

Глава 2

Программное обеспечение STEP 7

Содержание главы 2

2	<u>Программное обеспечение STEP 7</u>	4
2.1	<u>Базовый пакет STEP 7</u>	4
2.1.1	<u>Инсталляция</u>	4
2.1.2	<u>Авторизация</u>	5
2.1.3	<u>SIMATIC-менеджер</u>	5
2.1.4	<u>Проекты и библиотеки</u>	8
2.1.5	<u>Онлайновая помощь</u>	10
2.2	<u>Редактирование проектов</u>	11
2.2.1	<u>Создание проектов</u>	11
2.2.2	<u>Управление, перегруппировка, архивирование</u>	13
2.2.3	<u>Версии проектов</u>	15
2.3	<u>Конфигурирование станций</u>	17
2.3.1	<u>Компоновка модулей</u>	19
2.3.2	<u>Адресация модулей</u>	19
2.3.3	<u>Параметризация модулей</u>	20
2.3.4	<u>Объединение модулей в сеть с помощью MPI</u>	21
2.3.5	<u>Наблюдение и модифицирование модулей</u>	21
2.4	<u>Конфигурирование сети</u>	23
2.4.1	<u>Настройка просмотра сети</u>	25
2.4.2	<u>Конфигурирование системы DP-мастера в Network Configuration</u>	26
2.4.3	<u>Конфигурирование соединений</u>	27
2.4.4	<u>Сетевые переходы</u>	32
2.4.5	<u>Загрузка данных соединения</u>	33
2.5	<u>Создание программы S7</u>	35
2.5.1	<u>Введение</u>	35
2.5.2	<u>Таблица символов</u>	36
2.5.3	<u>Редактор программ</u>	38
2.5.4	<u>Обновление или генерирование исходных файлов</u>	41
2.5.5	<u>Приоритет адреса</u>	42
2.5.6	<u>Ссылочные данные</u>	43
2.5.7	<u>Мультиязыковые комментарии и отображение текста</u>	45
2.6	<u>Онлайновый (интерактивный) режим</u>	48
2.6.1	<u>Подключение PLC</u>	48
2.6.2	<u>Защита программы пользователя</u>	49
2.6.3	<u>Информация CPU</u>	50
2.6.4	<u>Загрузка пользовательской программы в CPU</u>	51
2.6.5	<u>Обработка блоков</u>	52
2.7	<u>Тестирование программы</u>	55
2.7.1	<u>Диагностирование аппаратных средств</u>	55
2.7.2	<u>Определение причины перехода в состояние STOP</u>	56
2.7.3	<u>Наблюдение и модифицирование переменных</u>	56
2.7.4	<u>Принудительная установка переменных (функция Force)</u>	58
2.7.5	<u>Разблокировка периферийных выходов</u>	61
2.7.6	<u>Статус программы LAD/FBD</u>	62

2 Программное обеспечение STEP 7

2.1 Базовый пакет STEP 7

Эта глава содержит описание базового программного пакета STEP 7 версии 5.1 (V5.1). Если первая глава представила обзор свойств программируемого контроллера, то в данной главе рассказывается о том, как устанавливать эти свойства.

В базовый пакет входят языки программирования списка операторов (statement list, STL), контактного плана (ladder logic, LAD) и функционального плана или диаграммы функциональных блоков (function block diagram, FBD). Кроме базового пакета также доступны такие пакеты, как S7-SCL (structured control language – структурированный язык управления), S7-GRAPH (sequence planning – последовательное планирование) и S7-HiGraph (state-transition diagram – диаграммы состояний-переходов).

2.1.1 Установка

STEP 7 V5 – это 32-битное приложение, требующее в качестве операционной системы Windows 95 с Service Pack 1 по крайней мере (версия 4.00.950a), Windows 98 или Windows NT с Service Pack 3 по крайней мере (версия 4.00.1381). Для работы с программным обеспечением STEP 7 под Windows 95/98 вам потребуется программирующее устройство (programming device, PG) или PC с процессором 80486 или выше с минимум 32 МБ RAM; рекомендуется процессор Pentium с 64 МБ. Для Windows NT вам потребуется процессор Pentium и минимум 32 МБ RAM; у вас должна быть авторизация администратора, чтобы установить STEP 7 под Windows NT.

Если вы работаете в STEP 7 над большими проектами, включающими, скажем, несколько станций автоматизации с более чем 100 модулями, вы должны использовать программирующее устройство или PC с современной мощностью обработки.

STEP 7 V5 занимает приблизительно от 200 до 300 МБ на жестком диске с использованием одного языка (например, английского) в зависимости от операционной и файловой систем компьютера. Также необходим файл подкачки (файл swap-out). Размер этого файла составляет примерно от 128 до 256 МБ минус сконфигурированный объем главной памяти.

Вы должны убедиться в том, что на диске, содержащем данные вашего проекта, достаточно памяти. Для некоторых операций, таких как копирование проекта, потребности в памяти могут увеличиться. Если для файла подкачки не хватает пространства, могут возникать такие ошибки, как сбои программы. Не рекомендуется хранить данные проекта на диске, содержащем файл подкачки Windows.

Для установки используется находящаяся на первой дискете программа установки SETUP для Windows 95/98/NT. На устройстве программирования STEP 7 уже установлен производителем.

Кроме STEP 7 V5 компакт-диск содержит также программу авторизации (смотрите ниже), программы NCM для конфигурирования устройств CP и электронные руководства по STEP 7 с Acrobat Reader V3.01.

Для онлайн-соединения требуется MPI-интерфейс. Устройства программирования уже оснащены встроенным многоточечным интерфейсом, но PC должны быть дооборудованы модулем MPI. Если вы хотите использовать карты памяти PC, вам потребуется проммер (prommer).

STEP 7 V5 имеет многопользовательские возможности, заключающиеся в том, что проект, который хранится, скажем, на центральном сервере, может редактироваться одновременно с нескольких станций. Необходимые установки производятся в Панели управления Windows (Windows Control Panel) с помощью программы «SIMATIC Workstation» («Рабочая станция SIMATIC»). В появляющемся диалоговом окне вы можете определить рабочую станцию как однопользовательскую систему или многопользовательскую систему с используемыми протоколами.

2.1.2 Авторизация

Для работы со STEP 7 требуется авторизация (право пользования). Авторизация поставляется на дискете. При выполнении инсталляции STEP 7 вам будет предложено установить вашу авторизацию, если жесткий диск уже не содержит авторизацию. Авторизацию можно поставить позже.

Вы также можете перенести авторизацию на другое устройство путем обратного копирования на (исходную) авторизационную дискету и последующего переноса ее на новое устройство.

Если вы утратили вашу авторизацию по какой-либо причине, например, из-за дефекта жесткого диска, то вы можете воспользоваться экстренной лицензией, также имеющейся на авторизационной дискете. Данная лицензия ограничена по времени действия до получения новой авторизации.

2.1.3 SIMATIC-менеджер

Утилита SIMATIC-менеджер (SIMATIC Manager) является главным инструментом в STEP 7; его пиктограмму (иконку) вы найдете на рабочем столе Windows или в меню Пуск (Start).



SIMATIC Manager

SIMATIC-менеджер запускается двойным щелчком левой клавишей мыши на его иконке.

При первом запуске появляется окно Мастера проектов (Project Wizard). Он может быть использован для упрощенного создания новых проектов. Вы можете деактивировать его при помощи переключателя (флажка, check box) «Display Wizard on starting the SIMATIC Manager» («При запуске SIMATIC-менеджера Отобразить Мастера»), так как при необходимости его можно вызвать по команде меню File → New Project Wizard (Файл → Мастер новых проектов).

Программирование начинается с открытия или создания проекта (project). Предоставляемые примеры проектов являются хорошей основой для ознакомления.

Когда вы откроете пример проекта, выбрав пункт меню File → Open (Файл → Открыть), вы увидите разделенное окно проекта: в левой части расположена структура открытого проекта (иерархия объектов), а справа отображается выделенный объект (рисунок 2.1). Щелчок на квадратике со знаком «плюс» в левом окне приведет к отображению дополнительных уровней структуры; выбор объекта в левой половине окна отобразит его содержимое в правой половине окна.

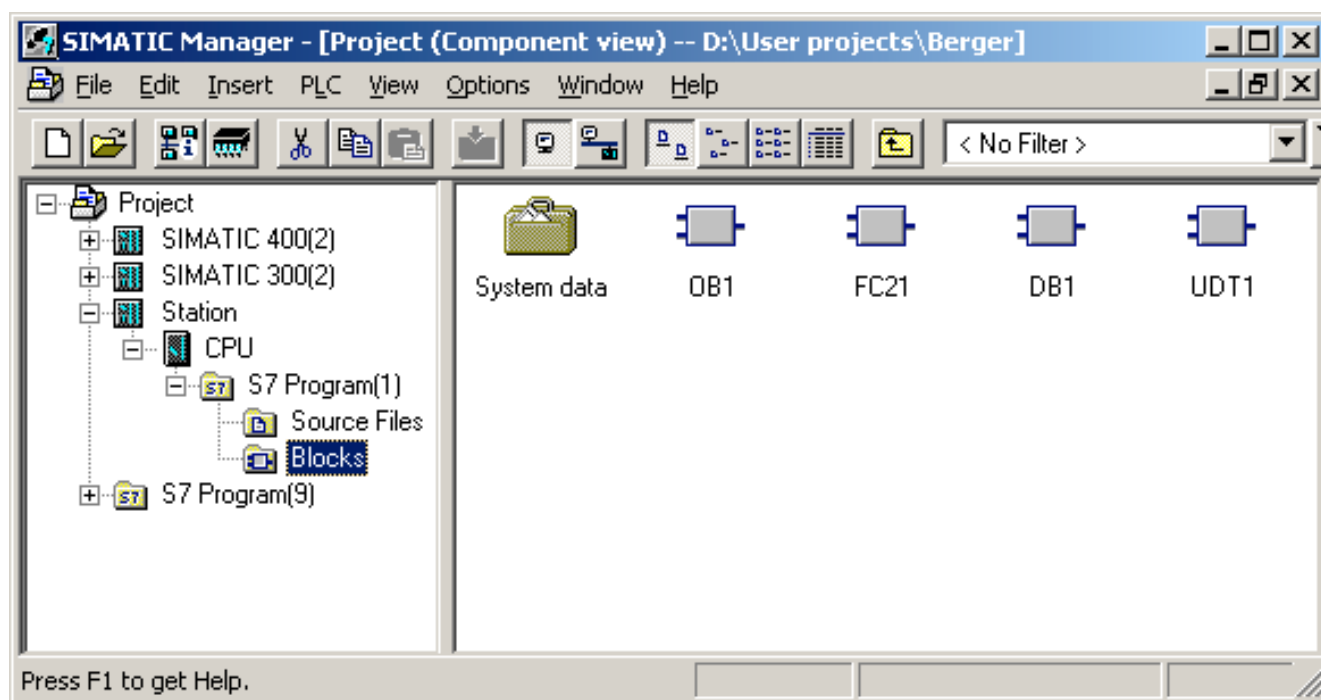


Рисунок 2.1 Пример окна SIMATIC-менеджера

Ваша работа с объектами мира STEP 7 происходит под управлением SIMATIC-менеджера. Данные «логические» объекты соответствуют «реальным» объектам вашего предприятия. Проект содержит предприятие в целом, станция соответствует программируемому контроллеру. Проект может содержать несколько станций, соединенных между собой, к примеру, через подсеть MPI. Станция содержит CPU, а CPU хранит программу, в вашем случае S7-программу. Эта программа, в свою очередь, является «контейнером» («container») для других объектов, таких как Blocks (Блоки), которые содержат, помимо других составляющих, скомпилированные блоки.

Объекты STEP 7 объединены посредством древовидной структуры. На рисунке 2.2 показана самая важная часть древовидной структуры (или «главная ветвь»), когда вы работаете с базовым пакетом STEP 7 для S7-приложений в режиме автономного просмотра. Выделенные (слева) объекты являются контейнерами для других объектов. Все объекты на рисунке доступны для автономного просмотра. Это те объекты, которые находятся на жестком диске устройства программирования. Если ваше программирующее устройство подключено к CPU (обычно целевая система PLC), вы можете переключиться в режим онлайн-просмотра выбором опции меню View → Online (Вид → Онлайн). Кроме того, эта опция отображает окно другого проекта, содержащее объекты удаленного устройства; объекты, выделенные на рисунке, более не отображаются.

Структура	Объект	Описание
Проект	Контейнер для всех объектов проекта	
MPI [PTP, PROFIBUS, Ethernet]	Подсеть	Содержит установки сетевых параметров для подсети
Станция SIMATIC 300/400	Контейнер для всех объектов станции	
Аппаратура	Конфигурационная таблица	Содержит конфигурационные данные для станции и параметры для модулей
CPU xxx	Контейнер для всех объектов CPU	
Соединения	Таблица соединений	Содержит определения коммуникационных соединений между узлами сети
Программа S7	Контейнер для всех объектов программы пользователя	
Символы	Таблица символов	Содержит назначения символов (= имена) абсолютным адресам глобальных данных
Исходники	Контейнер для всех объектов программных исходников	
Исходные файлы	Исходные программы	Содержит исходники для пользовательских программ (например, STL- и SCL-программы)
Блоки	Контейнер для всех скомпилированных объектов программы пользователя	
OB n	Организационные блоки	Содержат откомпилированный код и данные пользовательской программы
FB n	Функциональные блоки	
FC n	Функции	
DB n	Блоки данных	
SFC n	Системные функции	Содержат интерфейс вызовов для системных блоков, интегрированных в CPU
SFB n	Системные функцион. блоки	
Системные данные	Блоки системных данных	Содержат скомпилированные данные для конфигурационной таблицы
UDT n	Типы данных	Содержат определения пользовательских типов данных
VAT n	Таблицы переменных	Содержат переменные для наблюдения (мониторинга) и модифицирования
Программа S7	Контейнер для программы пользователя, не назначенной ни одному CPU (имеет ту же структуру, что и любая S7-программа, назначенная CPU)	

Рисунок 2.2 Иерархия объектов проекта STEP 7

В заголовке окна активного проекта вы можете увидеть, в каком режиме вы работаете – онлайнном или офлайнном (автономном). Для более ясного различия прямоугольник с заголовком и заголовок окна могут быть окрашены в другой цвет, нежели автономное окно. Для этой цели выберите Options → Customize (Опции → Настроить) и измените установки на вкладке «View» («Просмотр»).

Выберите Options → Customize (Опции → Настроить) для того, чтобы изменить основные установки SIMATIC-менеджера, такие как язык сессии (сеанса), программу архивирования и место хранения проектов и библиотек (libraries), а также, чтобы сконфигурировать программу архивирования.

Этапы редактирования

Следующее относится к общим понятиям редактирования объектов:

Выбрать (отметить) объект означает щелкнуть на нем один раз мышью, так чтобы он стал выделенным, подсвеченным (это возможно в обеих частях окна проекта).

Назвать объект означает щелкнуть на имени выбранного объекта (вокруг имени появится рамка, и вы сможете изменить имя в окне) или выбрать пункт меню Edit → Object Properties (Правка → Свойства объекта) и в диалоговом окне изменить имя. В случае некоторых объектов, таких как CPU, вы можете сменить имя только с помощью соответствующего инструмента (приложения), в данном случае – с помощью Hardware Configuration (Конфигурирование аппаратуры).

Чтобы *открыть объект*, дважды щелкните на этом объекте. Если объект является контейнером для других объектов, SIMATIC-менеджер отобразит содержимое объекта в правой половине окна. Если объект расположен на самом нижнем иерархическом уровне, SIMATIC-менеджер запускает соответствующий инструмент для редактирования объекта (например, двойной щелчок на блоке запускает редактор, позволяющий модифицировать блок).

В настоящей книге пункты меню в стандартной линейке меню в верхней части окна описываются как *последовательности действий оператора*. Программисты, имеющие опыт в использовании операторского интерфейса, используют иконки панели инструментов (toolbar). Очень эффективным является использование *правой клавиши мыши*. Единичный щелчок правой клавишей мыши на объекте отобразит меню с текущими опциями редактирования.

2.1.4 Проекты и библиотеки

В STEP 7 главные объекты (main objects) в верхней части иерархии объектов – это проекты и библиотеки.

Проекты (Projects) используются для систематизированного хранения данных и программ, требующихся для решения задачи автоматизации. По существу это

- конфигурационные данные аппаратного обеспечения,
- данные параметризации модулей,
- конфигурационные данные сетевых коммуникаций,
- программы (код и данные, символы, исходники).

Объекты в проекте упорядочены иерархически. Открытие проекта является первым шагом по редактированию всех (подчиненных) объектов, которые этот объект содержит. Следующие разделы посвящены тому, как редактировать эти объекты.

Библиотеки (Libraries) применяются для хранения повторно используемых программных компонентов. Библиотеки организованы иерархически. Они могут содержать программы STEP 7, которые, в свою очередь, могут содержать пользовательскую программу (контейнер для откомпилированных блоков), контейнер для исходных программ и таблицу символов (symbol table). За исключением онлайн-соединений (отладка невозможна) создание программы или раздела программы в библиотеке предоставляет ту же функциональность, что и в проекте.

Поставляемый пакет STEP 7 V5 предоставляет Стандартную Библиотеку (Standard Library), содержащую следующие программы:

- System Function Blocks (системные функциональные блоки)
Содержит интерфейсы вызовов системных блоков, встроенных в CPU, для автономного программирования;
- S5-S7 Converting Blocks (конвертирование блоков S5-S7)
Содержит загружаемые функции для конвертера S5/S7 (замена стандартных функциональных блоков S5 в связи с переходом на другую версию программы);
- TI-S7 Converting Blocks (конвертирование блоков TI-S7)
Содержит дополнительные загружаемые функции и функциональные блоки для конвертера TI-S7;
- IEC Function Blocks (функциональные блоки IEC)
Содержит загружаемые функции для редактирования переменных сложных типов данных DATE_AND_TIME и STRING;
- Communication Blocks (коммуникационные блоки)
Содержит загружаемые функции для управления CP 342-5 (CP-модулями);
- PID Control Blocks (блоки PID-управления)
Содержит загружаемые функциональные блоки для замкнутого управления;
- Organization Blocks (организационные блоки)
Содержит шаблоны (templates) организационных блоков (по существу описание переменных для стартовой информации).

