доступа в плане занятости

Обозначение доступа	Значение
.	к операнду нет обращений, он не занят
○	операнд используется непосредственно
X	операнд обрабатывается косвенно (доступ через байт, слово или двойное слово)

Сортировка

Этот список рассортирован по алфавиту.

Записи можно упорядочить, щелкнув на заголовке столбца.

12.6. Отображение свободных операндов (неиспользуемых символов)

Отображение свободных операндов

задействуйте соответствующий символ на панели

После выбора прикладной программы, для которой Вы хотели бы отобразить свободные операнды, активизируйте это отображение командой меню Ansicht → Freie Operanden (Вид → Свободные операнды) или инструментом.

Использование

Вы получаете обзор всех символов, которые определены в таблице символов, но внутри прикладной программы не используются.

Структура списка свободных операндов

в столбцах объяснено в табл. 12–4.

Каждая отображенная в окне строка соответствует записи из списка. Она состоит из столбцов Symbol (символ), Operand (операнд), Typ (тип) и Operandenkommentar (комментарий к операнду). Значение отдельных записей

Таблица 12–4. Столбцы отображения свободных операндов

Столбец	Содержание/значение
Symbol	Имя используемого операнда
Operand	Абсолютный адрес используемого операнда
Typ	Тип данных операнда, напр., BOOL, INT и т.д.
Operandenkommentar	Комментарий из таблицы символов

Сортировка

Сортировку можно выполнить, щелкнув на наименовании столбца, по операндам или по символическим именам.

12.7. Отображение отсутствующих обозначений (отсутствующих символов)

Отображение отсутствующих символов

После выбора прикладной программы, для которой Вы хотели бы вывести список отсутствующих обозначений, активизируйте этот список командой меню Ansicht → Fehlende Bezeichner (Вид → Отсутствующие обозначения) или задействуйте соответствующий символ на панели инструментов.

Использование

Вы получаете обзор всех абсолютных адресов (абсолютных операндов), которые используются в прикладной программе. но для которых в таблице символов отсутствует определение символа.

Структура списка отсутствующих обозначений

Каждая отображенная в окне строка соответствует записи в списке. Она состоит из столбцов Operand (операнд) и Anzahl der Verwendungen (количество применений). Значение записей для отдельных столбцов объяснено в табл. 12–5.

Таблица 12–5. Столбцы в списке отсутствующих обозначений

Столбец	Содержание/значение
Operand	Абсолютный адрес используемого операнда
Anzahl	Указание того, как часто адрес используется в прикладной программе

Сортировка

Сортировка выполнена по операндам.

Обзор

После завершения конфигурации, параметризации и разработки программы Вы можете прикладную программу или отдельные блоки перенести в контроллер.

Для переноса в контроллер системных данных, возникших после конфигурирования аппаратного обеспечения, проектирования сетей или создания таблицы соединений, загрузите объект "Systemdatenbausteine" ("Системные блоки данных").

Вы можете пометить названные объекты в окне проекта и загрузить из Администратора SIMATIC (SIMATIC Manager) (меню: Zielsystem). Наряду с этим существует возможность для загрузки объекта через строку меню окна, в котором Вы обрабатываете содержимое названных объектов.

Обзор главы

В раз-деле	Вы найдете	на стр.
13.1	Определение и изменение рабочего режима	13–2
13.2	Концепция запоминания и загрузки	13–4
13.3	Общее стирание CPU контроллера	13–6
13.4	Загрузка прикладных программ из PG в контроллер	13–7
13.5	Загрузка блоков из PG в контроллер	13–8
13.6	Стирание блоков в CPU контроллера	13–9
13.7	Дополнительная загрузка блоков из PG в контроллер	13–10
13.8	Редактирование блоков из CPU в PG	13–11
13.9	Сжатие памяти пользователя (RAM)	13–12

13.1. Определение и изменение рабочего режима

Обзор	Рабочие режимы описывают поведение CPU в любой момент времени. Важнейшими рабочими режимами являются RUN (РАБОТА), ANLAUF (ПУСК) (новый пуск и повторный пуск) , HALT (ОСТАНОВ), и STOP. Точное определение рабочих режимов и переходов между ними Вы найдете в руководстве по программированию для S7-300/400 /234/.
Изменения режимов	Изменения рабочих режимов происходят в результате событий при исполнении программы или вмешательства пользователя.
RUN → STOP	Переведите контроллер из RUN в STOP перед <ul style="list-style-type: none">• загрузкой прикладной программы в CPU разделы 13.5, 13.4),• инициализацией общего стирания CPU (раздел 13.3),• сжатием памяти пользователя (раздел 13.9).
STOP → RUN (новый пуск)	Если Вы из состояния STOP инициализируете новый пуск, программа (новый) запускается заново и в первую очередь в режиме ПУСК обрабатывается программа запуска (в блоке OB 100). Если запуск успешен, CPU переходит в режим RUN.
	Новый пуск требуется после: <ul style="list-style-type: none">• общего стирания,• загрузки прикладной программы в режиме STOP,• переполнения USTACK/BSTACK,• устранения ошибки, после которой CPU из-за ошибки программирования в прикладной программе перешел в режим STOP,• прерывания нового пуска (из-за отключения напряжения сети или переключения режима работы на CPU),• превышение границы времени перерыва в работе для повторного пуска.
STOP → RUN (повторный пуск)	Если Вы из состояния STOP инициализируете повторный пуск, программа продолжает работу с места прерывания. Если повторный пуск успешен, то CPU переходит в режим RUN. Повторный пуск возможен только в CPU S7-400. Повторный пуск здесь допустим только тогда, когда программа не менялась в состоянии STOP.
Приоритет рабочих режимов	Если одновременно поступает несколько требований на изменение рабочего режима, то осуществляется переход в рабочий режим с наивысшим приоритетом. Последовательность приоритетов в убывающем порядке следующая: STOP – HALT – ANLAUF – RUN.

Если, например, переключатель режимов работы на CPU стоит в положении RUN, а Вы пытаетесь из PG перевести CPU в состояние STOP, то CPU переходит в STOP, так как этот режим имеет наивысший приоритет.

Наоборот, если переключатель режимов работы на CPU стоит в положении STOP, то нет никакой возможности перевести его из PG в любое другое состояние.

Определение и переключение режима работы

Действуйте следующим образом:

1. Создайте с контроллером соединение online, например,
 - переключением вида (Ansicht) окна проекта в Online или
 - через окно "Erreichbare Teilnehmer" ("Доступные абоненты"), которое можно вызвать кнопкой на панели интерфейсов Администратора SIMATIC (SIMATIC Manager).
2. Отметьте CPU.
3. Вызовите соответствующее диалоговое окно командой меню **Zielsystem → Betriebszustand...** (Контроллер → Рабочий режим ...) Отображается текущий рабочий режим.

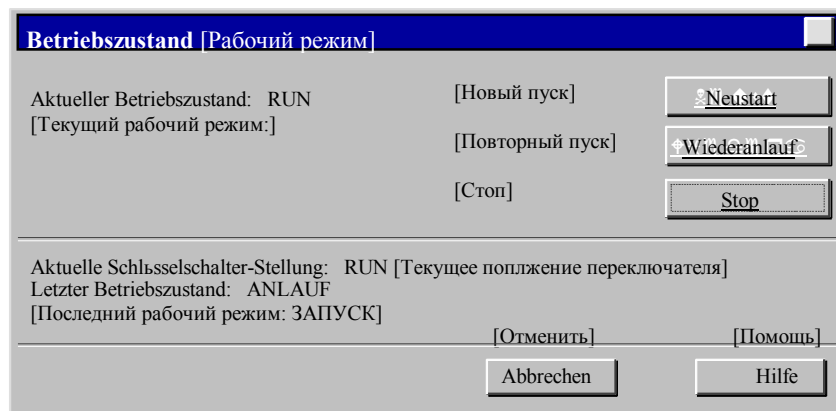


Рис. 13-1. Диалог "Рабочий режим"

4. Переключите рабочий режим нажатием соответствующей кнопки. Если некоторое изменение режима в текущей ситуации недопустимо, то соответствующая кнопка представляется с пониженной контрастностью.

13.2. Концепция запоминания и загрузки

Разделение областей памяти

Память S7-CPU можно разделить на три области:

- **Загрузочная память** служит для приема прикладной программы без таблицы символов и комментариев (они остаются в области памяти PG). В случае загрузочной памяти речь может идти об ОЗУ или СППЗУ. Блоки, помеченные как неисполняемые, принимаются исключительно в загрузочную память.
- **Рабочая память** (встроенное ОЗУ) служит для приема исполняемых частей прикладной программы. Обработка программы происходит исключительно во взаимодействии с рабочей и системной памятью.
- **Системная память** содержит дополнительные элементы памяти, которые каждый CPU предоставляет в распоряжение прикладной программе, например, отображение процесса на входах и выходах, меркеры, таймеры и счетчики. Кроме того, здесь содержатся стеки блоков, прерываний и локальных данных.

Особенности S7-300

Загрузочная память кроме встроенного ОЗУ может иметь и встроенное ЭСППЗУ.

Особенности S7-400

Использование платы памяти (ОЗУ или ЭСППЗУ) для расширения памяти пользователя недопустимо. Встроенная загрузочная память представляет собой ОЗУ и служит в сущности для повторной загрузки и коррекции блоков.



Указание

В связи с описанным выше могут применяться СППЗУ с идентификатором приложения MC5+_BST. Идентификатор приложения наносится автоматически при стирании платы памяти в PG для новой записи. СППЗУ с идентификатором приложения M7-DOS предусмотрены для использования в системах M7 и ведут себя как дисководы DOS. Идентификатор приложения наносится автоматически, когда Вы форматируете плату памяти программой **ftlforms** (возможно только на PG 720/740/760 или в системе M7).

Следствия из структуры загрузочной памяти

Разделение загрузочной памяти на ОЗУ и ЭСППЗУ влечет за собой последствия для возможностей загрузки прикладной программы или блоков прикладной программы. В табл. 13–1 показаны возможности загрузки:

Таблица 13–1. Структура загрузочной памяти и возможности загрузки

Вид памяти	Возможности загрузки	Вид загрузки
ОЗУ	Загрузка и стирание отдельных блоков	Связь PG–контроллер
	Загрузка и стирание всей прикладной программы	Связь PG–контроллер
	Дополнительная загрузка отдельных блоков	Связь PG–контроллер
Встроенное (только в S7–300) или вставляемое СППЗУ	Загрузка всей прикладной программы	Связь PG–контроллер
Вставляемое СППЗУ	Загрузка всей прикладной программы	Внешняя загрузка СППЗУ и вставление платы памяти

Программы, хранящиеся в ОЗУ, теряются при общем стирании CPU или вытаскивании платы памяти с ОЗУ.

Программы на платах памяти ЭСППЗУ при общем стирании не теряются и могут переноситься путем вставления и вытаскивания платы памяти.

Реманентность

Чтобы предотвратить потерю данных при отключении питания, можно создать реманентные области памяти. Дальнейшую информацию по этому вопросу Вы найдете в главе “Области памяти S7-CPU” Руководства по программированию для S7-300/400 /234/, а также в online-помощи по параметризации CPU.

13.3. Общее стирание CPU контроллера

Использование

Перед загрузкой прикладной программы следует произвести общее стирание CPU, чтобы обеспечить отсутствие в CPU “старых” блоков.

В качестве побочного эффекта при общем стирании можно также загрузить в CPU с платы памяти новую прикладную программу (см. ниже “Результат”).

Последовательность Командой меню **Zielsystem → Urlöschen...** (Контроллер → Общее действий

стирание...) можно произвести общее стирание CPU с PG в режиме online.

Для этого CPU должен находиться в режиме STOP.

Функция общего стирания может быть запущена также переключателем режимов работы на CPU.

Если Вы хотите произвести общее стирание CPU с помощью STEP 7, действуйте следующим образом:

1. Выберите CPU в окне проекта online или в окне “Erreichbare Teilnehmer” (“Доступные абоненты”) (вызывается через панель интерфейсов).
2. Переведите CPU в STOP через диалог, вызываемый командой меню **Zielsystem → Betriebszustand...** (Контроллер → Режим работы...).
3. Вызовите функцию “Общее стирание” командой меню **Zielsystem → Urlöschen** (Контроллер → Общее стирание...).

Результат общего стирания

При общем стирании в CPU происходит следующий процесс:

- CPU сбрасывается, и вся прикладная программа в рабочей памяти и в ОЗУ загрузочной памяти стирается.
- Системные параметры, а также параметры CPU и модулей сбрасываются на установки по умолчанию.
- CPU стирает все существующие соединения.
- Если данные находятся на СППЗУ (на плате памяти или встроенном СППЗУ), то после общего стирания CPU копирует содержимое СППЗУ в область ОЗУ.

Диагностический буфер и часы не сбрасываются.

При загрузке прикладной программы вся необходимая информация переносится в контроллер.

Параметры интерфейса MPI

Если при общем стирании вставлена плата памяти, то действительны параметры MPI, находящиеся на плате памяти.

Если плата памяти не вставлена, то сохраняются и действительны параметры MPI на модуле. Благодаря этому сохраняется коммуникационная способность модуля.

13.4. Загрузка прикладных программ из PG в контроллер

Применение

Вся прикладная программа загружается в контроллер через SIMATIC Manager, например, в конечной фазе тестирования программы или запуска на исполнение готовой прикладной программы.



Указание

Прикладные программы можно загружать в CPU в режимах STOP и RUN-P. Обратите, однако, внимание на то, что при загрузке в режиме RUN-P программа переносится поблочно. Таким образом, если Вы переписываете “старую” программу, то в процессе загрузки могут возникнуть конфликты.

Функция

Вся прикладная программа загружается в загрузочную память, а исполняемые части программы - также в рабочую память.

Рис. 13-2 показывает загрузку программ из PG в CPU:

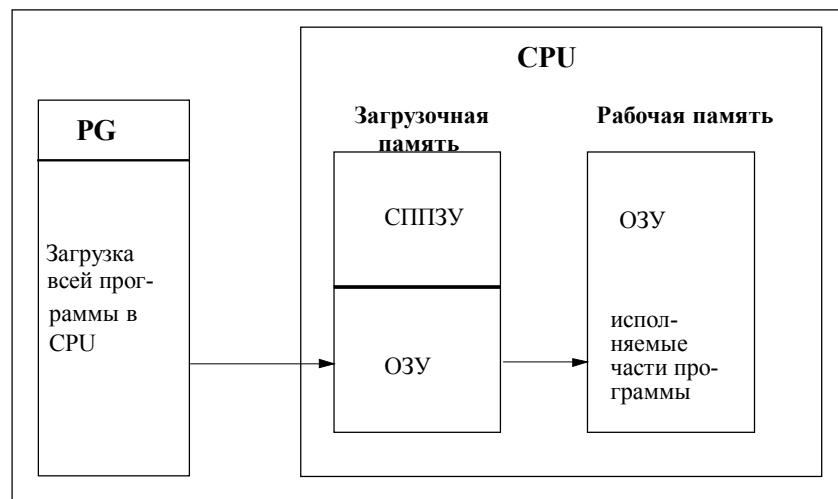


Рис. 13-2. Загрузка прикладной программы из PG в CPU

Предпосылки

Для загрузки должны быть выполнены следующие предпосылки:

- Существует связь между Вашим PG и контроллером.
- Программа, подлежащая загрузке, скомпилирована без ошибок.

Последовательность действий Для загрузки программы из PG в CPU пометьте в окне проекта прикладную программу, которую Вы хотите загрузить, а затем вызовите команду меню **Zielsystem → Laden** (Контроллер → Загрузить).

13.5. Загрузка блоков из PG в контроллер

Применение

В фазе разработки и тестирования часто возникает необходимость загружать отдельные кодовые блоки и блоки данных в CPU, чтобы потом запустить их на исполнение в условиях тестирования.



последовательность вызова блоков. CPU переходит в STOP, если он пытается вызвать отсутствующие блоки.

Указание

Блоки из PG с CPU можно загружать в режиме STOP или RUN-P. Однако, если в режиме RUN-P Вы изменяете параметры блока, то CPU переходит в STOP. При загрузке в режиме RUN-P обратите также внимание на

Функция

Отдельные блоки всегда загружаются в ОЗУ загрузочной памяти. исполняемые части блоков загружаются также в рабочую память.

Объем загрузки

Для тестирования необходимо загрузить по меньшей мере OB, а также вызываемые в нем FB и FC и используемые DB.

Загрузка в Администраторе SIMATIC

Командой меню **Zielsystem → Laden** (Контроллер → Загрузить) можно загрузить в контроллер помеченные компоненты прикладной программы.

Можно загрузить в контроллер отдельно блоки или объект "Systemdatenbausteine" ("Системные блоки данных"). Блоки, подлежащие загрузке, должны быть скомпилированы без ошибок.

Если блок уже находится в ОЗУ CPU, то при загрузке будет задан вопрос, перезаписывать этот блок или нет.

Загрузка при редактировании отдельных объектов

При конфигурировании аппаратного обеспечения и при программировании блоков Вы можете загрузить редактируемый в данный момент объект непосредственно через меню в главном окне, относящемся к Вашей деятельности (команда **Zielsystem → Laden in AS** (Контроллер → Загрузить в AS)).

Например, Вы можете загрузить в контроллер открытый в данный момент блок в окне "Bausteine programmieren" ("Программирование блоков") (команда меню **Zielsystem → Laden in AS** (Контроллер → Загрузить в AS)).

13.6. Стирание блоков в CPU контроллера

Применение	Стирание блоков в CPU может стать необходимым в фазе тестирования прикладной программы.
Блоки в CPU	Блоки хранятся в памяти пользователя CPU в СППЗУ или в ОЗУ (в зависимости от CPU и процесса загрузки).
Стирание блоков из СППЗУ	<p>Блоки, хранящиеся в СППЗУ, можно</p> <ul style="list-style-type: none">• удалить путем<ul style="list-style-type: none">- стирания содержимого платы памяти и- общего стирания CPUили• объявить недействительными (в CPU со встроенным СППЗУ), загрузив в ОЗУ CPU блок с тем же именем и затем его стерев.
Стирание блоков	Блоки, хранящиеся в ОЗУ, можно стереть непосредственно, удалив их из из ОЗУ открытого CPU. Занятые ячейки в загрузочной и в рабочей памяти освобождаются.



Указание

Блоки можно стирать в режимах STOP и RUN-P.
Однако при стирании в режиме RUN-P обратите внимание на следующее: если прикладная программа пытается получить доступ к стертому блоку, то CPU переходит в STOP или вызывается OB ошибок.

Последовательность действий Чтобы стереть блоки непосредственно в CPU, откройте окно проекта, переключите Вид (Ansicht) на online и пометьте подлежащие стиранию блоки в окне проекта online. Затем выберите команду меню **Datei → Löschen** (Файл → Стереть) или нажмите клавишу DEL.

13.7. Дополнительная загрузка блоков из PG в контроллер

Применение	Блоки, находящиеся в загрузочной памяти (ОЗУ или СППЗУ) или в рабочей памяти, можно переписать новой версией (дополнительно загрузить). При этом имеющееся состояние становится недействительным.
Блоки в ОЗУ	Имеющийся в ОЗУ блок при дополнительной загрузке стирается, а измененный блок заново загружается в ОЗУ. При этом в загрузочной и рабочей памяти возникают пустые места (см. раздел 13.9 “Сжатие памяти пользователя (ОЗУ)).
Блоки в СППЗУ	Блок, находящийся в СППЗУ, не может быть стерт, но при дополнительной загрузке он становится недействительным. Заменяющий его блок загружается в ОЗУ.
Последовательность действий	Она соответствует последовательности действий при загрузке блока (см. раздел 13.5).



Указание

Обратите, пожалуйста, внимание, что при небуферизованном отключении напряжения сети или при общем стирании “старые” блоки снова становятся действительными.

13.8. Редактирование блоков из CPU в PG

Применение

Возможность редактирования блоков в PG дает следующие преимущества:

- на этапе тестирования можно корректировать блок прямо в CPU и документировать результат
- через функцию загрузки можно текущее содержимое блоков загружать из ОЗУ загрузочной памяти CPU в PG.

Различные случаи

При загрузке блоков из CPU в устройство программирования необходимо различать два случая:

- первый случай: прикладная программа, к которой относятся блоки, находится в устройстве программирования
- второй случай: прикладная программа, к которой относятся блоки, не находится в устройстве программирования. Следствием этого является то, что части программы, которые не могут быть загружены в CPU, становятся недоступными. такими частями программы являются:
 - таблица символов с символическими именами для операндов, а также комментарии
 - комментарии к сети программы на языке KOP
 - комментарии к строкам программы на AWL
 - типы данных, определенные пользователем

Прикладная программа в PG

Для загрузки блоков из CPU в PG, в котором находится соответствующая прикладная программа, откройте окно проекта и включите Вид (Ansicht) в online. Если теперь в окне проекта online пометить модуль, то отобразится список загруженных блоков. Теперь Вы можете выбрать, открыть и редактировать блок. Командой меню **Datei → Speichern unter...** (Файл → Сохранить как...) можно сохранить изменение на PG, а командой **Zielsystem → Laden in AS** (Контроллер → Загрузить в AS) можно загрузить измененный блок в контроллер.

Прикладной программы нет в PG

Для загрузки блоков из CPU в PG, в котором нет соответствующей прикладной программы, действуйте следующим образом: щелкните на кнопке "Erreichbare Teilnehmer" ("Доступные абоненты") в SIMATIC Manager. Выберите из появившегося на экране списка CPU и откройте прикладную программу, чтобы отобразить блоки. Теперь Вы можете открывать блоки и при необходимости их редактировать, контролировать или копировать. Затем вызовите команду меню **Datei → Speichern unter...** (Файл → Сохранить как...) и внесите в соответствующем диалоговом окне путь, обозначающий желаемое место памяти на PG.

13.9. Сжатие памяти пользователя (ОЗУ)

Использование

При многократном стирании и дополнительной загрузке модулей в памяти пользователя (загрузочной и рабочей) возникают пустые места, уменьшающие полезный объем памяти. Благодаря сжатию имеющиеся блоки размещаются в памяти пользователя без пробелов и в памяти появляется связанное свободное пространство. На рис. 13–3 схематически показано, как занятые блоки памяти сдвигаются функцией "Speicher komprimieren" ("Сжатие памяти").

