

Программное обеспечение

3

3.1 Общие эксплуатационные принципы

Общие сведения Программное обеспечение предназначено для конфигурирования и программирования SIMATIC S7/M7/C7, устройств и соответствует современным эргономическим требованиям.

3.1.1 Требования к установке

Операционная система	Microsoft Windows 95.
Стандартное аппаратное обеспечение	Программаторы или PC со следующими техническими условиями и оборудованием: <ul style="list-style-type: none">• 80486 процессор (или выше)• Минимум 16 Мегабайт RAM (32 Мегабайт рекомендуется)• VGA монитор или другой тип монитора, поддерживаемый Windows 95• Клавиатура, и необязательно, но рекомендуется, мышь,
Требования к свободному месту	Требуемое свободное место на жестком диске: <ul style="list-style-type: none">• Стандартный пакет с одним установленным языком занимает 105 Мегабайт на жестком диске. Точное количество требуемого пространства зависит от количества стандартного установленного программного обеспечения.• STEP 7 должен также иметь приблизительно 64 Мегабайта общей памяти, необходимой для сохранения файла подкачки. Например, если Вы имеете 32 Мегабайта оперативной памяти, Вам будет необходимо дополнительных 32 Мегабайта виртуальной памяти.• Приблизительно 50 мегабайт должны быть доступны для рабочих данных.• Минимум 1 Мегабайт свободной памяти должен быть доступен на жестком диске для установки. (Файлы установки удаляются, по окончании инсталляции.)

3.1.2 Установка программного обеспечения STEP 7

Общие сведения

STEP 7 содержит программу установки, что позволяет проводить инсталляцию автоматически. Подсказки пользователю появляются в экранном руководстве при поэтапном выполнении Вами всего процесса установки.

Авторизация

Особенностью программного продукта является установка авторизации для работы с программным обеспечением STEP 7. Для полноценной работы пакета авторизация должна быть перенесена на тот же диск, где установлен STEP 7.

Эта дискета защиты от тиражирования должна входить в комплект поставки. Эта дискета также содержит программу AUTHORS, которая необходима для визуализации, инсталляции и деинсталляции авторизации STEP 7.

Процедура передачи и снятия авторизации описана в руководстве пользователя /231/.

Примечание

Программаторы Siemens (такие как PG 740) поставляют с уже установленным на жесткий диск программным обеспечением STEP7.

Для полной информации по установке STEP 7, смотри руководство пользователя /231/.

3.1.3 Запуск программного обеспечения STEP 7.

Запуск После запуска Windows 95/NT Вы можете найти значок SIMATIC Manager в пользовательском интерфейсе Windows.

Двойной щелчок на значке “SIMATIC Manager” является самым быстрым способом запустить STEP 7. Этот значок открывает окно SIMATIC Manager. Отсюда Вы имеете доступ к стандартной системе, всем настройкам программного обеспечения и ко всем дополнительным пакетам, которые Вы установили.

С другой стороны, Вы можете загрузить SIMATIC Manager, щелкнув на кнопке “Start” панели задач Windows 95/NT. Заголовок меню для него находится в “Simatic/STEP 7.”

SIMATIC Manager SIMATIC Manager является начальным пользовательским окном для конфигурирования и программирования. Здесь Вы можете сделать следующее:

- Установить проекты
- Конфигурировать и назначать параметры для аппаратных средств
- Конфигурировать коммуникации связи
- Создавать программы
- Проверять программы и затем их загружать

Доступ ко всем функциям, объектноориентированный, интуитивный и простой в обучении.

Вы можете работать с SIMATIC Manager следующими способами:

- OFFline режим (без подключения к контроллеру), или
- ONline режим (с подключением к контроллеру)

(При работе ONline, убедитесь, что соблюдены соответствующие рекомендации по безопасности.)

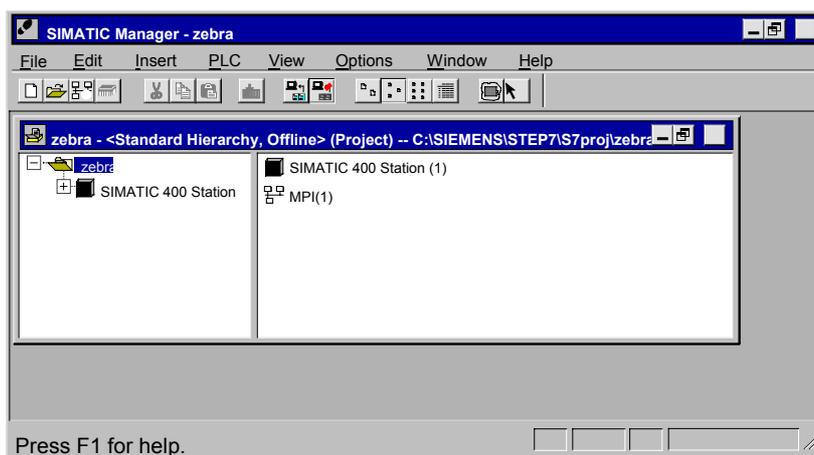


Рис.3-1 SIMATIC Manager с открытым проектом

3.2 Структура проекта S7

Описание	Проекты содержат все данные и программы для устройства автоматизации. Их цель состоит в том, чтобы обеспечить организованное хранение данных и программ, созданных для таких приложений.
Проекты в STEP 5	<p>Вы уже знакомы с термином "проект" при работе с STEP 5. В STEP 5 проект содержит все файлы STEP 5, созданные для одной пользовательской программы, в проектном файле.</p> <p>Этот проектный файл содержит информацию, необходимую для удобного редактирования и обслуживания пользовательской программы, типа параметров настройки, а так же каталоги и имена файлов.</p>
Проекты в STEP 7	В STEP 7, проект содержит все программы и данные, необходимые для решения задач автоматизации, независимо от применяемых центральных процессоров и являются ли они сетевыми. Таким образом, проект уже не ограничен пользовательской программой, используемой для одного программируемого модуля; вместо этого он содержит несколько пользовательских программ, используемых для нескольких программируемых модулей, которые объединены вместе под общим названием проекта.
Замечание	<p>Как и в STEP 5, в STEP 7 также можно создавать простую пользовательскую программу, предназначенную только для одного центрального процессора. В этом случае проект ограничен одним центральным процессором.</p> <p>В следующем разделе рассматривается структура каталогов, которые STEP 7 предоставляет для пользовательских программ, и данных, которые Вы создаете.</p>

Составные части проекта Проект в STEP 7 по существу состоит из объектов, изображенных на рисунке 3-2. Эти объекты перечислены и описаны ниже.