Обзор содержимого этих библиотек вы можете найти в главе 25 «Библиотеки блоков». Например, вы намерены использовать модуль S7 со стандартными блоками, соответствующая инсталляционная программа устанавливает на жесткий диск стандартные блоки как библиотеку. Затем вы можете скопировать эти блоки из библиотеки в ваш проект. Библиотека открывается по команде File → Open (Файл → Открыть) и может редактироваться также как проект. Также вы можете создавать ваши собственные библиотеки.

Пункт меню File → New (Файл → Новый) генерирует новый объект на вершине иерархии объектов (проект, библиотеку). Месторасположение в структуре директории, где SIMATIC-менеджер должен создать проект или библиотеку, должно быть определено в пункте меню Options → Customize (Опции → Настроить) или в диалоговом окне «New» («Новый»).

Меню Insert (Вставка) используется для добавления новых объектов к существующим (например, добавление нового блока к программе). Однако, перед выполнением этих действий вы сначала должны в левой части окна SIMATIC-менеджера выбрать контейнер объектов, в который вы хотите вставить новый объект.

Копирование контейнеров объектов и самих объектов выполняется с помощью опций меню Edit → Copy (Правка → Копировать) и Edit → Paste (Правка → Вставить) или, как обычно в Windows, перетаскиванием выбранных объектов мышью из одного окна в другое. Обратите внимание на то, что в SIMATIC-менеджере нельзя отменить (undo) удаление объекта или контейнера объектов.

2.1.5 Онлайновая помощь

Онлайновая (интерактивная) помощь предоставляет информацию, необходимую вам во время вашего сеанса программирования, не прибегая к обращению к руководствам. Вы можете выбрать тему (topic), по которой требуются данные, в пункте меню Help (Помощь). К примеру, опция онлайновой помощи GETTING STARTED (Начало) предоставляет краткое резюме о том, как работать в SIMATIC-менеджере.

Help → Contents (Помощь → Содержание) запускает центральную функцию Помощи STEP 7 из любого приложения. Здесь содержится вся основная информация.

Help → Context-Sensitive Help F1 (Помощь → Контекстно-зависимая помощь F1) предоставляет контекстно-зависимую помощь, то есть если вы нажмете F1, то получите информацию, касающуюся отмеченного мышью объекта или текущего сообщения об ошибке.

В линейке символов имеется кнопка со стрелкой и знаком вопроса. Если вы нажмете эту кнопку, то к указателю мыши добавится знак вопроса. Теперь с помощью этого информационного указателя мыши вы можете выбрать объект на экране, к примеру, символ или команда меню, щелкнув на нем, и получить соответствующую онлайн-подсказку.

2.2 Редактирование проектов

При настройке проекта вы создаете «контейнеры» для результирующих данных, затем вы генерируете данные и заполняете эти контейнеры. Обычно вы создаете проект, используя соответствующие аппаратные средства, конфигурируете аппаратуру или, по крайней мере, CPU и в ответ получаете контейнеры для пользовательской программы. Кроме того, вы можете поместить S7-программу непосредственно в контейнер проекта без применения какого-либо оборудования. Заметьте, что инициализация модулей (изменение адресов, установки CPU, конфигурирование соединений) возможно только с помощью инструментария Hardware Configuration.

Мы настоятельно рекомендуем, чтобы весь процесс редактирования проекта выполнялся с использованием SIMATIC-менеджера. Создание, копирование или удаление директорий или файлов, а также изменение имен (!) с помощью Windows Explorer (Проводника Windows) внутри структуры проекта могут вызвать проблемы при работе с SIMATIC-менеджером.

2.2.1 Создание проектов

Мастер проектов (Project wizard)

Начиная со STEP 7 V3.2, для оказания помощи в создании новых проектов применяется *STEP 7 Wizard (Мастер STEP 7)*. Вы определяете CPU, а Мастер создает за вас проект со станцией S7 и выбранным CPU, а также контейнер S7-программы, контейнер исходников и контейнер блоков с выбранными организационными блоками.

Создание проекта со станцией S7

Если вы хотите создать проект «вручную», то описание необходимых действий с вашей стороны вы найдете в этом разделе. Общая информация о командах оператора по редактированию объектов содержится в параграфе 2.1.3 «SIMATIC-менеджер».

Создание нового проекта

Выберите пункт меню **File → New (Файл → Новый)**, в диалоговом окне введите имя, при необходимости измените тип и место хранения и подтвердите нажатием кнопки «ОК» или клавиши RETURN.

Вставка в проект новой станции

Выберите проект и вставьте станцию с помощью **Insert → Station → SIMATIC 300 Station (Вставка → Станция → Станция SIMATIC 300)** (в случае S7-300).

Конфигурирование станции

В левой части окна проектов щелкните мышью на квадратике с плюсом, следующим за проектом, и выберите станцию; SIMATIC-менеджер отобразит объект аппаратных средств в правой части окна. Двойной щелчок на *Hardware (Apparatur)* запустит инструмент Hardware Configuration, с помощью которого вы можете отредактировать конфигурационные таблицы (configuration table). Если каталог модулей не выведен на экран, вызовите его командой меню View → Catalog (Вид → Каталог).

Конфигурирование начинается с выбора направляющей рейки с помощью мыши, например, под «SIMATIC 300» и «RACK 300», «удерживая», перетащите ее на свободное пространство верхней части окна станции и «отпустите» (операция drag & drop). После этого вы увидите таблицу с представлением слотов на направляющей рейке.

Затем из каталога модулей выберите требуемые модули и, используя описанную выше процедуру, перетащите и поместите их в соответствующие слоты. Для дальнейшего редактирования структуры проекта для станции требуется, по крайней мере, один CPU, например, CPU 314 в слоте 2. Все остальные модули вы можете добавить позднее. Редактирование аппаратного обеспечения подробно обсуждается в параграфе 2.3 «Конфигурирование станций».

Сохраните и откомпилируйте станцию, затем закройте и вернитесь в SIMATIC-менеджер. В дополнение к конфигурации аппаратных средств теперь открытая станция показывает также CPU.

При конфигурировании CPU SIMATIC-менеджер также создает папку S7 Program (S7-программы) со всеми объектами. Создание структуры проекта теперь завершено.

Просмотр содержимого папки S7 Program

Откройте CPU; в правой части окна проекта вы увидите значки S7-программы и таблицы соединений (connection table) – «S7 Program» и «Connection».

Откройте папку S7-программы; SIMATIC-менеджер в правой половине окна отобразит значки откомпилированной пользовательской программы (Blocks – Блоки), контейнер исходных программ и таблицу символов.

Откройте пользовательскую программу (Blocks); SIMATIC-менеджер в правой половине окна покажет значки откомпилированных конфигурационных данных (System data – Системные данные) и пустой организационный блок для главной программы (OB 1).

Редактирование объектов пользовательской программы

На данный момент мы достигли нижнего уровня иерархии объектов. OB 1 открыт в первый раз, отображается окно со свойствами объектов, и открыт редактор, необходимый для редактирования программы в организационном блоке. Добавьте другой

пустой блок для пошагового редактирования командой меню Insert → S7 Block → ... (Вставка → Блок S7) (*Blocks – Блоки* – должен быть подсвечен) и выберите из предоставляемого списка нужный тип блока.

Объект *System data* (*Системные данные*) в раскрытом виде показывает список доступных системных блоков данных. Вы получаете откомпилированные конфигурационные данные. Эти системные блоки данных редактируются с помощью объекта *Hardware* (*Аппаратура*) в контейнере *Station* (*Станция*). Вы можете переместить *System data* (*Системные данные*) в CPU, выбрав опцию меню PLC → Download (PLC → Загрузить), и параметризовать таким образом CPU.

Контейнер объектов *Source Files* (*Исходные файлы*) пуст. Предварительно отметив контейнер *Source Files* (*Исходные файлы*), выберите команду Insert → S7 Software → STL Source File (Вставка → Программное обеспечение S7 → Исходные файлы STL), чтобы вставить пустой исходный текстовый файл. Или вы можете выбрать Insert → External Source File (Вставка → Внешний исходный файл), чтобы переместить исходный текстовый файл, созданный, скажем, в другом редакторе в формате ASCII, в контейнер *Source Files* (*Исходные файлы*).

Создание проекта без станции S7

При желании вы можете сформировать программу без необходимости сначала настроить станцию. Для этого сами сгенерируйте контейнер для вашей программы. Выберите проект и сгенерируйте папку S7-программы путем Insert → Program → S7 Program (Вставка → Программа → S7-программа). Под этой папкой S7-программы SIMATIC-менеджер создаст контейнеры объектов *Sources* (*Исходники*) и *Blocks* (*Блоки*). *Blocks* (*Блоки*) содержит пустой OB 1.

Создание библиотеки

Программу вы также можете создать в виде библиотеки, если вы хотите, к примеру, использовать ее более одного раза. И таким образом, стандартная программа будет всегда доступна, ее можно скопировать полностью или частично в текущую программу.

Заметьте, что в библиотеке вы не можете устанавливать онлайнное соединение, а это значит, что вы можете отлаживать программу STEP 7 только в рамках проекта.

2.2.2 Управление, перегруппировка, архивирование

Утилита SIMATIC-менеджер осуществляет поддержку списка всех известных «главных объектов»: пользовательских проектов, библиотек и примеров проектов. Вы устанавливаете примеры проектов вместе со STEP 7, а пользовательские проекты и ваши собственные библиотеки вы устанавливаете самостоятельно.

При выполнении команды меню File → Rearrange (Файл → Перегруппировка) SIMATIC-менеджер покажет вам список имен и путей всех известных проектов и библиотек. Вы можете удалить из списка проекты или библиотеки, которые вы не хотите отображать («скрыть»), или включить в список новые проекты и библиотеки («показать»).

Когда исполняется File → Rearrange (Файл → Перегруппировка), SIMATIC-менеджер устраняет пробелы, образовавшиеся от удалений, и оптимизирует память данных подобно тому, как программа дефрагментации оптимизирует память данных на жестком диске. Реорганизация может занять некоторое время в зависимости от выполняемого перемещения данных.

Также вы можете заархивировать проект или библиотеку (File → Archive, Файл → Архивировать). В этом случае SIMATIC-менеджер сохраняет выбранный объект (директорию проекта или библиотеки со всеми поддиректориями и файлами) в сжатом виде в архивном файле.

Для архивирования проекта или библиотеки вам нужна программа архивации. STEP 7 содержит архиваторы ARJ и PKZIP 2.50, но вы можете использовать также другие программы-архиваторы (WinZip с версии 6.0, Pkzip с версии 2.04g, JAR с версии 1.02 или LHARC с версии 2.13).

Проекты и библиотеки не могут редактироваться в архивном (сжатом) состоянии. Вы можете распаковать заархивированный объект путем File → Retrieve (Файл → Восстановить) и затем продолжить редактирование. Восстановленные объекты принимаются системой управления проектом или библиотеки автоматически.

Установки для архивирования и восстановления вы можете сделать во вкладке «Archive» («Архив»), выбрав предварительно команду Options → Customize (Опции → Настроить); например, имеется установка целевой директории для архивирования и восстановления или «Generate archive path automatically» («Генерировать путь архива автоматически») (тогда при архивировании не требуется дополнительных определений, потому что имя архивного файла генерируется из имени проекта).

Архивирование проекта в CPU

STEP 7, начиная с версии 5.1, с соответствующим образом построенными CPU S7-400 позволяет вам хранить проект в архивном (сжатом) виде в загрузочной памяти CPU, то есть на карте памяти. Таким образом, вы можете сохранить все данные проекта, требуемые для полного выполнения пользовательской программы, такие как символы или исходные файлы, непосредственно в объекте-машине или объекте-предприятии. При необходимости изменить или дополнить программу вы можете загрузить хранимые локально данные на жесткий диск, скорректировать пользовательскую программу и снова сохранить в CPU обновленные проектные данные.

При загрузке данных проекта на карту памяти, подключенную к CPU, откройте проект, отметьте CPU и выберите PLC → Save Project on Memory Card (PLC → Записать объект на карту памяти). В обратном направлении: переместите сохраненные данные обратно на жесткий диск с помощью PLC → Retrieve Project from Memory Card (PLC

→ Получить объект из карты памяти). Обратите внимание на то, что когда вы записываете на карту памяти, подключенную к CPU, все содержимое загрузочной памяти записывается в CPU, включая системные данные и программы пользователя.

Если вы хотите осуществить обратную выборку проектных данных, сохраненных в CPU, без создания проекта на жестком диске, выберите соответствующий CPU с помощью опции меню PLC → Display Accessible Nodes (PLC → Отобразить доступные узлы). Если карта памяти вставлена в разъем модуля программирующего устройства, то перед передачей выберите карту памяти при помощи команды File → S7 Memory Card → Open (Файл → Карта памяти S7 → Открыть).

2.2.3 Версии проектов

С момента выхода STEP 7 V5 различных версий SIMATIC-проектов стало три. STEP 7 V1 создает проекты версии 1, STEP 7 V2 создает проекты версии 2, и STEP 7 V3/V4/V5.0 может быть использован для создания и редактирования проектов версий 2 и 3. С помощью STEP 7 V5.1 вы можете сформировать и отредактировать V3-проекты и V3-библиотеки.

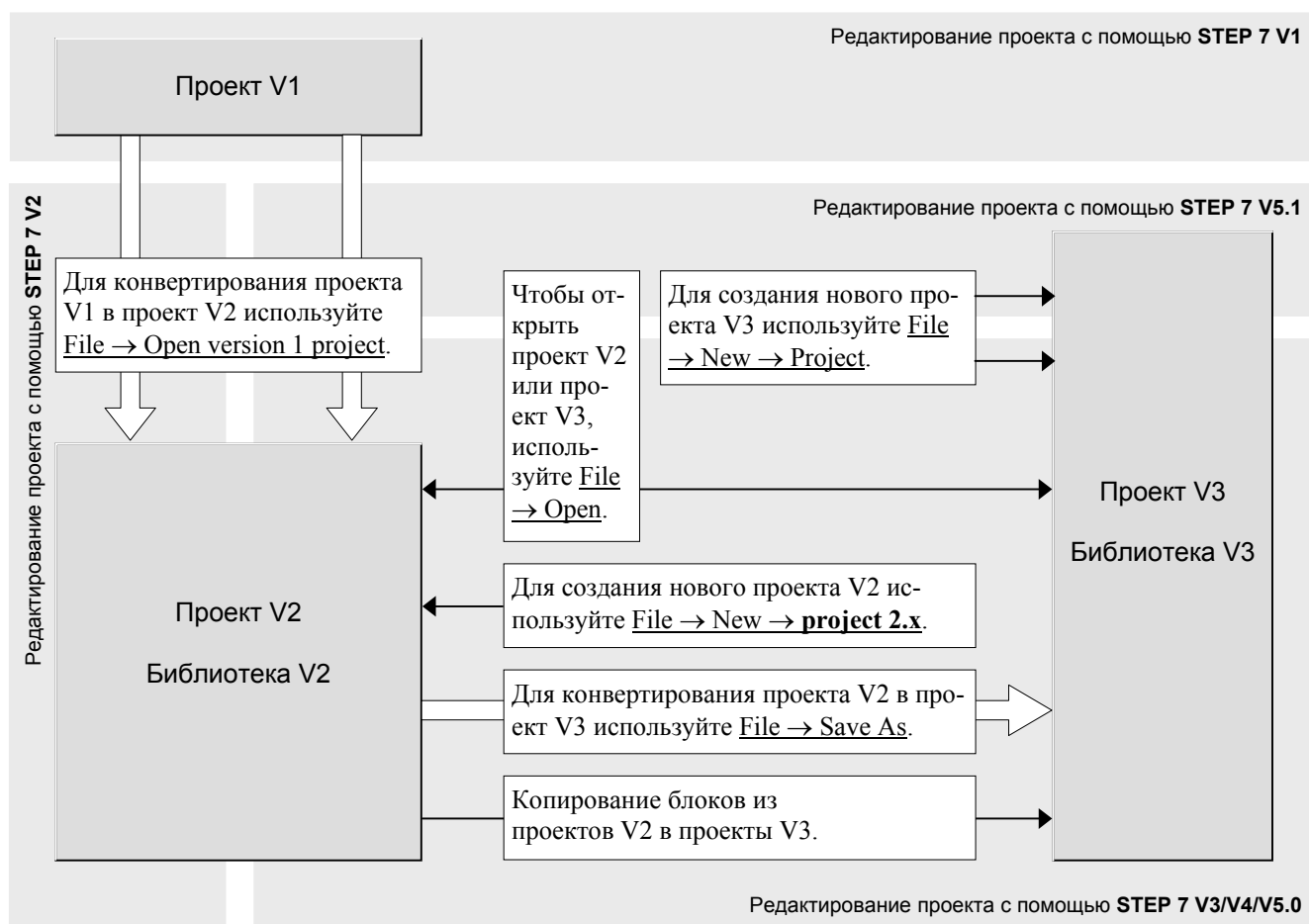


Рисунок 2.3 Редактирование проектов различных версий

Если у вас есть проект версии 1, вы можете преобразовать его в проект версии 2 с помощью File → Open Version 1 Project (Файл → Открыть проект версии 1). Структура проекта с программами, откомпилированными блоками версии 1, исходными STL-программами, таблицей символов и конфигурацией аппаратных средств остаются неизменными.

Вы можете создавать и редактировать проекты версии 2 при помощи STEP 7 версий V2, V3, V4 и V5.0 (рисунок 2.3).

STEP 7 V5.1 работает только с проектами версии 3. Однако, вы можете конвертировать V1-проект в V2-проект с помощью File → Open Version 1 Project (Файл → Открыть проект версии 1), а также открыть V2-проект, выбрав File → Open (Файл → Открыть). Создать V2-проект или записать проект в формате версии 2 невозможно.