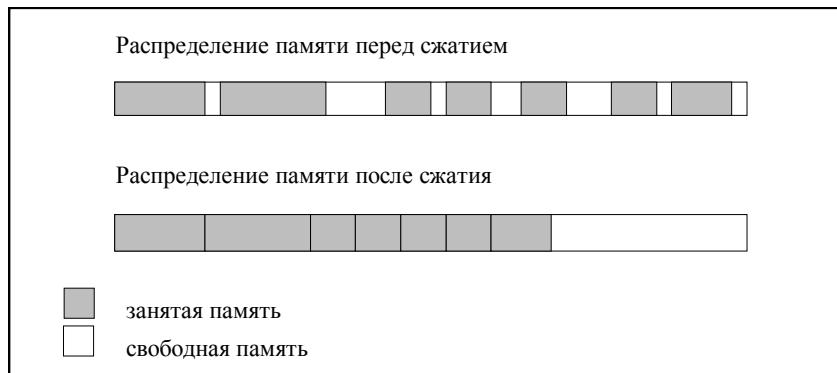


Рис. 13-3. Сдвиг занятых блоков памяти с помощью функции "Speicher komprimieren" ("Сжатие памяти")

Возможности для сжатия

Для сжатия памяти пользователя имеются две возможности:

- если при загрузке обнаруживается нехватка памяти в контроллере, то появляется диалоговое окно, указывающее на появление ошибки. Щелкнув на соответствующей кнопке в диалоговом окне, можно сжать память
- в качестве предупредительной меры можно отобразить загрузку памяти и сжать ее, если в распределении памяти есть пустые места..

Выбор рабочего режима STOP

Все пробелы в памяти можно закрыть только сжатием в режиме STOP. Если сжатие производится в режиме RUN–P (позиция переключателя режимов работы), то непосредственно обрабатываемые блоки не сдвигаются, так как они открыты. В режиме RUN (позиция переключателя режимов работы) функция сжатия не выполняется (защита от записи!).

"Упреждающее"

На вкладке "Speicher" ("Память") диалогового окна, вызываемого **сжатие** командой меню **Zielsystem → Baugruppenzustand...** (Контроллер → Состояние модулей...), можно отобразить загрузку памяти и, при известных условиях, "подтолкнуть" сжатие (см. раздел 15.7).

Обзор

Эта глава информирует о том, как

- отобразить текущие значения переменных из своей прикладной программы или CPU,
- назначать переменным прикладной программы или CPU жесткие значения (управлять),
- создать таблицу переменных для переменных, которые Вы хотите отобразить или которыми хотите управлять.

При пуске установки в эксплуатацию Вы получаете эффективную поддержку и быстрый обзор состояния переменных Вашей системы и текущего состояния CPU.

Обзор главы

В раз-деле	Вы найдете	на стр.
14.1	Введение	14–2
14.2	Создание таблицы переменных	14–3
14.3	Редактирование таблицы переменных	14–4
14.4	Наблюдение и управление переменными	14–6

14.1. Введение

Применение

К переменным, которые можно отобразить и которыми можно управлять, относятся: входы, выходы, меркеры, таймеры, счетчики, периферийные входы, периферийные выходы и элементы блоков данных.

- Если прикладная программа уже скомпилирована и загружена в CPU, то опросы отдельных блоков данных и переменных нужны, например, для того, чтобы
 - пустить в эксплуатацию установку или ее часть,
 - протестировать взаимодействие с другими частями установки или прикладной программы.
- Если прикладная программа еще не загружена в CPU, то можно дополнительно проверить правильность структуры аппаратного обеспечения (напр., наличие / отсутствие периферии).



Осторожно

Перед выполнением функции "Steuern" ("Управление") убедитесь в невозможности возникновения опасных ситуаций!

Выполнение функции управления возможно только тогда, когда переключатель режимов работы на CPU установлен на "RUN-P".

Использование

Умелой установкой точки и условий контроля можно быстро получить обзор

- состояния переменных системы,
- текущее состояние CPU.

Последовательность действий Для использования этих функций действуйте следующим образом:

1. Создайте новую или откройте уже существующую таблицу переменных.
2. Отредактируйте или проверьте таблицу переменных.
3. Создайте online-соединение между текущей таблицей переменных и желаемым CPU командой меню **Zielsystem → Verbindung herstellen zu → ...** (Контроллер → Создать соединение с → ...).
4. Выберите командой меню **Variable → Trigger** (Переменная → Пуск) желаемую точку контроля и установите условие контроля.
5. Команды меню **Variable → Status beobachten** (Переменная → Наблюдение состояния) и **Variable → Steuern** (Переменная → Управление) включается и вновь выключается соответствующая функция.
6. Сохраните созданную таблицу переменных командой меню **Tabelle → Sichern** (Таблица → Сохранить) или **Tabelle → Sichern als** (Таблица → Сохранить как), чтобы ее можно было вызвать в любое время.

14.2. Создание таблицы переменных

Применение В таблицу переменных вносятся переменные, за которыми Вы хотите наблюдать или которыми Вы хотите управлять.

Использование Таблицу переменных можно сохранять, распечатывать и вновь использовать. Вы можете установить,

- в каком формате (двоичном, десятичном, шестнадцатиричном и т.д.) должно отображаться значение переменной и
- какими значениями должны управляться выбранные переменные.

Последовательность действий Для создания таблицы переменных имеются следующие возможности:

- Выберите в SIMATIC Manager прикладную программу offline и создайте соответствующий объект командой меню **Einfügen → S7-Baustein → Variablentabelle** (Вставить → S7-блок → Таблица переменных). В последующем диалоге Вы можете дать таблице имя. Таблицу переменных можно открыть двойным щелчком на объекте.
- Выберите в представлении online S7-программу. Командой меню **Zielsystem → Variable beobachten/steuern** (Контроллер → Наблюдать/управлять переменной) создайте таблицу переменных без имени.
- Выберите соединение из списка доступных абонентов. Командой меню **Zielsystem → Variable beobachten/steuern** (Контроллер → Наблюдать/управлять переменной) создайте таблицу переменных без имени.
- Если Вы уже работаете в окне "Variable beobachten und steuern" ("Наблюдение и управление переменными"), Вы можете командой меню **Tabelle → Neu** (Таблица → Новая) создать новую таблицу переменных, которая еще не поставлена в соответствие ни одному из CPU. Уже существующие таблицы открывайте командой меню **Tabelle → öffnen** (Таблица → Открыть).
- Если Вы уже работаете в окне "Variable beobachten und steuern" ("Наблюдение и управление переменными"), Вы можете также использовать соответствующие символы на панели инструментов для создания и открытия таблицы переменных.



Указание

Таблица переменных "VAT0" зарезервирована для внутренних целей и поэтому не может быть предоставлена.

14.3. Редактирование таблицы переменных

Пример таблицы переменных На рис. 14–1 показан пример заполненной таблицы переменных.

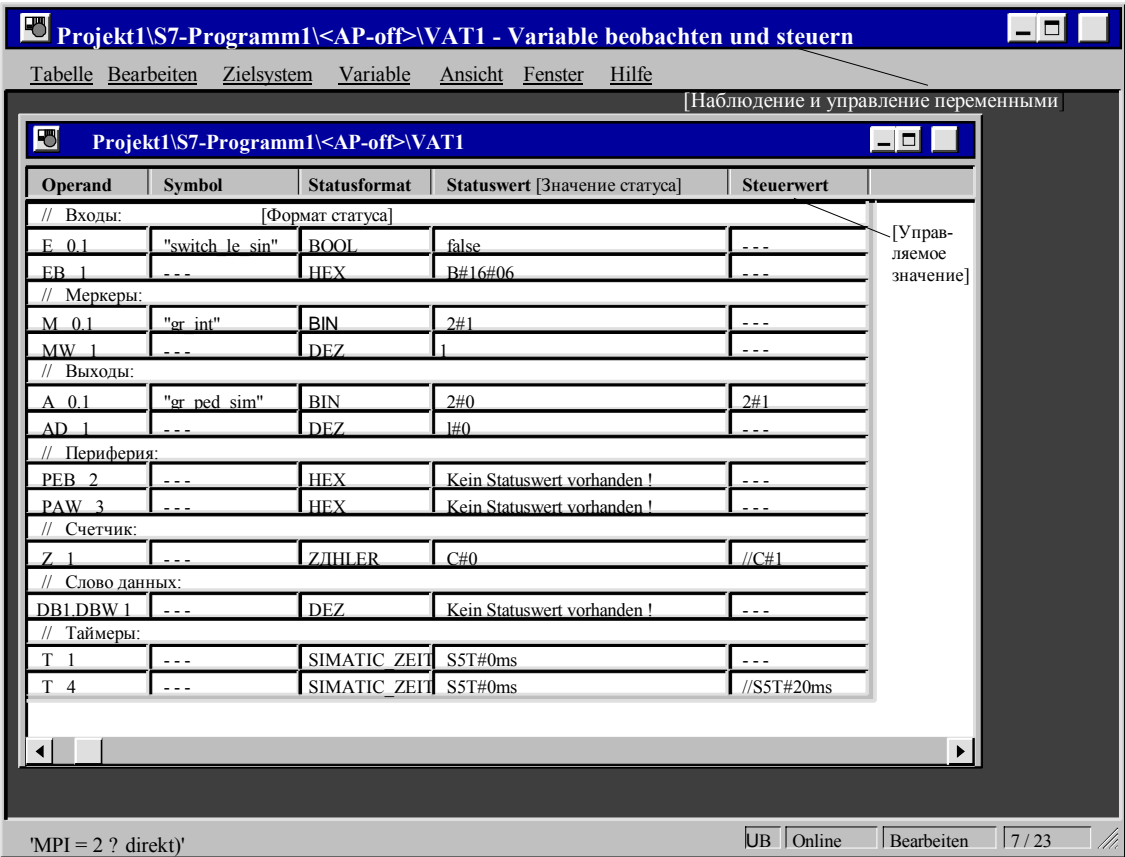


Рис. 14-1. Пример заполненной таблицы переменных

Редактирование таблицы

Внутри таблицы переменных Вы можете редактировать поля Operand (операнд), Symbol (символ), Statusformat (формат статуса) и Steuerwert (управляемое значение).

- Переменная, подлежащая управлению, задается с помощью операнда или символа. Если соответствующий символ определен в таблице символов, то запись в столбце Symbol (символ) или в столбце Operand (операнд) автоматически дополняется.
- Statusformat (формат статуса) определяет, в каком формате должно отображаться выявленное значение в столбце справа рядом. Формат выбирается командой меню **Ansicht → Statusformat wählen → ...** (Вид → Выбор формата статуса →...) или многократными щелчками мыши на поле таблицы, пока не появится желаемый формат.

Проверка синтаксиса	При внесении переменных в таблицу переменных перед покиданием строки производится проверка синтаксиса. Если при внесении переменной была сделана синтаксическая ошибка, то появляется сообщение об ошибке.
Строки комментариев	Строки комментариев начинаются знаком комментария “//”. Командой меню Bearbeiten → Zeile als Kommentar (Редактировать → Строку как комментарий) или с помощью соответствующего символа на панели инструментов Вы можете строку таблицы представить как строку комментариев.
Управляемое значение действительно/недействительно	Если в столбце “Steuerwert” перед значением, подлежащим управлению, поставить знак комментария “//”, то это значение становится недействительным. Если знак комментария убрать, то значение снова становится действительным и им можно управлять.
Выбор столбца/ширина столбца	<p>Командой меню Ansicht → ... (Вид → ...) можно отображать и убирать с экрана отдельные столбцы таблицы, если они не нужны. Отображаются только столбцы, отмеченные галочкой.</p> <p>Ширину столбца можно изменить командой меню Ansicht → Spaltenbreite (Вид → Ширина столбца).</p>
Максимальный размер	<p>Таблица переменных может включать не более 255 знаков на строку.</p> <p>Переход в следующую строку невозможен. Длина таблицы определена в 211 строк. Тем самым достигается ее максимальный размер.</p>

14.4 Наблюдение и управление переменными

Применение

С помощью функции **Status beobachten** (Наблюдение состояния) можно наблюдать выбранные переменные в заранее заданных точках контроля.

С помощью функции **Steuern** (Управление) можно задавать значения переменным в заранее заданных точках контроля и, таким образом, целенаправленно управлять ими.



Осторожно

Перед выполнением функции "Steuern" ("Управление") убедитесь в невозможности возникновения опасных состояний!

Выполнение функции управления возможно только тогда, когда переключатель режимов работы на CPU стоит в положении "RUN-P".

"Немедленное выполнение"

Значения выбранных переменных можно однократно отобразить командой меню **Variable → Statuswerte aktualisieren** (Переменная → Актуализировать значения состояния), а командой меню **Variable → Steuerwerte aktivieren** (Переменная → Активизировать управляемые значения) их можно однократно изменить. Это задание выполняется с максимально возможной скоростью как "Немедленное выполнение" безотносительно к определенному месту прикладной программы.

Установка точек и условия контроля

Командой меню **Variable → Trigger** (Переменная → Пуск) можно

- выбрать одну из следующих точек контроля: "Beginn Zyklus" ("Начало цикла"), "Ende Zyklus" ("Конец цикла") или "Übergang von RUN nach STOP" ("Переход из RUN в STOP") (см. рис. 14-2)
- выбрать в качестве условия контроля "einmalig" ("однократно") или "permanent" ("постоянно").



Указание

При управлении переменными относительно точек контроля имеют силу следующие утверждения:

Если в качестве условия контроля установлено "Einmalig" ("Однократно"), то в случае невозможности управления выбранными переменными, выдается соответствующее сообщение.

При условии контроля "Permanent" ("Постоянно") Вы никакого подтверждения не получаете.

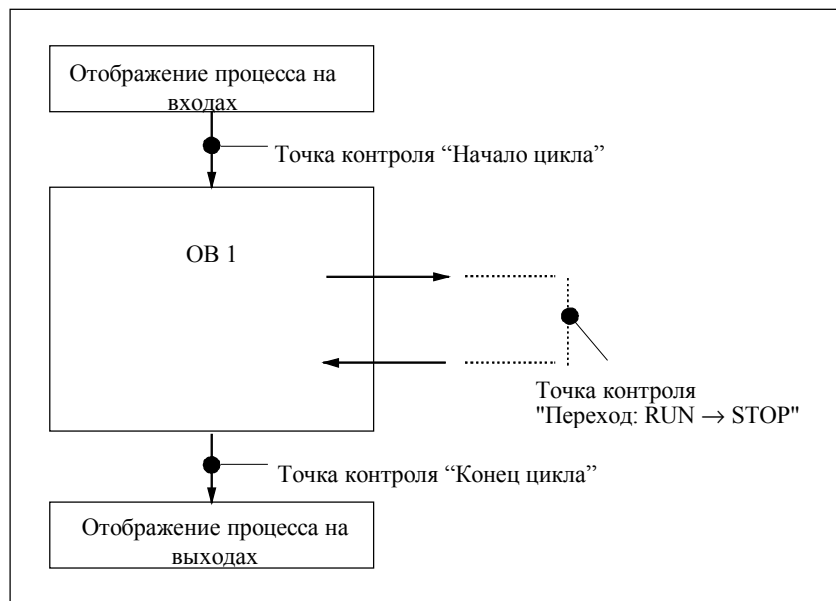


Рис. 14-2. Точки контроля

Количество достигнутых точек контроля отображается циклически от 0 до 255 в возрастающем порядке справа внизу строки состояний.

Создание связи online с CPU

Чтобы наблюдать за переменными или управлять ими, необходимо создать связь online с CPU. Если открыто несколько различных таблиц переменных, то для каждой из них должна быть построена связь online с CPU.

При этом каждая таблица переменных может быть связана с любым CPU.

Создана связь online или нет, можно узнать из строки состояний (online/offline).

- Если связь online отсутствует, определите ее командой меню **Zielsystem → Verbindung herstellen zu → ...** (Контроллер → Создать связь с → ...) для желаемого CPU, чтобы наблюдать за переменными или управлять ими.
- Командой меню **Zielsystem → Verbindung abbauen** (Контроллер → Демонтировать связь) связь между таблицей переменных и CPU снова разрушается.

Наблюдение за переменными

Для наблюдения за переменными имеются следующие возможности:

- Включите функцию "Status beobachten" ("Наблюдение за состоянием") командой меню **Variable → Status beobachten** (Переменная → Наблюдение за статусом). Значения выбранных переменных будут отображены в таблице переменных в зависимости от установленной точки и установленного условия контроля.
Если установлено условие "Permanent" ("Постоянно"), выключите функцию "Status beobachten" ("Наблюдение за состоянием") той же командой меню **Variable → Status beobachten**.
- Актуализируйте значения выбранных переменных однократно и немедленно командой меню **Variable → Statuswerte aktualisieren** (Переменная → Актуализировать значения статуса). В таблице переменных будут отображены текущие значения выбранных переменных.

Управление переменными

Для управления переменными имеются следующие возможности:

- Включите функцию "Steuern" ("Управление") командой меню **Variable → Steuern** (Переменная → Управление). Прикладная программа принимает управляемые значения выбранных переменных из таблицы переменных в зависимости от смотря по обстоятельствам установленных точки и условия контроля.
Если установлено условие "Permanent" ("Постоянно"), выключите функцию "Steuern" ("Управление") той же командой меню **Variable → Steuern** (Переменная → Управление).
- Актуализируйте значения выбранных переменных однократно командой меню **Variable → Steuerwert aktivieren** (Переменная → Активизировать управляемое значение).

Обзор

В этой главе описано, как

- отобразить online-информацию из CPU,
- установить причины сбоя в исполнении прикладной программы с помощью диагностического буфера и содержимого стеков и
- проверить, исполняема ли прикладная программа в определенном CPU.

Информация CPU оказывает Вам поддержку при системной диагностике без затрат на программирование и дает возможность быстро распознавать, точно локализовать и устранять ошибки. Благодаря этому в случае сбоя время остановки существенно сокращается.

Обзор главы

В раз-деле	Вы найдете	на стр.
15.1	Обзор	15–2
15.2	Отображение сообщений CPU	15–3
15.3	Отображение состояния модулей	15–5
15.4	Система времени и установка времени CPU	15–6
15.5	Отображение данных о CPU	15–7
15.6	Отображение имеющихся в наличии блоков (только S7–CPU)	15–8
15.7	Отображение загрузки памяти пользователя	15–9
15.8	Отображение времени цикла CPU	15–10
15.9	Отображение коммуникационных соединений	15–11
15.10	Отображение содержимого диагностического буфера	15–12
15.11	Отображение содержимого стеков (только S7–CPU)	15–15

15.1. Обзор

Сообщения CPU

С помощью функции “Сообщения CPU” можно выводить online асинхронные события в CPU (записи в диагностическом буфере), переводящие CPU в состояние STOP. В STEP 7 Вы можете установить, должны ли эти сообщения подавляться или выводиться на экран или накапливаться в буфере. Одновременно Вы принимаете решение архивировать эти сообщения или нет.

Справочные функции

Следующая таблица дает обзор того, какую информацию о CPU Вы можете получить через регистровый диалог и когда Вы эти функции можете рационально использовать.

Таблица 15–1. Обзор справочных функций

Справочная функция	Информация	Использование
Zielsystem (Контроллер)	База времени и коэффициент коррекции, с которым работает этот CPU, а также дата и время выбранного CPU. Информация о синхронизации часов, счетчик рабочего времени.	Для отображения времени и даты CPU и для контроля синхронизации часов
Allgemein (Общие сведения)	такая информация о CPU, как версия, емкость памяти и размер области отображения процесса	Перед переносом прикладной программы в CPU и во время работы с пакетом STEP 7, если Вам нужна специфическая информация о CPU.
Bausteine (Блоки)	Отображение всех видов блоков, доступных в наборе функций выбранного CPU. Список OB, SFB и SFC, которые Вы можете использовать в прикладной программе для этого CPU.	До и во время разработки прикладной программы и для дополнительной проверки, совместима ли существующая прикладная программа с конкретным CPU.
Speicher (Память)	Текущая загрузка рабочей и загрузочной памяти.	Перед переносом новых или расширенных блоков в CPU.
Zykluszeit (Время цикла)	Длительность самого длинного, самого короткого и последнего цикла.	Для контроля параметризованного минимального времени цикла, а также максимального и текущего времени цикла
Kommunikation (Связь)	Скорость передачи и обзор коммуникационных соединений CPU.	Для определения того, сколько и какие соединения CPU возможны и заняты.
Diagnosepuffer (Диагностический буфер)	Отображение диагностического буфера	Для оценки причины останова CPU и для оценки предшествующих событий.
Stacks (Стеки)	Отображается содержимое В–стека, U–стека и L–стека. Дополнительно можно редактировать блоки.	Для определения причины перехода в STOP и для коррекции блока.

15.2. Отображение сообщений CPU

Какие сообщения CPU отображаются? С помощью функции “Сообщения CPU” можно вывести **асинхронные сообщения о событиях, приводящих к ошибкам, и сообщения, определенные пользователем** (см. также описание системной функции SFC 52 в Справочном руководстве /235/.)

Отличие по отношению к отображению диагностических событий в диагностическом буфере (см. гл. 15.10) состоит в том, что здесь, за исключением сообщений пользователя, сообщается только о событиях, переводящих CPU в STOP.

Использование С помощью функции “Сообщения CPU” можно отобразить online сообщения для выбранных CPU.

Вы можете установить отображение “Сообщений CPU” так,

- чтобы сообщения собирались в буфере в фоновом режиме,
- чтобы окно с сообщениями выводилось на передний план, как только появляется новое сообщение.