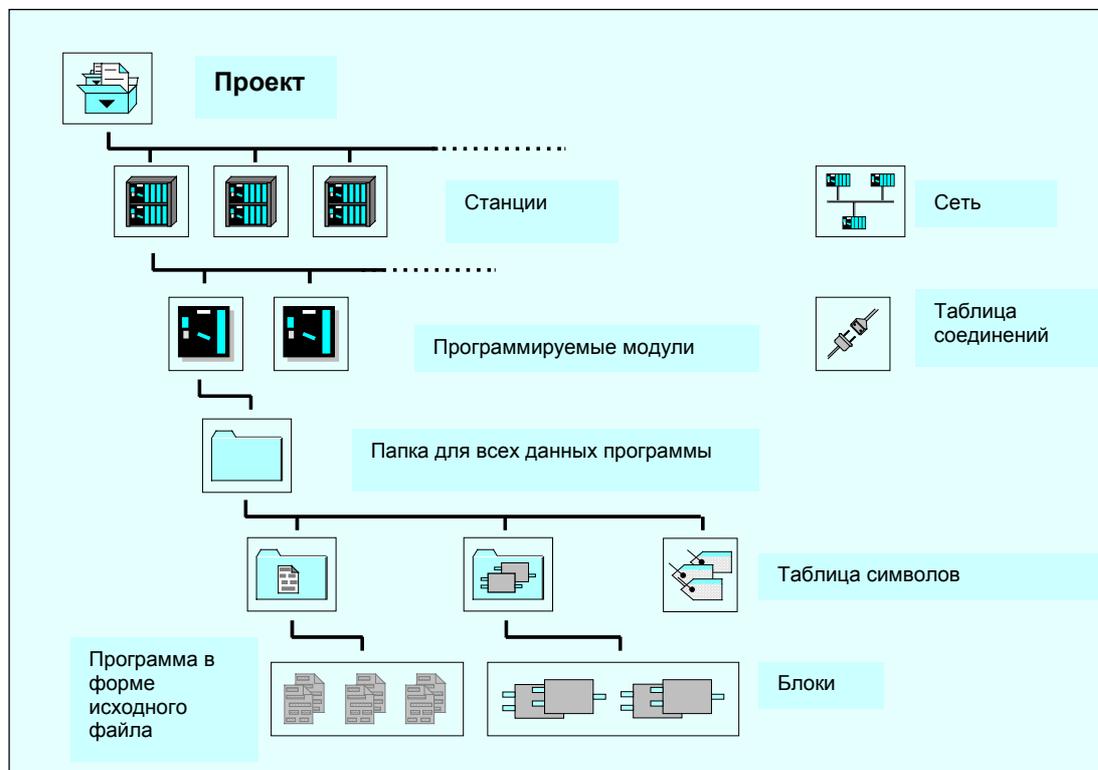


Рис.3-2 Основные объекты проекта STEP 7 в своей иерархической структуре

Сеть Объект “Сеть” содержит основные свойства для подсети типа MPI или PROFIBUS. Подключение станции или коммуникационного модуля станции к сети позволяет STEP 7 проверять параметры связи на соответствие (консистентность).

Станции Станция представляет собой структуру программируемого контроллера вместе с принадлежащими ему стойками. Если модуль с DP интерфейсом является частью станции, тогда взятая в целом мастер-система (так как DP слэйвы принадлежит ей) также является частью этой станции.
Станция состоит из одного или более программируемых модулей, таких как центральный процессор.

Аппаратные средства Аппаратные средства объекта содержат конфигурационные данные и параметры станции. Конфигурационные данные и параметры станции хранятся в системном блоке данных (SDBs).

Программируемые модули	<p>В отличие от других модулей, программируемые модули содержат пользовательские программы. В папках (известных как “containers” в STEP 7) представленных в программируемых модулях, вы найдете все данные, принадлежащие программе для модуля. Примеры таких программ следующие:</p> <ul style="list-style-type: none">• Программы в форме исходного файла (созданы в текстовом редакторе). <p>Когда исходная программа скомпилирована, выполняемые блоки появляются в папке “Blocks”.</p> <ul style="list-style-type: none">• Блоки, загружаемые в программируемый модуль• Таблица символов
Таблица соединений	<p>Таблица соединений отображает все подключения для программируемого модуля, типа CPU, в станции. Соединение определяет свойства связи между двумя узлами и получает идентификационный номер соединения. Это соединение ID – всё, что Вы можете запрограммировать с использованием стандартизированных коммуникационных блоков, которые являются подобными блокам обработки, используемыми в STEP 5.</p>
Исходные файлы	<p>В программировании S7, исходные файлы - основа для создания блоков. Исходные файлы не могут быть загружены в центральный процессор S7.</p>
Блоки	<p>Блоки - различные части пользовательской программы, они отличаются выполняемыми функциями, структурой и характером использования. Блоки могут быть загружены в центральные процессоры S7.</p> <p>В дополнение к исполняемым блокам, папка "Blocs" также содержит таблицы переменных.</p>
Таблица символов	<p>Таблица символов показывает назначение символьных имен, например, для входов, выходов, меркерной памяти и блоков.</p>

3.3 Редактирование проектов с помощью SIMATIC Manager

3.3.1 Создание проектов

Новый проект Для создания проекта следуйте указанным ниже шагам:

1. Выберите команду меню **File ► New** в SIMATIC Manager.
2. Выберите опцию “New Project” в диалоговом окне “New”.
3. Введите имя своего проекта и подтвердите свой выбор, нажав на кнопку “OK.”

Альтернативная процедура При создании проекта Вы свободны в выборе пути, которым Вы будете следовать при выполнении большинства задач. После создания проекта Вы можете выбрать один из следующих методов:

- Сначала конфигурируйте аппаратные средства, а затем создаете под него программное обеспечение, или
- Сначала создаете программное обеспечение, независимое от конфигурации любых аппаратных средств. Необязательно создавать конфигурацию аппаратных средств станции перед вводом программы.

Таблица 3–1 Альтернативные процедуры

Вариант 1	Вариант 2
Сконфигурируйте сначала аппаратные средства (см. также раздел 3.4)	Создайте сначала программное обеспечение
Сконфигурируйте ваши аппаратные средства (см. Раздел 3.4).	
Как только конфигурация закончена, папка “S7 Program” необходимая для создания программного обеспечения, уже установлена и доступна.	Создайте необходимые программные папки (S7 Programs) в Вашем проекте (см. раздел 3.6).
Создайте программу для программируемых модулей (см. раздел 3.6).	Создайте программу для программируемых модулей (см. раздел 3.6).
	Сконфигурируйте ваши аппаратные средства (см. Раздел 3.4).
	После конфигурирования аппаратных средств, свяжите программу S7 с центральным процессором.