2.3 Конфигурирование станций

Для создания конфигурации программируемого контроллера используется инструмент Hardware Configuration. Конфигурирование выполняется автономно без подключения к CPU. Также вы можете использовать этот инструмент для адресации и параметризации модулей. Вы можете создать конфигурацию аппаратного обеспечения на стадии планирования или подождать до установки аппаратуры.

Конфигурирование аппаратного обеспечения начинается с выбора станции, затем следует обратиться к меню Edit → Open Object (Правка → Открыть объект) или дважды щелкнуть мышью на объекте Hardware в открытом контейнере SIMATIC 300/400 Station (станция SIMATIC 300/400). Основные установки производятся с помощью Options → Customize (Опции → Настроить).

Когда конфигурирование окончено, Station → Consistency Check (Станция → Проверка согласованности) проверит введенную вами информацию на наличие ошибок. Команда Station → Save (Станция → Сохранить) записывает на жесткий диск конфигурационные таблицы со всеми данными значений параметров в вашем проекте.

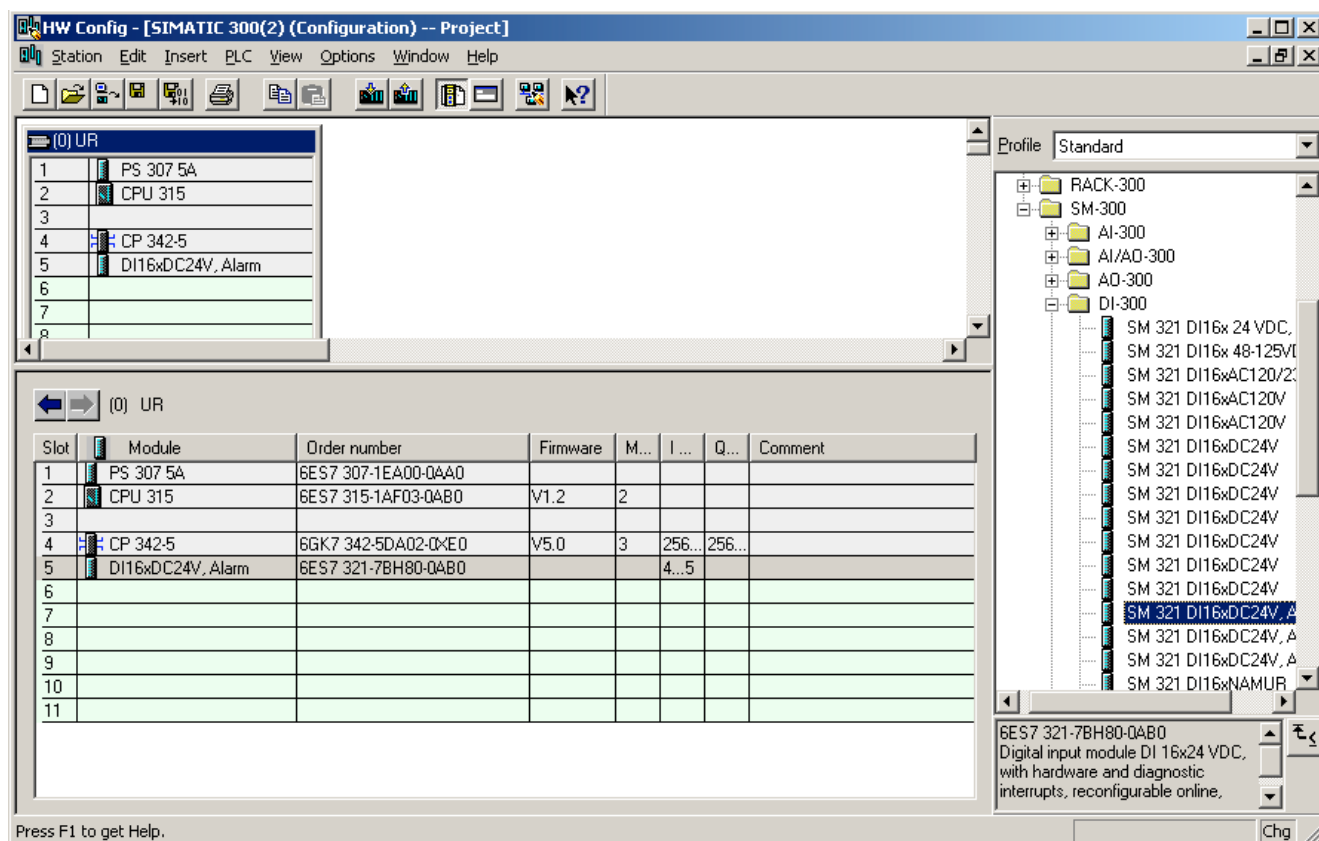


Рисунок 2.4 Пример окна станции в Hardware Configuration

Команда Station → Save and Compile (Станция → Сохранить и скомпилировать) не только сохраняет, но и компилирует конфигурационные таблицы и записывает скомпилированные данные в объект *System data* (Системные данные) в автономный контейнер *Blocks* (Блоки). После компиляции вы можете переместить конфигурационные данные в CPU с помощью PLC → Download (PLC → Загрузить). Объект *System data* (Системные данные) в онлайн-контейнере *Blocks* (Блоки) представляет текущие конфигурационные данные в CPU. Вы можете «вернуть» эти данные на жесткий диск с помощью PLC → Upload (PLC → Выгрузить).

Экспортирование данных конфигурации аппаратного обеспечения осуществляется пунктом меню Station → Export (Станция → Экспорт). STEP 7 затем создает файл в формате ASCII, содержащий конфигурационные данные и данные параметризации модулей. Вы можете выбрать между текстовым форматом с данными в виде «читаемых» английских букв и компактным форматом с шестнадцатеричными данными. Предусмотрена возможность импорта соответствующим образом структурированного ASCII-файла.

Контрольная сумма

Hardware Configuration генерирует контрольную сумму (checksum) корректно скомпилированной станции и сохраняет ее в системных данных. Идентичные системные конфигурации имеют такую же контрольную сумму, так что вы можете, к примеру, легко сравнить онлайн-ую и автономную конфигурации.

Контрольная сумма является свойством объекта *System data* (Системные данные). Чтобы считать контрольную сумму, откройте в S7-программе контейнер *Blocks* (Блоки), выберите объект *System data* (Системные данные) и откройте его с помощью Edit → Open Object (Правка → Открыть объект). Пользовательская программа также имеет соответствующую контрольную сумму. Она находится вместе с контрольной суммой системных данных в свойствах *Blocks* (Блоки): выберите контейнер *Blocks* (Блоки) и затем Edit → Object Properties (Правка → Свойства объектов) на вкладке «Checksums» («Контрольные суммы»).

Окно станции

При работе Hardware Configuration отображает окно станции и каталог аппаратного обеспечения (рисунок 2.4). Увеличьте или максимизируйте окно станции для облегчения редактирования. В верхнем разделе окна показаны монтажные стойки в форме таблиц и DP-станции в виде символов. Если имеются несколько стоек, то вы видите здесь соединения между интерфейсными модулями, и, если используется PROFIBUS, вы видите конфигурацию системы DP-мастера. Нижняя часть окна станции отображает конфигурационную таблицу, в подробностях показывающую стойку или DP-ведомого, отмеченных в верхнем разделе.

Каталог аппаратного обеспечения

Каталог оборудования вы можете вызывать и закрывать с помощью View → Catalog (Вид → Каталог). Он содержит все монтажные стойки, модули и интерфейсные подмодули, известные STEP 7. Выбрав Options → Edit Catalog Profile (Опции → Редактировать профиль каталога), вы можете откомпилировать ваш собственный каталог аппаратных средств, который показывает только те модули, с которыми вы хотите работать, - в выбранной вами структуре. Двойным щелчком клавишей мыши на линейке заголовка вы можете «пристыковать» каталог оборудования к правой границе окна станции или снова его отпустить.

Конфигурационная таблица

Инструмент Hardware Configuration работает с таблицами, каждая из которых представляет монтажную стойку, модуль или DP-станцию. Конфигурационная таблица показывает слоты с расположенными в них модулями или свойства модулей, такие как их адреса и порядковые номера. Двойной щелчок на строке модуля открывает окно свойств модуля и позволяет задать параметры модуля.

2.3.1 Компоновка модулей

Конфигурирование начинается с того, что вы выбираете направляющую рейку из каталога модулей, например, под «SIMATIC 300» или «RACK 300», «удерживая» ее, перетаскиваете мышью в верхнюю часть окна станции и помещаете где-нибудь в этом окне (drag&drop). Отображается пустая конфигурационная таблица для центральной стойки. Затем из каталога модулей выберите требуемые модули и описанным выше способом поместите их в соответствующие слоты. Символ «No Parking» («Парковки нет») информирует вас о невозможности поместить выбранный модуль в намеченный слот.

В случае однорядной конфигурации станции S7-300 слот 3 остается пустым; он зарезервирован для интерфейсного модуля стойки расширения.

Вы можете сгенерировать конфигурационную таблицу для другой стойки путем перетаскивания выбранной стойки из каталога и помещения ее в окно станции. В системах S7-400 неподключенной стойке назначается интерфейс (или точнее: соответствующие стойки получают интерфейсный модуль) на вкладке «Link» («Связь») окна «Properties» («Свойства») модуля Send IM (выберите модуль и нажмите Edit → Object Properties, Правка → Свойства объекта).

Компоновка распределенных I/O-станций описывается в параграфе 20.4.2 «Конфигурирование распределенных входов/выходов».

2.3.2 Адресация модулей

При компоновке модулей инструмент Hardware Configuration автоматически назначает модулям стартовый (начальный) адрес. Вы можете увидеть этот адрес в нижней

части окна станции в свойствах объектов соответствующих модулей. В случае CPU S7-400 и CPU S7-300 со встроенным DP-интерфейсом вы можете изменять адреса модулей. Выполняя это, соблюдайте правила адресации для систем S7-300 и S7-400, а также адресную емкость отдельных модулей.

Существуют модули, имеющие и входы и выходы, для которых вы можете (теоретически) зарезервировать различные стартовые адреса. Тем не менее, внимательно изучайте специальную информацию в руководствах по продукту; для большинства функциональных и коммуникационных модулей требуются одинаковые адреса для входов и выходов.

Когда вы назначаете стартовый адрес модулей в S7-400, вы также можете инициализировать вспомогательный (subsidiary) образ процесса (каждому модулю назначить частичный образ процесса – Part process image). Если в центральной стойке расположено более одного CPU, то автоматически устанавливается мультипроцессорный режим, и вы должны назначить модуль центральному процессору.

С помощью View → Address Overview (Вид → Обзор адресов) вызывается окно, содержащее все адреса модулей, которые используются в настоящий момент выбранным CPU.

Модули на MPI-шине или коммуникационной шине имеют MPI-адрес. Вы также можете изменить эти адреса. Однако, имейте в виду, что новый MPI-адрес становится действующим после передачи в CPU конфигурационных данных.

Символы для адресов пользовательских данных

В Hardware Configuration вы можете назначить входам и выходам символы (имена), которые передаются в таблицу символов.

После того, как вы скомпоновали и адресовали цифровые и аналоговые модули, сохраните данные станции. Затем выберите модуль (строку) и нажмите Edit → Symbols (Правка → Символы). В окне, которое затем откроется, вы можете назначить символ, тип данных и комментарий абсолютному адресу для каждого канала (побитно для цифровых модулей и каждому слову для аналоговых модулей).

Кнопка «Add Symbol» («Добавить символ») вводит абсолютные адреса как символы вместо абсолютных адресов без символов. Кнопка «Apply» («Применить») передает символы в таблицу символов. «OK» также закрывает диалоговое окно.

2.3.3 Параметризация модулей

При параметризации модулей вы определяете их свойства. Параметризовать модули необходимо только тогда, когда вы хотите изменить параметры по умолчанию. Обязательным условием параметризации является расположение модуля.

Дважды щелкните в конфигурационной таблице на модуле или выделите модуль и нажмите Edit → Object Properties (Правка → Свойства объекта). В диалоговом окне

откроются несколько таблиц с задаваемыми параметрами для этого модуля. Когда вы используете этот метод для параметризации CPU, вы определяете характеристики исполнения вашей (пользовательской) программы.

Некоторые модули позволяют вам устанавливать их параметры во время исполнения через пользовательскую программу с помощью системных функций SFC 55 WR_PARM, SFC 56 WR_DPARM и SFC 57 PARM_MOD.

2.3.4 Объединение модулей в сеть с помощью MPI

Для MPI (подсеть) узлы определяются дополнительно с помощью свойств модуля (Module Properties). В конфигурационной таблице выберите CPU или карту MPI-интерфейса, если CPU оснащен таковой, и откройте выбранный объект, нажав Edit → Object Properties (Правка → Свойства объекта). В появляющемся затем диалоговом окне в секции «Interface» («Интерфейс») вкладки «General» («Общие») есть кнопка «Properties» («Свойства»). Если вы нажмете на эту кнопку, вы переместитесь в другое диалоговое окно с вкладкой «Parameter» («Параметр»), где сможете найти подходящую подсеть.

Здесь также есть возможность установить MPI-адрес, предусмотренный вами для данного CPU. Заметьте, что в более старших CPU S7-300 модули FM или CP с MPI-соединением автоматически получают MPI-адрес от CPU.

Наибольший MPI-адрес должен быть больше или равен наибольшему MPI-адресу, назначенному в подсети (примите во внимание автоматическое назначение FM и CP!). Он должен иметь одинаковое значение для всех узлов подсети.

Совет: если у вас есть несколько станций с одинаковым типом CPU, присвойте CPU различных станций разные имена (идентификаторы). По умолчанию все они имеют имя «CPUxxx(1)», поэтому в подсети они могут различаться по их MPI-адресам. Если вы не хотите присваивать имена самостоятельно, вы можете, например, сменить идентификатор по умолчанию «CPUxxx(1)» на «CPUxxx(n)», где «n» - MPI-адрес.

При назначении MPI-адреса также примите во внимание возможность более позднего подключения к MPI-сети устройства программирования или панели оператора (operator panel, OP) для осуществления поддержки или обслуживания. Установленные на постоянной основе программирующие устройства или OP должны быть подключены к сети MPI напрямую; для подключаемых через шнур (отвод) устройств имеется MPI-коннектор с сокетом (гнездом) с большим сечением и резьбовым соединением. Совет: зарезервируйте адрес 0 для сервисного программирующего устройства, адрес 1 – для сервисной OP и адрес 2 – для резервного CPU (соответствует адресам по умолчанию).

2.3.5 Наблюдение и модифицирование модулей

При помощи Hardware Configuration вы можете провести проверку монтажных соединений (разводки) машины или предприятия без пользовательской программы. Для этого требуется, чтобы программирующее устройство было подсоединено к станции (онлайн) и конфигурация сохранена, откомпилирована и загружена в CPU. Теперь вы можете адресовать любой каждый цифровой и аналоговый модуль. Выбе-

рите модуль, затем PLC → Monitor/Modify (PLC → Наблюдение/модифицирование) и установите операционные режимы наблюдения (Monitor) и модифицирования (Modify) и условия вызова.

Hardware Configuration по нажатию кнопки «Status Value» («Значение статуса») показывает сигнальные состояния значений каналов модулей. Кнопка «Modify Value» («Изменение значения») предназначена для записи в модуль значения, определенного в столбце Modify Value.

Если окошко метки (флажок) «I/O Display» («Отображение I/O») активировано, то периферийные входы/выходы (память модуля) отображаются вместо входов / выходов (образ процесса). Флажок «Enable Perif Outputs» («Разблокировать периферийные выходы») отменяет блокировку модулей выходов, если CPU находится в режиме STOP (Останов) (обратитесь к параграфу 2.7.5 «Разблокировка периферийных выходов»).

Вы можете ознакомиться с другими методами наблюдения и модифицирования входов и выходов в параграфах 2.7.3 «Наблюдение и модифицирование переменных» и 2.7.4 «Принудительная установка переменных».

2.4 Конфигурирование сети

Сетевая система станций S7 является коммуникационной базой SIMATIC-системы. Требуемые объекты – это подсети и модули с коммуникационными возможностями в станциях. Вы можете с использованием SIMATIC-менеджера создать в рамках иерархии проекта новые подсети и станции. Затем с помощью инструмента Hardware Configuration могут быть добавлены модули с коммуникационными возможностями (устройства CPU и CP); одновременно настраиваемой сети назначаются коммуникационные интерфейсы этих модулей. После этого определяются коммуникационные отношения между этими модулями – соединения – с помощью инструмента Network Configuration (Конфигурирование сети) в таблице соединений.

Network Configuration позволяет графически представить сконфигурированные сети и их узлы. Также с помощью Network Configuration вы можете создать все необходимые подсети и станции; затем назначить станции подсетям и параметризовать узловые свойства модулей с коммуникационными возможностями.

Чтобы определить коммуникационные отношения посредством инструмента конфигурирования сети, вы можете действовать следующим образом:

- Откройте стандартную подсеть MPI в контейнере проекта; если она более не доступна, просто создайте новую подсеть с помощью Insert → Subnet (Вставка → Подсеть).
- Используйте Network Configuration для создания необходимых станций и – если требуется – других подсетей.
- Откройте станции и обеспечьте их модулями с коммуникационными возможностями.
- При необходимости измените параметры сети.
- Если требуется, определите коммуникационные соединения в таблице соединений.

Также в рамках Network Configuration можно сконфигурировать коммуникацию глобальных данных: выберите подсеть MPI и затем Options → Define Global Data (Опции → Определить глобальные данные) (обратитесь к параграфу 20.5 «Коммуникация глобальных данных»).

Network → Save (Сеть → Сохранить) сохраняет данные во время работы с Network Configuration. Вы можете проверить согласованность сетевых настроек с помощью Network → Consistency Check (Сеть → Проверка согласованности).

Network Configuration закрывается по нажатию Network → Save and Compile (Сеть → Сохранить и скомпилировать).

Окно Network (окно сети)

Перед запуском утилиты Network Configuration у вас должен быть создан проект. Вместе с проектом SIMATIC-менеджер автоматически создает подсеть MPI.