В окне “CPU Melden” (“Сообщения CPU”) можно пролистать последние 10 поступивших сообщений. Некоторые примеры представлены на рис. 15–1:

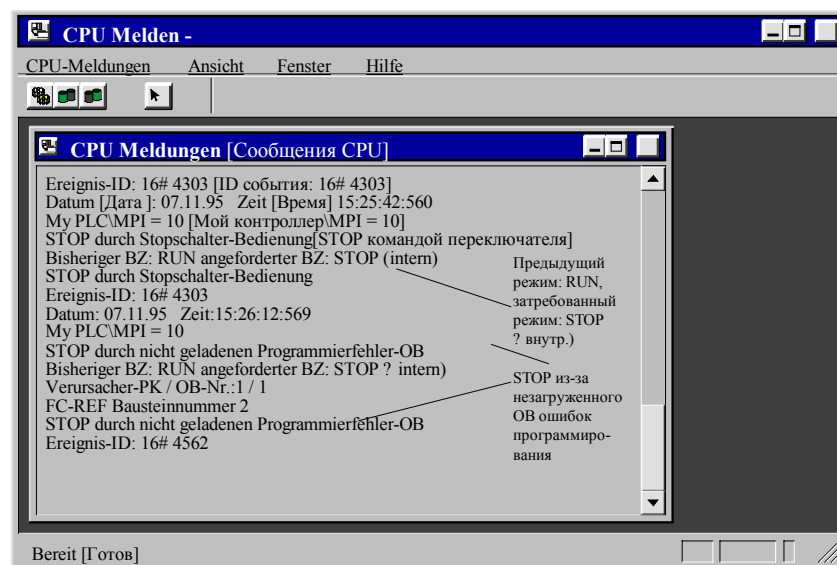


Рис. 15-1. Пример сообщений CPU

Функция архивирования Для архивирования сообщений имеется в распоряжении новый и старый архив. В целом в каждом архиве может быть заархивировано максимум 1000 сообщений CPU. Если установленная величина архива превышена, наличные сообщения переносятся в “Старый архив”, а самое последнее сообщение остается одно в “Новом архиве”.

Отображение архивных сообщений

Archiv (Вид → Старый архив) Вы получаете сообщения CPU из

При выборе команды меню **Ansicht → Neues Archiv** (Вид → Новый архив) в окне "Neues Archiv" ("Новый архив") отображаются самые последние сообщения CPU. Точно также, командой меню **Ansicht →**

Altes

старого архива.

Если Вы хотите обновить отображение сообщений CPU в окне "Neues Archiv" ("Новый архив"), закройте это окно и откройте снова.

Конфигурирование сообщений CPU

Если Вы хотите отобразить сообщения CPU для выбранных модулей, действуйте следующим образом:

1. Отметьте в SIMATIC Manager прикладную программу online или интерфейс, с которым существует коммуникационное соединение (напр., "Erreichbare Teilnehmer" ("Доступные абоненты")).
2. Выберите команду меню **Zielsystem → CPU Meldungen...** (Контроллер → Сообщения CPU...) для объявления модуля.
3. Появляется диалоговое окно "CPU Melden konfigurieren" ("Конфигурирование сообщений CPU"). В левом поле выводятся все заявленные модули. Список модулей можно расширить, повторив шаги 1. и 2. для других программ или интерфейсов.

С помощью управляющих окошек перед записями в списке установите, должны ли приниматься сообщения модуля (CPU).
4. Установите, в каком режиме должны отображаться сообщения:
 - **"Vordergrund"** ("На переднем плане"): Окно с сообщениями CPU появляется на переднем плане. Оно выводится на экран при каждом новом сообщении.
 - **"Hintergrund"** ("В фоновом режиме"): Прием сообщений CPU происходит в фоновом режиме. При новых сообщениях окно остается на заднем плане и может выводиться на передний план при необходимости.
 - **"Ignorieren"** ("Игнорировать"): Сообщения CPU **не** отображаются и, в противоположность двум первым случаям **не** архивируются.
5. Установите размер архива.
6. По завершении всех установок закройте диалоговое окно.

15.3. Отображение состояния модулей

Предпосылка

Для получения возможности отображать состояние модуля CPU необходимо установить связь online.

Последовательность Текущее состояние модуля CPU отображается, если Вы действуете **действий** следующим образом:

1. Выберите в SIMATIC Manager S7–программу (online) проекта или в окне "Erreichbare Teilnehmer" ("Доступные абоненты") выберите CPU.
2. Командой меню **Zielsystem → Baugruppenzustand** (Контроллер → Состояние модуля) откройте регистровый диалог.

Регистровый диалог состоит из восьми вкладок. При первом вызове отображается как предварительная установка вкладки "Allgemein" ("Общие сведения").

Актуализация отображения

При каждом переходе на другую вкладку данные с модуля считываются заново. Однако во время отображения вкладки данные автоматически не актуализируются. Если щелкнуть на кнопке "Aktualisieren" ("Актуализировать"), то данные будут считаны из модуля заново - без смены вкладки.



Указание

Отображаемые тексты, в которые CPU не посылает данных, представляются неактивными и не отображают никаких значений.

Отображение одновременно нескольких CPU

Вы можете опрашивать состояния различных CPU и одновременно их отображать. Для этого Вы должны перейти в соответствующий список модулей и действовать, как описано выше. Вам будет предоставлен очередной регистровый диалог. Однако, для одного соединения online (т.е. для одного CPU) можно открыть только один регистровый диалог. Поэтому сравнение во времени состояний одного и того же CPU невозможно.

15.4. Система времени и установка времени CPU

Вкладка "Zeitsystem" На вкладке "Zeitsystem" ("Система времени") отображаются следующие ("Система времени") данные:

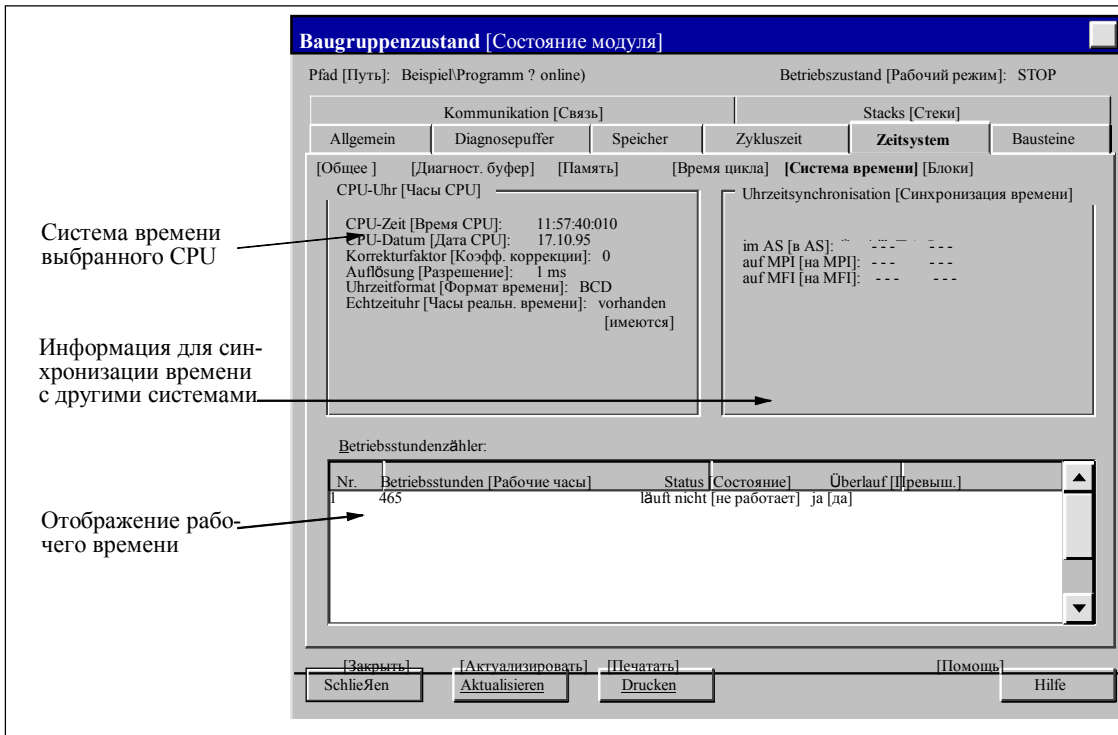


Рис. 15-2. Вкладка "Zeitsystem" ? "Система времени"

Использование

Отображаются база времени и коэффициент коррекции, с которыми работает этот CPU, а также время суток и дата выбранного CPU. Кроме того, Вы получаете информацию для синхронизации часов и для счетчика рабочего времени.

Установка времени и даты

Командой меню **Zielsystem → Uhrzeit stellen** (Контроллер → Установка времени) можно установить время и дату на выбранном CPU. Внесите желаемые значения в следующие поля в диалоге:

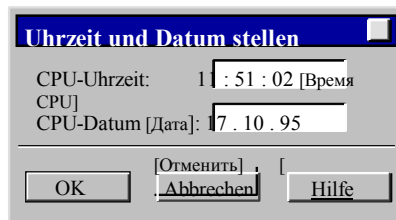


Рис. 15-3. Установка времени и даты CPU

15.5. Отображение данных о CPU

Вкладка "Allgemein" Вкладка "Allgemein" ("Общие сведения") отображает технические ("Общие сведения") данные выбранного CPU (см. рис. 15-4):

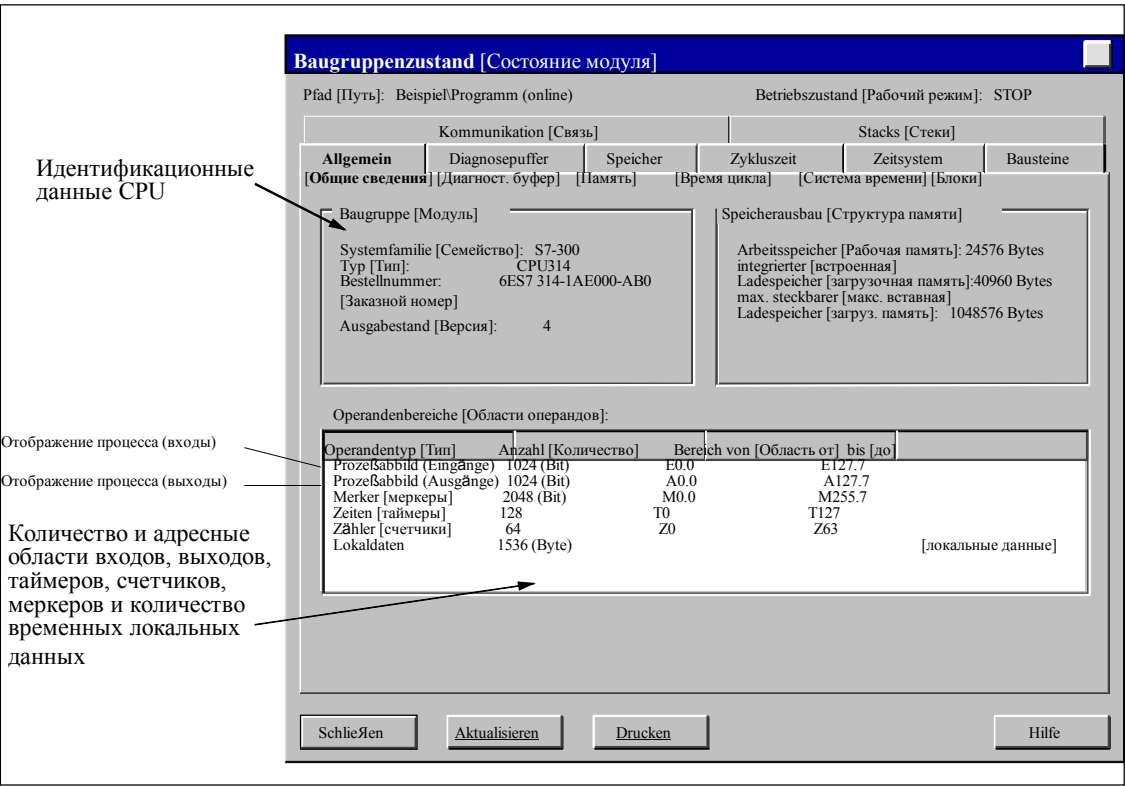


Рис. 15-4. Вкладка "Allgemein" ("Общие сведения")

Использование

Эти функции можно использовать в следующих ситуациях:

- если Вам нужны тип, версия, заказной номер данного CPU
- если Вы хотите загрузить прикладную программу в CPU и дополнительно проверить перед этим, обладает ли CPU надлежащими предпосылками, например, в отношении загрузочной памяти или величины отображения процесса.

15.6. Отображение имеющихся в наличии блоков (только S7–CPU)

Вкладка “Bausteine” (“Блоки”)

На вкладке ”Bausteine” (“Блоки”) отображаются online для каждого CPU следующие данные:

- **Блоки пользователя:**
Максимальное количество OB, FB, FC и DB, а также их максимально допустимая общая длина. Имеющиеся в наличии OB перечисляются с описанием их функций.
- **Системные блоки:**
Количество SFC и SFB, а также их максимально допустимая общая длина. Имеющиеся в наличии SFC и SFB перечисляются с символическими именами и указанием семейства блоков.

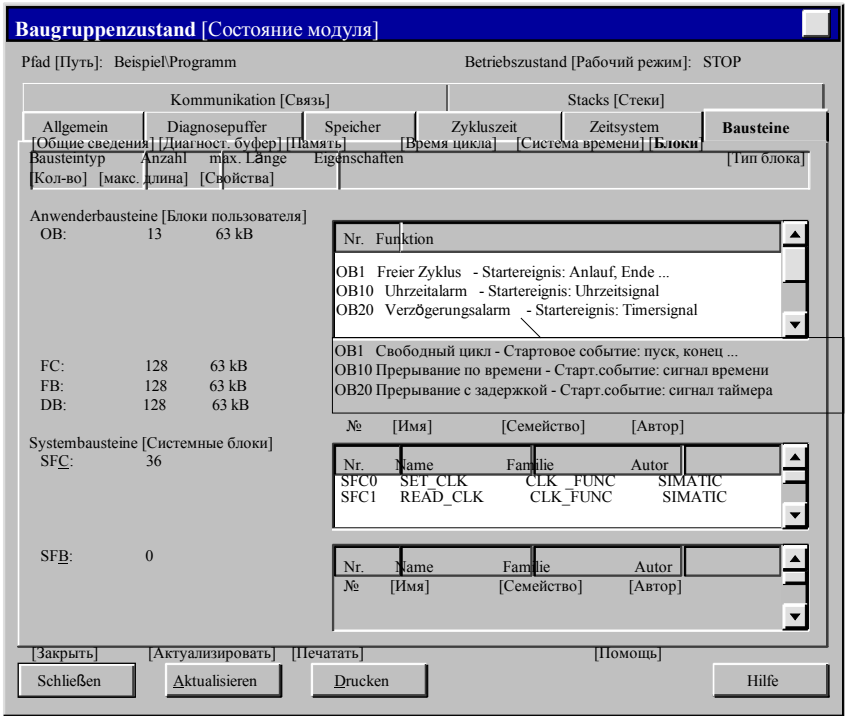


Рис. 15-5. Вкладка "Bausteine" (“Блоки”)

Использование

Отображаются все виды блоков, доступные в наборе функций выбранного CPU. Кроме того, список OB, SFB и SFC, которые можно использовать в прикладной программе для этого CPU.

Перед загрузкой новой прикладной программы в CPU Вы, например, хотите проверить, какие блоки может содержать или вызывать ваша программа, чтобы она могла исполняться на выбранном CPU.

15.7. Отображение степени использования памяти пользователя

Вкладка "Speicher" ("Память") На вкладке "Speicher" ("Память") отображаются online для каждого CPU степень использования рабочей и загрузочной памяти, а также величина самой большой связной свободной области памяти (в зависимости от CPU).

Степень использования памяти представляется как в процентах в виде столбиковой диаграммы, так и в абсолютных величинах в виде таблицы. Значение, представленное неактивным, означает, что величина этой области выбранного CPU не может быть установлена.

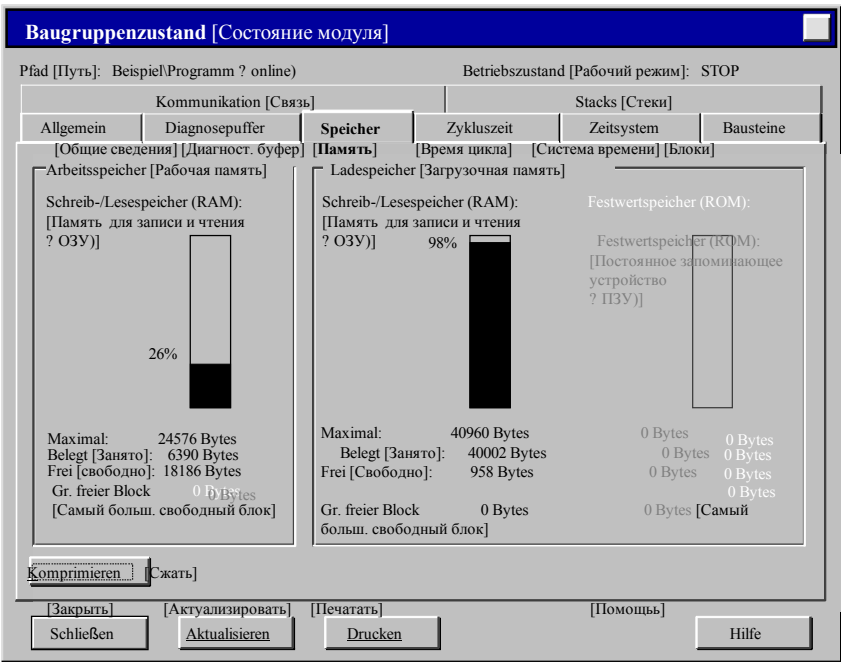


Рис. 15-6. Вкладка "Speicher" ("Память")

Использование Вы хотите перенести прикладную программу в CPU и дополнительно проверить, достаточно ли загрузочной памяти в этом CPU или

Вы хотите добавить к существующему проекту дополнительный объект и предварительно проверить, достаточно ли для этого расширения свободной связной памяти.

Сжатие При копировании и стирании блоков в памяти возникают пустые промежутки. С помощью функции сжатия блоки в рабочей и загрузочной памяти сдвигаются таким образом, что пробелы закрываются и в памяти появляется свободное связное пространство.

Все пробелы в памяти закрываются только при сжатии в состоянии STOP. В режиме RUN-P непосредственно обрабатываемые блоки не сдвигаются, так что и после сжатия некоторые пробелы сохраняются.

15.8. Отображение времен цикла CPU

Вкладка “Zykluszeit” На вкладке ”Zykluszeit” (“Время цикла”) отображается online следующая информация:

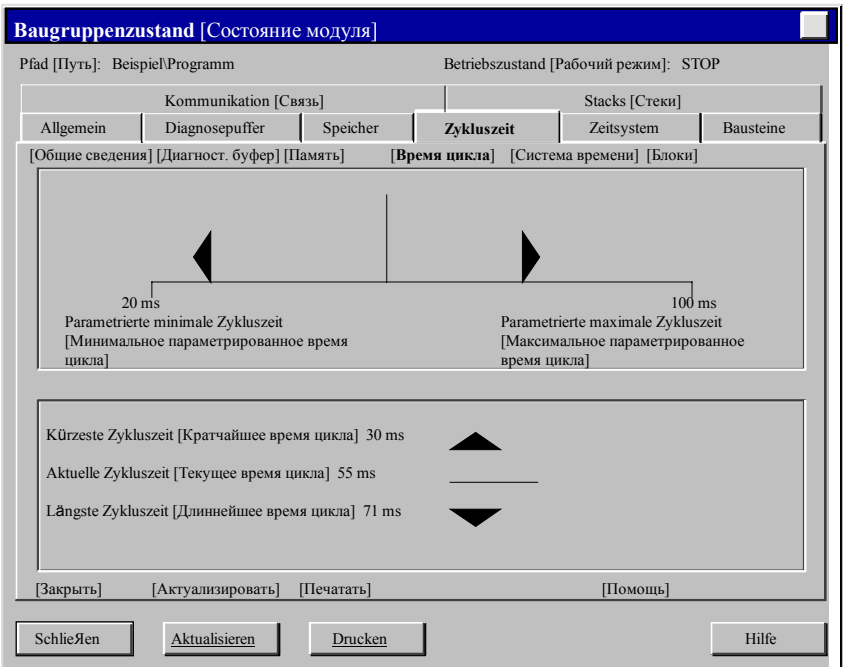


Рис. 15-7. Вкладка "Zykluszeit" (“Время цикла”)

Использование

Эта вкладка дает справку о временах цикла прикладной программы.

Если продолжительность длиннейшего цикла близка к времени контроля цикла, то существует опасность, что при колебаниях времени цикла возникнет ошибка времени. Этого можно избежать, увеличив максимальное время цикла прикладной программы.

Если бы продолжительность кратчайшего цикла стала меньше запараметрированного минимального времени цикла, то CPU автоматически продлило бы цикл до величины минимального запараметрированного времени цикла, так что в этом случае ошибка времени возникнуть не может.

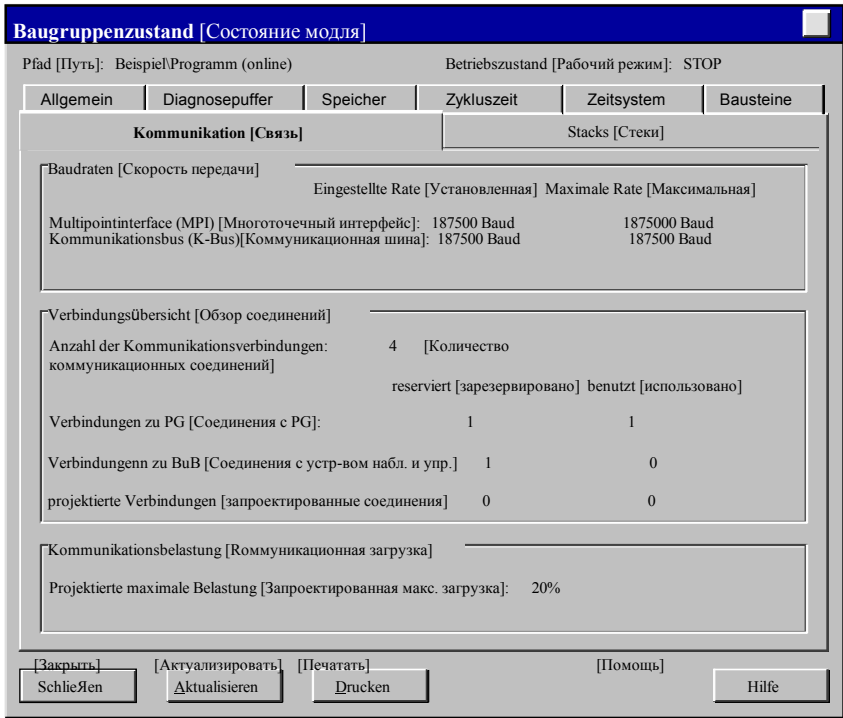
Установка
времени цикла

Максимальное и минимальное время цикла можно установить при конфигурировании аппаратного обеспечения. Для этого щелкните дважды в конфигурационной таблице на CPU, чтобы определить свойства. На вкладке ”Zyklus/Taktmerker” (“Цикл/тактовые меркеры”) можно установить эти значения.