Процедура загрузки и проверки Вашей программы без использования конфигурации аппаратных средств описана в руководстве пользователя /231/.

3.3.2 Сохранение проектов

- Общие сведения** Для создания резервной копии проекта, Вы можете сохранить копию проекта под другим именем или поместить ее в архив.
- Сохранить как...** Сохранения проекта под другим именем выполняйте следующим образом:
1. Откройте проект.
 2. Выберите команду меню **File ► Save As**. Будет выведено на экран диалоговое окно "Save As".
 3. Выберите один из двух вариантов сохранения "с" или "без" проверки совместимости и закройте диалоговое окно, нажав на "ОК." Будет выведено на экран диалоговое окно "Save As".
 4. В поле "Save In" выберите директорию, в которой должен быть сохранен проект.
 5. В поле "File Name" введите имя файла вместо звездочки (*). Не изменяйте расширение файла.
 6. Закройте диалоговое окно, нажав "ОК."
- Удостоверьтесь, что на выбранном диске есть достаточно свободного места. Например, не желательно выбрать дисковод "А" для сохранения проекта, потому что проект является слишком большим, чтобы разместиться на дискете. Вы должны архивировать проекты перед сохранением их на дискетах. Архив тогда может быть разбит на несколько дискет.
- Архивация** Вы можете сохранить индивидуальные проекты или библиотеки в сжатой форме в файле архива, расположенном на жестком диске или дискете.
- Чтобы иметь доступ к компонентам заархивированного проекта или библиотекам, проект должен сначала быть извлечен из архива. Архивирование подробно рассмотрено в руководстве пользователя *1231*.

3.4 Конфигурирование аппаратных средств с помощью STEP 7

SIMATIC S5 не обеспечивал функцию для конфигурирования аппаратных средств с использованием программного обеспечения. В S7 адресация и определение параметров модулей и конфигурирование связи выполняется посредством приложения STEP 7. Преимущество этого метода состоит в том, что пользователь больше не должен делать никаких аппаратных заданий параметров настройки модулей, так как конфигурирование и назначение параметров могут теперь быть сделаны централизованно с использованием программаторов.

Предварительное условие Чтобы конфигурировать аппаратные средства, проект должен быть уже создан.

Установка станции

Для создания в проекте новой станции откройте проект (если это не было сделано).

1. Выделите проект.
2. Создайте объект для требуемых аппаратных средств, выбрав команду меню **Insert ► Station**.

В подменю Вы можете выбрать один из следующих вариантов:

- SIMATIC 300 station
- SIMATIC 400 station
- PC/PG
- SIMATIC S5
- Other stations, имеется ввиду не SIMATIC S7/M7 и SIMATIC S5

Станции PC/программатор, SIMATIC S5 и другие станции перечислены для конфигурирования коммуникационной связи. Конфигурирование и программирование станций S5 невозможны.

Нажмите на символ “+” на передней стороне значка проекта в проектном окне, если станция не показана ниже.

Конфигурирование аппаратных средств Конфигурирование аппаратных средств выполняйте следующим образом:

1. Выберите с помощью "мыши" новую станцию, которую Вы создали; она находится в папке "Hardware".
2. Откройте папку "Hardware". Откроется окно "HWConfig".
3. В окне "Hardware Configuration" спроектируйте структуру станции. Чтобы помочь Вам сделать это, доступен каталог модулей. Если он не показан, выберите команду меню **View ► Catalog** для вывода его.
4. Вставьте стойку из каталога модулей в пустое окно. Затем выберите модули и поместите их в ячейки стойки. Не менее одного CPU должно быть сконфигурировано для каждой станции. В процессе выполнения этой процедуры HWConfig автоматически проверяет все вводимые Вами данные.

Для дальнейшей информации относительно конфигурирования аппаратных средств смотри руководство пользователя /231/.

Результат конфигурации

Для каждого CPU после создания конфигурации программа S7 и таблица подключений (объект "Connections") создаются автоматически, после сохранения и выхода из конфигурации аппаратных средств. Программа S7 содержит объекты "Source Files" и "Blocks" как программные папки, так же как и таблицу символов.

Папка "Blocks" содержит объект для OB1 и объект "System Data" со скомпилированными конфигурационными данными.

3.5 Конфигурирование соединений в таблице соединений

В S5 соединения конфигурируются с помощью COM NCM. Здесь есть программный пакет COM для каждого коммуникационного процессора (CP). В S7 все соединения конфигурируются в таблице соединений.

Общие сведения

Конфигурирование соединений – это предварительное условие для использования SFB коммуникационных функций в пользовательской программе.

Соединения определяются следующим:

- Партнеры по соединению должны быть объединены в проект S7,
- Задан тип соединений (например, S7 соединение или FDL соединение),
- Специальными свойствами типа активного или пассивного состояния соединений или необходимостью отправки операционных сообщений.

Когда Вы сконфигурируете соединения, для каждого соединения будет создан уникальный локальный идентификатор (известный как локальный ID). Эти локальные ID являются всем, что Вам необходимо при назначении параметров соединений.

Каждый центральный процессор, который может быть конечным пунктом соединения, имеет его собственную таблицу соединений.

Специальные свойства

Если оба партнера связи - станции S7-400, локальные ID автоматически создаются для обоих конечных пунктов связи. Только один локальный ID генерируется для станции S7-400 при соединении со станцией S7-300.

Загрузка конфигурационных данных

Локальные конфигурационные данные для соединения конечных точек станции S7 должны быть по отдельности загружены в каждую целевую станцию.

Пустая таблица соединений (объект "Connections") автоматически создается для каждого центрального процессора. Эта таблица соединений используется для определения соединений связи между центральными процессорами в сети. После ее открытия появляется окно, содержащее таблицу для определения соединений между программируемыми модулями (Для получения дополнительной информации об определении соединений, см. руководство пользователя /231/).

**Пример:
Соединение с S5**

Этот пример показывает Вам, как сконфигурировать соединение со станцией SIMATIC S5. Он предполагает, что Вы уже вставили станции SIMATIC 400 в ваш проект.