Двойной щелчок на этой или любой другой подсети запускает Network Configuration. Также можно запустить Network Configuration, если открыть объект Connections (Соединения) в контейнере CPU.

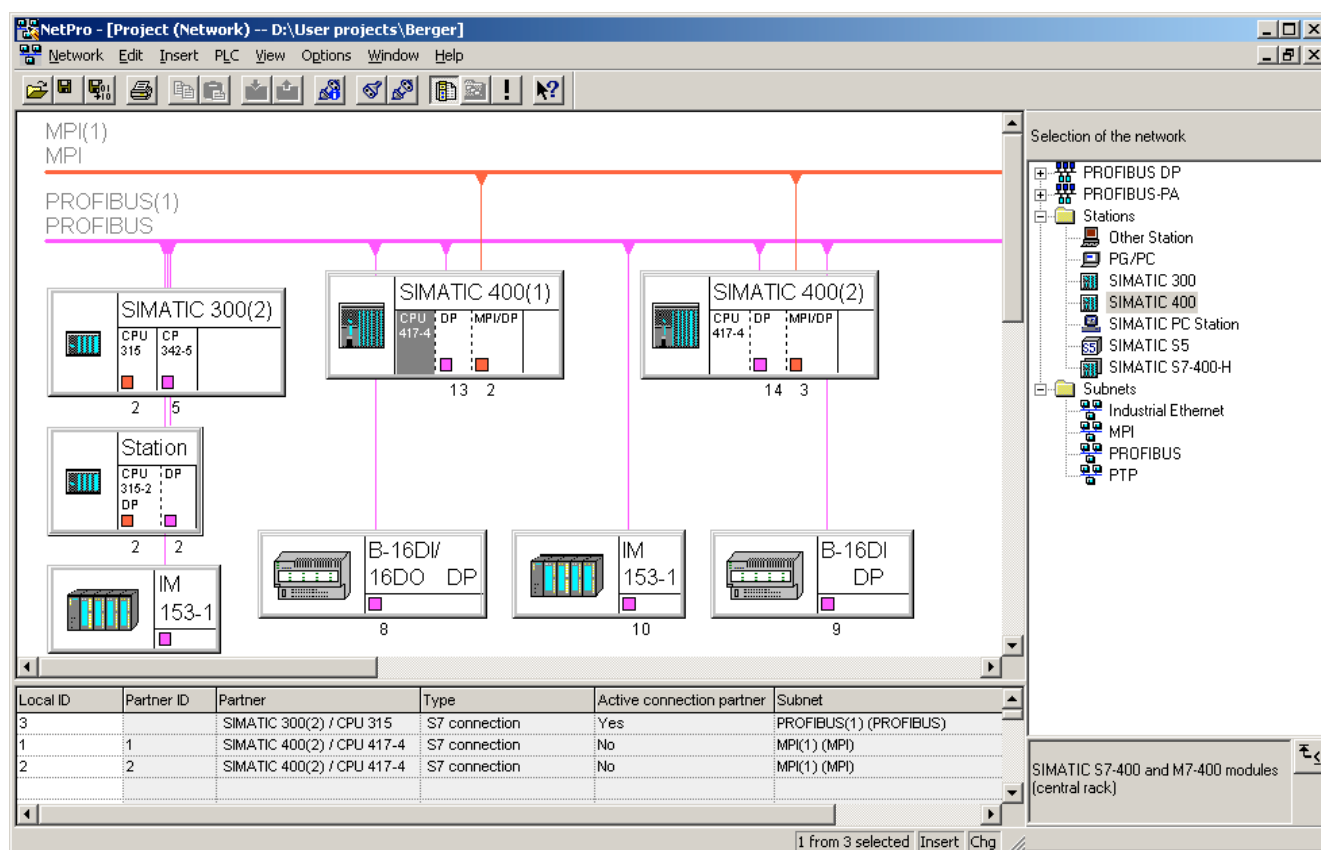


Рисунок 2.5 Пример Network Configuration

В верхней части окна Network Configuration показаны все ранее созданные в проекте подсети и станции (узлы) со сконфигурированными соединениями.

Если в верхней секции окна выбран модуль с «коммуникационными возможностями», например, CPU S7-400, то в нижней части окна отобразится таблица соединений.

Второе окно показывает каталог сетевых объектов (Network Object Catalog) с выбором доступных станций SIMATIC, подсетей и DP-станций. Вы можете скрыть каталог и вывести его на экран с помощью пункта меню View → Catalog (Вид → Каталог), а также «пристыковать» его к правому краю окна сети (двойным щелчком на линейке заголовка).

При помощи команд View → Zoom In (Вид → Увеличить масштаб), View → Zoom Out (Вид → Уменьшить масштаб) и View → Zoom Factor... (Вид → Выбрать масштаб...) вы можете настроить масштаб графического представления.

2.4.1 Настройка просмотра сети

Работа в Network Configuration начинается с выбора подсети. При помощи мыши вы должны выбрать из каталога подсеть, и, удерживая, перетащить ее в окно сети. Подсеть в окне представляется в виде горизонтальной линии. Недопустимые позиции индицируются запрещающим указателем мыши.

Идентичные действия выполняются по отношению к требуемым станциям, для начала без подключения к сети. Станции еще «пустые». Двойной щелчок на станции откроет инструментарий Network Configuration, позволяющий вам сконфигурировать станцию или, по крайней мере, модуль(и) с сетевым соединением. Сохраните станцию и вернитесь к Network Configuration.

Интерфейс модуля с коммуникационной возможностью изображен в Network Configuration в виде маленького квадрата под представлением модуля. Нажмите на этот квадрат, и, удерживая, перетащите его к соответствующей подсети. Соединение с подсетью представлены в виде вертикальной линии.

То же самое проделайте с другими узлами.

В окне сети вы можете передвигать созданные подсети и станции. Таким образом, вы также можете визуальнo представить вашу аппаратную конфигурацию.

Установка коммуникационных свойств

После формирования графического представления происходит параметризация подсетей: выберите подсеть и нажмите Edit → Object Properties (Правка → Свойства объекта). Окно свойств, которое затем появляется, во вкладке «General» («Общие») содержит ID (идентификатор) подсети S7. ID состоит из двух шестнадцатеричных чисел, номера проекта и номера подсети. Этот ID подсети S7 необходим, если вы хотите работать в онлайн-режиме с программирующим устройством без соответствующего проекта с целью достижения по сети других узлов. Сетевые свойства, например, скорость передачи данных или наибольший адрес узла, устанавливаются во вкладке «Network Settings» («Сетевые установки»).

При выборе сетевого соединения (подключения к сети) узла вы можете определить сетевые свойства узла с помощью Edit → Object Properties (Правка → Свойства объекта), например, адрес узла и сеть, к которой он подключен, или вы можете создать новую подсеть.

На вкладке «Interfaces» («Интерфейсы») свойств станции вы можете увидеть обзор всех модулей с коммуникационными возможностями, содержащий адреса узлов и используемые типы подсетей.

Определение свойств модулей узлов осуществляется тем же способом (теми же действиями оператора, что и в инструменте Hardware Configuration).

2.4.2 Конфигурирование системы DP-мастера в Network Configuration

Network Configuration также можно использовать для конфигурирования распределенных входов/выходов. Выберите опцию меню View → With DP Slaves (Вид → С DP-ведомыми), чтобы отобразить или скрыть DP-ведомых в окне просмотра сети.

Чтобы сконфигурировать систему DP-мастера, вам потребуется следующее:

- Подсеть PROFIBUS (если она еще не доступна, перетащите подсеть PROFIBUS из каталога сетевых объектов в окно сети).
- DP-мастер станции (если он еще не доступен, перетащите станцию из каталога сетевых объектов в окно сети, откройте станцию и выберите DP-мастер с помощью Hardware Configuration либо интегрированный в CPU, либо как автономный модуль).
- Соединение из DP-мастера к подсети PROFIBUS (либо выберите подсеть в Hardware Configuration, либо щелкните на сетевом соединении к DP-мастеру в Network Configuration, «удерживая», перетащите в сеть PROFIBUS).

В окне сети выберите DP-мастера, которому должен быть назначен ведомый. Найдите DP-ведомого в каталоге сетевых объектов под «PROFIBUS» и соответствующим подкаталогом, перетащите его в окно сети и заполните появившееся окно свойств.

Параметризация DP-ведомого осуществляется путем выбора этого объекта и последующего нажатия Edit → Open Object (Правка → Открыть объект). Запускается Hardware Configuration. Теперь вы можете установить адреса пользовательских данных или, в случае модульных ведомых, выбрать модули входов/выходов (обратитесь к параграфу 2.3 «Конфигурирование станций»).

Если DP-ведомый с развитой логикой (I-slave) был вами предварительно создан, то можете только подключить его к подсети (смотрите параграф 20.4.2 «Конфигурирование распределенных входов/выходов»). Тип интеллектуального DP-ведомого вы можете найти в каталоге сетевых объектов в разделе «Already created stations» («Уже созданные станции»); перетащите его, предварительно отметив DP-мастер, в окно сети и заполните появляющееся окно свойств (как в инструменте Hardware Configuration).

С помощью View → Highlight → Master System (Вид → Подсветить → Система мастера) можно выделить назначение узлов системы DP-мастера; сначала выделите мастера или ведомого этой системы мастера.

2.4.3 Конфигурирование соединений

Соединения описывают коммуникационные отношения между двумя устройствами. Соединения должны быть сконфигурированы, если

- вы хотите установить SFB-коммуникации между двумя устройствами SIMATIC S7 («Коммуникация посредством сконфигурированных соединений») или
- коммуникационный партнер не является устройством SIMATIC S7.

Примечание: сконфигурированное соединение для непосредственного онлайнного подключения программирующего устройства к сети MPI для программирования и отладки не требуется. Если вы хотите получить доступ к другим узлам, которые объединены в другую подключенную сеть, с помощью программирующего устройства, то вы должны сконфигурировать соединение программирующего устройства. Для этого в каталоге сетевых объектов выберите двойным щелчком объект *PG/PC* (устройства программирования) под *Stations (Станции)*, в окне сети откройте двойным щелчком *PG/PC* и выберите интерфейс и назначьте его подсети.

Таблица соединений

Коммуникационные соединения конфигурируются в таблице соединений. Требование: у вас есть созданный проект со всеми станциями, между которыми предполагается обмен данными, и вы назначили модули с коммуникационными возможностями сети.

Объект *Connections (Соединения)* в контейнере *CPU* представляет таблицу соединений. Двойной щелчок на *Connections (Соединения)* запускает Network Configuration, также как двойное нажатие на подсети в контейнере проекта.

Чтобы сконфигурировать соединения, выберите в Network Configuration CPU S7-400. В нижней части окна сети размещена таблица соединений (таблица 2.1; если она не видна, поместите указатель мыши на нижнюю границу окна до смены его формы и перетащите границу окна вверх). Ввод нового коммуникационного соединения осуществляется путем Insert → New Connection (Вставка → Новое соединение) или двойным щелчком на пустой строке.

Соединение создается для каждого «активного» CPU. Заметьте, что нельзя создать таблицу соединений для CPU S7-300; CPU S7-300 могут быть только «пассивными» партнерами в S7-соединении.

Выбор коммуникационного партнера осуществляется в диалоговых окнах «Station» («Станция») и «Module» («Модуль») окна «New Connection» («Новое соединение») (рисунок 2.6); станция и модуль должны к этому моменту существовать. В данном окне также определяется тип соединения.

При желании установить другие свойства соединения активируйте флажок «Show Properties Dialog» («Показать диалог свойств»).

Таблица соединений содержит все данные о сконфигурированных соединениях. Для отображения ее в требуемом виде используйте команду View → Display Columns (Вид → Отобразить столбцы) и выберите интересующую вас информацию.

ID (идентификатор) соединений

Количество возможных соединений определяется CPU. STEP 7 определяет ID соединения для каждого отдельного соединения и каждого партнера. Данная спецификация требуется вам при использовании в программе коммуникационных блоков. Вы можете изменить **локальный ID (local ID)**, ID соединения открытого в настоящий момент модуля). Это необходимо, если вы уже запрограммировали коммуникационные блоки и хотите использовать для соединения определенный в них локальный ID.

Новый локальный ID вводится в виде шестнадцатеричного числа. Оно не должно быть уже присвоено и должно оставаться в следующих границах значений, зависящих от типа соединений:

- Область значений для S7-соединений:
от 0001_{hex} до 0FFF_{hex}
- Область значений для PtP-соединений:
от 1000_{hex} до 1400_{hex}

ID партнера (partner ID) изменяется путем обращения к таблице соединений CPU-партнера и изменения (в этих условиях) локального ID: выберите линию соединения и нажмите Edit → Object Properties (Правка → Свойства объекта). Если STEP 7 не вводит ID партнера, то данное соединение является односторонним (смотрите ниже).

Таблица 2.1 Пример таблицы соединений

Local ID / Локаль- ный ID	Partner ID / ID парт- нера	Partner / Партнер	Type / Тип	Active connection partner / Актив- ный партнер со- единения	Send operating mode messages / Посылает сообщения рабочего режима
1	1	Станция 416 / CPU416(5)	S7- соединение	да	нет
2	2	Станция 416 / CPU416(5)	S7- соединение	да	нет
3		Станция 315 / CPU315(7)	S7- соединение	да	нет
4	1	Станция 417 / CPU414(4)	S7- соединение	да	нет

Партнер

Столбец отображает коммуникационного партнера. Если вы хотите зарезервировать соединительный ресурс без наименования устройства-партнера, в диалоговом окне под станцией (Station) введите «unspecified» («неопределено»).

В **одностороннем соединении (one-way connection)** коммуникация может быть инициирована одним партнером; пример: SFB-коммуникации между CPU S7-400 и S7-300. Хотя функции SFB-коммуникаций не доступны в S7-300, обмен данными может быть осуществлен CPU S7-400 при помощи SFB 14 GET и SFB 15 PUT. В S7-300 для этой коммуникации пользовательская программа не выполняется, а поддержка обмена данными осуществляется операционной системой.

Одностороннее соединение конфигурируется в таблице соединений «активного» CPU. Только после этого STEP 7 назначает «локальный ID». Также загрузка этого соединения происходит только в локальную станцию.

В случае **двухстороннего соединения (two-way connection)** оба партнера активно участвуют в коммуникации; например, два CPU S7-400 с помощью коммуникационных функций SFB 8 SEND и SFB 9 BRCV.

Для одного из двух партнеров конфигурирование двухстороннего соединения происходит только один раз. Затем STEP 7 назначает «локальный ID» и «ID партнера» и генерирует данные соединения для обеих станций. Загрузка каждого партнера должна осуществляться с помощью его собственной таблицы соединений.

Тип соединений

Базовый пакет STEP 7 предоставляет вам в Network Configuration следующие типы соединений:

PtP-соединение (двухточечное соединение) принято для использования в подсети RTP (процедуры 3964(R) и RK 512) с SFB-коммуникациями. PtP-соединение (соединение point-to-point, двухточечное) является последовательным соединением между двумя партнерами. Это могут быть устройства SIMATIC S7 с соответствующими CP или устройство SIMATIC и устройство другого производителя (не Siemens), например, принтер или устройство считывания штрихового кода.

S7-соединение принято к использованию в подсетях MPI, PROFIBUS и промышленного Ethernet с SFB-коммуникациями. S7-соединение – это соединение между устройствами SIMATIC S7 и может включать программирующие устройства и устройства интерфейса человек-машина. Обмен данными происходит через S7-соединение, или исполняются программные и управляющие функции.

Отказоустойчивое S7-соединение принято для подсетей PROFIBUS и промышленного Ethernet с SFB-коммуникациями. Отказоустойчивое S7-соединение создается между устойчивыми к сбоям устройствами SIMATIC S7, а также может быть использовано в соответствующим образом оборудованном PC.

Дополнительные пакеты «NCM S7 for PROFIBUS» и «NCM S7 for Industrial Ethernet» применяются для **параметризации CP**. В зависимости от установленного программного обеспечения NCM можно выбрать дополнительные типы соединений: FMS-, FDL-, транспортное ISO-, ISO-on-TCP-, TCP-, UDP- и email-соединение.

Построение активного соединения

Перед фактической передачей данных соединение должно быть образовано (инициализировано). Если партнеры по соединению имеют такую возможность, то определяется, какое устройство устанавливает соединение. Вы можете это сделать с помощью флажка «Active Connection Buildup» («Построение активного соединения») в окне свойств соединения (выберите соединение, затем нажмите Edit → Object Properties, Правка → Свойства Объекта).

Отправление сообщений рабочего режима

Партнеры по соединению с настроенным двухсторонним соединением могут обмениваться сообщениями рабочего состояния. Если локальный узел будет посылать свои сообщения рабочего состояния, активируйте соответствующий флажок в окне свойств соединения. В пользовательской программе CPU-партнера эти сообщения могут быть получены с помощью SFB 23 USTATUS.

Путь соединения

В качестве пути соединения (connection path) в окне свойств соединения показаны конечные точки соединения и подсети, через которые оно проходит. Если на выбор имеются несколько подсетей, STEP 7 выбирает их в таком порядке:

- 1) промышленный Ethernet;
- 2) промышленный Ethernet/TCP-IP;
- 3) MPI;
- 4) PROFIBUS.

Станция и CPU, через которые проходит соединение, показаны как конечные точки соединения. Список модулей с коммуникационными возможностями приведен в пункте «Interface» («Интерфейс»), определяя номер стойки слот. Если оба CPU расположены в одной стойке (например, процессоры CPU S7-400 в мультипроцессорном режиме), то окно просмотра отображает «PLC-internal» («PLC-внутренний»).

В пункте «Types» («Типы») вы должны выбрать подсеть, по которой будет проходить соединение. Если оба партнера, к примеру, подключены к одной подсети MPI и одной подсети PROFIBUS, то в пункте «Types» («Типы») ставится «MPI». Вы можете изменить данное определение на «PROFIBUS», и STEP 7 автоматически подберет остальные установки. После этого в пункте «Address» («Адрес») вы увидите MPI-адрес или PROFIBUS-адрес.