15.9. Отображение коммуникационных соединений

Вкладка
“Kommunikation”
 (“Связь”)

На вкладке “Kommunikation” (“Связь”) отображаются, например, количество и текущие состояния возможных коммуникационных соединений CPU:



Рисд 15-8. Вкладка "Kommunikation" (“Связь”)

Использование

Эта функция дает возможность проверить коммуникационные соединения и получить, например, информацию о возможных или уже занятых соединениях между выбранным CPU и подключенными устройствами.

Соединения

Зарезервированные соединения могут быть установлены исключительно между CPU и соответствующей указанной системой. Остальные соединения (=разность между числом коммуникационных соединений и зарезервированными соединениями) могут подключаться произвольно. Соединения через глобальные данные не относятся к показанным соединениям.

Коммуникационная нагрузка

Запроектированная максимальная нагрузка CPU коммуникационными функциями может находиться между 5 и 50%. Она устанавливается при параметризации CPU. Установленное здесь значение указывает, до какой максимальной доли CPU может быть загружен коммуникационными задачами.

15.10. Отображение содержимого диагностического буфера

Диагностический буфер У каждого CPU имеется диагностический буфер, в который для всех диагностических событий вносится подробная информация по мере ее поступления. Содержимое диагностического буфера сохраняется также после общего стирания.

Диагностические события В качестве диагностических событий отображаются, например, следующие записи:

- ошибки в модуле
- ошибки в электрическом монтаже
- системные ошибки в CPU
- переходы из одного рабочего режима в другой (напр., из RUN в STOP)
- ошибки в прикладной программе
- сообщения пользователя

Использование Ошибки в системе благодаря диагностическому буферу могут анализироваться и в более позднее время, чтобы установить причину перехода в состояние STOP или проследить в обратном направлении и сопоставить отдельные диагностические события.

Организация диагностического буфера Диагностический буфер - это кольцевой буфер, максимальное количество записей в котором зависит от CPU. Если максимальное количество записей достигнуто, то при наступлении нового диагностического события самая старая запись стирается. Все остальные записи соответственно сдвигаются. Благодаря этому самое последнее диагностическое событие всегда стоит на первом месте. Для CPU 314 семейства S7-300 емкость буфера составляет 100 записей (см. рис. 15-9):

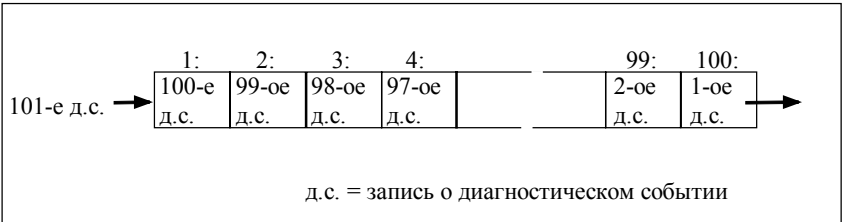


Рис. 15-9. Организация записей в диагностическом буфере

Количество отображенных записей в диагностическом буфере зависит от CPU и его текущего состояния.

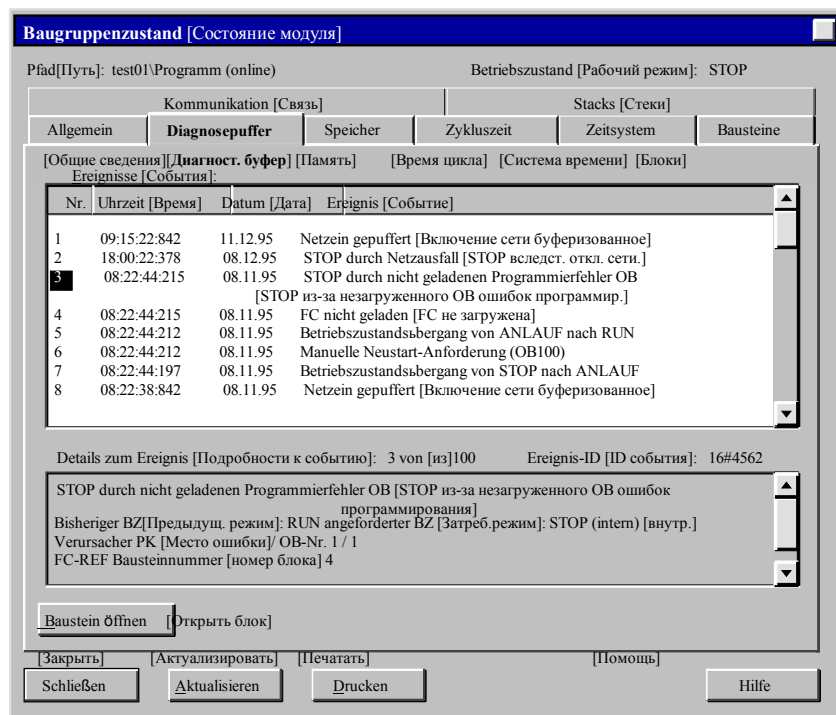


Рис. 15-10. Вкладка "Diagnosepuffer" ("Диагностический буфер")

Отображение диагностического буфера

Верхнее окно содержит список всех наступивших диагностических событий со следующей информацией:

- Текущий номер записи (последняя запись имеет № 1)
- Время и дата диагностического события

Отображает время и дату CPU. Чтобы рационально использовать эти данные о времени, важно установить и проверить время и дату на CPU (см. гл. 15.4).

- Текст события (краткое описание)

В нижнем текстовом поле отображается дополнительная информация к событию, выбранному в верхнем окне. Сюда относятся, например:

- номер события,
- идентификатор события,
- ОБ, в котором наступило событие или который вследствие этого события был вызван,
- изменение рабочего режима, вызванное этим событием,
- указание на место ошибки в блоке (тип, номер блока, относительный адрес), приведшей к записи о событии,
- следующее или предыдущее событие.

Устранение ошибок

При наличии в диагностическом буфере записей, указывающих на место ошибки (тип и номер блока, относительный адрес), соответствующий блок, обусловивший появление события, можно открыть для устранения причины ошибки.

Выберите в верхнем окне диагностическое событие и щелкните на кнопке "Baustein öffnen" ("Открыть блок"). Блок открывается в редакторе (напр., в AWL), и курсор устанавливается на месте, вызвавшем событие. (Чтобы открыть блок в редакторе на месте, вызвавшем событие, внутри должно быть произведено преобразование адресов.)

Особые случаи

Диагностический буфер хранит все диагностические записи вплоть до максимального количества. Они сохраняются и в том случае, если на этом интервале была загружена другая прикладная программа.

Поэтому может произойти так, что более старые диагностические записи относятся к блокам, которых больше нет в CPU. В неблагоприятном случае среди одинаковых имен блоков находится блок, не имеющий отношения к диагностическому сообщению.

В редких случаях могут наступить следующие состояния:

- **Диагностическое событие старше, чем дата последнего изменения блока:**

Вы получаете диалоговое окно "Baustein öffnen" ("Открыть блок") с указанием, что блок был изменен. Возможно, что это просто одноименный блок другой программы.

- Вы можете, несмотря на это, открыть этот блок online в CPU и при определенных обстоятельствах изменить - или
- Вы можете выбрать этот блок offline в соответствующей программе и изменить его offline.

- **Блок, обусловивший появление события, отсутствует в CPU:**

Вы получаете диалоговое окно "Baustein öffnen" ("Открыть блок") с указанием, что данный блок в CPU отсутствует. После диагностической записи блок был стерт.

Вы можете этот блок выбрать offline в соответствующей программе и offline его изменить. Для выбранного блока выполняется, если возможно, пересчет адресов, и блок открывается на рассчитанном адресе.



Указание

Если Вы изменили блок offline, то затем его нужно загрузить в CPU, чтобы сделать действительным его изменение в программе.

15.11. Отображение содержимого стеков (только S7-CPU)

Вкладка "Stacks" ("Стеки") Вкладка "Stacks" ("Стеки") отображает для каждого CPU, с которым есть связь online, содержимое В-стека, L-стека и U-стека. Для этого CPU должен быть переведен в состояние STOP вследствие ошибки программирования или командой останова.

Использование Содержимое стеков дает информацию о том, какая команда в каком блоке привела CPU в состояние STOP.
Дальнейшую информацию о событии, приведшем CPU в STOP, Вы найдете в диагностическом буфере (см. гл. 15.10).

В-стек В В-стеке приведены все блоки, которые были вызваны до перехода в состояние STOP и еще не обработаны до конца. В табличной форме указывается, на каком месте (сеть, команда) был прерван соответствующий блок.

Открытие блока Выберите в таблице блок и щелкните на кнопке "Baustein öffnen" ("Открыть блок"). Блок открывается в редакторе программ, курсор стоит на месте программы, вызвавшей ошибку, и Вы можете выполнить изменения.

L-стек Для каждого из блоков, перечисленных в В-стеке, можно вывести на экран соответствующие локальные данные, выбрав этот блок и щелкнув на кнопке "L-Stack".
Стек локальных данных (L-стек) содержит значения локальных данных блока, с которыми прикладная программа работала вплоть до момента прерывания.
Для интерпретации и анализа отображенных локальных данных требуются очень хорошие системные знания.

U-стек При щелчке на кнопке "U-Stack" отображаются данные о месте прерывания.
Стек прерываний (U-стек) содержит данные или состояния, действовавшие на момент прерывания, например,

- содержимое аккумуляторов и регистров
- открытые DB и их величину
- содержимое слова состояния
- класс приоритета
- прерванный блок (сеть и номер команды)
- следующий блок (сеть и номер команды)

Обзор

С помощью STEP 7 и дополнительного программного обеспечения M7 Вы можете использовать такие языки высокого уровня, как С или С++, а также графические программные пакеты, как CFC (Continuous Funktion Chart), для разработки прикладных программ для управляющих ЭВМ M7-300/400.

В этой главе Вы узнаете свои возможности по разработке прикладных программ для управляющих ЭВМ M7-300/400.

Для разработки этих программ Вам в дополнение к STEP 7 нужна операционная система M7, а также среда разработки для M7-программ. Эти программные компоненты Вы найдёте в дополнительном программном обеспечении M7.

Обзор главы

16.1	Дополнительный программный пакет M7	16-2
16.2	Операционные системы для M7-300/400	16-5

16.1. Дополнительный программный пакет M7

Обзор

STEP 7 предлагает Вам основной набор функций, который необходим для

- создания проектов и управления ими
- конфигурирования и параметризации аппаратуры целевой системы
- проектирования сетей и соединений
- управления символьными данными.

Этот набор функций не зависит от того, является ли целевой системой SIMATIC S7 или SIMATIC M7.

Целевые системы SIMATIC S7 или SIMATIC M7 из-за различия в операционных и исполняющих системах оказывают влияние главным образом на программирование приложений.

Для создания прикладных M7–программ дополнительно к STEP 7 нужен программный пакет M7.

Таблица 16–1. Дополнительное программное обеспечение для программирования M7

Дополнительное программное обеспечение	Содержание
M7–SYS	<ul style="list-style-type: none">• Операционная система M7 RMOS32• Системная библиотека M7–API• Поддержка MPI
CFC	Программное обеспечение для программирования CFC–приложений (CFC = Continuous Function Chart - непрерывная функциональная диаграмма)
M7–ProC/C++	<ul style="list-style-type: none">• Включение среды разработки Borland в STEP 7• Редактор и генератор для импорта символов• Отладчик Organon xdb386
Borland C++	Среда разработки Borland C/C++

Кроме того, STEP 7 совместно с дополнительным пакетом M7 поддерживает следующие операции:

- перенос данных в целевую систему через MPI
- получение информации о целевой системе
- выполнение определенных установок на целевой системе и общее стирание целевой системы

Зависимости

На следующем рисунке показаны зависимости в дополнительном программном пакете М7:

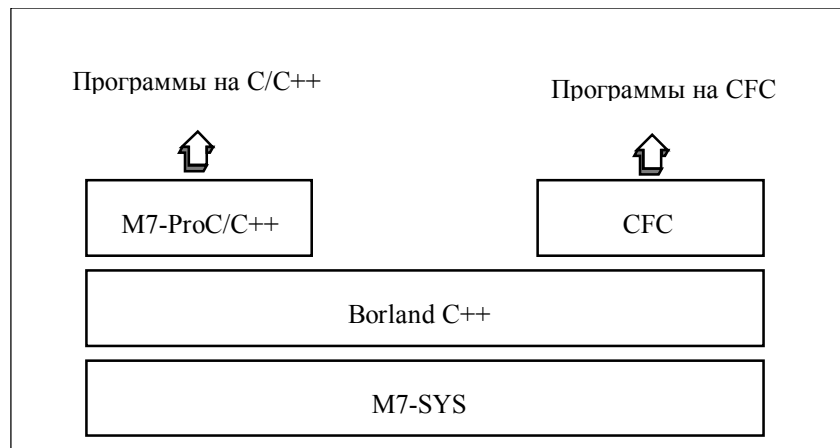


Рис. 16-1. Зависимости в дополнительном программном пакете М7

Таблица 16-2. Резюме

Для создания ...	Вам нужно дополнительное программное обеспечение М7
программ на C/C++	1. М7-SYS 2. М7-ProC/C++ 3. Borland C++
программ на CFC	1. М7-SYS 2. CFC 3. Borland C++

**Где Вы найдете
поддержку**

следующей таблицы Вы узнаете, какую поддержку можно найти в

Специфические инструменты для разработки М7-приложений встроены частично в STEP 7, а частично в дополнительный программный пакет М7. Из отдельных программных пакетах:

Таблица 16–3. Поддержка при разработке прикладных М7–программ

Программный пакет...	окажет Вам поддержку...
STEP 7	<ul style="list-style-type: none"> • при инсталляции операционной системы М7, • управлении целевой системой М7, • переносе, запуске и стирании прикладных М7–программ, • вызове данных о состоянии и диагностике, • общем стирании CPU
М7–SYS	<p>с помощью утилит операционной системы М7 и системного программного обеспечения М7 при:</p> <ul style="list-style-type: none"> • управлении исполнением программы, • управлении памятью и ресурсами, • доступе к аппаратуре ЭВМ и SIMATIC, • управлении прерываниями, • диагностике, • контроле состояния и • связи
М7–ProC/C++	<ul style="list-style-type: none"> • посредством встроенной генерации кодов (включение в STEP 7 среды разработки Borland) • включением символики проекта в исходный код и • встроенным набором функций отладки
Borland C++	<ul style="list-style-type: none"> • при разработке программ на языках С и C++
CFC	<ul style="list-style-type: none"> • при разработке, тестировании и отладке программ на CFC и • при запуске и исполнении CFC–программ

16.2. Операционные системы для М7–300/400

Введение

Благодаря стандартизированной архитектуре ЭВМ АТ управляющая ЭВМ М7-300/400 образует свободно программируемое расширение платформы автоматизации SIMATIC. Прикладные программы для SIMATIC S7 можно программировать на таких языках высокого уровня, как С, или графически с помощью CFC.

Стандартные операционные системы

Для приложений, создаваемых на языках высокого уровня С и С++, имеются утилиты операционной системы различного назначения. Операционная система берет на себя следующие задачи:

- доступ к аппаратуре
- управление ресурсами
- встраивание системы
- связь с другими компонентами системы

Для АТ-совместимых компьютеров в качестве стандартных операционных систем нашли признание **MS-DOS** и **MS-Windows**.

Операционная система реального времени

Однако для решения задач автоматизации эти стандартные операционные системы непригодны. С помощью одних только MS-DOS и MS-Windows невозможно решать задачи в реальном времени, использовать специальную аппаратуру SIMATIC S7 и М7 или обращаться к системным данным. Поэтому мы используем в управляющих ЭВМ SIMATIC М7 операционную систему реального времени RMOS (RMOS=Realtime-Multitasking-Operating-System - многозадачная операционная система реального времени). Для включения в систему SIMATIC RMOS расширена интерфейсом вызова М7-API (API = Application Programming Interface - прикладной программный интерфейс).

Конфигурации операционной системы для М7

Операционная система реального времени М7 RMOS32 используется для 32-битных приложений для решения критичных к времени задач реального времени и в многозадачных случаях. Она применяется для модулей М7 в следующих конфигурациях:

- М7 RMOS32
- М7 RMOS32 с MS-DOS
- М7 RMOS32 с MS-Windows (и MS-DOS)

Какую конфигурацию операционной системы Вы выберете для своей системы М7, зависит от многих факторов:

- Какого вида прикладные программы должны исполняться в системе М7.
- Какие модули М7 используются (см. табл. 16–4).

Прикладные

программы для М7

В системах М7 в зависимости от выбранной конфигурации операционной системы могут исполняться следующие виды прикладных программ:

- чистые М7 RMOS32–программы для решения критичных к времени задач реального времени и для случая многозадачности
- MS–DOS и/или MS–Windows–программы. Это значит, что в Вашей системе автоматизации могут исполняться кроме RMOS–программ также и собственные приложения MS-DOS и MS-Windows.

Какая операционная система на каком модуле М7?

Таблица 16–4. Возможные конфигурации операционных систем на модулях М7

Модуль М7	М7 RMOS32	М7 RMOS32 с MS–DOS	М7 RMOS32 с MS–Windows
М7–300			
FM356-4 (4 MB)	Да	ограниченно	Nein
FM356-4 (8 MB)	Да	Да	Да
CPU388-4 (8 MB)	Да	Да	Да
М7–400			
FM456-4 (4 MB)	Да	ограниченно	нет
FM456-4 (8 MB)	Да	Да	Да
FM456-4 (16 MB)	Да	Да	Да
CPU488-4 (4 MB)	Да	ограниченно	нет
CPU488-4 (8 MB)	Да	Да	Да
CPU488-4 (16 MB)	Да	Да	Да
CPU488-5 (8 MB)	Да	Да	Да
CPU488-5 (16 MB)	Да	Да	Да
CPU488-5 (32 MB)	Да	Да	Да

Ограниченно означает, что горячий старт MS–DOS (Ctrl–Alt–Del) невозможен.

Дополнительная аппаратура

М7 RMOS32 с MS–Windows в принципе применима только на модулях М7, имеющих в своем распоряжении следующую дополнительную аппаратуру:

- жесткий диск через модуль расширения MSM378 или MSM478 и
- монитор VGA и клавиатуру через интерфейсный модуль IF962–VGA.

Массовые накопители

Центральные и прикладные модули М7 имеют в своем распоряжении следующие виды массовых накопителей (см. табл. 16–5):

- платы памяти
- жесткие диски и дискеты

Все программируемые модули М7 могут быть оснащены по выбору посредством модулей расширения MSM жестким диском и дисководом для дискет 3,5". К дискете, как и к плате памяти, можно обращаться как на PC/PG, так и на целевой системе М7.

- Onboard Silicon Disk (OSD) - встроенный кремниевый диск.
Этот массовый накопитель ведет себя, как дисковод жесткого диска, из которого можно загружать операционную систему и на котором можно хранить прикладные программы. Некоторые модули М7–400 оснащаются им по выбору.

Таблица 16–5. Массовые накопители для систем М7 (текущее состояние)

Массовый накопитель	Емкость	Модуль М7–300	Модуль М7–400
Жесткий диск	512MB	MSM378	MSM478
Флоппи-диск 3,5"	1,44MB	MSM378	MSM478
Плата памяти	1, 2, 4, 8, 16 MB	CPU388-3 FM356-4	CPU488-4/5 FM456-4
OSD	4MB	-	по выбору в CPU488-4, FM456-4

Плата памяти на 1 Мб не пригодна для инсталляции операционной системы. Используйте ее только для переноса программ.