- Вставьте станцию SIMATIC S5 в ваш проект и затем задайте свойства станции.
- Откройте таблицу соединений для станции S7 и выберите команду меню **Insert ► Connections** для вставки соединения. Будет отображено диалоговое окно, в котором Вы можете ввести партнера связи (станция SIMATIC S5) и тип соединений.
- Как только Вы ввели эту информацию, соединение появляется в таблице подключений. Параметры соединения должны быть введены в соответствующий COM NCM для станции S5.

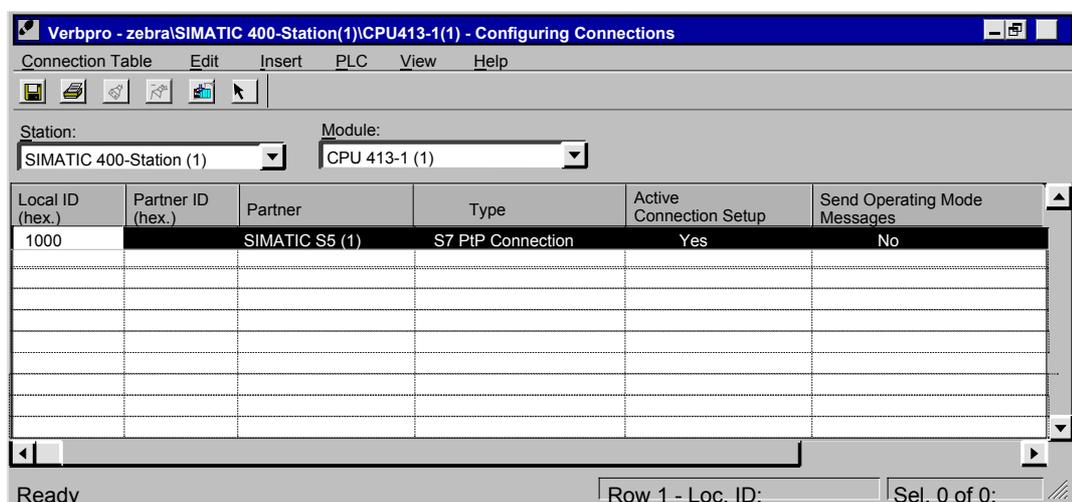


Рис.3-3 Таблица соединений

3.6 Вставка и редактирование программы

Методика, описанная в этом разделе, применяется для создания новой программы.

3.6.1 Основная методика создания программы

Общие сведения

Программа для центральных процессоров сохраняется в программных папках. В модулях SIMATIC S7 такие объекты называются "S7 Program."

На рисунке, приведенном ниже, показана папка "S7 program" в CPU станции SIMATIC 300.

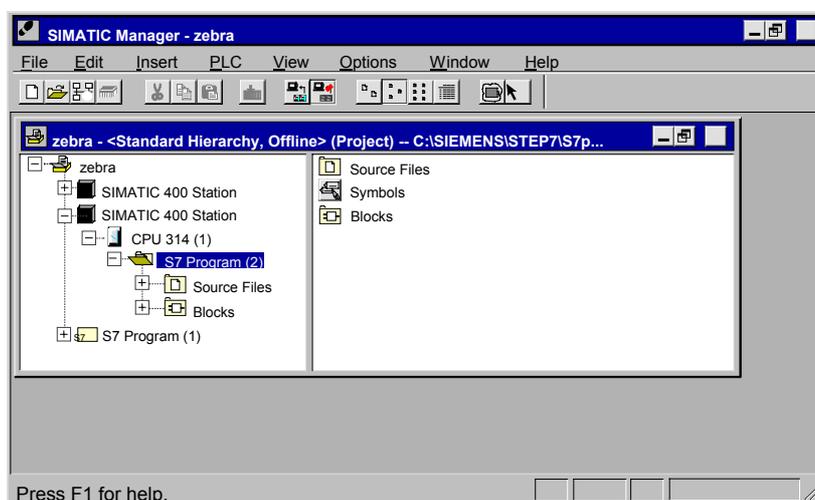


Рис. 3-4 Открытая S7 программа в SIMATIC Manager

Методика

При создании программного обеспечения для Вашего проекта выполните следующее:

1. Откройте программу S7.
2. Откройте объект “Symbols” в S7 программе и задайте символы. (Этот шаг может быть сделан позже.) Подробную информацию относительно определения символов Вы найдете в разделе 3.13.2.
3. Откройте папку “Blocks”, если Вы хотите создать блоки или откройте папку “Source Files”, если Вы хотите создать исходный файл.
4. Вставьте блок или исходный файл, соответственно. (Для детальной информации, см. Раздел 3.6.2). Для этого используются следующие команды меню:
 - **Insert ► S7 Software ► Block**, или
 - **Insert ► S7 Software ► Source File**
5. Откройте блок или исходный файл и введите программу. Подробную информацию относительно программ Вы найдете в Руководствах по программированию /232/-/236/.

В зависимости от вашей задачи, Вы можете выполнять не все эти шаги.

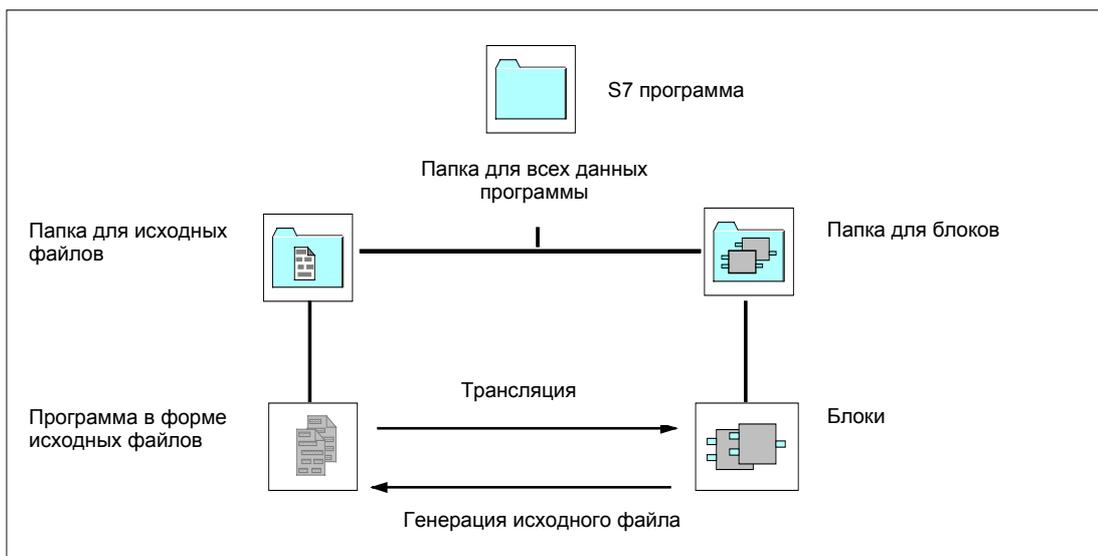


Рис.3-5 Основные объекты проекта S7 и их иерархическая структура

3.6.2 Вставка компонентов для создания программного обеспечения в S7 и M7 программах