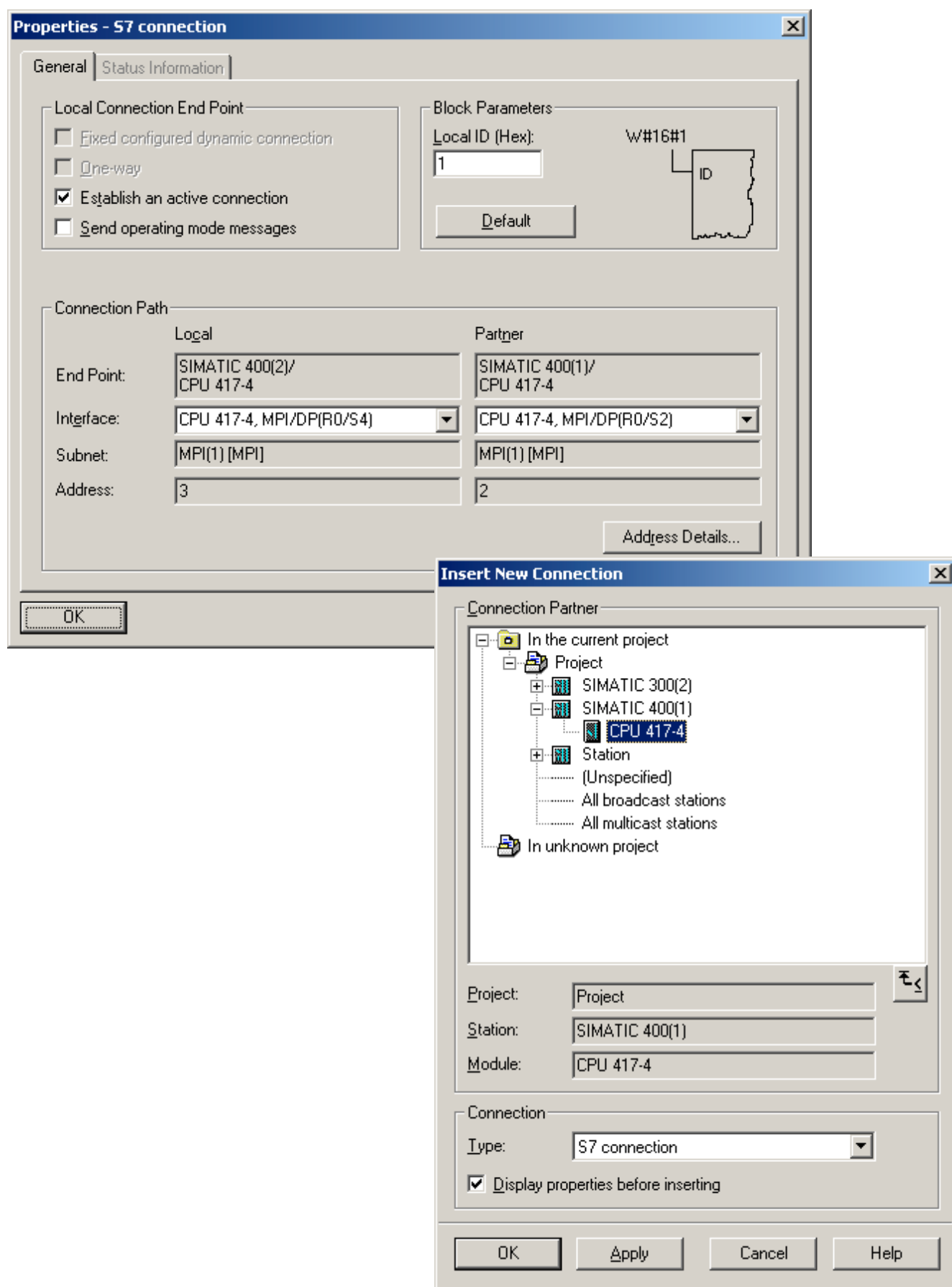


Рисунок 2.6 Конфигурирование коммуникационных соединений

Соединения между проектами

Для обмена данными между двумя модулями S7, принадлежащими различным SIMATIC-проектам, в таблице соединений для партнера по соединению вводится «unspecified» («неопределено») (в локальной станции в обоих проектах).

Убедитесь, что данные соединения в обоих проектах согласованы (STEP 7 не проверяет это). После сохранения и компиляции загрузите данные соединения в локальную станцию каждого проекта.

Подключение к станции, не являющейся станцией S7

В рамках проекта вы также можете определить в качестве партнеров по соединению станции, не являющиеся станциями S7:

- Другие станции (устройства производства не Siemens, а также станции S7 в другом проекте);
- Программирующие устройства/PC;
- Станции SIMATIC S5.

Для конфигурирования соединения требуется, чтобы подобная станция присутствовала в качестве объекта в контейнере проекта, и она была подключена к соответствующей подсети в свойствах станции. Например, выберите станцию в Network Configuration, нажмите Edit → Object Properties (Правка → Свойства объекта) и соедините станцию с нужной подсетью во вкладке «Interfaces» («Интерфейсы»).

2.4.4 Сетевые переходы

Если устройство программирования подключено к подсети, то оно может получить доступ ко всем другим узлам этой подсети. Например, из одной точки соединения вы можете программировать и отлаживать все станции S7, подключенные к подсети MPI.

Если иная подсеть, такая как подсеть PROFIBUS, подключена к станции S7, программирующее устройство также может получить доступ к станциям другой подсети. Для этого требуется, чтобы станция с подсетевым переходом имела возможность маршрутизации, то есть она проведет передаваемые фреймы сообщений.

Когда конфигурация сети откомпилирована, для станций с подсетевыми переходами автоматически генерируются таблицы маршрутизации (routing tables), содержащие всю необходимую информацию.

Все доступные коммуникационные партнеры в сети предприятия в рамках S7-проекта должны быть сконфигурированы и снабжены «знанием» о том, к каким станциям можно получить доступ через какие подсети и подсетевые переходы.

Если вы хотите получить доступ ко всем узлам подсети с использованием программирующего устройства из одной точки соединения, то вы должны сконфигурировать точку соединения. Введите «заполнитель» («placeholder»), PG/PC-станцию из каталога сетевых объектов в конфигурации сети соответствующей подсети. Вы должны сконфигурировать PG/PC-станцию на каждой подсети, к которой вы хотите подключить устройство программирования.

Во время выполнения операций вы должны подключить программирующее устройство к подсети и выбрать команду меню PLC → Assign PG/PC (PLC → Назначить PG/PC). Это адаптирует интерфейсы программирующего устройства к заданным установкам подсети. Перед отключением устройства программирования от подсети выберите PLC → Undo PG/PC Assignment (PLC → Отменить назначение PG/PC).

Если вы работаете в онлайн-режиме с программирующим устройством, не содержащим правильного проекта, для доступа в сеть вам потребуется ID подсети S7. ID подсети S7 включает в себя два номера: номер проекта и номер подсети. Вы можете получить ID подсети в Network Configuration путем выбора подсети и Edit → Object Properties (Правка → Свойства объекта) на вкладке «General» («Общие»).

2.4.5 Загрузка данных соединения

Чтобы активировать соединения, вы должны загрузить таблицу соединений в PLC после сохранения и компиляции (все таблицы соединений во все «активные» CPU).

Требование: вы находитесь в окне сети, и таблица соединений видима. Программирующее устройство является узлом подсети, по которой данные соединений должны быть загружены в модули с коммуникационными возможностями.

Всем узлам подсети присвоены уникальные узловые адреса. Модули, которым данные соединений будут переданы, находятся в режиме STOP.

С помощью опции меню PLC → Download → ... (PLC → Загрузить...) вы можете передать данные соединений и конфигураций доступным модулям. В зависимости от того, какой объект отмечен и какая команда меню выбрана, вы можете выбрать между следующим:

<u>→ Selected Stations</u>	(<u>→ Отмеченные станции</u>),
<u>→ Selected and Partner Stations</u>	(<u>→ Отмеченные и станции-партнеры</u>),
<u>→ Selected Connections</u>	(<u>→ Отмеченные соединения</u>),
<u>→ Stations on Subnet</u>	(<u>→ Станции в подсети</u>),
<u>→ Connections and Routers</u>	(<u>→ Соединения и маршрутизаторы</u>).

Откомпилированные данные соединений также являются составляющей частью *System data* (Системных данных) в контейнере *Blocks* (Блоки). Передача системных данных и последующего запуска центральных процессоров также эффективно передает данные соединений модулям с коммуникационными возможностями.

Для работы в онлайн-режиме через MPI дополнительных аппаратных средств не требуется. Если вы подключаете PC к сети или осуществляете включение программирующего устройства в сеть Ethernet или PROFIBUS, то вам потребуется соответствующий интерфейсный модуль. Параметризация модуля выполняется в приложении «Set PG/PC Interface» («Установка интерфейса PG/PC») панели управления Windows (Windows Control Panel).

2.5 Создание программы S7

2.5.1 Введение

Пользовательская программа создается в папке *S7 Program* (*S7-программа*). Вы можете назначить этот объект в иерархии проекта центрального процессора (CPU) или можете создать его независимо от CPU. Он содержит объект *Symbols* (*Символы*), контейнеры *Source Files* (*Исходные файлы*) и *Blocks* (*Блоки*) (рисунок 2.7).

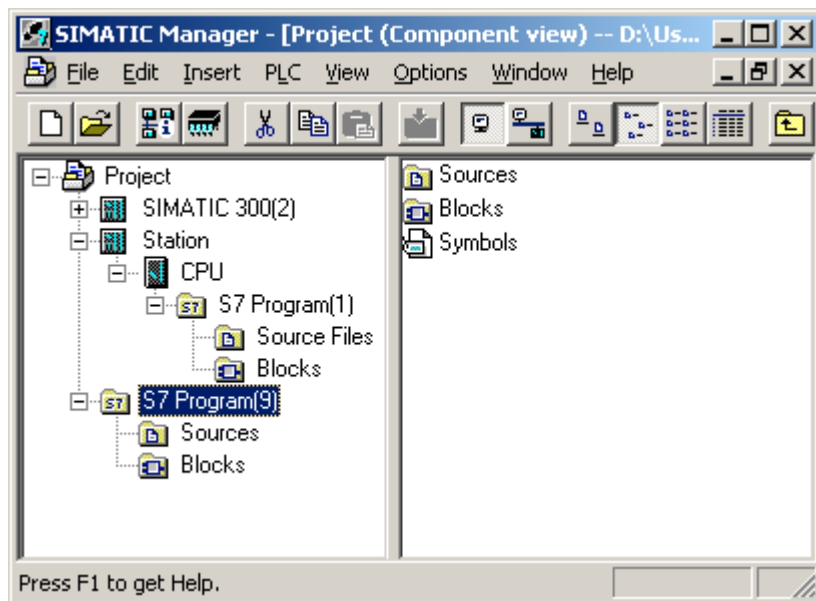


Рисунок 2.7 Объекты для программирования

Используя **пошаговое** создание программ, вы вводите программу непосредственно блок-за-блоком (*block-by-block*). Синтаксис вводимой информации немедленно проверяется. Блок компилируется одновременно с записью и сохранением его в контейнере *Blocks* (*Блоки*). С помощью пошагового программирования вы также можете редактировать блоки в CPU в онлайн-режиме, даже во время выполнения операций. Пошаговое программирование доступно для всех основных языков.

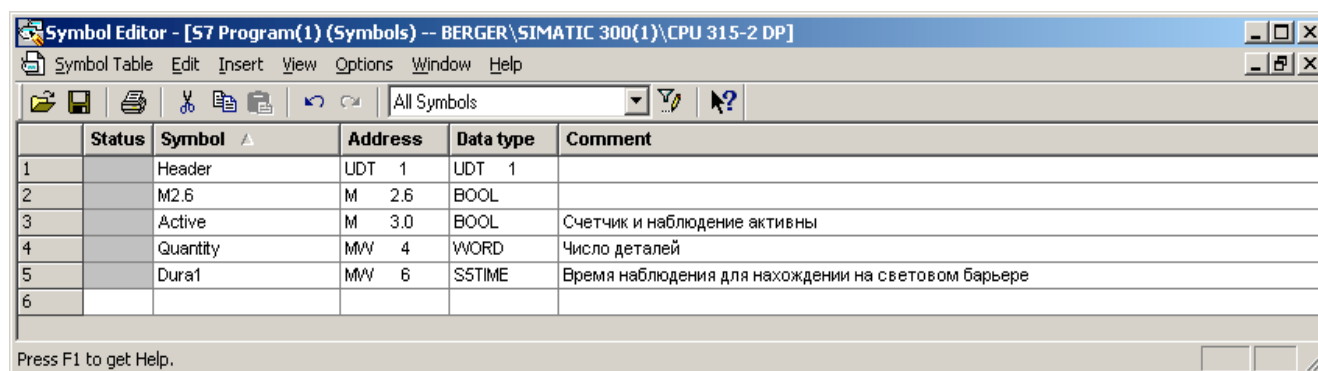
В случае способа создания программ, **ориентированного на источник (исходные файлы)**, вы создаете один или более исходных файлов программы и сохраняете их в контейнере *Source Files* (*Исходные файлы*). Исходные программные файлы представляют собой текстовые файлы в формате ASCII, содержащие программные операторы для одного или более блоков, возможно даже для всей программы. Они компилируются, и откомпилированные блоки помещаются в контейнер *Blocks* (*Блоки*). Ориентированное на исходные файлы создание программ используется в STL; вы не можете воспользоваться данным способом программирования в LAD или FBD, но программы, созданные с помощью LAD или FBD, могут быть сохранены как исходные файлы.

В программе обрабатываются сигнальные состояния значений адресов. Адресом является, например, вход I 1.0 (*абсолютная адресация – absolute addressing*). С помощью таблицы символов под объектом *Symbols* (*Символы*) вы можете назначить адресу символ (буквенно-цифровое имя, например, «Switch motor on» или «Включить мотор») и затем обращаться к нему по этому имени (*символическая адресация – symbolic addressing*). В свойствах контейнера автономных объектов *Blocks* (*Блоки*) определяется, абсолютный адрес или символ является определяющим для уже откомпилированных блоков при следующем сохранении в случае изменения в таблице символов (*приоритет адреса – address priority*).

2.5.2 Таблица символов

В управляющей программе вы работаете с адресами; это входы, выходы, таймеры, блоки. Вы можете присвоить абсолютные адреса (например, I1.0) или символические адреса (например, Start signal). Символическая адресация использует вместо абсолютного адреса имени. Применяя осмысленные имена, вы можете сделать вашу программу более читаемой.

В символической адресации различают *локальные (local)* и *глобальные (global)* символы. Локальный символ известен только в блоке, в котором он был определен. Вы можете использовать одинаковые локальные символы в разных блоках для различных целей. Глобальный символ известен во всей программе и имеет одинаковое значение во всех блоках. Глобальные символы определяются в таблице символов (объект *Symbols* (*Символы*) в контейнере *S7 Program* (*S7-программа*)).



	Status	Symbol	Address	Data type	Comment
1		Header	UDT 1	UDT 1	
2		M2.6	M 2.6	BOOL	
3		Active	M 3.0	BOOL	Счетчик и наблюдение активны
4		Quantity	MW 4	WORD	Число деталей
5		Dura1	MW 6	S5TIME	Время наблюдения для нахождения на световом барьере
6					

Рисунок 2.8 Пример таблицы символов

Глобальный символ начинается с буквы, и его длина может составлять до 24 знаков. Он также может включать в себя пробелы, специальные литеры и национальные буквы, такие как умляут. Исключениями являются знаки 00_{hex}, FF_{hex} и кавычки ("). При написании программ символы со специальными литерами вы должны заключать в кавычки. В скомпилированных блоках все глобальные символы программный редактор отображает в кавычках. Комментарий символа может включать до 80 знаков.

В таблице символов вы можете присвоить имена следующим адресам и объектам:

- Входам I, выходам Q, периферийным входам PI и периферийным выходам PQ;
- Меркерам M, таймерам T и счетчикам C;
- Кодовым блокам OB, FB, FC, SFC, SFB и блокам данных DB;
- Определенным пользователем типам данных UDT;
- Таблице переменных (variable table) VAT.

Адреса данных и блоки данных включены в число локальных адресов; ассоциированные символы определяются в разделе описаний блока данных в случае глобальных блоков данных и разделе описаний функционального блока в случае блоков данных экземпляров.

При создании программы S7 SIMATIC-менеджер создает также пустую таблицу символов *Symbols* (*Символы*). Вы можете ее открыть, определить глобальные символы и назначить их абсолютным адресам (рисунок 2.8). В программе S7 может быть только одна таблица символов.

Тип данных является частью определения символа. Он определяет специфические свойства данных за символом, по существу представление содержимого данных. Например, тип данных BOOL (логический) идентифицирует двоичную переменную, а тип данных INT (целый) обозначает цифровую переменную, чье содержимое представляет 16-битное целое. Обратитесь к параграфу 3.5 «Переменные, константы и типы данных», в нем приводится подробная информация о типах данных.

Используя пошаговое программирование, вы создаете таблицу символов перед вводом программы; также вы можете добавить или откорректировать отдельные символы во время ввода программы. В случае ориентированного на источник программирования завершенная таблица символов должна быть доступна, когда компилируется исходная программа.

Импортирование, экспортирование

Таблицы символов могут быть импортированы и экспортированы. «Экспорт» означает создание файла с содержимым вашей таблицы символов. Вы можете выбрать для этого либо всю таблицу символов, либо ограниченное фильтрами подмножество, либо только отдельные строки. В качестве формата данных вы можете выбрать между чистым ASCII-текстом (расширение *.asc), последовательным списком назначений (*.seq), форматом системных данных (System Data Format, *.sdf для Microsoft Access) и форматом обмена данными (Data Interchange Format, *.dif для Microsoft Excel). Экспортированный файл можно откорректировать в соответствующем редакторе. Таблицу символов в одном из названных выше форматов можно также импортировать.

Свойства специальных объектов

С помощью команды Edit → Special Object Properties → ... (Правка → Свойства специальных объектов → ...) вы можете установить атрибуты для каждого символа в таблице символов. Атрибуты или свойства используются

- В функциях интерфейса человек-машина (HMI-функциях) для наблюдения с использованием WinCC;
- При конфигурировании коммуникаций;
- При конфигурировании сообщений;
- При наблюдении процесса с использованием S7-PDIAG.