Обзор

При разработке проектов, содержащих программируемые модули (CPU и FM) управляющей ЭВМ SIMATIC M7, STEP 7 оказывает поддержку в следующих операциях управления:

- установка операционной системы
- перенос прикладной программы в целевую систему
- стирание компонентов программного обеспечения целевой системы
- опрос и изменение состояния модулей

Обзор главы

В раз-деле	Вы найдете	на стр.
17.1	Подготовка инсталляции	17–2
17.2	Установка M7 RMOS32 на плату памяти	17–7
17.3	Установка M7 RMOS32 на жесткий диск	17–8
17.4	Установка M7 RMOS32 с MS–DOS на жесткий диск	17–9
17.5	Установка M7 RMOS32 с MS–Windows на жесткий диск	17–11
17.6	Установка M7 RMOS32 на OSD	17–12
17.7	Дополнительная инсталляция операционной системы M7	17–14
17.8	Актуализация программ, записанных в ПЗУ	17–15
17.9	Перенос и стирание программ в системе M7	17–17
17.10	Функции опроса и управления для M7–300/400	17–22

17.1. Подготовка инсталляции

Цель инсталляции	<p>Цель инсталляции состоит в переносе полной конфигурации операционной системы, включая программное обеспечение M7, на целевую среду, массовую память, системы M7.</p> <p>Этот раздел дает обзор возможностей инсталляции и принципиальной последовательности действий. Последовательные указания по инсталляции Вы найдете в следующих разделах.</p>
Возможности инсталляции	<p>В зависимости от массовой памяти системы M7 различают:</p> <ol style="list-style-type: none">1. Инсталляцию на жесткий диск или OSD. При первой инсталляции в системе M7 еще нет работоспособной операционной системы; связь через MPI еще невозможна.2. Инсталляцию на плату памяти. На плате памяти размещается полная операционная система M7 RMOS32 с прикладными программами (см. табл. 16–4).
Принципиальная последовательность действий	<p>Для инсталляции операционной системы (ОС):</p> <ol style="list-style-type: none">1. Выберите в своем проекте объект "M7-Programm" ("M7-программа").2. Вызовите команду меню Zielsystem → M7-Zielsystem verwalten (Целевая система → Управление системой M7).3. Откройте вкладку "BS Installieren" ("Инсталляция ОС").4. Произведите выбор в следующих полях (см. рис. 17–1):<ul style="list-style-type: none">- конфигурация операционной системы- среда- локальный дисковод и дисковод-партнер, если используется среда "MPI/RFS"5. Щелкните на кнопке "Installieren" ("Инсталлировать"). <p>Все дальнейшие операции зависят от выбранной конфигурации операционной системы и от целевой среды.</p>

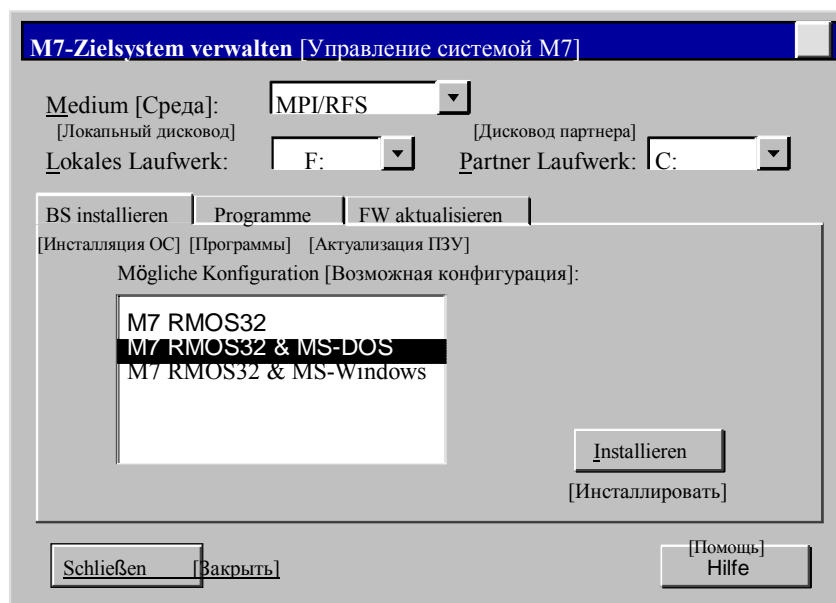


Рис. 17-1. Вкладка "BS Installieren" ("Инсталлировать ОС")

Выбор операционной системы

Выберите конфигурацию операционной системы из поля "Mögliche Konfiguration" ("Возможная конфигурация"). Выбор конфигурации операционной системы зависит от вида прикладной программы, которая должна исполняться в системе M7. В таблице 17–1 показано, когда какую операционную систему следует выбирать. Обратите, пожалуйста, внимание также на зависимость от аппаратуры в табл. 16–4.

Таблица 17–1. Конфигурации операционной системы

Прикладные программы	Конфигурация операционной системы
только приложения M7 RMOS32	M7 RMOS32
Приложения M7 RMOS32 и MS-DOS	M7 RMOS32 & MS-DOS
Приложения M7 RMOS32, MS-DOS и MS-Windows	M7 RMOS32 & MS-Windows

Потребность в памяти M7 RMOS32 без MS-DOS/Windows составляет максимум 2 Мб в целевой среде. Сюда следует добавить также потребность в памяти прикладных программ. В табл. 17–2 показано, какие массовые накопители можно выбрать для отдельных конфигураций операционной системы.

Таблица 17–2. Возможные массовые накопители для операционных систем M7

Операционная система	Массовая память		
	Жесткий диск	Плата памяти	OSD
M7 RMOS32	Да	Да	Да
M7 RMOS32 с MS-DOS	Да	Да	Да
M7 RMOS32 с MS-Windows	Да	Нет	Нет

Выбор средства

В поле выбора "Medium" предлагаются следующие **средства инсталляции**:

1. MPI/RFS:

Выберите "MPI/RFS" (RFS = **R**emote **F**ile **S**ystem - Удаленная файловая система), если операционная система должна быть установлена на жестком диске или OSD системы M7. Для возможности использования среды инсталляции, необходимо установить связь через MPI между инструментальной и целевой системой.

Как правило, операционная система устанавливается на жестком диске системы M7 через MPI/RFS. Для менее объемных конфигураций операционных систем, например, только M7 RMOS32, достаточно платы памяти или OSD.

При инсталляции через MPI/RFS всегда требуется также **средство загрузки** (см. стр. 17–5).

2. Плата памяти

Выберите "Memory Card", если операционная система должна быть установлена на плате памяти. Операционная система и прикладные программы переносятся с PG на плату памяти. Затем плата памяти вставляется в систему M7 и целевая система с нее загружается.

Для использования платы памяти необходим PG 720/740/760.



как средство загрузки или носитель данных для прикладных программ.

Указание

Дискета емкостью 1,44 Мб может содержать минимальную систему M7 RMOS32, однако она не мыслится как целевая среда для инсталляции операционной системы на M7–300/400. Дискету можно использовать

Выбор локального дисковода и дисковода-партнера

Если для инсталляции используется среда передачи "MPI/RFS" (RFS = Remote File System - Удаленная файловая система), то между локальным дисководом PC/PG и дисководом системы M7 строится коммуникационное соединение через MPI.

Локальный дисковод:

В поле выбора отображается список обозначений свободных дисководов на PG/PC, из которого можно выбрать любой дисковод.

Дисковод-партнер:

В поле выбора отображается список обозначений свободных дисководов в целевой системе M7, из которого можно выбрать желаемую массовую память. Обычно (если не установлено иначе) дисководы распределены следующим образом:

Таблица 17–3. Распределение дисководов (предварительная установка) в системе M7

Дисковод	Обозначение дисковода-партнера	
	MS-DOS	M7 RMOS32
Гибкий диск	A:	A: или B:
Плата памяти	B:	M0:
Жесткий диск	C:, D:, ...	C:, D:, ...
Onboard Silicon Disk (Встроенный кремниевый диск)	D:, E:, ... с жестким диском C: без жесткого диска	M1:

Средство загрузки

Если операционная система устанавливается на жестком диске или OSD системы M7, то Вам дополнительно нужно средство загрузки. Средством загрузки мы называем носитель данных, с которого система загружается после включения напряжения питания. Средство загрузки содержит минимальную операционную систему M7 RMOS32. При загрузке части операционной системы, необходимые для исполнения прикладных программ и коммуникации, загружаются в рабочую память.

После загрузки может быть установлена связь через MPI между PC/PG и системой M7.

Загружаемыми носителями данных для M7 являются:

- дискеты 3,5"/1,44 Мб или
- плата памяти на 2 Мб

Инсталляция MS-DOS и MS-Windows

Перед инсталляцией одной из конфигураций операционной системы с MS-DOS или MS-Windows необходимо эти операционные системы инсталлировать с дискеты непосредственно на целевую систему M7 в следующей последовательности (см. также /282/:

1. Инсталляция MS-DOS V6.22, если устанавливается RMOS с MS-DOS или с MS-Windows
2. Инсталляция MS-Windows V3.1X, если устанавливается RMOS с MS-Windows

Затем можно инсталлировать M7 RMOS32, как описано в следующих разделах.

Форматирование целевой среды

целевую среду необходимо форматировать в следующих случаях:

Операционная система	Целевая среда форматируется...
M7 RMOS32	перед каждой новой или повторной инсталляцией, так как M7 RMOS32, если она работает без MS-DOS, всегда должна быть записана в начале памяти.
M7 RMOS32 с MS-DOS/ Windows	перед первой инсталляцией MS-DOS.

Во время инсталляции M7 RMOS32 без MS-DOS или MS-Windows Вы должны отформатировать целевую среду - жесткий диск или OSD. Пожалуйста, следуйте для этого указаниям, выводимым в диалоговом окне.

17.2. Установка M7 RMOS32 на плате памяти

Исходное состояние На Вашей целевой системе M7 нет дисководов для жесткого диска и дискет.

Предпосылка

В этом случае Вы можете использовать плату памяти в качестве целевой среды. На плате памяти находится полная операционная система M7 RMOS32 с прикладными программами (см. табл. 17–2).

Вам нужны:

- флэш-диск на Вашем PG 720/740/760
- плата памяти на 2 Мб

Процедура

Для запуска операционной системы M7 RMOS32 на плате памяти, необходимо выполнить следующие шаги:

1. Выберите в своем проекте M7-программу, соответствующую M7–CPU/FM.
2. Запустите управление M7 командой меню **Zielsystem → M7–Zielsystem verwalten** (Целевая система → Управление системой M7).
3. Откройте вкладку "BS installieren" ("Инсталляция ОС").
4. Установите операционную систему M7 RMOS32 локально на плате памяти, выбрав:
 - Medium (Средство): "Memory Card" ("Плата памяти")
 - Mögliche Konfiguration (Возможная конфигурация): "M7 RMOS32"
5. Выберите кнопку "Installieren" ("Инсталлировать"). В диалоговом окне Вы будете получать сообщения о текущих процессах.
6. Перенесите локально свою прикладную программу со всеми соответствующими данными проекта на плату памяти. Для этого переключитесь на вкладку "Programme" ("Программы") и действуйте, как описано на стр. 17-19 под заголовком "Перенос M7–программы через носитель данных". Этот шаг является необязательным.
7. Вставьте плату памяти в целевую систему M7 и запустите ее переключателем режимов работы. При необходимости произведите настройку BIOS.

Результат: целевая система M7 загружается с новой операционной системой. Ваша прикладная программа запускается.

Установка RMOS с MS-DOS

Инсталляция M7 RMOS32 с MS-DOS на плате памяти описана в /282/.

17.3. Установка M7 RMOS32 на жестком диске

Исходное состояние Первоначально на целевой системе не установлено никакой работоспособной операционной системы, связь через MPI невозможна.

Предпосылка Для установки M7 RMOS32 на жестком диске целевой системы M7 Вам нужны:

- модуль массовой памяти MSM 378/478 на Вашей целевой системе M7
- средство загрузки (дискета на 1,44 Мб или плата памяти на 2 Мб)

Процедура

Требуется выполнение следующих шагов:

1. Выберите в своем проекте M7–программу, соответствующую M7–CPU/FM.
2. Запустите управление M7 командой меню **Zielsystem → M7–Zielsystem verwalten** (Целевая система → Управление системой M7).
3. Откройте вкладку ”BS installieren” (“Инсталляция ОС”).
4. Выберите:

- Medium (средство): ”MPI/RFS”
- Mögliche Konfiguration (возможная конфигурация: ”M7 RMOS32”
- Lokales Laufwerk (локальный дисковод): первый свободный дисковод, например, F:
- Partner Laufwerk (дисковод-партнер): C: для жесткого диска

5. Выберите кнопку ”Installieren” (“Инсталлировать”).

Затем в диалоговом окне Вы будете получать информацию о текущих событиях и указания о дальнейших действиях. В основном Вы должны сделать следующее:

6. Выбрать средство загрузки (гибкий диск или плату памяти).
Результат: на выбранном средстве загрузки устанавливается минимальная операционная система M7 RMOS32.
7. Вставить средство загрузки в дисковод целевой системы M7 и запустить систему M7. **Результат:** целевая система M7 загружается с новой операционной системой устанавливается связь через MPI между PC/PG и целевой системой M7.
8. Отформатировать жесткий диск через RTI (Remote Terminal Interface - интерфейс удаленного терминала) или на локальной консоли целевой системы M7 (см. /282/).
Результат: жесткий диск форматируется. Затем операционная система M7 RMOS32 и возможные прикладные программы устанавливаются на жестком диске M7–300/400 через MPI.

Для переноса в целевую систему M7 прикладной программы, откройте вкладку ”Programme” (“Программы”) и действуйте, как описано на стр. 17-18 под заголовком ”Перенос M7–программ через MPI/RFS”.

9. Переключателем режимов работы выполните новый пуск целевой системы M7 и при необходимости произведите настройку BIOS. **Результат:** целевая система M7 загружается с новой операционной системой. Прикладная программа - если имеется - запускается.

17.4. Установка M7 RMOS32 с MS-DOS на жестком диске

Исходное состояние Первоначально на целевой системе не установлено никакой работоспособной операционной системы, связь через MPI невозможна.

Предпосылка Для установки M7 RMOS32 с MS-DOS на жестком диске целевой системы у Вас должна быть уже установлена на жестком диске **MS-DOS V6.22**. Кроме того, Вам нужны:

- модуль массовой памяти MSM 378/478 на Вашей целевой системе M7
- средство загрузки (дискета на 1,44 Мб или плата памяти на 2 Мб)
- инсталляционные дискеты MS-DOS

Процедура

Для установки операционной системы M7 RMOS32 с MS-DOS на целевой системе M7 необходимо выполнить следующие шаги:

1. Выберите в своем проекте M7-программу, соответствующую M7-CPU/FM.
2. Запустите управление M7 командой меню **Zielsystem → M7-Zielsystem verwalten** (Целевая система → Управление системой M7).
3. Откройте вкладку "BS installieren" ("Инсталляция ОС").
4. Выберите:
 - Medium (средство): "MPI/RFS"
 - Mögliche Konfiguration (возможная конфигурация: "M7 RMOS32& MS-DOS"
 - Lokales Laufwerk (локальный дисковод): первый свободный дисковод, например, F:
 - Partner Laufwerk (дисковод-партнер): C: для жесткого диска
5. Выберите кнопку "Installieren" ("Инсталлировать").

Затем в диалоговом окне Вы будете получать информацию о текущих событиях и указания о дальнейших действиях. В основном Вы должны сделать следующее:

6. Выбрать средство загрузки (гибкий диск или плату памяти).
Результат: на выбранном средстве загрузки устанавливается минимальная операционная система M7 RMOS32.
7. Вставить средство загрузки в дисковод целевой системы M7 и запустить систему M7. **Результат:** целевая система M7 загружается с новой операционной системой устанавливается связь через MPI между PC/PG и целевой системой M7. Затем на жесткий диск M7-300/400 через MPI устанавливается M7 RMOS32 для MS-DOS и, возможно, прикладные программы.

Для переноса своей прикладной программы в систему M7 откройте вкладку "Programme" ("Программы") и действуйте, как описано на стр. 17-18 под заголовком "Перенос M7-программ через MPI/RFS".

8. Переключателем режимов работы выполните новый пуск целевой системы M7 и при необходимости произведите настройку BIOS. **Результат:** целевая система M7 загружается с новой операционной системой. Прикладная программа - если имеется - запускается.

17.5. Установка M7 RMOS32 с MS–Windows на жестком диске

Исходное состояние Первоначально на целевой системе не установлено никакой работоспособной операционной системы, связь через MPI невозможна.

Предпосылка Для установки M7 RMOS32 с MS–Windows на жестком диске целевой системы у Вас должны быть уже установлены на жестком диске **MS–DOS V6.22** и **MS–Windows V3.x**. Кроме того, Вам нужны:

- модуль массовой памяти MSM 378/478 на Вашей целевой системе M7
- средство загрузки (дискета на 1,44 Мб или плата памяти на 2 Мб)
- инсталляционные дискеты MS–DOS и MS–Windows

Процедура

Для установки операционной системы M7 RMOS32 с MS–DOS на целевой системе M7 необходимо выполнить следующие шаги:

1. Выберите в своем проекте M7–программу, соответствующую M7–CPU/FM.
2. Запустите управление M7 командой меню **Zielsystem → M7–Zielsystem verwalten** (Целевая система → Управление системой M7).
3. Откройте вкладку ”BS installieren” (“Инсталляция ОС”).
4. Выберите:

- Medium (средство): ”MPI/RFS”
- Mögliche Konfiguration (возможная конфигурация: ”M7 RMOS32& MS–Windows”
- Lokales Laufwerk (локальный дисковод): первый свободный дисковод, например, F:
- Partner Laufwerk (дисковод-партнер): C: для жесткого диска

5. Выберите кнопку ”Installieren” (“Инсталлировать”).

Затем в диалоговом окне Вы будете получать информацию о текущих событиях и указания о дальнейших действиях. В основном Вы должны сделать следующее:

6. Выбрать средство загрузки (гибкий диск или плату памяти).
Результат: на выбранном средстве загрузки устанавливается минимальная операционная система M7 RMOS32.
7. Вставить средство загрузки в дисковод целевой системы M7 и запустить систему M7. **Результат:** целевая система M7 загружается с новой операционной системой устанавливается связь через MPI между PC/PG и целевой системой M7. Затем на жесткий диск M7-300/400 через MPI устанавливается M7 RMOS32 для MS– Windows и, возможно, прикладные программы.

Для переноса своей прикладной программы в систему M7 откройте вкладку ”Programme” (“Программы”) и действуйте, как описано на стр. 17-18 под заголовком ”Перенос M7–программ через MPI/RFS”.

8. Переключателем режимов работы выполните новый пуск целевой системы M7 и при необходимости произведите настройку BIOS. **Результат:** целевая система M7 загружается с новой операционной системой. Прикладная программа - если имеется - запускается.

17.6. Установка M7 RMOS32 на OSD

Исходное состояние У Вашей целевой системы M7-400 с OSD нет дисководов для жесткого и гибкого дисков. Первоначально на целевой системе не установлено никакой работоспособной операционной системы, связь через MPI невозможна.

Предпосылка

Для установки M7 RMOS32 на OSD целевой системы M7 Вам нужна:

- плата памяти емкостью не менее 2 Мб как средство загрузки на PG 720/740/760



Указание

Если в целевой системе M7 имеется модуль массовой памяти MSM 478, то операционную систему **нельзя** загрузить с OSD.

Процедура

Требуется выполнить следующие шаги:

1. Выберите в своем проекте M7–программу, соответствующую M7–CPU/FM.
2. Запустите управление M7 командой меню **Zielsystem → M7–Zielsystem verwalten** (Целевая система → Управление системой M7).
3. Откройте вкладку "BS installieren" ("Инсталляция ОС").

4. Выберите:

- Medium (средство): "MPI/RFS"
- Mögliche Konfiguration (возможная конфигурация: "M7 RMOS32"
- Lokales Laufwerk (локальный дисковод): первый свободный дисковод, например, F:
- Partner Laufwerk (дисковод-партнер): M1:

5. Выберите кнопку "Installieren" ("Инсталлировать").

Затем в диалоговом окне Вы будете получать информацию о текущих событиях и указания о дальнейших действиях. В основном Вы должны сделать следующее:

6. Выбрать средство загрузки (гибкий диск или плату памяти).

Результат: на выбранном средстве загрузки устанавливается минимальная операционная система M7 RMOS32.

7. Вставить средство загрузки в дисковод целевой системы M7 и запустить систему M7. **Результат:** целевая система M7 загружается с новой операционной системой устанавливается связь через MPI между PC/PG и целевой системой M7.

8. Отформатировать OSD как средство загрузки через RTI (Remote-Terminal-Interface - Интерфейс удаленного терминала) или на локальной консоли командой **ftlform m1: /s** (см. также /282/). **Результат:** OSD форматируется, и операционная система M7 RMOS32 устанавливается на OSD M7 через MPI.
9. Перенесите Вашу прикладную программу со всеми соответствующими проектными данными через MPI/RFS на OSD. Для этого перейдите на вкладку "Programme" ("Программы") и действуйте, как описано на стр. 17-18 под заголовком "Перенос M7-программ через MPI/RFS". Этот шаг не является обязательным.
10. Переключателем режимов работы выполните новый пуск целевой системы M7 и при необходимости произведите настройку BIOS. **Результат:** целевая система M7 загружается с новой операционной системой. Прикладная программа - если имеется - запускается.

Установка RMOS с MS-DOS

Установка M7 RMOS32 с MS-DOS на OSD описана в /282/.

17.7. Дополнительная инсталляция операционной системы М7

Исходное состояние Если на жестком диске целевой системы М7 уже имеется операционная система, то можно выполнить **дополнительную установку** через "MPI/RFS", т.е. изменить, расширить или заменить новой версией операционную систему на Вашей целевой системе М7.

Процедура дополнительной инсталляции

В табл. 17–4 показано, что необходимо делать в различных случаях дополнительной инсталляции на жестком диске. Эта процедура похожа на новую инсталляцию и уже описана в разделах 17.3, 17.4 и 17.5.

Таблица 17–4. Дополнительная инсталляция

В наличии	Если Вы дополнительно устанавливаете		
	М7 RMOS32	М7 RMOS32 с MS–DOS	М7 RMOS32 с MS–Windows
М7 RMOS32	Как новая инсталляция М7 RMOS32	Отформатировать целевую среду, локально установить MS–DOS и заново инсталлировать М7 RMOS32	Отформатировать целевую среду, локально установить MS–DOS и MS–Windows и заново инсталлировать М7 RMOS32
М7 RMOS32 с MS–DOS	Как новая инсталляция М7 RMOS32	Дополнительно устанавливается только новый компонент М7 RMOS32	Локально установить MS–Windows и заново инсталлировать М7 RMOS32
М7 RMOS32 с MS–Windows	Как новая инсталляция М7 RMOS32	Отформатировать целевую среду, локально установить MS–DOS и заново инсталлировать М7 RMOS32	Дополнительно устанавливается только новый компонент М7 RMOS32

Дополнительная инсталляция на плате памяти или OSD

Дополнительная инсталляция М7 RMOS32 на плате памяти или OSD всегда является новой инсталляцией. В случае RMOS32 с MS–DOS как правило дополнительно устанавливаются только компоненты RMOS.

Обратите, пожалуйста, внимание, что в обоих видах памяти возможно лишь ограниченное количество обращений для записи.

Дальнейшая информация

Подробное описание отдельных шагов, а также дополнительную информацию по инсталляции конфигураций операционных систем Вы найдете

- online–помощи управления целевой системой М7
- руководстве пользователя дополнительного пакета М7–SYS /282/.

17.8. Актуализация программ, записанных в ПЗУ

Введение

На центральных и прикладных модулях M7-300/400 находятся специфические для модулей программы, записанные в ПЗУ, например, BIOS. Актуализацию этих программ можно производить через управление целевой системой (Zielsystemverwaltung).

Программы в ПЗУ могут актуализироваться независимо или совместно с системным программным обеспечением для M7-300/400.