Имеющиеся компоненты	<p>Программа S7/M7 создается автоматически для каждого программируемого модуля как папка для программного обеспечения:</p> <p>Следующие объекты уже существуют в новой программе S7:</p> <ul style="list-style-type: none">• Таблица символов (объект "Symbols")• Папка "Blocks" для блоков с начальным OB1• Папка "Source Files" для программ в форме исходных файлов
Создание S7 блоков	<p>Если Вы хотите создать программу на одном из языков STEP 7 (STL, FBD или LAD), выделите объект "Blocks" и затем выберите с помощью "мыши" команду меню Insert ► S7 Software ► Block. В подменю Вы можете выбрать тип блока, который Вы хотите создать (блок данных, определяемый пользователем тип данных (UDT), функции, функциональный блок, организационный блок или таблица переменных (VAT)).</p> <p>Теперь Вы можете открыть блок (пустой) и начать ввод программы. Подробную информацию Вы найдете в Руководстве по программированию <i>Statement List I232I</i>, <i>Ladder Logic I233I</i>, и <i>Function Block Diagram I236I</i>.</p> <p>Объект "System Data" (SDB), который может существовать в пользовательской программе, создается системой. Вы можете открыть его для просмотра содержания, но Вы не можете делать в нем изменения. Он используется для изменений в конфигурации, при загрузке программу и при загрузке изменений в программируемый контроллер.</p>
Использование блоков от стандартных библиотек	<p>Вы также можете использовать блоки из стандартных библиотек, поставляемых с программным обеспечением, для создания пользовательских программ. Для обращения к библиотекам используйте команду меню File ► Open. Подробную информацию относительно использования стандартных библиотек и создания ваших собственных библиотек Вы найдете в интерактивной справке.</p>
Создание исходных файлов	<p>Если Вы хотите создать исходный файл в Statement List, выберите объект "Source Files" или "Charts" в программе S7, а затем выберите команду меню Insert ► S7 Software ► Source File. В подменю Вы можете выбрать исходный файл, соответствующий Вашему языку программирования. Вы можете теперь открыть пустой исходный файл и начать ввод Вашей программы.</p>
Создание таблицы символов	<p>Таблица символов (пустая) (объект "Symbols") создается автоматически при создании программы S7. Когда Вы открываете таблицу символов, открывается окно "Symbol Editor" таблицы символов, где Вы можете задавать символы (для полной информации обратитесь к разделу 3.13.2).</p>

**Вставка внешних
исходных
файлов**

Вы можете создавать и редактировать исходные файлы с помощью любого ASCII редактора. Затем Вы можете импортировать эти файлы в любой проект и компилировать в исполняемые блоки. Для вставки внешнего исходного файла выполните следующее:

1. Выберите папку "Source Files", в которую Вы хотите импортировать исходный файл.
2. Выберите команду меню **Insert ► External Source File**.
3. Введите имя исходного файла в появившееся диалоговое окно.

Блоки создаются, когда импортированные исходные файлы откомпилированы и сохранены в папке "Blocks".

3.7 Блоки

3.7.1 Сравнение

В следующей таблице представлено сравнение блоков в STEP 5 и STEP 7. Эта таблица предназначена для ответа на вопрос, “Какой блок STEP 7 можно использовать вместо блока STEP 5?”

Никаких фиксированных заданий

Эта таблица не должна интерпретироваться как установленный однозначный набор назначений, так как новый состав блоков делает доступными дополнительные варианты программирования. Информационные элементы таблицы должны быть поняты как набор рекомендаций для начала программирования STEP 7.

Таблица 3–2 Сравнение блоков: STEP 5 / STEP 7

Блок STEP 5	Блок STEP 7	Объяснение
Организационный блок (OB)	Организационный блок (OB)	Интерфейс к операционной системе
Интегрированные специальные организационные блоки	Системные функции (SFC) Системные функциональные блоки (SFBs)	Системные функции в STEP 7 заменяют специальные организационные блоки (STEP 5), которые могут быть вызваны в пользовательской программе.
Функциональные блоки (FB, FX)	Функции (FC)	Функции (FCs) в STEP 7 имеют те же самые свойства, как и функциональные блоки в STEP 5.
Программные блоки (PB)	Функциональные блоки (FB)	Программные блоки соответствуют функциональным блокам в STEP 7. Функциональные блоки в STEP 7 имеют совершенно новые свойства по сравнению с блоками в STEP 5, имея то же названия; таким образом, они теперь обеспечивают новые варианты программирования. Замечание: При преобразовании программные блоки преобразовываются в функции (FCs).
Шаговый блок (SB)	-	В STEP 7 нет никаких шаговых блоков
Блоки данных (DB, DX)	Блоки данных (DB)	В STEP 7 блоки данных более длинные, чем в STEP 5 (в S7–300 до 8 Килобайт, в S7–400 до 64 Килобайт).
Блоки данных DX0, DB1 со специальными функциями	Системные блоки данных (SDB) (назначение параметров CPU)	Новые системные блоки данных содержат все аппаратные конфигурационные данные, включаемые CPU параметры назначения определяют обработку программы.
Блоки комментариев DK, DKX, FK, FKX, PK	-	In STEP 7 нет больше никаких блоков комментария. Комментарии содержатся в соответствующем блоке в автономной базе данных.

3.7.2 Функции и функциональные блоки

Функции (FCs) Функции (FCs) - логические блоки без "памяти". Выходные параметры содержат расчетные функциональные значения после того, как функция обработана. Они используются как фактические параметры, и пользователь решает, где они сохраняются после обработки FC.

Не путайте функции с функциональными блоками! В STEP 7 это разные типы блоков.

Функциональные блоки (FBs) Функциональные блоки (FB) - логические блоки, которые имеют "память." Память находится в форме экземпляра блока данных, который связан с функцией. В нем сохраняются фактические параметры и статические данные функционального блока.