Видимыми установки делает команда меню View → Columns O, M, C, R (Вид → Столбцы O, M, C, R). С помощью Options → Customize (Опции → Настроить) вы можете установить, будут ли копироваться свойства специальных объектов, и определить поведение при импортировании символов.

2.5.3 Редактор программ

Для создания пользовательских программ базовый пакет STEP 7 содержит редактор программ (Program Editor) для языков программирования LAD, FBD и STL. С LAD и FBD вы будете программировать пошагово, то есть будете вводить непосредственно исполняемый блок; на рисунке 2.9 показаны возможные действия для этого.

Если вы используете символическую адресацию для глобальных адресов, то в случае пошагового программирования символы уже должны быть присвоены абсолютным адресам; тем не менее, вы можете ввести новые символы или изменить их во время ввода программы.

Блоки LAD/FBD могут быть «декомпилированы», то есть из кода MC7 может быть воссоздан читаемый блок без использования автономной базы данных (вы можете прочитать любой блок из CPU с помощью программирующего устройства без ассоциированного проекта). Кроме того, из любого откомпилированного блока может быть создана исходная программа STL.

Запуск редактора программ

Доступ к редактору программ можно получить при открытии блока в SIMATIC-менеджере, например, по двойному щелчку на автоматически генерируемом символе организационного блока OB 1 или через панель задач Windows, нажав Start → Simatic → STEP 7 → LAD, STL, FBD – Program S7 Blocks (Пуск → Simatic → STEP 7 → LAD, STL, FBD → Программировать блоки S7).

каждый блок, в конечном счете, содержит исполняемый код МС7 независимо от языка программирования, использованного для его написания.

Используя пошаговое программирование, вы можете выполнять все функции программирования с одним исключением: если вы хотите предусмотреть защиту блоков (KNOW_HOW_PROTECT), сделать это вы сможете только через исходный программный файл (за более подробной информацией по этой теме обратитесь к параграфу 24.1 «Защита блоков»).

Пошаговое программирование

Применяя пошаговое программирование, вы можете редактировать блоки как в автономном, так и в онлайн-контейнере *Blocks* (Блоки). Редактор проверит введенные вами в пошаговом режиме данные, как только вы выйдете из сети (network). При закрытии блока он немедленно компилируется, так что только не содержащие ошибок блоки могут быть сохранены.

На вкладке «Create Blocks» («Создание блоков»), нажав Options → Customize (Опции → Настроить), можно установить автоматическое обновление ссылочных данных при записи блока.

Блоки могут редактироваться как автономно в базе данных устройства программирования, так и онлайн в CPU, обычно называемом «программируемый контроллер» или «PLC». Для этих целей SIMATIC-менеджер предоставляет автономное и онлайн-новое окно; одно отличается от другого метками в линейке заголовка.

В автономном окне происходит редактирование блоков непосредственно в базе данных PG. Если вы находитесь в редакторе, можете сохранить измененный блок в автономной базе данных с помощью опции меню File → Save (Файл → Сохранить) и передать его в CPU, нажав PLC → Download (PLC → Загрузить). Если вы хотите сохранить открытый блок под другим номером или в другом проекте, или если вы хотите переместить его в библиотеку или в другой CPU, воспользуйтесь командой меню File → Save As (Файл → Сохранить как).

Чтобы редактировать блок в CPU, откройте этот блок в онлайн-окне. Это переместит блок из CPU в программирующее устройство, и, таким образом, он может редактироваться. Вы можете записать отредактированный блок обратно в CPU, выбрав PLC → Download (PLC → Загрузить). Если CPU находится в режиме RUN (исполнение), то центральный процессор обработает измененный блок в следующем цикле программного сканирования. Если вы хотите сохранить отредактированный в режиме онлайн блок также в автономной базе данных, то это можно выполнить с помощью команды File → Save (Файл → Сохранить).

Параграфы 2.6.4 «Загрузка пользовательской программы в CPU» и 2.6.5 «Обработка блоков» продолжают рассказ об онлайн-программировании. Параграфы 3.3 «Программирование кодовых блоков» и 3.4 «Программирование блоков данных» показывают, как вводить блоки LAD/FBD.

2.5.4 Обновление или генерирование исходных файлов

На вкладке «Source Files» («Исходные файлы»), вызываемой командой Options → Customize (Опции → Настроить), вы можете выбрать опцию «Generate source file automatically» («Автоматически генерировать исходный файл») с тем, чтобы при сохранении (пошагово созданного) блока исходный файл программы обновлялся или создавался, если он еще не был сгенерирован.

Имя нового исходного файла может быть образовано на основе абсолютного адреса или символического адреса. Адреса могут быть переданы в исходный файл в абсолютной или символической форме.

Нажмите кнопку «Execute» («Выполнить») и в появившемся диалоговом окне выберите блоки, из которых вы хотите сгенерировать программный исходный файл.

Вы можете экспортировать исходные файлы из проекта путем выбора в SIMATIC-менеджере Edit → Export Source File (Правка → Экспортировать исходный файл). После этого вы можете произвести дальнейшую обработку ASCII-файлов, например, в другом текстовом редакторе. Исходные файлы также могут быть импортированы обратно в контейнер *Source Files* (Исходные файлы) с помощью Insert → External Source File (Вставка → Внешний исходный файл).

Если вы генерируете исходный файл из блока, созданного с помощью LAD или FBD, то вы сможете из этого исходного файла вновь сгенерировать LAD- или FBD-блок. Компиляция исходного файла осуществляется после его открытия в SIMATIC-менеджере по двойному щелчку и выбора в редакторе программ File → Compile (Файл → Компилировать). В контейнере *Blocks* (Блоки) создается STL-блок. Откройте этот блок и перейдите к обычному для вас представлению, нажав View → LAD (Вид → LAD) или View → FBD (Вид → FBD). После записи блок сохраняет это свойство.

Если при создании исходного файла вы выбрали установку «Addresses – Symbolic» («Адреса – символические»), для компилирования исходного файла вам потребуется завершенная таблица символов. Таким образом, вы можете определить в таблице символов различные абсолютные адреса и, после компиляции, получить в итоге программу, например, с различными входами и выходами. Это позволяет вам адаптировать программу к различным конфигурациям аппаратных средств. Для этой цели лучше сохранить эти исходные файлы (которые не зависят от аппаратной адресации), например, в библиотеке.

Перемонтаж (переадресация)

Функция *Rewire* (Перемонтаж) позволяет вам заменить адреса отдельно в откомпилированных блоках или во всей программе пользователя. Например, вы можете заменить входные биты с I0.0 по I0.7 на входные биты I 16.0, ..., I 16.7.

Допустимыми адресами являются входы, выходы, биты памяти, таймеры и счетчики, а также функции FC и функциональные блоки FB.

В SIMATIC-менеджере вы должны выбрать объекты, в которых вы хотите выполнить перезапись; выбираете один блок, группу блоков, отмечая их мышью при нажатой клавише Ctrl, или всю папку *Blocks* (Блоки) программы пользователя. Нажав Options → Rewire (Опции → Перемонтировать), вы попадете в таблицу, в которой вы можете ввести заменяемые старые адреса и новые адреса. После подтверждения кнопкой «OK» SIMATIC-менеджер производит обмен адресов.

При «перемонтаже» блоков сначала измените номера блоков, затем выполните переадресацию, которая модифицирует вызовы соответствующим образом.

В отображаемом позже файле указано, в каких блоках сделаны изменения и сколько.

Другие возможные методы переадресации:

- С отдельно откомпилированными блоками можно использовать функцию *Address priority* (Приоритет адреса);
- Если имеется программный исходный файл с символической адресацией, вы должны изменить абсолютные адреса в таблице символов перед компиляцией, и в вашей программе после компиляции не будет «защитных» адресов, то есть она будет связана с другими адресами.

2.5.5 Приоритет адреса

В окне свойств автономного контейнера объектов *Blocks* (Блоки), во вкладке «Blocks» («Блоки») вы можете установить, абсолютный или символический адрес будет иметь приоритет для уже записанных блоков, когда они отображаются и сохраняются вновь после изменений в таблице символов или в описании (объявлении) или назначении глобальных блоков данных.

Присваиваемое по умолчанию значение – «Absolute address has priority» («Приоритет имеет абсолютный адрес») (так же, как и в предыдущих версиях STEP 7). Это означает, что когда в таблице символов сделано изменение, абсолютный адрес в программе сохраняется, а символ соответственно меняется. Если установлено «Symbol has priority» («Приоритет имеет символ»), меняется абсолютный адрес, символ сохраняется.

Пример:

Таблица символов содержит следующее:

```
I 1.0 "Limit_switch_up"  
I 1.1 "Limit_switch_down"
```

В программе уже скомпилированного блока вход I1.0 сканируется:

```
I 1.0 "Limit_switch_up"
```

Если назначения для входов I1.0 и I1.1 в таблице символов изменены на:

```
I 1.0 "Limit_switch_down"
I 1.1 "Limit_switch_up"
```

и уже откомпилированный блок считан, то программа содержит:

```
I 1.1 "Limit_switch_up"
```

если установлено «Symbol has priority» («Приоритет имеет символ») и установлено «Absolute address has priority» («Приоритет имеет абсолютный адрес»), то программа содержит:

```
I 1.0 "Limit_switch_down"
```

Если в результате изменения в таблице символов удалены все связи между абсолютным адресом и символом, то в случае установленного флажка «Absolute address has priority» («Приоритет имеет абсолютный адрес») оператор (statement) будет содержать абсолютный адрес (даже с символическим отображением, потому что символа, разумеется, не будет). Если установлено «Symbol has priority» («Приоритет имеет символ»), оператор отклоняется как ошибочный (потому, что отсутствует обязательный абсолютный адрес).

Если установлено «Symbol has priority» («Приоритет имеет символ»), то пошагово запрограммированные блоки с символической адресацией в случае изменения таблицы символов сохраняют свои символы. Таким образом, уже запрограммированный блок может быть «перешит» сменой назначения адреса.

Заметьте, что эта «перешивка» не происходит автоматически, потому что уже откомпилированные блоки содержат исполняемый MC7-код операторов с абсолютными адресами. Изменения касаются только соответствующих блоков – следующих за соответствующим сообщением – после того, как они были открыты и сохранены снова.

2.5.6 Ссылочные данные

В качестве дополнения к самой программе SIMATIC-менеджер показывает вам ссылочные данные (reference data), которые вы можете использовать как основу для исправлений или тестов. Эти ссылочные данные включают следующее:

- Перекрестные ссылки
- Резервированные адреса (I, Q, M и T, C)
- Структура программы
- Неиспользуемые символы
- Адреса без символов

Чтобы сгенерировать ссылочные данные, выберите объект *Blocks* (Блоки) и команду меню Options → Reference Data → Display (Опции → Ссылочные данные → Отобразить). Представление ссылочных данных может быть установлено отдельно для каждого рабочего окна с помощью View → Filter... (Вид → Фильтр...); вы можете сохранить установки для более позднего редактирования путем выбора опции «Save as Standard» («Сохранить в качестве стандартного»). Вы можете открыть и просмотреть несколько списков одновременно.

С помощью Options → Customize (Опции → Настроить) в редакторе программ на вкладке «Create Blocks» («Создание блоков») задайте, будут ли обновляться ссылочные данные при компилировании программного исходного файла или при сохранении пошагово написанного блока.

Обратите внимание, что ссылочные данные доступны только при автономном управлении данными; автономные ссылочные данные отображаются даже в случае, если функция вызвана в блоке, открытом в режиме онлайн.

Перекрестные ссылки

Список перекрестных ссылок показывает, как используются адреса и блоки в пользовательской программе. Он включает абсолютный адрес, символ (если есть), блок, в котором задействован адрес, как он использовался (чтение или запись) и связанная с языком информация. Для STL столбец с языковой информацией отображает сеть, в которой использовался адрес; для SCL – номер строки и столбца. Щелкните на заголовке столбца, чтобы отсортировать таблицу по содержимому столбца.

Если адрес выбран, команда меню Edit → Go To → Line (Правка → Перейти к → Строка) запускает редактор программ и отображает адрес в программной среде.

Список перекрестных ссылок показывает адреса, выбранные вами при помощи View → Filter... (Вид → Фильтр...), например, память меркеров. По двойному щелчку на адресе редактор открывает блок, отображенный в той строке, в которую занесен адрес. К тому же STEP 7 использует фильтр, сохраненный как «Standard» («стандартный»), каждый раз, когда открывает список перекрестных ссылок.

Преимущество: перекрестные ссылки показывают, были ли также отсканированы или сброшены ссылочные адреса. Кроме того, они показывают, в каких блоках используются адреса (возможно более одного раза).

Назначения

Список ссылок I/Q/M показывает, какие биты в областях адресов I, Q и M назначены в программе. В каждой строке отображается один байт, разбитый на биты. Также показано, как осуществляется доступ – побайтно, по слову или по двойному слову. Список ссылок T/C показывает таймеры и счетчики, задействованные в программе. В строке отображаются десять таймеров или счетчиков.

Преимущество: список демонстрирует, были ли определенные области адресов (неправильно) назначены или где еще есть доступные адреса.

Структура программы

Структура программы показывает иерархию вызовов блоков в пользовательской программе. «Начальный блок» для иерархии вызовов определяется посредством установок фильтра. Вы можете выбрать между двумя различными представлениями: *Древовидная структура* показывает все вложенные уровни вызовов блоков. Управление отображением вложенных уровней осуществляется с помощью квадратов со знаками «+» и «-». Показаны потребности во временных локальных данных для полного пути, следующего за начальным блоком, и/или на каждый путь вызова. Щелкните правой кнопкой мыши для вызова меню, в котором вы можете открыть блок, переключиться в область вызова или в окно с дополнительной информацией о блоке.

Структура предок/потомок показывает два уровня вызовов с одним вызовом блока. Связанная с языком информация также включена.

Преимущество: какие блоки были задействованы? Все ли запрограммированные блоки были вызваны? Каковы потребности блоков во временных локальных данных? Достаточно ли заданных требований по локальным данным на приоритетный класс (на организационный блок)?

Неиспользуемые символы

Этот список отображает все адреса, которые размещены в таблице символов, но не используются в программе. В списке приведены символ, адрес, тип данных и комментарий из таблицы символов.

Преимущество: какие адреса из списка были нечаянно забыты в начале написания программы? Или они, возможно, являются лишними и не нужны?

Адреса без символов

В этом списке собраны все используемые в программе адреса, которым символы не назначены. В списке приводятся эти адреса и как часто они использовались.

Преимущество: были ли адреса использованы ошибочно (случайно или из-за опечатки)?

2.5.7 Мультиязыковые комментарии и отображение текста

SIMATIC-менеджер позволяет управлять несколькими языковыми версиями комментариев и текста на дисплее.

Выбор языка комментариев и отображаемого текста влияет на тексты, вводимые пользователем. Язык сеанса, назначенный для меню и сообщений об ошибках, например, устанавливается STEP 7 и выбирается в SIMATIC-менеджере во вкладке «Language» («Язык») после нажатия Options → Customize (Опции → Настроить). На

этой вкладке также устанавливается мнемоника, то есть язык, который STEP 7 использует для операндов и операций. Все три установки независимы друг от друга.

Общая процедура

Вы ввели тексты на исходном языке, например, английском и хотите сгенерировать немецкую версию вашей программы. Чтобы это сделать, экспортируйте требуемый текст или типы текстов. Экспортированный файл имеет расширение *.csy, и вы можете отредактировать его, к примеру, в Microsoft Excel. Можно ввести перевод для каждого текста. Импортируйте законченную таблицу перевода обратно в ваш проект. Теперь вы можете переключать языки. Это можно сделать с несколькими языками.

Экспорт и импорт текстов

В SIMATIC-менеджере выберите объекты, содержащие комментарии, которые вы хотите перевести, например, таблицу символов, контейнер блоков, несколько блоков или один блок. Выберите команду меню Options → Manage Multilingual Texts → Export (Опции → Управление мультиязыковыми текстами → Экспорт). В появившемся диалоговом окне введите местоположение для сохранения экспортированного файла и требуемый язык. Выберите типы текстов, которые вы хотите перевести (таблица 2.2).

Для каждого текстового типа генерируется отдельный файл, к примеру, файл SymbolComment.csv для комментариев из таблицы символов.

Откройте экспортированный файл(ы) в Microsoft Excel с помощью диалогового окна File → Open (Файл → Открыть) (не двойным щелчком клавишей мыши). Экспортированные тексты отображаются в первом столбце, а перевод вы можете ввести во втором столбце.

Ввести переведенные тексты обратно в проект вы можете с помощью Options → Manage Multilingual Texts → Import (Опции → Управление мультиязыковыми текстами → Импорт). Регистрационный файл (log-файл) предоставляет информацию об импортированном файле и любых ошибках, которые могут возникнуть.

Обратите внимание, что имя импортируемого файла не должно изменяться, так как имеется прямая связь между именем и содержащимися в файле текстовыми типами.

Таблица 2.2 Текстовые Типы переводимых текстов (выбор)

Тип текста	Значение
BlockTitle	Заголовок блока
BlockComment	Комментарий блока
NetworkTitle	Заголовок сегмента (сети, схемы)
NetworkComment	Комментарий к сегменту (сети, схемы)
LineComment	Комментарий строки
InterfaceComment	Комментарий в ➤ таблице описаний блоков кода ➤ блоках данных ➤ пользовательских типах данных UDT
SymbolComment	Комментарий символа

Выбор и удаление языка

В SIMATIC-менеджере вы можете переключаться на любой импортированный язык при помощи Options → Manage Multilingual Texts → Change Language (Опции → Управление мультязыковыми текстами → Сменить язык). Смена языка выполняется для объектов (блоков, таблицы символов), для которых были импортированы соответствующие тексты. Эта информация содержится в log-файле.

Импортированные языки можно удалить с помощью команды меню Options → Manage Multilingual Texts → Delete Language (Опции → Управление мультязыковыми текстами → Удалить язык).