Указание

Обратите, пожалуйста, внимание на указания по совместимости в информации о продукте.

Предпосылка

Для актуализации CPU или FM M7-300/400 нужно средство загрузки (дискета или плата памяти).

Актуализация должна производиться в контексте проекта, содержащего M7-станции (CPU или FM), с выбранной M7-программой.



Осторожно

В процессе актуализации ни в коем случае нельзя отключать напряжение питания, так как из-за этого модуль может быть поврежден.

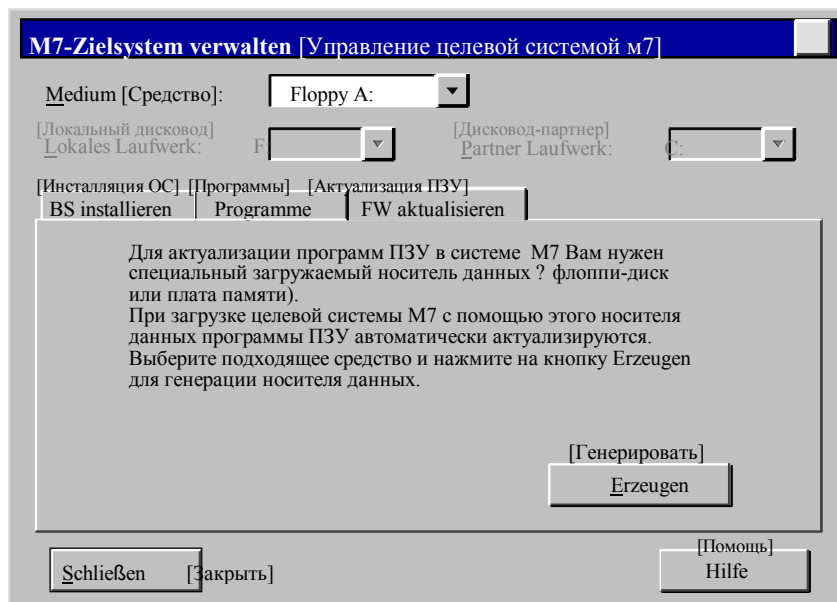


Рис. 17-2. Вкладка "FW aktualisieren" ("Актуализация ПЗУ")

Процедура

Для актуализации программ ПЗУ целевой системы M7 действуйте следующим образом:

1. Выберите в своем проекте M7-программу, соответствующую M7-CPU/FM.
2. Запустите управление M7 командой меню: **Zielsystem → M7-Zielsystem verwalten** (Целевая система → Управление системой M7).
3. Откройте в диалоговом окне вкладку "FW aktualisieren" ("Актуализация ПЗУ") (см. рис. 17-2).
4. Выберите средство загрузки - флоппи-диск или плату памяти.
5. Щелкните на кнопке "Erzeugen" ("Генерировать").

Результат: средство загрузки форматируется (с предварительным предупреждением) и на нее устанавливается новая программа ПЗУ.

6. Вставьте средство загрузки в целевую систему M7 и запустите M7-CPU/FM.
Если система M7 загружается с помощью этого носителя данных, то программы ПЗУ актуализируются автоматически. Актуализация завершена, если непрерывно горят следующие светодиоды:
 - USR в M7-300 CPU/FM
 - USR1 в M7-400 CPU/FM
7. Удалите средство загрузки из системы M7 и загрузите ее с предварительно установленной массовой памяти (платы памяти, жесткого диска и т.д.).



INTF у M7-400).

Указание

Если программы ПЗУ в средстве загрузки несовместимы с типом модуля или старше уже имеющейся версии программ ПЗУ, то актуализация **не выполняется** и с помощью светодиодов индицируется ошибка (светодиод SF у M7-300 и

Если произошла ошибка

В случае ошибки сделайте следующее:

1. Удалите средство загрузки из системы M7.
2. Проверьте, не выше ли в данном случае версия BIOS модуля M7-300/400, чем версия новой программы ПЗУ. Если это так, то актуализация не требуется и невозможна.
3. Проверьте, совпадает ли станция текущего проекта с типом модуля системы M7. Если это не так, адаптируйте соответствующим образом свой проект и проведите актуализацию снова.

17.9. Перенос и стирание программ в системе M7

Применение	<p>STEP 7 предлагает возможность, используя управление M7,</p> <ul style="list-style-type: none">• переносить прикладные M7-программы со всеми соответствующими данными проекта отдельно или вместе с операционной системой в целевую систему M7• стирать любые компоненты программного обеспечения (M7-программы) из целевой системы. <p>Переносить M7-программы в целевую систему M7 и управлять ими можно также с помощью некоторых средств разработки, содержащихся в дополнительном программном пакете M7, например, CFC или отладчика Organon (см. документацию соответствующего дополнительного пакета). Однако, таким способом программы нельзя перенести постоянно на массовую память, а можно только временно загрузить в главную память системы M7.</p>
Предпосылка	<p>Для получения возможности переноса прикладных программ через MPI в целевую систему M7 должна уже быть в наличии операционная система, с помощью которой M7-300/400 может запускаться и устанавливать связь через MPI с PC/PG.</p> <p>В противном случае прикладную программу можно установить также и вместе с операционной системой.</p>
Принципиальная последовательность действий	<p>Для переноса в целевую систему M7 программ на языках C, C++ или других программ, предназначенных для MS-DOS/Windows, действуйте следующим образом:</p> <ol style="list-style-type: none">1. Выберите контейнер с M7-программой, соответствующий модулю M7 (CPU или FM).2. Вызовите в контексте своего проекта команду меню Zielsystem → M7–Zielsystem verwalten (Целевая система → Управление системой M7).3. Откройте вкладку "Programme" ("Программы").4. Выберите (см. рис. 17–3):<ul style="list-style-type: none">- программы в инструментальной системе (Erstellsystem)- средство переноса (Medium)- локальный дисковод и дисковод-партнер, если используется средство "MPI/RFS"5. Нажмите на кнопку "Installieren" ("Инсталлировать"). <p>Все дальнейшие действия зависят от выбранного средства.</p>
Выбор программ	<p>В поле "Erstellsystem" ("Инструментальная система") перечислены все программы на языках C и C++, соответствующие внутри Вашего проекта целевой системе M7. Для переноса можно выбрать одну или несколько из этих программ. В поле "Zielsystem" ("Целевая система") отображаются программы, находящиеся в целевой системе.</p>

Средства переноса

У Вас есть два способа для переноса прикладных программ из STEP 7:

- Online через MPI/RFS
- Offline через гибкий диск или плату памяти

Выбор локального дисковода и партнера

Если для инсталляции используется средство переноса "MPI/RFS", то, как и при инсталляции операционной системы, Вы можете выбрать **ДИСКОВОДА-локальный дисковод** и **дисковод-партнер** (см. рис. 17-5).

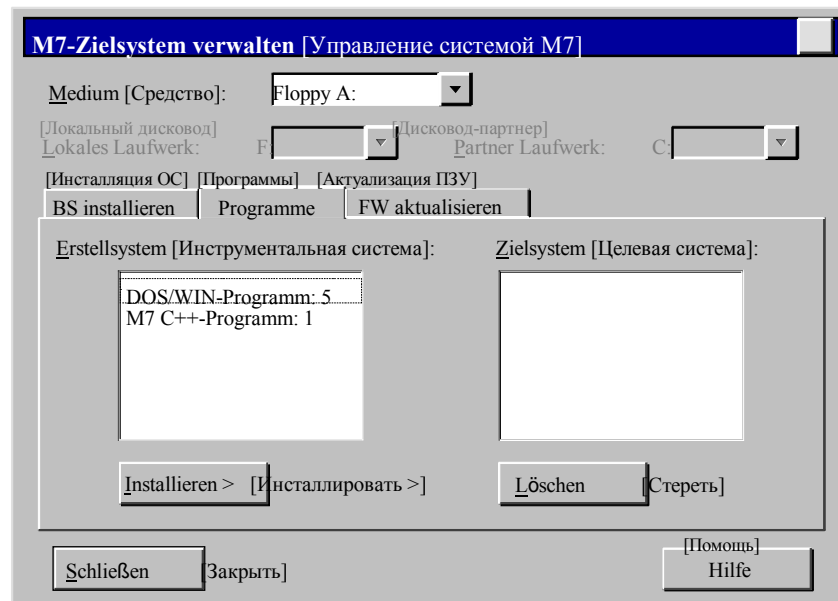


Рис. 17-3. Вкладка "Programme" ("Программы")

Перенос M7-программ через MPI/RFS

При переносе в режиме online необходимые части программы переносятся через MPI непосредственно на массовый накопитель целевой системы и вносятся соответствующие стартовые командные файлы в `\etc\inittab` целевой системы, так что при следующем запуске системы программы запускаются автоматически. Дополнительно для каждой M7-программы на языке C/C++ переносится специальный файл описаний, содержащий всю необходимую информацию для отображения и стирания программы. Имя этого файла создается автоматически из имени программы, причем гарантируется, что это имя файла в целевой системе является уникальным.

Для переноса M7-программ через MPI/RFS в целевую систему M7 действуйте следующим образом:

1. Запустите систему M7. Для запуска системы можно также использовать загрузочную дискету или плату памяти.
2. Запустите управление M7 командой меню: **Zielsystem → M7-Zielsystem verwalten** (Целевая система → Управление системой M7).
3. Откройте в диалоговом окне вкладку "Programme" ("Программы").
4. Выберите (см. рис. 17–3):
 - Medium (средство): "MPI/RFS"
 - Lokales Laufwerk (локальный дисковод): первый свободный дисковод, например, F:
 - Partner Laufwerk (дисковод-партнер): C: для жесткого диска
 - желаемые прикладные программы из списка "Erstellsystem" ("Инструментальная система").
5. Выберите кнопку "Installieren" ("Инсталлировать")

Результат: устанавливается связь через MPI с целевой системой и выбранные компоненты программного обеспечения переносятся на дисковод целевой системы. Перенесенные программы отображаются в поле "Zielsystem" ("Целевая система").

Программы автоматически запускаются при следующем старте системы.

При выполнении этих шагов в диалоговом окне выводится информация о текущих событиях.



Осторожно

Если в целевой системе находятся одноименные файлы, то в процессе переноса они переписываются! Автоматического переименования и автоматического создания резервных копий не предусмотрено!

Перенос M7-программ с помощью носителя данных

При переносе в режиме offline все файлы сначала копируются на дискету или плату памяти. Дополнительно на носителе данных создается установочный файл **m7swins.bat**, с помощью которого выбранные в последний раз программы переносятся с дискеты или платы памяти в массовую память целевой системы M7. файл m7swins.bat должен быть выполнен через CLI (Call Level Interface - Спецификация формулировки запроса) операционной системы M7 RMOS32.

Для переноса M7-программы посредством носителя данных действуйте следующим образом:

1. Запустите управление M7 командой меню: **Zielsystem → M7-Zielsystem verwalten** (Целевая система → Управление системой M7).
2. Вставьте носитель данных в дисковод PC/PG.
3. Откройте в диалоговом окне вкладку "Programme" ("Программы").
4. Выберите (см. рис. 17–3):

- "Floppy Disk" ("Гибкий диск") или "Memory Card" ("Плата памяти") в качестве средства (Medium).
- желаемые прикладные программы из списка для выбора "Erstellsystem" ("Инструментальная система").

5. Нажмите кнопку "Installieren" ("Инсталлировать").

Результат: выбранные компоненты программного обеспечения переносятся на носитель данных.

6. Вставьте носитель данных в целевую систему M7.
7. Запустите CLI локально на M7-300/400 или через удаленный терминал (Remote Terminal).
8. Вызовите находящийся на носителе данных командный файл **m7swins.bat** для копирования компонентов программного обеспечения на жесткий диск. Командный файл m7swins.bat всегда копирует на активный в данный момент дисковод. Т.е. для переноса с дискеты на жесткий диск введите, например, следующую последовательность команд:

```
cd c:\
A:\m7swins.bat
```



Осторожно

Если в целевой системе имеются одноименные файлы, то в процессе переноса они переписываются!

9. Командные файлы для запуска программ не вносятся автоматически в \etc\inittab целевой системы M7. Вместо этого на носителе данных создается временный файл \etc\inittab.ins, содержащий все необходимые записи. Для автоматического запуска программ при следующем старте системы M7-300/400 Вы должны перенести эти записи с помощью редактора в файл \etc\inittab целевой системы M7.

Правила переноса в режиме offline

отображения и стирания программы. Имя этого файла создается автоматически из имени программы. Для обеспечения

При переносе M7-программ в целевую систему на носителе данных в режиме offline для каждой M7-программы на носителе сохраняется специальный файл описаний, содержащий всю необходимую информацию для уникальности этого имени файла в целевой системе должно быть выполнено одно из следующих условий:

- Имена программ, определенных для M7-CPU или M7-FM различаются в первых 5 знаках.

Все программы, относящиеся к одному CPU или FM, копируются на носитель данных и оттуда переносятся в целевую систему M7.



система”) и вследствие этого не может быть стерт.

Указание

Если не принять это во внимание, то возникает опасность, что при дальнейшем обращении через MPI некоторый компонент программного обеспечения более не отображается в списке для выбора ”Zielsystem” (“Целевая

Стирание M7-программ

Для стирания M7-программ с целевой системы M7 в режиме online действуйте следующим образом:

1. Выполните шаги с 1-го по 4-ый, как при переносе прикладных программ через MPI/RFS.
2. Выберите из списка ”Zielsystem” (“Целевая система”) программные компоненты, которые необходимо стереть.
3. Нажмите кнопку ”Löschen” (“Стереть”).

Результат: выбранные программные компоненты удаляются с дискового целевой системы.

Запуск M7-программ

Для запуска прикладных программ в системе M7 имеются следующие возможности:

1. Во время эксплуатации управлением через локальную консоль или через Remote Terminal Interface (RTI, интерфейс удаленного терминала). Управление через RTI описано в руководстве пользователя M7–SYS.
2. При запуске системы при помощи записи в файле **\etc\inittab**. Этот файл читается сразу после запуска операционной системы. Он содержит вызовы всех программ, которые должны автоматически выполняться при старте системы.

Если прикладные программы устанавливаются вместе с операционной системой или через "MPI/RFS", то они автоматически вносятся в файл **\etc\inittab**.

При переносе программ без операционной системы через носитель данных (дискету или плату памяти) Вы должны эти записи сделать самостоятельно, если программы должны автоматически выполняться при запуске системы. Соответствующие записи Вы найдете во временном файле **\etc\inittab.ins**.

17.10. Функции опроса и управления для M7–300/400

Справочные функции

Командой меню **Zielsystem → Baugruppenzustand** (Целевая система → Состояние модуля) можно получить на PC/PG следующую информацию о центральных модулях M7:

- система времени и время CPU
- данные M7–CPU
- степень использования памяти пользователя
- времена циклов CPU
- состояние коммуникационных соединений
- содержимое диагностического буфера

В чем отличие?

В отличие от модулей S7 Вы не можете запрашивать следующую информацию для M7-CPU:

- данные блоков и
- содержимое стеков

Правда, соответствующие вкладки и поля на рабочей панели доступны, но не содержат никакой информации.

Сообщения CPU

С помощью функции **”CPU Melden”** (”Сообщения CPU”) могут быть выведены асинхронные сообщения о сбойных событиях и сообщения, определенные пользователем (см. гл. 15: ”Отображение сообщений CPU”).

Установки

На M7-CPU, как и на S7-CPU, можно выполнить следующие установки:

- изменить рабочий режим, произвести общее стирание CPU (см. гл. 13: ”Загрузка прикладных программ”)
- установить время (см. гл. 15: ”Отображение данных о CPU”, стр. 15–6)

Опрос и управление переменными

Командой меню **Zielsystem → Variable beobachten/steuern** (Целевая система → Наблюдение/управление переменными) Вы получаете следующие функции для обработки таблицы переменных (см. гл. 14: ”Наблюдение и управление переменными”):

- чтение содержимого блоков данных, входов, выходов и меркеров
- запись содержимого блоков данных, входов, выходов и меркеров.

Обзор

Сохранение прикладных программ производится в SIMATIC Manager.

Распечатка составной части проекта для проектной документации производится всегда в той же среде, где эта часть была разработана.

Обзор главы

В раз-деле	Вы найдете	на стр.
18.1	Архивирование и деархивирование проектов и библиотек	18–2
18.2	Сохранение содержимого ОЗУ CPU во встроенном СППЗУ	18–4
18.3	Сохранение блоков и прикладных программ на плате памяти	18–5

18.1. Архивирование и деархивирование проектов и библиотек

Применение

Отдельные проекты или библиотеки можно сохранять в сжатой форме в архивном файле. Такое сжатое хранение возможно на жестком диске или на переносимых носителях данных (дискетах).

Функция архивирования предлагает Вам интерфейс для вызова предпочитаемой Вами программы-архиватора.

Используемые архиваторы

Вы можете использовать следующие программы-архиваторы:

- *pkzip* начиная с версии 2.04g
- *arj* начиная с версии 2.41a
- *lha* с версии 2.13

Названные программы-архиваторы сконфигурированы в стандартной конфигурации STEP 7. Если сконфигурированные программы-архиваторы не были установлены в пути поиска, то при первом обращении появляется сообщение, что программа не найдена. После этого Вы можете дополнительно установить этот архиватор или - если он уже установлен вне пути поиска - зарегистрировать путь к месту, где он хранится.

Предпосылки

Вы должны были установить архиватор в своей системе.

- Каталог, в котором находится архиватор, должен содержаться в пути поиска DOS.

Вы можете отобразить путь поиска DOS в окне "MS-DOS—Eingabeaufforderung" ("Требуется ввод MS-DOS") командой PATH.

- Все проектные данные без исключения должны быть помещены в каталоге или подкаталогах проекта. Правда, при работе со средой разработки С данные можно хранить и в другом месте. Но тогда эти данные не будут включены в архивный файл.
- Имена файлов должны соответствовать соглашению об именах DOS (8 знаков с добавлением 3 знаков для расширения имени файла), так как архивированная программа является программой DOS.



Указание

Обратите внимание, что подкаталоги создаются автоматически! !

При архивировании/деархивировании открывается окно DOS. Вы можете продолжать работу в SIMATIC Manager, если только это окно закрыто. В свойствах программы-архиватора Вы можете установить, чтобы окно DOS по завершении процесса архивирования/деархивирования автоматически закрывалось. Для этого пометьте, например, в WIN 95— Explorer программу-архиватор и выберите команду меню **Datei → Eigenschaften...** (Файл → Свойства...).

Архивирование проектов/ библиотек

Для сохранения проекта/библиотеки в архиве действуйте следующим образом:

1. Обеспечьте, чтобы для проекта/библиотеки не было открытых окон.
2. Выберите команду меню **Datei → Archivieren...** (Файл → Архивировать...).
3. Установите в следующем диалоговом окне целевой каталог, имя и тип архивного файла. С помощью типа файла STEP 7 выясняет подлежащую применению программу-архиватор (напр., "zip" для PKZIP).
4. Выберите в следующем диалоге каталог подлежащего архивированию проекта/библиотеки и подтвердите ввод.
5. В следующем диалоговом окне можно сделать специальные установки для архивирования (напр., сохранение на нескольких дисках).

Открывается окно DOS, в котором происходит архивирование. Проект/библиотека сжимается и сохраняется в целевом каталоге.

Копирование на дискету

Вы можете заархивировать проект/библиотеку, как описано выше, а затем скопировать архивный файл на дискету. Можно также в диалоге "Archivieren" ("Архивирование") сразу выбрать дисковод для дискет.

Обработка архивированных библиотек

Проекты в архивах не могут обрабатываться непосредственно. Для обработки архивированного проекта/библиотеки Вы должны извлечь **проектов/данные** из архива. Действуйте следующим образом:

1. Выберите в SIMATIC Manager команду меню **Datei → Dearchivieren...** (Файл → Деархивировать...).
2. Выберите в появившемся диалоговом окне архивный файл, содержащий сжатый проект/библиотеку.
3. В следующем диалоговом окне выберите целевой каталог, в котором должен быть сохранен проект/библиотека. Указание: если Вы не укажете новый каталог, то будет переписано содержимое старого!
4. В следующем диалоговом окне можно установить параметры для деархивирования, например, поведение при переписывании файлов или как должны использоваться содержащиеся в архиве данные о путях.

После завершения диалога открывается окно DOS, в котором происходит деархивирование. При этом создается проект/библиотека, куда переписывается содержимое архива. теперь можно открыть и обрабатывать проект/библиотеку или копировать его части и вставлять в другой проект.



Указание

Деархивированные проекты/библиотеки при первом открытии в соответствующем диалоговом окне еще не отображаются для выбора. В этом случае нажмите на кнопку "Durchsuchen" ("Поиск") и выберите таким способом проект/библиотеку. Если после этого Вы снова откроете проект/библиотеку, Вам будет предложен для выбора и деархивированный объект.

18.2. Сохранение содержимого ОЗУ CPU во встроенном СППЗУ

Применение

В модулях CPU, имеющих встроенное СППЗУ, можно туда копировать содержимое ОЗУ, чтобы не потерять данные при исчезновении напряжения питания или общем стирании.

Последовательность Для копирования содержимого ОЗУ во встроенное СППЗУ действуйте **действий** следующим образом:

1. Создайте доступ к CPU, например, выбором этого модуля в отображении проекта в режиме online или в окне "Erreichbare Teilnehmer" ("Доступные абоненты").
2. Выберите команду меню **Zielsystem → RAM nach ROM kopieren...** (Целевая система → Копировать ОЗУ в ПЗУ).

Окно с видом online для открытого проекта можно вывести на экран командой меню **Ansicht → online** (Вид → online), а окно "Erreichbare Teilnehmer" ("Доступные абоненты") - командой меню **Ansicht → Schnittstellen → Erreichbare Teilnehmer** (Вид → Интерфейсы → Доступные абоненты).

18.3. Сохранение блоков и прикладных программ на плате памяти

Применение Платы памяти - это портативные носители данных. Они содержат в качестве блоков памяти электрически стираемые флэш-СППЗУ. Запомненные данные сохраняются при исчезновении напряжения питания и при общем стирании CPU. Платы памяти записываются в инструментальной системе и затем используются в CPU.