3.7.3 Блоки данных

Блоки данных сохраняют данные пользовательской программы. Различие сделано между блоками общих данных и блоками данных образца, как объяснено в следующем:

- Глобальные блоки данных не связаны с конкретным блоком (как в STEP 5).
- Блоки-экземпляры данных образца связаны с функциональным блоком (FB) и содержат, в дополнение к данным FB, данные других экземпляров, которые могут быть определены как мультиэкземпляры.

Каждый блок данных может быть или глобальным блоком данных или блоком-экземпляром.

3.7.4 Системные блоки

Системные функции (SFCs) и системные функциональные блоки (SFBs)

Вы не должны программировать каждую функцию самостоятельно. Кроме того, например, Вы можете вызывать коммуникационные функции, например, используя предустановленные CPU, предоставляемые операционной системой. Это следующие:

- **Системные функции (SFC)**, со свойствами подобными таким функциям (FC)
- **Системные функциональные блоки (SFB)**, со свойствами подобными функциональным блокам (FB).

Блоки системных данных (SDB)

Предыдущее обсуждение было сконцентрировано вокруг блоков, содержащих программы или данные пользовательской программы. В дополнение к этим блокам есть другие блоки, содержащие параметры настройки типа параметров модуля или адресов. Их называют **системными блоками данных (SDBs)** и они создаются специальными приложениями STEP 7, например, при вводе аппаратной конфигурации или создании таблиц соединений.

3.7.5 Организационные блоки

Организационные блоки (OBs) формируют интерфейс между операционной системой и пользовательской программой. Различные организационные блоки выполняют свои определенные задачи.

Распределение организационных блоков Вы связываете пользовательскую программу для центрального процессора вашего S7 CPU с помощью организационных блоков (OBs), необходимых для вашего решения автоматизации.

Таблица 3–3 Сравнение организационных блоков в S5 и S7

Функция		S5	S7
Основная программа	Циклическая обработка	OB1	OB1
Прерывания	Прерывание с задержкой	OB6	OB20 до OB23
	Управление по времени	OB9	OB10 до OB17
	Аппаратные прерывания	OB2 до OB5	OB40 до OB47
	Прерывания процесса	OB2 до OB9 (IB 0)	Заменен аппаратными прерываниями
	Циклические (по времени) прерывания	OB10 до OB18	OB30 до OB38
	Прерывания в многопроцессорной системе	-	OB60
Запуск	Ручной (холодный) новый старт	OB21 (S5–115U) OB20 (от S5–135U)	OB100
	Ручной (теплый) новый старт	OB21 (от S5–135U)	OB101
	Автоматический (теплый) новый старт	OB22	OB101
Ошибки	Ошибка	OB19 до OB35	OB121, OB122, OB80 до OB87
Прочие	Обработка в режиме STOP	OB39	Исключено
	Фоновая обработка	-	OB90

Обработка ошибок

Организационные блоки обработки ошибок

Организационные блоки обработки ошибок вызываются системой при возникновении ошибки при выполнении программы. Вы можете использовать их для реакции программы на ошибки. Если имеется ошибка, и нет соответствующего организационного блока, центральный процессор переходит в режим STOP (кроме ошибки буферной батареи).

Table 3–4 Обработка ошибок в S5 и S7

Функция	S5	S7
Вызов блока, который не загружен	OB19	OB121
Ошибка прямого доступа к модулям ввода - вывода	OB23	OB122
Ошибка обновления процесса отображения	OB24	OB122
Ошибка адресации	OB25	OB122
Превышение времени цикла	OB26	OB80
Ошибка подстановки	OB27	Пропущено
Остановка оператором	OB28 (S5–135U)	Пропущено
Ошибка ожидания входного байта IB 0	OB28 (S5–155U)	OB85
Некорректная команда	OB29 (S5–135U)	STOP
Ошибка прямого доступа к модулям ввода - вывода в расширенной области адреса	OB29 (S5–155U)	OB122
Некорректный параметр	OB30 (S5–135U)	Пропущено
Ошибка четности или превышение времени обращения к пользовательской памяти	OB30 (S5–155U)	OB122
Специальная функциональная групповая ошибка	OB31	Пропущено
Ошибка загрузки или передача в блоки данных	OB32	OB121
Конфликт временных прерываний	OB33	OB80
Ошибка контроллера	OB34 (S5–135U)	Отсутствует
Ошибка генерации блока данных	OB34 (S5–155U)	SFC обратная связь
Коммуникационная ошибка	OB35	OB84

Поиск неисправностей в S5 и S7

Превышение уровня сигнала

Как в S5, Вы также можете использовать биты слова состояния OV и OS для оценки сообщения о переполнении. Различие в характере изменений в этих двух системах незначительное.

Для полной информации о характере изменений состояния битов со ссылкой на инструкции, смотри инструкцию по программированию *Statement List Programming Manual I232/*.

Встроенные специальные функции

Интерфейс между пользовательской программой и системной программой в центральных процессорах S5 состоит в создании доступа в область операционной системы и с помощью специальных организационных блоков.

В центральных процессорах S7 этот интерфейс имеет два новых типа блоков (“системные функции” и “системные функциональные блоки”) в дополнение к организационным блокам.

Системные функции / Системные функциональные блоки

Системные функции (SFCs) и системные функциональные блоки (SFBs) являются блоками, интегрированными в операционную систему центрального процессора, и могут быть необходимыми в пользовательской программе STEP 7. Если ошибка происходит во время обработки системной функции (SFC), эта ошибка может быть оценена в пользовательской программе при помощи возвращаемого значения RET_VAL.

Таблица 3–5 Специальные функции в S5 и S7

Функция	Блок S5	Замена в S7
Перезапуск контроля времени цикла	OB31	SFC43 RE_TRIGR
Отказ батареи	OB34	OB81 (Реакция на ошибку может быть запрограммирована пользователем)
Доступ к битам состояния	OB110	команда STEP 7: L STW/T STW
Обнуление ACCU 1 - 4	OB111	STEP 7 последовательность команд: L 0; PUSH; PUSH; PUSH
Сдвиг вверх ACCU	OB112	Не идентичная функция: STEP 7 инструкция: PUSH
Сдвиг вниз ACCU	OB113	Функция не идентична инструкции STEP 7: POP
Запрет всех прерываний	OB120	SFC41 DIS_AIRT SFC42 EN_AIRT
Запрет временных прерываний	OB121	SFC39 DIS_IRT SFC40 EN_IRT
Задержка всех прерываний, вкл\выкл	OB122	SFC41 DIS_AIRT SFC42 EN_AIRT
Индивидуальная задержка циклических (установленных по времени) прерываний вкл\выкл	OB123	SFC39 DIS_IRT SFC40 EN_IRT
Установка\чтение процессорного времени (продолжение на следующей странице)	OB150	SFC0 SET_CLK SFC1 READ_CLK