2.6 Онлайновый (интерактивный) режим

Вы создаете конфигурацию аппаратного обеспечения и пользовательскую программу на устройстве программирования, обычно называемом «инженерная система» («engineering system», ES). Программа S7 автономно сохранена здесь на жестком диске, а также в скомпилированном виде.

Чтобы передать программу в CPU, вы должны подключить к нему устройство программирования. Вы устанавливаете «онлайновое» соединение. Этим соединением можно воспользоваться для определения рабочего (операционного) состояния CPU и назначенных модулей, то есть выполнить функции диагностики.

2.6.1 Подключение PLC

Для онлайн-соединения требуется механическое соединение между MPI-интерфейсом программирующего устройства и MPI-интерфейсом CPU. Соединение является единственным, когда CPU – единственный подключенный программируемый модуль. Если в подсети MPI имеется несколько CPU, каждому из них должен быть присвоен уникальный узловой номер (MPI-адрес). Установка MPI-адреса происходит при инициализации CPU. Пред объединением всех CPU в одну сеть, подключите устройство программирования к одному CPU и передайте объект *System Data* (*Системные данные*) из автономного *Blocks* (*Блоки*) пользовательской программы или направьте с помощью редактора Hardware Configuration, используя команду меню PLC → Download (PLC → Загрузить). Это назначит CPU его собственный уникальный MPI-адрес («присвоение имени») наряду с другими свойствами.

MPI-адрес CPU в MPI-сети может быть изменен в любой момент путем передачи в CPU новой записи данных параметризации, содержащей новый MPI-адрес. Обратите особое внимание на то, что новый MPI-адрес вступает в силу немедленно. Так как программирующее устройство сразу устанавливает новый адрес, вы должны адаптировать другие приложения, такие как коммуникации глобальных данных, к новому MPI-адресу.

MPI-параметры сохраняются в CPU даже после сброса памяти. Таким образом, к CPU можно обращаться и после сброса памяти. Устройство программирования всегда может работать в онлайн-режиме в CPU, даже с независимой от модулей программой и без загруженного проекта.

Если нет загруженного проекта, вы должны установить соединение с CPU с помощью PLC → Display Accessible Nodes (PLC → Отобразить доступные узлы). Эта команда отобразит окно проекта со структурой «*Accessible Nodes*» – «Module (MPI=n)» – «Online User Program (*Blocks*)» или «*Доступные узлы*» – «Модуль (MPI=n)» – «Онлайновая программа пользователя (*Блоки*)». При выборе объекта *Module* (*Модуль*) вы можете использовать интерактивные функции, такие как изменение рабочего режима и проверка состояния модуля. Выбор объекта *Blocks* (*Блоки*) отобразит блоки в пользовательской памяти CPU. И вы можете редактировать (изменять, удалять, вставлять) отдельные блоки.

Вы можете считать обратно системные данные из подключенного CPU с целью, скажем, продолжения работы на основе имеющейся конфигурации без наличия соответствующего проекта в системе управления данными программирующего устройства. Создайте новый проект в SIMATIC-менеджере, отметьте проект и нажмите PLC → Upload Station (PLC → Выгрузить станцию). После определения нужного CPU в появляющемся диалоговом окне онлайнные системные данные сгружаются на жесткий диск.

Если в проектном окне имеется **независимая от CPU программа**, создайте соответствующее (ассоциированное) онлайнное окно проекта. Если к MPI-сети подключены и доступны несколько CPU, при отмеченной S7-программе выберите Edit → Object Properties (Правка → Свойства объекта) и на вкладке «Addresses Module» («Адреса модуля») установите номер монтажной стойки и слота CPU.

Если вы в онлайнном окне выберите *S7 Program (S7-программу)*, все онлайнные функции подключенных CPU будут для вас доступны. *Blocks (Блоки)* показывает блоки, расположенные в пользовательской памяти CPU. Если блоки в автономной программе согласованы с блоками онлайнной программы, то вы можете отредактировать блоки в пользовательской памяти, используя информацию из системы управления данными устройства программирования (символический адрес, комментарии).

Когда вы переводите **назначенную CPU программу** в онлайнный режим, вы можете модифицировать ее, как если бы это была независимая от CPU программа. Кроме того, теперь вы можете сконфигурировать SIMATIC-станцию, то есть установить параметры и адрес CPU и параметризовать модули.

2.6.2 Защита программы пользователя

С помощью соответствующим образом оснащенных CPU доступ к программе пользователя может быть защищен паролем. Каждый, кто владеет паролем, имеет неограниченный доступ к пользовательской программе. Для тех, кто пароля не знает, вы можете определить три уровня защиты. Установка уровней защиты производится в Hardware Configuration, на вкладке «Protection» («Защита») во время параметризации CPU.

Уровень защиты 1: блокировочное положение переключателя

Данный уровень защиты установлен по умолчанию (без пароля). При этом пользовательская программа защищена переключателем выбора режимов, который расположен на лицевой стороне центрального процессора. В положениях RUN-P (исполнение-P) и STOP вы имеете неограниченный доступ к пользовательской программе; в положении RUN возможен только доступ для чтения через программирующее устройство. В этом положении вы можете вынуть блокирующий переключатель, так что с помощью переключателя нельзя будет изменить режим.

Вы можете обойти защиту положением RUN блокирующего переключателя путем выбора опции «Can be revoked with password» («Может быть отменен паролем»), к

примеру, если к CPU, а вместе с ним и к блокирующему переключателю доступ затруднен или они находятся на расстоянии.

Уровень защиты 2: защита от записи

На этом уровне защиты программу пользователя можно только прочитать, несмотря на положение блокирующего переключателя.

Уровень защиты 2: защита от чтения/записи

Доступа к пользовательской программе нет, несмотря на положение блокирующего переключателя.

Защита паролем

Если вы выберете уровень защиты 2 или 3 или уровень 2 с «Can be revoked with password», вы получите запрос на задание пароля. Длина пароля может быть до 8 символов.

При попытке получить доступ к пользовательской программе, защищенной паролем, будет выведен запрос на ввод пароля. Перед доступом к защищенному CPU также можно ввести пароль через PLC → Access Rights (PLC → Права доступа). Сначала выберите соответствующий CPU или S7-программу.

В диалоговом окне «Enter Password» («Введите пароль») вы можете выбрать опцию «Use password for other protected modules» («Использовать пароль для других защищенных модулей») для того, чтобы получить доступ ко всем модулям, защищенным тем же паролем.

Авторизация доступа с паролем действует до завершения последнего S7-приложения.

Каждый, кто знает пароль, имеет неограниченный доступ к пользовательской программе в CPU, несмотря на установленный уровень защиты и положения блокировки.

2.6.3 Информация CPU

В онлайн-режиме доступна информация CPU, приведенная ниже. Задействованы следующие команды меню при выбранном модуле (в режиме онлайн и без проекта) или программе S7 (в онлайн-режиме проектного окна).

- PLC → Diagnose Hardware (PLC → Диагностировать аппаратуру)
Обратитесь к параграфу 2.7.1 «Диагностирование аппаратных средств».

- PLC → Module Information (PLC → Информация о модуле)
Общая информация (версия, например), диагностический буфер, память (текущее отображение рабочей и загрузочной памяти, сжатие), время цикла (длительность последнего, самый продолжительный и самый короткий программный цикл), система отсчета времени (свойства часов CPU, синхронизация часов, счетчик рабочего времени), рабочие параметры (конфигурация памяти, размеры адресных областей, число доступных блоков, SFC и SFB), коммуникация (скорость передачи данных и коммуникационные связи), стеки в состоянии STOP (В-стек, I-стек и L-стек).
- PLC → Operating Mode (PLC → Рабочий режим)
Отображает текущий рабочий режим (к примеру, RUN или STOP), изменение рабочего режима.
- PLC → Clear/Reset (PLC → Очистить/Сбросить)
Очистка (сброс) памяти CPU, перевод его в режим STOP.
- PLC → Set Time and Date (PLC → Установка времени и даты)
Установка внутренних часов CPU.
- PLC → CPU Messages (PLC → Сообщения CPU)
Отчет об асинхронных системных ошибках и об определенных пользователем сообщениях, сгенерированных в программе с помощью SFC 52 WR_USMSG, SFC 18 ALARM_S и SFC 17 ALARM_SQ.
- PLC → Display Force Values (PLC → Отобразить принудительные значения), PLC → Monitor/Modify Variables (PLC → Наблюдение/модифицирование переменных)
(Обратитесь к параграфам 2.7.3 «Наблюдение и модифицирование переменных» и 2.7.4 «Принудительная установка переменных»).

2.6.4 Загрузка пользовательской программы в CPU

При передаче пользовательской программы (скомпилированные блоки и конфигурационные данные) в CPU она грузится в загрузочную память CPU. Физически загрузочная память может быть модулем RAM или флэш-EPROM, и либо встроенной в CPU либо на (дополнительной) карте памяти.

Если используется карта памяти флэш-EPROM, то вы можете записывать на нее в программирующем устройстве и использовать ее в качестве среды данных. Подключение карты памяти к CPU происходит в выключенном состоянии; после включения питания и обнуления памяти соответствующие данные карты памяти передаются в рабочую память CPU. С соответствующим образом оснащенными CPU можно также перезаписать карту флэш-EPROM, если она подключена к CPU, но только записав на нее программу полностью.

В случае загрузочной памяти RAM передача всей программы пользователя осуществляется путем переключения CPU в состояние STOP, выполнения сброса памяти и собственно передачи пользовательской программы. Конфигурационные данные так-

же перемещаются. При обнулении памяти или отключении резервной батареи программа в RAM теряется.

Если вы хотите только изменить конфигурационные данные (свойства CPU, сконфигурированные соединения, GD-коммуникации, параметры модулей и так далее), вам потребуется лишь загрузить объект *System Data* (*Системные данные*) в CPU (выберите объект и передайте его с помощью команды меню PLC → Download - PLC → Загрузить). Параметры CPU вступают в силу немедленно; CPU передает параметры оставшимся модулям во время запуска.

Обратите внимание, что конфигурация полностью загружается в PLC с объектом *System Data* (*Системные данные*). Если вы используете PLC → Download (PLC → Загрузить) в приложении, например, в коммуникациях глобальных данных, передаются только данные, редактируемые приложением.

Замечание: для загрузки сжатого архивного файла выберите PLC → Save Project on Memory Card (PLC → Записать проект на карту памяти) (смотрите параграф 2.2.2 «Управление, перегруппировка, архивирование»). Проект в архивном файле не может быть отредактирован ни с помощью устройства программирования, ни из CPU.

2.6.5 Обработка блоков

Передача блоков

В случае загрузочной памяти RAM в дополнение к передаче всей программы в онлайн-режиме вы можете также модифицировать, удалять или перезагружать отдельные блоки.

Передача отдельных блоков в CPU начинается с их выбора в автономном окне и нажатия PLC → Download (PLC → Загрузить). При одновременно открытых автономном и онлайн-окнах вы можете перетащить блоки из одного окна и поместить их в другое.

Особое внимание требуется при передаче отдельных блоков во время работы. Если в блоке вызываются блоки, отсутствующие в данный момент в памяти, то CPU перейдет в состояние STOP, поэтому сначала вы должны загрузить блоки «нижнего уровня». Это также относится к блокам данных, чьи адреса используются в загруженном блоке. Загрузка блока «высшего уровня» происходит в последнюю очередь. Тогда, после вызова, он будет немедленно исполнен в следующем программном цикле.

SIMATIC-менеджер также предоставляет вам возможность в SCL передать отдельные блоки или программу целиком из автономного контейнера *Blocks* (*Блоки*) в CPU. Нет смысла перемещать их обратно из CPU на жесткий диск, так как скомпилированные блоки не могут редактироваться в редакторе SCL. Вы можете лишь отредактировать исходный файл SCL-программы и, сформировав его, сгенерировать скомпилированные блоки.

Модифицирование блоков в онлайнном режиме

Вы можете пошагово редактировать блоки с помощью STL в онлайнной программе пользователя (в CPU), точно так же, как в автономной пользовательской программе. Однако если онлайнная и автономная обработки данных отличаются, это может привести к тому, что в редакторе нельзя будет отобразить дополнительную информацию об автономной базе данных; в таком случае эти данные могут быть потеряны (символические идентификаторы, метки переходов, комментарии, определенные пользователем типы данных).

Блоки, модифицированные в онлайнном режиме, лучше хранить автономно на жестком диске, чтобы избежать несовместимости данных (например, «конфликт временных меток», при котором интерфейс вызванного блока имеет более позднюю метку времени, чем программа в вызывающем блоке).

Удаление блоков

Если загрузочная память состоит исключительно из RAM, блоки можно модифицировать и удалять. Если пользовательская программа расположена на флэш-EPROM, модули также можете редактировать и удалять, что обеспечивается достаточной емкостью дополнительно доступной RAM. Блоки во флэш-EPROM отмечаются как «неверные» («invalid»). Тем не менее, после сброса памяти или включения питания без резервной батареи блоки вновь передаются из загрузочной памяти флэш-EPROM в рабочую память.

Очистить карту памяти флэш-EPROM можно только в устройстве программирования.

Сжатие

Когда вы загружаете новый или измененный блок в CPU, центральный процессор помещает блок в загрузочную память и передает соответствующие данные в рабочую память. Если блок с таким же номером уже имеется, то данный «старый блок» объявляется неверным (после запроса на подтверждение), а новый блок «добавляется в конец» памяти. Даже удаленный блок «только лишь» объявляется неверным, а не удаляется фактически из памяти.

Результатом этого становятся пробелы в пользовательской памяти, которые все больше и больше уменьшают объем доступной памяти. Эти пробелы могут быть заполнены только с помощью функции *Compress* (Сжатие). Когда вы производите сжатие в режиме RUN, исполняемые в текущий момент блоки не перемещаются; истинное сжатие без пробелов может быть достигнуто только в режиме STOP.

Текущее распределение памяти может быть отображено в процентах по команде меню PLC → Module Information (PLC → Информация о модуле), на вкладке *Memory* (Память). Диалоговое окно, которое затем появляется, также снабжено кнопкой для предварительного сжатия.

Вы можете инициировать сжатие, зависящее от событий, через программу с вызовом SFC 25 COMPRESS.

Блоки данных в режимах офлайн/онлайн

Адреса данных в блоках данных могут быть назначены *начальному значению* (*initial value*) и *фактическому значению* (*actual value*) (смотрите также параграф 3.4 «Программирование блоков данных»). Если блок данных загружен в CPU, начальные значения передаются загрузочную память, а фактические – в рабочую. Каждое изменение значения по отношению к адресу данных, сделанное через программу, соответствует изменению фактического значения.

При загрузке из CPU блока данных его значения берутся из рабочей памяти, в которой находятся все фактические значения. Вы можете просмотреть фактические значения в момент их считывания по команде View → Data View (Вид → Просмотр данных). Если вы изменяете фактическое значение в блоке данных и записываете блок обратно в CPU, измененное значение помещается в рабочую память.

Когда в качестве загрузочной памяти используется карта памяти флэш-EPROM, блоки, находящиеся на карте памяти, передаются в рабочую память после сброса памяти CPU. Блоки данных сохраняют начальные значения, изначально запрограммированные для них. То же происходит при включении питания без резервной батареи. В S7-300 вы можете защитить отдельные области в блоках данных, объявив ее сохраняемыми.

Блок данных, сгенерированный со свойством UNLINKED (несвязанный), в рабочую память не передается; он остается в загрузочной памяти. Блок данных с этим свойством может быть прочитан только при помощи SFC 20 BLKMOV.

2.7 Тестирование программы

После установки соединения с CPU и загрузки пользовательской программы вы можете протестировать (отладить) программу, как всю целиком, так и частично, к примеру, отдельные блоки. Вы должны инициализировать переменные с использованием сигналов и значений, например, с помощью модулей симулятора, и оценить возвращаемую вашей программой информацию. Если CPU переходит в состояние STOP в результате ошибки, то помощь в поиске источника ошибки может оказать, наряду с другим, информация CPU.

Большие программы отлаживаются по частям. Если, к примеру, вы хотите отладить только один блок, загрузите этот блок в CPU и вызовите его в OB 1. Если OB 1 организован таким образом, что программа может быть отлажена раздел за разделом «от начала до конца», то для отладки вы можете выбрать блоки или разделы программы, используя функции перехода, чтобы пропустить вызовы или программные разделы, которые отладки не требуют.

С помощью дополнительного программного обеспечения PLCSIM можно на устройстве программирования смоделировать CPU и таким образом отладить вашу программу без дополнительных аппаратных средств.

2.7.1 Диагностирование аппаратных средств

В случае сбоя вы можете получить диагностическую информацию о сбойных модулях с помощью функции «Diagnose Hardware» («Диагностика аппаратуры»). Программирующее устройство нужно подключить к шине MPI и запустить SIMATIC-менеджер.

Если в базе данных устройства программирования доступен проект, ассоциированный с конфигурацией предприятия, откройте онлайнное окно проекта командой меню View → Online (Вид → Онлайн). В противном случае нажмите PLC → Display Accessible Nodes (PLC → Отобразить доступные узлы) и выберите CPU. Теперь вы можете получить быстрый доступ к просмотру сбойных модулей с помощью PLC → Diagnose Hardware (PLC → Диагностировать аппаратуру) (по умолчанию). Программа Hardware Configuration во время онлайнного просмотра предоставляет подробную диагностическую информацию о модулях; этот режим может быть установлен в SIMATIC-менеджере на вкладке «View» («Просмотр») после выбора пункта меню Options → Customize (Опции → Настроить).

Информацию о статусе и рабочем состоянии модулей, доступных в онлайнном режиме, вы можете получить в виде просмотра проекта (отображение ошибок, сообщенных станцией), просмотра станции (ошибки, сообщенные модулями) и просмотра модуля (отображение доступной диагностической информации).