Содержимое платы памяти Если щелкнуть на панели интерфейсов в SIMATIC Manager на кнопке "S7-Memory Card" ("Плата памяти S7"), то в окне отображается структура объектов платы памяти. Плата памяти при этом должна находиться в слоте инструментальной системы.

Сохранение на плате памяти Для сохранения блоков или прикладной программы на плате памяти действуйте следующим образом:

1. Проверьте, вставлена ли плата памяти в PG, и щелкните на кнопке "Memory Card" ("Плата памяти") на панели интерфейсов SIMATIC Manager.
2. Пометьте блоки или прикладную программу, которые Вы хотели бы сохранить, в окне проекта offline или online.
3. Просто отбуксируйте помеченные объекты с помощью мыши в окно, в котором отображено содержимое платы памяти. Но Вы можете также скопировать объекты командой меню **Bearbeiten → Kopieren** (Редактировать → Копировать) в промежуточный файл и вывести в окно командой меню **Bearbeiten → Einfügen** (Редактировать → Вставить).
4. Если блок уже находится на плате памяти, то выводится сообщение об ошибке. В этом случае сотрите содержимое платы памяти и повторите шаги 3 и 4.

Результат: исполняемые блоки сохраняются на плате памяти.

Стирание платы памяти Содержимое платы памяти для S7 можно стереть только целиком. Стирание отдельных блоков здесь невозможно. Напротив, на платах памяти для M7 с системой флэш-файлов можно стирать также и отдельные объекты.

Для стирания платы памяти действуйте следующим образом:

1. Проверьте, вставлена ли плата памяти в PG, и щелкните на кнопке "Memory Card" ("Плата памяти") на панели интерфейсов в SIMATIC Manager.
В левой части появившегося окна виден контейнер, представляющий содержимое платы памяти. В правой части отображаются содержащиеся в нем объекты.
2. В S7 сотрите контейнер в левой части окна, а в M7 - контейнер или отдельные объекты контейнера.

Применение

Когда Вы завершили разработку программы для решения задачи автоматизации, Вы можете распечатать все важные данные для проектной документации.

Части проекта, которые Вы можете распечатать

Распечатать можно следующие составные части проекта:

- блоки
- таблицу символов с символическими именами абсолютных адресов
- конфигурационную таблицу с размещением модулей в AS и параметрами модулей
- содержимое диагностического буфера
- таблицу переменных с форматами статуса и текущими и управляемыми величинами
- справочные данные, т.е. списки перекрестных ссылок, планы занятости, таблицы структуры программы
- таблицу глобальных данных
- свойства программы и список блоков программы
- справочные данные о CPU с состоянием модуля

Принципиальная последовательность действий

Для распечатки действуйте следующим образом:

1. Откройте нужный объект, чтобы отобразить на экране подлежащую распечатке информацию.
2. Выведите на экран диалоговое окно "Drucken" ("Печать") командой меню **Datei → Drucken...** (Файл → Печатать...) в соответствующем окне.
В зависимости от окна первая запись в строке меню вместо "Datei" ("Файл") может быть и другой, например, "Symboltabelle" ("Таблица символов").
3. Измените при необходимости установки печати (напр., принтер, область печати, количество копий) в диалоговом окне и закройте его. Более подробную информацию об этом Вы найдете в online-помощи.

Блоки открывать не нужно. Вы можете их вывести непосредственно в SIMATIC Manager командой меню **Datei → Drucken...** (Файл → Печатать...).

Формат

Приспособьте, пожалуйста, применяемый для распечатки бланк, к желаемому формату бумаги. Если бланк слишком широк, правый край распечатывается на следующем листе. Вы можете установить формат страницы и текст колонтитулов.

Открытие проектов версии 1



Обзор

В SIMATIC Manager имеется возможность повторного использования проектов, созданных с помощью первой версии STEP 7. Для этого STEP 7 преобразует проекты версии 1 в новые проекты версии 2.

Сохраняются следующие компоненты проекта версии 1:

- структура проекта с программами
- блоки
- исходные тексты на AWL
- таблица символов

Конфигурация аппаратуры не конвертируется. Сохраненные компоненты программ можно скопировать в другие проекты или дополнить новый проект станцией, которую нужно соответствующим образом сконфигурировать и параметризовать.

Версия блока

Отдельные блоки остаются по своим свойствам блоками версии 1.

Сгенерированный в версии 1 код не меняется, поэтому эти блоки нельзя применять в связи с мультиэкземплярами.

Если Вы хотите эти блоки преобразовать в блоки версии 2 (пригодные для модели мультиэкземпляров), сгенерируйте из них исходный текст на AWL, а затем скомпилируйте его снова в блоки.

Последовательность действий Для открытия проекта версии 1 действуйте следующим образом:

1. Выберите в SIMATIC Manager команду меню **Datei → Version 1–Projekt öffnen** (Файл → Открыть проект версии 1).
2. В диалоге "S7 Projekt öffnen" ("Открытие проекта S7") выберите проект версии 1, который Вы хотите повторно использовать. Проект версии 1 Вы распознаете по расширению файла *.s7a (см. предварительную установку для "Тур" ("Тип")).
3. В следующем диалоге "Neues Projekt" ("Новый проект") введите имя проекта для версии 2.

Результат: STEP 7 преобразует проект версии 1 в проект версии 2. Новый проект открывается в SIMATIC Manager.

Список литературы

- /21/ Брошюра: *Automatisierungssystem S7/M7, Dezentralisieren mit PROFIBUS-DP und AS-I*
Система автоматизации S7/M7, Децентрализация с помощью PROFIBUS-DP и AS-I
- /30/ Fibel: *Automatisierungssystem S7-300, Einfach aufbauen und programmieren*
Букварь: *Система автоматизации S7-300, Простой монтаж и программирование*
- /70/ Handbuch: *Automatisierungssystem S7-300, Aufbauen der CPU-Daten*
Руководство: *Система автоматизации S7-300, Организация данных CPU*
- /100/ Installationshandbuch: *Automatisierungssystem S7-400, M7-400, Aufbauen*
Руководство по установке: *Система автоматизации S7-400, M7-400, Монтаж*
- /230/ Benutzerhandbuch: *Basissoftware für S7, STEP 5-Programme konvertieren*
Руководство пользователя: *Базовое программное обеспечение для S7, Конвертирование программ S5*
- /231/ Benutzerhandbuch: *Basissoftware für S7 und M7, STEP 7*
Руководство пользователя: *Базовое программное обеспечение для S7 и M7, STEP 7*
- /232/ Handbuch: *AWL für S7-300/400, Bausteine programmieren*
Руководство: *AWL для S7-300/400*
Программирование блоков
- /233/ Handbuch: *KOP für S7-300/400, Bausteine programmieren*
Руководство: *KOP для S7-300/400*
Программирование блоков
- /234/ Programmierhandbuch: *Systemsoftware für S7-300/400*
Programmentwurf
Руководство по программированию: *Системное программное обеспечение для S7-300/400*
Разработка программ
- /235/ Referenzhandbuch: *Systemsoftware für S7-300/400*
System- und Standardfunktionen
Справочное руководство: *Системное программное обеспечение для S7-300/400*
Системные и стандартные функции

- /250/ Handbuch: *SCL für S7-300/400*, Bausteine programmieren
 Руководство: *SCL для S7-300/400*, Программирование блоков
- /251/ Handbuch: *GRAPH für S7-300/400*,
 Ablaufsteuerungen programmieren
 Руководство: *GRAPH для S7-300/400*,
 Программирование систем управления исполнением
- /252/ Handbuch: *HiGraph für S7-300/400*,
 Zustandsgraphen programmieren
 Руководство: *HiGraph для S7-300/400*,
 Программирование графов состояний
- /253/ Handbuch: *C für S7-300/400*,
 C-Programme erstellen
 Руководство: *C для S7-300/400*,
 Разработка программ на языке C
- /254/ Handbuch: *CFC für S7 und M7*,
 Technologische Funktionen graphisch verschalten
 Руководство: *CFC для S7 и M7*,
 Графический монтаж технологических функций
- /280/ Programmierhandbuch: *Systemsoftware für M7-300/400*,
 Programmwurf
 Руководство по программированию: *Системное программное обеспечение для M7-300/400*,
 Разработка программ
- /281/ Referenzhandbuch: *Systemsoftware für M7-300/400*,
 System- und Standardfunktionen
 Справочное руководство: *Системное программное обеспечение для M7-300/400*,
 Системные и стандартные функции
- 282/ Benutzerhandbuch: *Systemsoftware für M7-300/400*,
 Installieren und Bedienen
 Руководство пользователя: *Системное программное обеспечение для M7-300/400*,
 Установка и управление
- /290/ Benutzerhandbuch: *ProC/C++ für M7-300/400, C-Programme erstellen*
 Руководство пользователя: *ProC/C++ для M7-300/400, Разработка программ на язык C*
- /291/ Benutzerhandbuch: *ProC/C++ für M7-300/400*,
 Debugger für C-Programme
 Руководство пользователя: *ProC/C++ для M7-300/400*,
 Отладчик для программ на языке C

Глоссарий

Адрес абонента	По адресу абонента производится обращение к устройству (напр., PG) или к программируемому модулю (напр., CPU) в сети (напр., MPI, L2).
Адрес, абсолютный	Абсолютный адрес - это характеристика определенного операнда, указывающая ячейку памяти, начиная с которой хранится этот операнд. Примеры: вход E 12.1; меркерное слово MW25; блок данных DB3.
Адресация, абсолютная	При абсолютной адресации указывается адрес подлежащего обработке операнда. Пример: адрес A 4.0 обозначает бит 0 в байте 4 отображения процесса на выходах.
Адресация, прямая	При прямой адресации операнд содержит адрес в памяти того значения, над которым должна выполняться операция. При этом операнд может адресоваться абсолютно или символически.
Адресация, символическая	При символической адресации операнд, подлежащий обработке, указывается символически (вместо адреса). Соответствие между символами и адресами устанавливается помощью таблицы символов.
Библиотека	Контейнер для многократно применяемых блоков, исходных текстов и планов.
Блок	<p>Блоки - это части прикладной программы, ограниченные своей функцией, структурой или целью применения. В STEP 7 имеются:</p> <ul style="list-style-type: none">• кодовые блоки (FB, FC, OB, SFB, SFC)• блоки данных (DB, SDB)• типы данных, определенные пользователем (UDT)
Блок данных (DB)	Блоки данных - это области данных в прикладной программе, содержащие данные пользователя. Имеются глобальные блоки данных, к которым можно обратиться из любого кодового блока, и экземпляры блоков данных, ставящиеся в соответствие определенному вызову FB.

Буферизация	<p>В SIMATIC S7 информация, находящаяся в областях ОЗУ (в рабочей памяти), может быть защищена:</p> <ul style="list-style-type: none"> • посредством буферной батареи; в этом случае содержимое рабочей памяти и областей чтения/записи загрузочной памяти всегда сохраняется, также как и счетчики, таймеры и меркеры (эта область параметризуется). • без буферной батареи (эксплуатация без обслуживающего персонала); в этом случае постоянно сохраняется максимально возможное (специфическое для CPU) количество данных из рабочей памяти, памяти записи/чтения загрузочной памяти, а также максимально возможное количество таймеров, счетчиков и меркеров в резервном буфере CPU.
Ввод, инкрементный Встречающиеся ошибки выделяются и должны быть исправлены до завершения ввода. Инкрементный ввод возможен,	<p>При инкрементном вводе блока каждая строка или каждый элемент немедленно проверяется на наличие ошибок ввода (например, синтаксис). Встречающиеся ошибки выделяются и должны быть исправлены до завершения ввода. Инкрементный ввод возможен, например, в языках программирования AWL, KOP, GRAPH и HiGraph.</p>
Ввод, ориентированный на исходный текст	<p>При вводе, ориентированном на исходный текст, блоки или вся прикладная программа редактируются в виде исходного текста. Проверка синтаксиса производится только при компиляции. Ввод, ориентированный на исходный текст, возможен, например, в языках программирования AWL и SCL.</p>
Время контроля цикла	<p>Если время обработки прикладной программы превышает установленное время контроля цикла, то операционная система генерирует сообщение об ошибке и CPU переходит в состояние STOP.</p>
Время цикла	<p>Время цикла - это время, необходимое CPU для однократной обработки прикладной программы.</p>
Глобальные данные	<p>Глобальные данные - это данные, к которым можно обратиться из любого кодового блока (FC, FB, OB). В частности, это меркеры (M), входы (E), выходы (A), таймеры, счетчики и элементы блоков данных (DB). К глобальным данным можно обращаться абсолютно или символически.</p>
Диагностический буфер	<p>Диагностический буфер - это буферизованная область памяти в CPU, в которой хранятся диагностические события в последовательности их возникновения.</p>
Диагностическое событие	<p>Диагностическое событие приводит к внесению записи в диагностический буфер CPU. Они делятся на:</p> <ul style="list-style-type: none"> • ошибки в модуле • неисправности в соединении с процессом • системные ошибки в CPU • переходы из одного режима работы CPU в другой • ошибки в прикладной программе • сообщения пользователя

Загрузка в AS	Загрузка загружаемых объектов (напр., кодовых блоков) из устройства программирования в загрузочную память программируемого модуля. Это может происходить как через непосредственно подключенное к CPU устройство программирования, так и через SINEC L2.
Загрузка в PG	Загрузка загружаемых объектов (напр., кодовых блоков) из загрузочной памяти центрального модуля в устройство программирования. Это может происходить как через непосредственно подключенное к центральному модулю устройство программирования, так и через SINEC L2.
Загрузочная память	Загрузочная память - это составная часть центрального модуля. Она содержит объекты, созданные устройством программирования (загружаемые объекты). Она реализуется либо в виде вставной платы памяти или в виде жестко встроенной памяти.
ЗАПУСК	ЗАПУСК - это режим работы CPU, имеющий место при переходе из состояния STOP в состояние RUN. ЗАПУСК вызывается с помощью переключателя режимов работы на CPU, после включения напряжения сети или командой на устройстве программирования.
Значение по умолчанию	Значение по умолчанию - это рациональная установка, используемая во всех случаях, когда не введено другое значение.
Идентификатор DP	Однозначный код мест установки ведомых DP (DP-Slaves). Кодированы вид модуля, длина адресного пространства и консистентность данных (байт, слово). Пример: 2DE для двухканального цифрового модуля ввода.
Инструментальная система	Инструментальная система служит для разработки прикладных программ для целевой системы. Инструментальной системой является, например, устройство программирования с программным обеспечением Windows и STEP 7.
Интерфейс, многоточечный (MPI)	Многоточечный интерфейс - это интерфейс устройства программирования SIMATIC S7. Он обеспечивает возможность одновременной работы нескольких устройств программирования, текстовых дисплеев, панелей оператора на одном или нескольких центральных модулях. Абоненты MPI связаны между собой через систему шин. (см. также Связь с помощью глобальных данных).
Исходный текст	Часть программы, создаваемая с помощью графического или текстового редактора, из которой путем компиляции возникает исполняемая прикладная S7-программа или машинный код для M. Исходный текст хранится в контейнере "Quellen" (SO) ("Исходные тексты") под S7- или M7-программой.

Кодовый блок	<p>Кодовый блок в SIMATIC S7 - это блок, содержащий часть прикладной программы STEP 7. В противоположность ему, блок данных содержит только данные. Имеются следующие кодовые блоки:</p> <ul style="list-style-type: none"> • организационные блоки (OB) • функциональные блоки (FB) • функции (FC) • системные функциональные блоки (SFB) • системные функции (SFC)
Команда управления FREEZE	Ведущий DP (DP–Master) посылает команду FREEZE группе ведомых DP (DP-Slaves) и побуждает их заморозить состояния их входов со значениями на данный момент времени.
Команда управления SYNC	Ведущий DP (DP–Master) посылает команду SYNC группе ведомых DP (DP-Slaves) и побуждает их заморозить состояния их выходов со значениями на данный момент времени.
компилировать	Создавать исполняемую прикладную программу из исходного текста.
Коммуникационный функциональный блок (CFB)	<p>Коммуникационные функциональные блоки - это системные функциональные блоки для обмена данными и управления программами.</p> <p>Примеры обмена данными:</p> <ul style="list-style-type: none"> • SEND • RECEIVE • GET <p>Примеры управления программами:</p> <ul style="list-style-type: none"> • перевод центрального модуля партнера по связи в состояние STOP • опрос STATUS'а центральных модулей партнера по связи
Консистентные данные	Консистентными называются данные, которые содержательно связаны друг с другом и не могут быть разделены, например, указание времени суток.
Контактный план (КОР)	Контактный план - это графический язык программирования. Синтаксис его команд возник из коммутационных схем.
Конфигурация станции	Объект с данными о конфигурации и параметрах станции. Данные о конфигурации и параметры станции хранятся в системных блоках данных (SDB).
Конфигурирование	Выбор и компоновка отдельных компонентов системы автоматизации или установка необходимого программного обеспечения (напр., операционной системы на управляющей ЭВМ М7) и адаптация к специальному использованию (напр., путем параметризации модулей).

Межсетевой шлюз	Переход между подсетями общей сети. При этом может также идти речь о переходе между (под)сетями с различными свойствами (например, переход между SINEC L2 и SINEC H1).
Меркеры (M)	Область в системной памяти SIMATIC S7 CPU. К ней возможен доступ на запись и чтение (битами, байтами, словами и двойными словами). Область меркеров может быть использована для хранения промежуточных результатов.
Модуль, программируемый	К программируемым модулям относятся центральные модули (CPU), функциональные модули (FM) и коммуникационные процессоры (CP). CPU, FM и CP могут обрабатывать прикладные программы, они обмениваются друг с другом данными через коммуникационную шину (К-шину).
Мультиэкземпляры	При применении мультиэкземпляров экземпляр блока данных содержит данные для нескольких функциональных блоков иерархии вызовов.
Новый пуск	При запуске центрального модуля (например, при переводе переключателя режимов работы из STOP в RUN или при включении напряжения сети) перед циклической обработкой программы (OB 1) сначала выполняется организационный блок OB 101 (повторный пуск; только у S7-400) или организационный блок OB 100 (новый пуск). При новом пуске считывается отображение процесса на входах и прикладная программа STEP 7 обрабатывается начиная с первой команды в OB 1.
Общее стирание	При общем стирании сбрасываются следующие виды памяти CPU: <ul style="list-style-type: none"> • рабочая память • область записи/чтения загрузочной памяти • системная память за исключением параметров MPI и диагностического буфера
ОЗУ (RAM)	ОЗУ (Оперативное Запоминающее Устройство), или RAM (Random Access Memory - память с произвольным доступом) - это память для чтения и записи, в которой каждая ячейка имеет отдельный адрес и может быть содержательно изменена. ОЗУ используются для запоминания данных и программ.
Операнд	Операнд - это часть команды STEP 7, указывающая на то, с чем процессор должен что-то сделать. Он может адресоваться абсолютно или символически.
Операционная система	Обобщенное наименование для всех функций, которые управляют и контролируют исполнение программ пользователя, распределение ресурсов между программами пользователя и поддержание режима работы во взаимодействии с аппаратными средствами (напр., MS-DOS).
Организационный	Организационные блоки образуют интерфейс между операционной блок (OB) системой CPU и прикладной программой. В организационных блоках определяется последовательность обработки прикладной программы.
Отображение процесса	Сигнальные состояния цифровых модулей ввода и вывода хранятся в CPU в отображении процесса. Различают отображение процесса на входах (PAE) и отображение процесса на выходах (PAA).

Отображение процесса на входах (РАЕ)	Отображение процесса на входах считывается операционной системой с модулей ввода перед обработкой прикладной программы.
Отображение процесса на выходах (РАА)	Отображение процесса на выходах в конце прикладной программы переносится операционной системой на модули вывода.
Параметризация	Под параметризацией понимается установка параметров модуля.
Переключатель режимов работы	Переключателем режимов работы на центральном модуле устанавливается желаемый рабочий режим.
Переменная	Переменная определяет некоторое данное с переменным содержанием, которое может быть использовано в прикладной программе STEP 7. Переменная состоит из операнда (напр., M 3.1), и типа данных (напр., Bool) и может быть обозначена одним символом (напр., BAND_EIN).
Периферия, децентрализованная	Децентрализованная периферия - это аналоговые и цифровые модули, пространственно удаленные от центрального носителя модулей. Для децентрализованной периферии характерна техника монтажа. Целью этой техники является экономия затрат на соединения благодаря использованию периферийных модулей рядом с процессом.
План	Специальный графический исходный текст, создаваемый с помощью языка программирования CFC. План хранится в контейнере "Pldne" (PL) ("Планы") под S7- или M7-программой.
Плата памяти	Модуль памяти формата чековой карточки, содержащий элементы ОЗУ или СППЗУ. Платы памяти - это средства для запоминания в формате чековых карточек для CPU и CP. Они реализуются как ОЗУ или флэш-СППЗУ.
Повторный пуск	При запуске центрального модуля (например, при переводе переключателя режимов работы из STOP в RUN или при включении напряжения сети) перед циклической обработкой программы (OB 1) сначала выполняется организационный блок OB 101 (повторный пуск; только у S7-400) или организационный блок OB 100 (новый пуск). При повторном пуске считывается отображение процесса на входах и обработка прикладной программы STEP 7 начинается с того места, где она была закончена при последнем прерывании (STOP, выключение напряжения сети).
Прерывание	В SIMATIC S7 имеется 10 различных классов приоритета, регулирующих обработку прикладной программы. К этим классам приоритета относятся в частности и прерывания, например, прерывания по сигналам процесса. При возникновении прерывания операционной системой автоматически вызывается соответствующий организационный блок, в котором пользователь может запрограммировать желаемую реакцию (например, в FB).