Таблица 3–5 Специальные функции в S5 и S7 (продолжение)

Функция	Блок S5	Замена в S7
Установка/чтение управляемого временем прерывания	OB151	SFC28 SET_TINT SFC30 ACT_TINT SFC31 QRY_TINT
Статистика цикла	OB152	Локальные данные в OB1
Счетчик	OB160 – 163 (S5–135U)	STEP 7 инструкция: LOOP
Временный цикл	OB160 (S5–115U)	SFC47 WAIT
Чтение стека блоков	OB170	Отсутствует
Доступ к переменным блока данных	OB180	Отсутствует
Тестирование блока данных	OB181	SFC24 TEST_DB
Копирование области данных	OB182	SFC20 BLKMOV
Передача меркеров блоков данных	OB190, OB192	SFC20 BLKMOV
Передача данных блоков в область меркеров	OB191, OB193	SFC20 BLKMOV
Функции для многопроцессорных коммуникаций	OB200 – 205	Отсутствует
Доступ к странице	OB216 – 218	Нет страницы адресации в S7
Переход к двойному циклу	OB220	S7 инструкция: ITD
Установка контрольного времени цикла	OB221	Параметр назначается с S7
Перезапуск контроля времени цикла	OB222	SFC43 RE_TRIGR
Сравнение типов запуска	OB223	Многопроцессорный старт только для одного типа запуска
Передача IPC флагов в блоки	OB224	Отсутствует
Чтение слов из системной программы	OB226	Отсутствует
Чтение CRC из системной программы	OB227	Отсутствует
Чтение информации об уровне обработки программы	OB228	SFC51 RDSYSST SFC6 RD_SINFO
Функции обслуживания блоков	OB230 – 237	Коммуникации с SFBs
Инициализация регистров сдвига	OB240	Отсутствует
Работа регистра сдвига	OB241	Отсутствует
Удаление регистра сдвига	OB242	Отсутствует
Контроль: Инициализация PID алгоритма Контроль: Выполнение PID алгоритма	OB250 OB251	Контроль обратной связи FBs: FB41 – FB43 or SFB41 – SFB43
Передача блоков данных (DB/DX) в DB RAM	OB254, OB255	Отсутствует

3.7.6 Представление блоков при преобразовании

Назначение блоков Структура блоков была изменена для S7. На рисунке ниже показан упрощенный пример назначения блоков для STEP 5 и STEP 7, следующий из процесса преобразования.

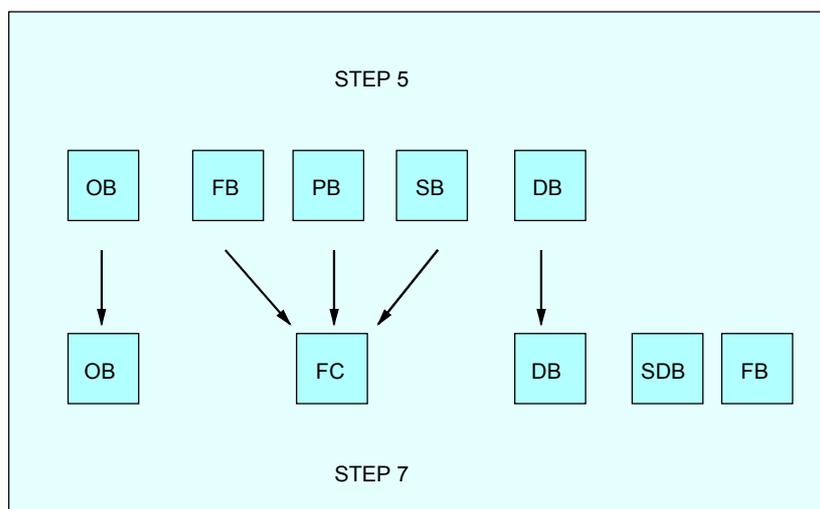


Рис.3-6 Блоки с сопоставимыми функциями в STEP 5 и STEP 7

В таблице 3–6 на следующей странице показано, как конвертируются вызываемые блоки.

Таблица 3–6 Типы блоков в S5 и S7

S5			S7	
OB	Фиксированный номер	Пользовательская программа	Соответствует S7 OB	Фиксированный номер
OB	Фиксированный номер	Специальная функция	Не конвертируемый, должен быть перепрограммирован с помощью S7	
PB	0 до 255	Пользовательская программа	FC блоки без параметров	Номер произвольный
FB/FX	0 до 255	Пользовательская программа	FC блоки с параметрами, названия которых сохранены	Номер произвольный
FB	Фиксированный номер	Интегрированные функциональные блоки	Загружаемый FCs, содержащийся в FBlib1 библиотеке, которая должна быть загружена к преобразованному файлу перед компиляцией	Фиксированный номер
FB/FX	Фиксированное имя	Стандартные функциональные блоки	Загружаемый FCs, содержащийся в FBlib1 библиотеке, которая должна быть загружена к преобразованному файлу перед компиляцией	Фиксированный номер
SB	0 до 255	Пользовательская программа	FC блоки без параметров (шаговые блоки не могут быть преобразованы и должны быть созданы в S7 GRAPH).	Номер произвольный
DB	2 до 255	Пользовательские данные	Глобальные блоки данных (DBs)	Номер взят в S5
DX	1 до 255	Пользовательские данные	Глобальные блоки данных (DBs)	Предложены номера от 256
DB1/ DX0		Блоки данных с системными параметрами настройки	Если блоки содержат специфичные для CPU компоненты, настройки параметров должны быть сделаны в STEP 7. Преобразованное содержание DB1 и DX0 является несоответствующим и может быть удалено.	