2.7.2 Определение причины перехода в состояние STOP

Если CPU из-за возникновения ошибки переходит в состояние STOP, первая предпринимаемая мера для определения причины перехода в STOP – вывод диагностического буфера. CPU помещает в диагностический буфер все сообщения, в том числе причину перехода в STOP и ошибки, приведшие к этому. Чтобы вывести диагностический буфер, переключите PG в режим онлайн, отметьте S7-программу и выберите вкладку *Diagnostic Buffer* (*Диагностический буфер*) с помощью команды меню PLC → Module Information (PLC → Информация о модуле). Последнее событие (оно имеет номер 1) вызвало переход в состояние STOP, например, «STOP because programming error OB not loaded» («STOP, потому что не загружен ОБ программных ошибок»). Ошибка, приведшая к STOP, описывается в предыдущем сообщении, например, «FC not loaded» («Не загружен FC»). Щелкнув на номере сообщения, вы можете вызвать дополнительные комментарии в следующем расположенном ниже поле дисплея. Если сообщения связаны с программными ошибками в блоке, то с помощью кнопки «Open Block» («Открыть блок») вы можете открыть и отредактировать блок.

Если состояние STOP вызвано, к примеру, программной ошибкой, вы можете выяснить сопутствующие обстоятельства на вкладке *Stacks* (*Стеки*). На открытой вкладке *Stacks* (*Стеки*) вы увидите В-стек (стек блоков), который показывает вам путь вызовов всех незавершенных блоков вплоть до блока, содержащего точку прерывания. Воспользуйтесь кнопкой «I-stack» («I-стек») для того, чтобы отобразить стек прерываний, который покажет вам содержимое регистров CPU (аккумуляторов, адресного регистра, регистра блока данных, слова статуса) в точке прерывания в момент обнаружения ошибки. L-стек (стек локальных данных) показывает временные локальные данные блока, который вы выбрали с помощью мыши в В-стеке.

2.7.3 Наблюдение и модифицирование переменных

Превосходным ресурсом для отладки программ пользователя является наблюдение и модифицирование переменных с помощью таблиц переменных VAT. Могут быть отображены сигнальные состояния или значения переменных простых типов данных. Если у вас есть доступ к пользовательской программе, вы также можете модифицировать переменные, то есть изменить сигнальное состояние или присвоить новые значения.

Внимание: вы должны быть уверены в том, что в результате модифицирования переменных не возникнет опасных состояний.

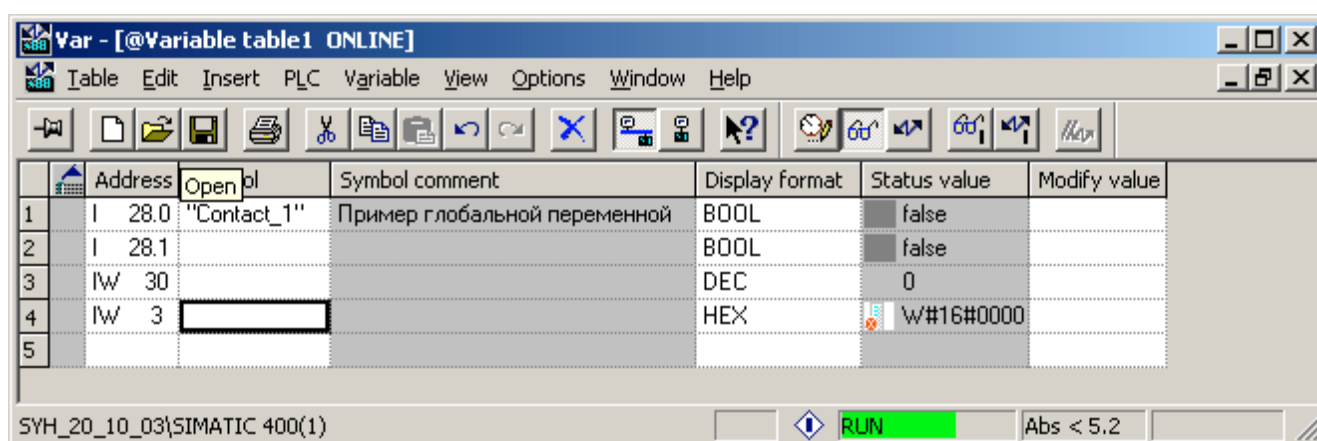
Создание таблицы переменных

Для наблюдения и изменения переменных вы должны создать таблицу переменных VAT, содержащую переменные и ассоциированные форматы данных. Можно сгенерировать до 255 таблиц переменных (от VAT 1 до VAT 255) и присвоить им имена в таблице символов. Максимальный размер таблицы переменных составляет 1024 строки длиной до 255 знаков (рисунок 2.10).

Вы можете сгенерировать VAT автономно, выбрав *Blocks (Блоки)* пользовательской программы и команду меню Insert → S7 Block → Variable Table (Вставка → Блок S7 → Таблица переменных), также можно сгенерировать VAT без имени в онлайн-режиме путем выбора *S7 Program (S7-программа)* и команды меню PLC → Monitor/Modify Variables (PLC → Наблюдение/модифицирование переменных).

Вы можете определить переменные либо по абсолютным, либо по символическим адресам и выбрать тип данных (формат отображения), с использованием которого переменная будет отображена и модифицирована (по команде меню View → Display Format – Вид → Отобразить формат или щелкнув правой кнопкой мыши прямо на «Display Format» или «Формате отображения»).

Используйте строки комментариев, чтобы дать заглавие определенным разделам таблицы. Также можно задать, какие столбцы должны быть отображены. Вы можете изменить переменную или формат отображения, добавить или удалить строки в любое время. Таблица переменных сохраняется по команде меню Table → Save (Таблица → Сохранить) в контейнере объектов *Blocks (Блоки)*.



	Address	Open pl	Symbol comment	Display format	Status value	Modify value
1	I 28.0	"Contact_1"	Пример глобальной переменной	BOOL	false	
2	I 28.1			BOOL	false	
3	IW 30			DEC	0	
4	IW 3			HEX	W#16#0000	
5						

SYN_20_10_03\SIMATIC 400(1) RUN Abs < 5.2

Рисунок 2.10 Пример таблицы переменных

Установка онлайн-соединения

Для работы с таблицей переменных, созданной автономно, переключите ее в онлайн-режим с помощью PLC → Connect To → ... (PLC → Подключить к → ...). Вы должны переключить каждую отдельную VAT в онлайн-режим, и вы можете вновь разъединить соединение по команде PLC → Disconnect (PLC → Отключить).

Условия триггера (пуска)

Находясь в таблице переменных, выберите Variable → Trigger (Переменная → Триггер), чтобы установить триггерную точку и условия триггера отдельно для наблюдения и модифицирования. Триггерная точка – это точка, в которой CPU считывает значения из системной памяти или записывает значения в системную память. Вы

должны определить, как будут осуществлены чтение и запись – один раз или периодически.

Если наблюдение и модифицирование имеют одинаковые условия триггера, то наблюдение выполняется перед модифицированием. Если вы для изменения выбираете триггерную точку «Start of circle» («Начало цикла»), переменные модифицируются после обновления входного образа процесса и перед вызовом ОБ 1. Если для наблюдения вы выбираете триггерную точку «End of circle» («Конец цикла»), значения статуса отображаются после завершения ОБ 1 и перед выводом выходного образа процесса.

Наблюдения переменных

Активируйте функцию наблюдения командой меню Variable → Monitor (Переменная → Наблюдение). Переменные в VАТ обновляются в соответствии с заданными условиями триггера. Постоянное наблюдение позволяет отслеживать изменения значений на экране. Значения отображаются в формате данных, установленном вами в столбце «Display format» («Формат отображения»). Клавиша ESC завершает постоянную функцию наблюдения.

Variable → Update Monitor Values (Переменная → Обновить наблюдаемые значения) обновляет отслеживаемые значения один раз и немедленно, несмотря на установленные условия триггера.

Модифицирование переменных

Для передачи заданных значений в CPU, зависящей от условий триггера, воспользуйтесь командой Variable → Modify (Переменная → Модифицирование). Значения вводите только в строках, содержащих переменные, которые вы хотите изменить. Вы можете расширить комментарий для значения при помощи «//» или Variable → Modify Values as comment (Переменная → модифицировать значения как комментарий); эти значения не участвуют в модифицировании. Вы должны определить значения в формате данных, заданном в столбце «Display format» («Формат отображения»).

Изменяются только значения, отображенные при запуске функции модифицирования. Клавиша ESC завершает постоянную функцию модифицирования.

Variable → Activate Modify Values (Переменная → Активировать модифицированные значения) передает измененные значения только один раз и немедленно, несмотря на установленные условия триггера.

2.7.4 Принудительная установка переменных (функция Force)

С помощью соответствующим образом оснащенных CPU вы можете задать фиксированные значения для определенных переменных. Пользовательская программа не

сможет изменить эти значения («принудительная установка»). Принудительная установка допустима в любом режиме работы CPU и выполняется немедленно.

Внимание: вы должны быть уверены в том, что в результате принудительной установки переменных не возникнет опасных состояний.

Стартовой точкой для принудительной установки является таблица переменных (VAT). Создайте VAT, введите адреса для принудительной установки значений и установите соединение с CPU. Выбрав Variable → Display Force Values (Переменные → Отобразить принудительные значения), можно открыть окно, содержащее принудительные значения.

Если в CPU принудительные значения уже активизированы, то они выделены в окне принудительных установок значений жирным шрифтом. Теперь вы можете переместить некоторые или все адреса из таблицы переменных в окно принудительных установок или добавить новые адреса. Сохранение содержимого окна принудительных установок в VAT осуществляется по команде меню Table → Save As (Таблица → Сохранить как).

Принудительные значения допустимы для следующих областей адресов:

- Входы I (образ процесса)
(S7-300 и S7-400)
- Выходы Q (образ процесса)
(S7-300 и S7-400)
- Периферийные входы PI
(только S7-400)
- Периферийные выходы PQ
(S7-300 и S7-400)
- Память меркеров M
(только S7-400)

Задание принудительной установки начинается с Variable → Force (Переменная → Присвоить принудительное значение). CPU принимает принудительные значения и изменений принудительно установленных адресов больше не допускает.

Пока активна функция принудительной установки, выполняется следующее:

- Все обращения для чтения к принудительно заданным адресам через пользовательскую программу (например, загрузка) и через системную программу (например, обновление образа процесса) всегда выдают принудительно установленное значение.
- В S7-400 все обращения для записи к принудительно заданным адресам через пользовательскую программу (например, передача) и через системную программу (например, через SFC – последовательные функциональные схемы) остаются

безрезультатными. В S7-300 пользовательская программа может переписать установленные принудительно значения.

Принудительная установка в S7-300 аналогична циклическому модифицированию: после обновления образа входов процесса CPU переписывает входы принудительными значениями; перед выводом образа выходов процесса CPU переписывает выходы принудительными значениями.

Примечание: принудительная установка не завершается закрытием окна принудительных значений или таблицы переменных или прерыванием соединения с CPU! Удалить задание принудительной установки можно только с помощью Variable → Delete Force (Переменная → Снять принудительную установку).

Принудительная установка также снимается путем сброса памяти или в случае сбоя в электропитании, при условии, что CPU не снабжено резервной батареей. По завершению принудительной установки адреса сохраняют принудительные значения до их перезаписи либо программой пользователя, либо системной программой.

Принудительная установка эффективна только на входах/выходах, назначенных CPU. Если после рестарта установленные принудительно периферийные входы и выходы более не назначены (к примеру, в результате повторной параметризации), то соответствующие периферийные входы и выходы не будут принудительно заданы.

Обработка ошибок

Если размер области, к которой обращаются при чтении, больше размера области, устанавливаемой принудительно (например, принудительно установленный байт в слове), не устанавливаемый компонент значения адреса читается как обычно. Если здесь возникает ошибка синхронизации (ошибка доступа или размера области), то считывается «значение, заменяющее ошибочное» («error substitute value»), определяемое программой пользователя или центральным процессором, или CPU переходит в состояние STOP.

Если при записи размер области, к которой происходит обращение, больше размера устанавливаемой принудительно области (например, принудительно установленный байт в слове), не устанавливаемый компонент значения адреса записывается как обычно. Ошибочный доступ при записи оставляет принудительно устанавливаемый компонент адреса без изменений, то есть ошибка синхронизации не отменяет защиты от записи.

Загрузка принудительно установленных периферийных входов выводит принудительно установленное значение. Если размер области, к которой происходит обращение, согласуется с устанавливаемой принудительно областью, то модули входов, в которых произошел сбой, или которые (еще) не подключены, могут быть «заменены» принудительным значением.

Вход I в образе процесса, который принадлежит принудительно установленному периферийному входу PI, принудительно не устанавливается; он предварительно не

назначается и все еще доступен для перезаписи. Когда обновляется образ процесса, вход получает принудительное значение периферийного входа.

Когда принудительно устанавливается периферийные выходы PQ, соответствующие выходы Q в образе процесса не обновляются и принудительно не устанавливаются (принудительная установка действует «внешне» только на выходы модулей). Выходы Q сохраняются и доступны для перезаписи; чтение выходов выводит записанное значение (не принудительное значение). Если модуль выхода принудительно установлен и если в нем возникает сбой или он удаляется, то он вновь немедленно получит принудительное значение при переподключении.

Модули выходов выдают сигнальное состояние «0» или заменяют значение на сигнал OD (блокировка модулей выходов в режиме STOP, HOLD или RESTART – ожидания, ожидания или рестарта) – даже если периферийные выходы принудительно установлены (исключение: аналоговые модули без определения OD продолжают выдавать принудительное значение). Если сигнал OD деактивирован, то принудительное значение вновь становится действующим.

Если в состоянии STOP активирована функция *Enable PQ (Активация PQ)*, принудительное значение в режиме STOP также становится действующим (благодаря деактивации сигнала OD). По завершении *Enable PQ (Активация PQ)* происходит обратная установка модулей в «безопасное» («safe») состояние (сигнальное состояние «0» или заменяемое значение); принудительное значение вновь становится действующим на переходе к режиму RUN.

2.7.5 Разблокировка периферийных выходов

В режиме STOP обычно модули выходов деактивированы сигналом OD; с помощью функции разблокировки периферийных выходов вы можете заблокировать сигнал OD и, таким образом, изменять модули выходов, даже если CPU находится в состоянии STOP. Модифицирование осуществляется посредством таблицы переменных. Изменяются только периферийные выходы, назначенные CPU. Возможное применение: тест монтажных соединений выхода в режиме STOP и без программы пользователя.

Внимание: вы должны быть уверены, что разблокировка периферийных выходов не вызовет опасных состояний!

Создайте таблицу переменных и введите периферийные выходы (PQ) и изменяющиеся значения. Переключите таблицу переменных в режим онлайн по команде меню PLC → Connect To → ... (PLC → Подключить к → ...), и, если требуется, остановите CPU, например, командой PLC → Operating Mode (PLC → Рабочий режим) и «STOP».

Блокировка сигнала OD осуществляется выбором пункта меню Variable → Enable Peripheral Outputs (Переменная → Разблокировать периферийные выходы); выходы модуля теперь имеют сигнальное состояние «0», заменяемое значение или принудительное значение. Модифицировать периферийные выходы вы можете с помощью

Variable → Activate Modify Values (Переменная → Активировать модифицированные значения). Вы можете сменить изменяющееся значение и модифицировать снова.

Отключить функцию можно повторным выбором пункта меню Variable → Enable Peripheral Outputs (Переменная → Разблокировать периферийные выходы) или нажатием клавиши ESC. После этого сигнал OD будет разблокирован, выходы модуля устанавливаются в «0», а заменяемое значение или принудительное значение сбрасывается.

Если при активизированной «разблокировке периферийных выходов» действует режим STOP, то все периферийные выходы сняты (удалены), сигнал OD активизирован на переходе к RESTART и деактивирован на переходе к RUN.

2.7.6 Статус программы LAD/FBD

С помощью функции *Program status* (Состояние программы) программный редактор предоставляет дополнительный метод тестирования программы пользователя. Редактор покажет поток двоичного сигнала и цифровых значений в сети (network).

Блок, чью программу требуется отладить, находится в пользовательской памяти CPU, вызывается и редактируется там. Откройте этот блок, к примеру, дважды щелкнув на нем в онлайн-окне SIMATIC-менеджера. Откроется редактор и отобразит программу в блоке.

Отметьте сеть (network), которую вы намереваетесь отладить. Активируйте функцию состояния программы по команде меню Debug → Monitor (Отладка → Наблюдение). Теперь в окне блока можно увидеть поток двоичного сигнала и следить за его изменением. Параметры отображения в программном редакторе (к примеру, цвет) настраиваются на вкладке «LAD/FBD» после вызова Options → Customize (Опции → Настроить). Деактивировать функцию состояния программы можно, вновь нажав Debug → Monitor (Отладка → Наблюдение).

Вы можете установить условия триггера, выбрав команду Debug → Call Environment (Отладка → Параметры вызова). Данная установка потребуется, если отлаживаемый блок вызывается в вашей программе более одного раза. Можно инициировать запись состояния либо путем определения порядка вызовов, либо поставив ее в зависимость от открытого блока данных. Если блок вызывается только один раз, отметьте «No condition» («Без условий»).

Функция состояния программы позволяет модифицировать переменные. Отметьте модифицируемый адрес и выберите Debug → Modify Address (Отладка → Модифицирование адреса).

Для записи информации по статусу программы требуется дополнительное рабочее время в программном цикле. По этой причине для целей отладки вы можете выбрать два рабочих режима: режим отладки и режим процесса. В *режиме отладки* (debug mode) все функции отладки могут использоваться без ограничения. Данный режим можно избрать, например, для отладки блоков без подключения к системе, так как это может значительно увеличить время исполнения цикла. В режиме обработки (process mode) внимание уделяется сведению к минимуму увеличения времени цик-

ла, что приведет к возникновению ограничений отладки, например, в программных циклах (отображаются не все циклические проходы).

В CPU, оснащенных соответствующим образом, установить рабочий режим вы можете во время параметризации CPU на вкладке «Protection» («Защита»). Если режим отладки выбран во время параметризации CPU, то изменить его вы сможете только путем повторной параметризации. В противном случае он может быть изменен в отображаемом диалоговом окне. Установленный режим работы отображается по команде меню Debug → Operation (Отладка → Операция).