Прикладная программа	Прикладная программа содержит все команды и описания, а также данные для обработки сигналов, с помощью которых можно управлять установкой или процессом. Она ставится в соответствие программируемому модулю (напр., CPU, FM) и может быть структурирована делением на более мелкие единицы (блоки).
Проверка синтаксиса	При инкрементном вводе программ STEP 7 после ввода каждой строки выполняется проверка синтаксиса, т.е. проверяется, например, полностью ли введена команда STEP 7. При вводе, ориентированном на исходный текст, проверка синтаксиса производится при компиляции.
Программа	Общее понятие для S7- и M7-программ.
Проект	Контейнер для всех объектов решения задачи автоматизации независимо от количества станций, модулей и их соединения в сеть.
Рабочая память	Рабочая память - это оперативное запоминающее устройство (ОЗУ, RAM) в CPU, к которому обращается процессор при обработке прикладной программы.
Рабочее состояние (режим)	В системах автоматизации SIMATIC S7 имеются следующие (рабочий режим) рабочие состояния: STOP, ANLAUF (ЗАПУСК), RUN (РАБОТА) и HALT (ОСТАНОВ).
Рабочий режим ANLAUF (ЗАПУСК)	Рабочий режим ANLAUF (ЗАПУСК) имеет место при переходе из состояния STOP в состояние RUN. Он может быть вызван с помощью переключателя режимов работы на CPU, после включения напряжения сети или командой на устройстве программирования. В S7-300 выполняется новый пуск. В S7- 400, в зависимости от положения переключателя режимов запуска, выполняется новый пуск или повторный пуск (рестарт).
Рабочий режим HALT (ОСТАНОВ)	Переход в рабочий режим HALT (ОСТАНОВ) осуществляется из рабочего режима RUN командой с PG. В этом режиме возможны специальные тестовые функции.
Рабочий режим RUN (РАБОТА)	В режиме RUN (РАБОТА) происходит обработка прикладной программы, отображение процесса циклически обновляется. Все цифровые выходы деблокированы.

Рабочий режим STOP

Рабочий режим STOP достигается:

- с помощью переключателя режимов работы
- в результате внутренней ошибки в центральном модуле
- с помощью команды на устройстве программирования

В рабочем режиме "STOP" прикладная программа не обрабатывается. Все модули переключены в безопасное состояние. При этом возможно выполнение некоторых программных функций, а также управление и наблюдение.

Реманентный

Данные называются реманентными, если после отключения напряжения питания они сохраняют то же значение, что и перед отключением напряжения.

Реманентность достигается двумя методами буферизации:

- а) буферизацией напряжения
 - б) резервированием
- (см. также Буферизация).

Режим работы

Переключателем режимов работы центрального модуля могут быть установлены следующие режимы работы:

- RUN с возможностью доступа к прикладной программе STEP 7, например, с помощью устройства программирования ("RUN-P"),
- RUN с защитой от доступа (RUN),
- STOP и
- общее стирание ("MRES").

Связь с помощью глобальных данных

Связь с помощью глобальных данных - это способ переноса глобальных данных между CPU (без CFB).

Сеть

Сеть - это соединение абонентов через соединительный кабель для целей коммуникации.

Символ

Символ - это имя, определенное пользователем при соблюдении определенных синтаксических предписаний. Это имя после определения того, взамен чего оно используется (напр., переменной, типа данных, метки перехода, блока), может быть применено при программировании и при управлении и наблюдении. Пример: операнд: E 5.0, тип данных: BOOL, символ: Taster Notaus (кнопка аварийного выключения).

Система с программным управлением от запоминающего устройства (SPS)	<p>Системы с программным управлением от запоминающего устройства (SPS)- это электронные системы, функции которых хранятся в виде программ в устройстве управления. Конструкция и электрический монтаж устройства, таким образом, не зависят от функции управления.</p> <p>Система с программным управлением от запоминающего устройства имеет структуру ЭВМ; она состоит из CPU (центральный модуль) с памятью, модулей ввода/вывода и встроенной системы шин. Периферия и язык программирования ориентированы на требования техники управления.</p>
Системная память	Системная память встроена в центральный модуль и выполнена как ОЗУ. В системной памяти находятся области операндов (напр., таймеры, счетчики, меркеры), а также внутренне необходимые области данных (напр., буфер для коммуникации).
Системная ошибка	Системные ошибки - это ошибки, которые могут возникнуть внутри системы автоматизации(т.е. не в управляемом процессе). Системными ошибками являются, например, программные ошибки в CPU и неисправности в модулях.
Системная функция (SFC)	Системная функция (SFC) - это функция, встроенная в операционную систему CPU, которая при необходимости может быть вызвана в прикладной программе STEP 7.
Системный блок данных (SDB)	Системные блоки данных - это области данных в центральном модуле, содержащие системные установки и параметры модулей. Системные блоки данных создаются и изменяются при конфигурировании.
Системный функциональный блок (SFB)	Системный функциональный блок (SFB) - это функциональный блок, встроенный в операционную систему CPU, который при необходимости может быть вызван в прикладной программе как функциональный блок. Соответствующий экземпляр блока данных находится в рабочей памяти.
Список команд (AWL)	Список команд (AWL) - это текстовый язык программирования, близкий к машинному языку.
Список перекрестных ссылок	Список перекрестных ссылок делает возможным обзор применения операндов областей памяти E, A, M, T, Z, P и DB внутри программы CPU.
Справочные данные	Справочные данные служат для контроля программы CPU и включают в себя список перекрестных ссылок, план занятости, структуру прикладной программы, список свободных операндов и список отсутствующих обозначений.
Станция	Устройство, которое как одно целое может быть подключено к одной или нескольким подсетям, например, система автоматизации, устройство программирования, пульт оператора.
Счетчики (Z)	Счетчики - это составная часть системной памяти CPU. Содержимое этих счетчиков может изменяться командами STEP 7 (напр., считать вперед, назад).

Таблица переменных (VAT)	В таблице переменных собираются переменные, которые нужно наблюдать и которыми нужно управлять, включая соответствующие указания о форматах.
Таблица символов	Таблица соответствия между символами (=именами) и адресами для глобальных данных и блоков. Примеры: Notaus (символ), E 1.7 (адрес), Regler (символ), SFB 24 (блок)
Таблица сообщений	Таблица для определения текстов сообщений и сопоставления этих текстов с событиями.
Таблица соединений	Таблица для определения коммуникационных соединений между программируемыми модулями в сети.
Таймеры (T)	Таймеры - это составные части системной памяти CPU. Содержимое этих таймеров обновляется операционной системой асинхронно по отношению к прикладной программе. Командами STEP 7 определяется точная функция таймера (напр., задержка включения) и производится его запуск.
Устройство программирования (PG)	Персональный компьютер в специальном компактном и пригодном для промышленного использования исполнении. PG полностью оснащен для программирования систем автоматизации SIMATIC.
Функциональный блок (FB)	Функциональный блок в соответствии с IEC 1131-3 - это кодовый блок со статическими данными. FB предоставляет возможность передачи параметров в прикладной программе. Благодаря этому функциональные блоки пригодны для программирования часто встречающихся сложных функций, например, регулирования, выбора режимов работы. Так как FB имеет в своем распоряжении память (экземпляр блока данных), то к его параметрам (напр., выходам) можно обратиться в любое время на любом месте прикладной программы.
Функциональный модуль (FM)	Функциональный модуль (FM) - это модуль, разгружающий центральный модуль (CPU) от критичных по времени или требующих интенсивного использования памяти задач обработки сигналов процесса. Как правило, FM используют внутреннюю коммуникационную шину для быстрого обмена данными с CPU. Примеры применений FM: счет, позиционирование, регулирование.

Функция (FC)	<p>Функции в соответствии с IEC 1131-3 - это кодовые блоки без памяти. Функция дает возможность передачи параметров в прикладной программе. Благодаря этому они пригодны для программирования часто встречающихся сложных функций, например, расчетов.</p> <p>Важно: из-за отсутствия памяти рассчитанные значения должны быть обработаны непосредственно после вызова FC.</p>
Целевая система	В качестве целевой системы обозначается система автоматизации или ее компонент, на котором выполняется прикладная программа. Целевыми системами являются, например, SIMATIC S7, M7 и C7.
Центральный модуль (CPU)	CPU (central processing unit) - это центральный модуль устройства автоматизации или системы автоматизации с управляющим и арифметическим устройством, памятью, системной программой и интерфейсами для периферийных модулей.
Язык программирования	Язык программирования служит для разработки прикладных программ и предоставляет для этого в распоряжение определенный языковый набор графических или текстовых команд. Эти команды вводятся пользователем в редакторе и компилируются в исполняемую прикладную программу.
ANLAUF	см. ЗАПУСК
CFC	Язык программирования для удобного описания непрерывных процессов с помощью графического “монтажа” сложных функций.
C-Programm (программа)	Часть M7-программы, состоящая из исходного текста на языке C и (C-сгенерированного из него машинного кода для программируемых модулей M7.
DP-Master DP)	DP-Master может, если он имеет право на передачу, посылать данные (ведущий другим абонентам сети SINEC L2-DP и запрашивать данные от других абонентов.
DP-Slave (ведомый DP)	DP-Slave может обмениваться данными с ведущим DP (DP-Master) через сеть SINEC L2-DP только по запросу ведущего DP.
M7-Programm (M7-программа)	Контейнер для планов и C-программ для программируемых модулей M7, содержащий также таблицу символов.
MPI-адрес	В сети MPI каждому программируемому модулю должен быть назначен собственный MPI-адрес.

MRES	<p>При общем стирании (MRES) сбрасываются следующие виды памяти CPU:</p> <ul style="list-style-type: none"> • рабочая память • область записи/чтения загрузочной памяти • системная память • резервная память
On-Board Silicon (OSD)	<p>OSD (On-Board Silicon Disk, встроенный кремниевый диск) - это специальный образом буферизованная рабочая память, не теряющая своего содержимого и при исчезновении напряжения питания. OSD встраивается непосредственно в M7-CPU.</p>
Disk	
Online/Offline	<p>В режиме online имеет место связь по данным между системой автоматизации и устройством программирования, в режиме offline - нет.</p>
RAM	<p>См. ОЗУ.</p>
S7-Programm (S7-программа)	<p>Контейнер для блоков, исходных текстов и планов для программируемых модулей S7, содержащий также таблицу символов.</p>
SCL	<p>Паскалеобразный язык высокого уровня в соответствии с IEC 1131-3 для программирования сложных задач в SPS, например, алгоритмов, задач обработки данных.</p>
SIMATIC Manager	<p>Графическая панель пользователя SIMATIC под Windows 95.</p>

А

Авторизация, 2–3
 исходная дискета, 2–4
 перенос, 2–4
Адрес абонента, изменение, 8–11

Б

База справочных данных, генерация, 12–3
Библиотека, 5–9
Блоки
 вставка, 5–9
 дополнительная загрузка, 13–10
 загрузка в AS, 13–8
 загрузка в PG, 13–11
 создание, 11–7
 создание в KOP/AWL, 11–6
 создание посредством S7–GRAPH, 11–8
 создание посредством S7–HiGraph, 11–10
 сохранение на плате памяти, 18–5
 стирание, 13–9

В

Версия блока, А–1
Вид offline, 5–11, 5–12
Вид online, 5–11, 5–12
Возможности загрузки, 13–5
Вставка
 блоков, 5–9
 компонентов аппаратуры, 5–7
 S7/M7–программы, 5–8

Г

Группы графов, 11–9

Д

Деинсталляция, STEP 7, 2–9

организация, 15–12
содержимое, 15–12, 15–13

Диагностическое событие, 15–12
Диалоговые окна, 3–4
Диалог для выбора, 3–12
Дополнительная загрузка
 блоков, 13–10
Дополнительное программное обеспечение, 16–2
Доступные абоненты, 5–14

З

Загрузка
 блоков в AS, 13–8
 блоков в PG, 13–11
 прикладной программы в AS, 13–7
 программы CPU в AS, 13–7
Загрузка в устройство программирования, 7–21
Записи в списке, сортировка, 12–3
Защита от копирования, 2–3
Знак комментария, 14–5

И

Идентификатор партнера, 10–6
Идентификатор DP
 параметризация, 7–18
 присвоение модульному DP–Slave, 7–15
Идентификационные данные CPU, 15–7
Иерархия объектов, 3–7
Изображение сети, 8–8
Импорт, внешних исходных текстов, 5–10
Инсталляция STEP 7, 2–7
Интерфейс PG, 2–10
Интерфейсный модуль, 7–7, 7–9
Информация о состоянии, 15–2
Исходные тексты, 11–2, 11–10
 создание на AWL/SCL, 11–7

К

Каталог модулей, 7–6
Кольцевой буфер (диагностический буфер), 15–12
Команда управления, 7–19
Коммуникационная загрузка CPU, 15–11
Коммуникационные блоки, 10–4
Коммуникационные соединения
 обзор, 10–1
 отображение, 15–11
 последовательность действий, 10–2
 проверка, 15–11
 создание, 10–5
Коммуникационный процессор, 7–9
Контактный план, 11–5
Контекстно зависимая помощь, 3–5
Конфигурационная таблица
 вызов, 7–4
 мастер-системы, 7–10, 7–12, 7–14, 7–15
Конфигурация
 загрузка в целевую систему, 7–20
 загрузка из AS, 7–21
 изменение, 7–22, 7–23
 копирование, 7–23
 расширение, 7–8
 сохранение, 7–20
Конфигурация связи, примеры, 9–7–9–10
Конфигурация сети, 8–1
 загрузка, 8–11
 последовательность действий, 8–3
 особенности подсетей MPI, 8–10
 с помощью NETPRO, 8–6
 с помощью STEP 7, 8–4
Конфигурирование, 7–3
 связи с помощью глобальных данных, 9–2
 сетей, 8–1
 центральной структуры, 7–5
Концепция памяти и загрузки в S7–300, 13–4
Копирование конфигурации, 7–23

Л

Локальный ID, 10–6

М

Мастер-система, создание, 7–11
Модули
 выбор и размещение, 7–6
 параметризация, 7–16
 подчинение модульному DP–Slave, 7–15
Модуль расширения (EXM), 7–7

Н

Наблюдение и управление переменными, 14–6
Носитель модулей, 7–5
 выбор и размещение, 7–6

О

Обучающая программа, вызов, 3–5
Общее стирание CPU, 13–6
Объект
 выбор, 3–12
 вырезание, копирование, вставка, 3–11
 открытие, 3–10
 свойства, 3–11
 создание, 3–10
Объекты, 3–7
ОЗУ, сохранение, 18–4
Окно списка, 3–4
Операционная система M7
 выбор, 17–3
 дополнительная установка, 17–13
 инсталляция, 17–2
 установка на жестком диске, 17–8, 17–9, 17–10
 установка на плате памяти, 17–7
 установка на OSD, 17–11
Организация диагностического буфера, 15–12
Отображение архивных сообщений, 15–4
Отображение данных о CPU, 15–7
Отображение занятости, 12–9
Отображение неприменяемых символов, 12–10
Отображение отсутствующих символов, 12–11
Отображение перекрестных ссылок, 12–7
Отображение содержимого стеков, 15–15
Отображение состояния модулей, 15–5
 M7–300/400, 17–22
Ошибка, во время инсталляции, 2–9

П

Память пользователя
 отображение степени использования, 15–9
 представление, 15–9

- сжатие, 13–12, 15–9
- степень использования, 15–9
- Панель инструментов, символы, 3–3
- Параметризация, 7–3, 7–16, 7–17
 - в прикладной программе, 7–16
 - идентификаторов, 7–18
 - модулей, 7–16
 - DP–Master, 7–18
 - DP–Slave, 7–18
 - SINEC L2–DP–Netz, 7–17
- Параметры шины, установка, 7–18
- Переименование проектов, 3–11
- Переменная, наблюдение и управление, 14–6
- Переменная, наблюдение и управление, M7–300/400, 17–22
- Перенос прикладных программ M7, 17–18
- Печать, 3–11
 - блоков, 19–1
 - конфигурационной таблицы, 19–1
 - свойств программ CPU, 19–1
 - содержимого диагностического буфера, 19–1
 - составных частей проекта, 19–1
 - списков блоков, 19–1
 - справочных данных, 19–1
 - таблицы глобальных данных, 19–1
 - таблицы переменных, 19–1
 - таблицы символов, 19–1
- Планы, 11–2
- Плата памяти, 18–5
 - стирание, 18–5
- Подсеть, 8–1
- Подсоединение к сети, свойства, 8–5
- Помощь (online)
 - вызов, 3–5
 - темы, 3–5
- Потребность в локальных данных, максимальная, 12–5
- Прикладная программа, 11–2
 - загрузка в AS, 13–7
 - создание, 11–1
- Прикладная программа M7
 - запуск, 17–21
 - перенос в целевую систему, 17–16
 - перенос с помощью носителя данных, 17–19
 - перенос через MPI, 17–18
 - стирание, 17–20
- Программа CPU, загрузка в AS, 13–7
- Программирование S7–CPU , 11–2
- Проекты, 11–2
 - архивирование, 18–3
 - открытие, 5–6
 - переименование, 3–11
 - создание, 5–4
- Прямой доступ к контроллеру, 5–14

Р

- Распечатка, для проектной документации, 19–1
- Регистровый диалог, 3–4
- Редакторы
 - инкрементный, 11–4
 - ориентированный на исходный текст, 11–4

С

- Свойства сети, 8–4
- Связь с помощью глобальных данных, 9–1
 - конфигурирование, 9–2
 - примеры, 9–7–9–10
 - принцип, 9–4
 - тестирование, 9–6
- Сети, 11–5
- Сеть SINEC L2–DP, 7–11, 7–12
 - конфигурирование, 7–9
 - параметризация, 7–17
- Сжатие
 - памяти пользователя, 15–9
 - в режиме RUN, 15–9
 - в режиме STOP, 15–9
- Символы
 - объектов STEP 7, 3–7
 - панели инструментов, 3–3
- Синхронизация часов, 15–6
- Системные блоки данных, 11–2
- Системы управления исполнением, 11–8
- Содержимое диагностического буфера, 15–13
- Сообщения, определенные пользователем, 15–3
- Сообщения CPU
 - величина архива, 15–3
 - обзор, 15–2
 - отображение, 15–3
 - последовательность действий, 15–4
 - M7–300/400, 17–22
- Сохранение
 - конфигурации, 7–20
 - на плате памяти, 18–5
 - содержимого ОЗУ CPU во встроенном СППЗУ, 18–4
- Список команд, 11–5
- Справочные данные, 12–2
 - генерация, 12–3
 - отображение, 12–4
 - применение, 12–2
- Средство загрузки, 17–5
- Средство переноса, 17–17
- Стандартная библиотека, 5–9
- Статусная строка, пример, 3–3
- Стек блоков (В–стек), 15–15

Стек локальных данных (L–стек), 15–15
Стек прерываний (U–стек), 15–15
Степень использования памяти, 15–9
Стирание
 блоков в CPU, 13–9
 платы памяти, 18–5
Строка комментариев, 14–5
Структура памяти CPU, 13–4
Структура программы
 в табличной форме, 12–6
 древовидная, 12–6
 отображение, 12–5
 построение, 12–5
Структура, централизованная, 7–5
Счетчик рабочего времени, 15–6

Т

Таблица глобальных данных, 9–3
Таблица переменных, 14–3
 использование, 14–3
 максимальный размер, 14–5
 отображение, 14–5
 проверка синтаксиса, 14–5
 редактирование, 14–4
 сохранение, 14–2
Таблица символов, 11–2
 импорт/экспорт, 6–7
 редактирование, 6–5
Текстовые редакторы, 11–4

У

Установка времени и даты, 15–6
Установки, сохранение, 12–4

Ф

Файл типов для DP–Slave, 7–24
Файлы исходных текстов
 на AWL, 11–7
 на S7–GRAPH, 11–8
 на SCL, 11–7
Форматирование целевой среды M7, 17–6
Функциональные единицы в S7–HiGraph, 11–9

Ц

Целевая система M7
 средство загрузки, 17–5
 целевая среда, 17–2, 17–4

Э

Этапы программирования
 M7, 4–3
 S7, 4–2

Я

Язык программирования, 11–9
Языки программирования, 11–4
 определение, 11–4
Языковые редакторы, 11–2
 запуск, 11–4

А

AUTHORS.EXE, 2–4
AWL, 11–5

В

В–стек, 15–15
BRCV, 10–4
Browser, 3–12
BSEND, 10–4

С

CFC–программа, 16–1

Д

DP–Master, 7–9
 выбор и размещение, 7–11
 параметризация, 7–18
DP–Slave, 7–9
 виды, 7–13
 включение в группы, 7–19
 включение в каталог модулей, 7–24
 выбор и размещение, 7–14
 модульный, 7–15
 назначение адресов входов/выходов, 7–15
 параметризация, 7–18
 семейство, 7–13
 файл типов, 7–24

Ф

FREEZE, 7–19

G

GD—связь. см. Связь с помощью глобальных данных
GD_RCV, 9–5
GD_SND, 9–5
GET, 10–4
GRAPH, 11–8

H

HiGraph, 11–9
HiGraph, исходный текст, 11–10

K

KOP, 11–5

L

L—стек, содержимое, 15–15
L2—адрес, 7–12, 7–14

M

M7–300/400
 модули, 16–6
 операционная система, 16–5
MPI—адрес, FM и CP, 8–10

N

NETPRO
 вызов, 8–7
 изображение сети, 8–8
 конфигурация сети, 8–6

O

Online—помощь
 вызов, 3–5
 темы, 3–5

P

PUT, 10–4

R

RESUME, 10–4

S

S7–GRAPH, 11–8
S7–HiGraph, 11–9
S7–SCL, 11–7
SCL, 11–7
SDB, 11–2
SFB 12 BSEND, 10–4
SFB 13 BRCV, 10–4
SFB 14 GET, 10–4
SFB 15 PUT, 10–4
SFB 19 START, 10–4
SFB 20 STOP, 10–4
SFB 21 RESUME, 10–4
SFB 22 STATUS, 10–4
SFB 23 USTATUS, 10–4
SFB 8 USEND, 10–4
SFB 9 URCV, 10–4
SFC 60 GD_SND, 9–5
SFC 61 GD_RCV, 9–5
SIMATIC Manager, 3–2
START, 10–4
STATUS, 10–4
STEP 7
 вызов обучающей программы, 3–5
 вызов пакета, 3–2
 инсталляция, 2–7
 ошибки при инсталляции, 2–9
 панели управления, 3–3
STOP, 10–4
SYNC, 7–19

U

U—стек, содержимое, 15–15
URCV, 10–4
USEND, 10–4
USTATUS, 10–4