3.8 Системные установки

Преобразование DB1 и DX0 Следующие таблицы показывают, как сделаны функции для параметров в DB1 и DX0 (системные установки):

Таблица 3–7 Преобразование системных установок для DB1

Блок параметра S5	Как выполнено в S7
Задержка запуска	Вызов SFC47 WAIT
IPC меркеры	Вызов функций передачи глобальных коммуникационных данных SFC60 GD_SND SFC61 GD_RCV
Адрес ошибочного кода определения	Система вводит сообщения об ошибках в диагностический буфер. Информация об адресе кода ошибки отсутствует
Замена номера интегрированных FBs	Отсутствует
Встроенные аналоговые входы	Установка в HWConfig свойств CPU
Встроенные прерывания	Установка в HWConfig свойств CPU
Встроенные счетчики	Установка в HWConfig свойств CPU
Смена приоритетов OBs	Установка в HWConfig свойств CPU
Вывод/запрет актуализации области отображения выходов	Вызов SFC27 UPDAT_PO
Вывод/запрет актуализации области отображения входов	Вызов SFC26 UPDAT_PI
Сохраняемые меркеры	Установка в HWConfig свойств CPU
Сохраняемые таймеры	Установка в HWConfig свойств CPU
Сохраняемые счетчики	Установка в HWConfig свойств CPU
SINEC L1	Замена на MPI шину (глобальные коммуникационные данные)
SINEC L2	Установка с помощью HWConfig
Защита программного обеспечения	В подготовке
Параметры часов	Установка в HWConfig свойств CPU или вызова SFC28 SET_TINT
Назначение параметров для установленного по времени прерывания OB	Установка в HWConfig свойств CPU
Контрольное время цикла	Установка в HWConfig свойств CPU

Таблица 3–8 Преобразование системных установок для DX0

Блок параметров S5	Как выполнено в S7
Контроль ошибок адресации	Вызов OB121
Обновление IPC меркеров	Глобальные коммуникационные данные
Типы запуска после включения питания	Установка в HWConfig свойств CPU
Синхронизация начала в многопроцессорной операции	Установка в HWConfig свойств CPU
Число таймеров	Фиксированные значения в зависимости от CPU (для S7–300) или установка в HWConfig использования свойств CPU (для S7–400)
Обработка ошибок	Вызов: SFC36 MSK_FLT SFC37 DMSK_FLT
Математические операции с плавающей точкой	Представлены
Деблокировка аппаратных прерываний	Установка в HWConfig свойств CPU
Установка циклических прерываний	Вызов SFC28 SET_TINT
Контроль времени цикла	Установка в HWConfig свойств CPU

3.9 Стандартные функции

При преобразовании, стандартный набор функций в S5 автоматически заменяется преобразованными функциями, имеющими те же самые функциональные возможности. В S7 большинство этих функций может быть заменено упрощенными последовательностями команд, которые экономят память и уменьшают время цикла.

Стандартные функции содержатся в “StdLib30” S7 библиотеки, расположенной в папке программы FBLib1.

Для дальнейшей информации относительно работы с библиотеками обратитесь к интерактивной справке.

3.9.1 Математические операции с плавающей точкой

STEP 5	STEP 7		STEP 5	STEP 7	
FB	Номер	Название	FB	Номер	Название
GP:FPGP	FC61	GP_FPGP	GP:MUL	FC65	GP_MUL
GP:GPFP	FC62	GP_GPFP	GP:DIV	FC66	GP_DIV
GP:ADD	FC63	GP_ADD	GP:VGL	FC67	GP_VGL
GP:SUB	FC64	GP_SUB	RAD:GP	FC68	RAD_GP

3.9.2 Сигнальные функции

STEP 5	STEP 7		STEP 5	STEP 7	
FB	Номер	Название	FB	Номер	Название
MLD:TG	FC69	MLD_TG	MLD:EZ	FC75	MLD_EZ
MELD:TGZ	FC70	MELD_TGZ	MLD:ED	FC76	MLD_ED
MLD:EZW	FC71	MLD_EZW	MLD:EZWK	FC77	MLD_EZWK
MLD:EDW	FC72	MLD_EDW	MLD:EDWK	FC78	MLD_EDWK
MLD:SAMW	FC73	MLD_SAMW	MLD:EZK	FC79	MLD_EZK
MLD:SAM	FC74	MLD_SAM	MLD:EDK	FC80	MLD_EDK

3.9.3 Интегрированные функции

STEP 5	STEP 7	
FB	Номер	Название
COD:B4	FC81	COD_B4
COD:16	FC82	COD_16
MUL:16	FC83	MUL_16
DIV:16	FC84	DIV_16

3.9.4 Основные функции

STEP 5	STEP 7		STEP 5	STEP 7	
FB	Номер	Название	FB	Номер	Название
ADD:32	FC85	ADD_32	REG:LIFO	FC93	REG_LIFO
SUB:32	FC86	SUB_32	DB:COPY	FC94	DB_COPY
MUL:32	FC87	MUL_32	DB:COPY	FC95	DB_COPY
DIV:32	FC88	DIV_32	RETTEN	FC96	RETTEN
RAD:16	FC89	RAD_16	LADEN	FC97	LADEN
REG:SCHB	FC90	REG_SCHB	COD:B8	FC98	COD_B8
REG:SCHW	FC91	REG_SCHW	COD:32	FC99	COD_32
REG:FIFO	FC92	REG_FIFO			

3.9.5 Аналоговые функции

STEP 5	STEP 7		STEP 5	STEP 7	
FB	Номер	Название	FB	Номер	Название
AE:460	FC100	AE_460_1	AE:466	FC106	AE_466_1
AE:460	FC101	AE_460_2	AE:466	FC107	AE_466_2
AE:463	FC102	AE_463_1	RLG:AA	FC108	RLG_AA1
AE:463	FC103	AE_463_2	RLG:AA	FC109	RLG_AA2
AE:464	FC104	AE_464_1	PER:ET	FC110	PER_ET1
AE:464	FC105	AE_464_2	PER:ET	FC111	PER_ET2

3.9.6 Математические функции

STEP 5	STEP 7		STEP 5	STEP 7	
FB	Номер	Название	FB	Номер	Название
SINE	FC112	SINE	ARCCOT	FC119	ARCCOT
COSINE	FC113	COSINE	LN X	FC120	LN_X
TANGENT	FC114	TANGENT	LG X	FC121	LG_X
COTANG	FC115	COTANG	B LOG X	FC122	B_LOG_X
ARCSIN	FC116	ARCSIN	E^X	FC123	E_H_N
ARCCOS	FC117	ARCCOS	ZEHN^X	FC124	ZEHN_H_N
ARCTAN	FC118	ARCTAN	A2^A1	FC125	A2_H_A1

3.10 Типы данных

STEP 7 использует новый формат данных. Таблица ниже сравнивает различные типы данных в S5 и S7:

Таблица 3–9 Типы данных в S5 и S7

Типы данных в S5	Типы данных в S7	Класс данных
BOOL, BYTE, WORD, DWORD, Integer, Double integer, Floating point, Значения времени, – ASCII символы	BOOL, BYTE, WORD, DWORD, INT, DINT, REAL, S5TIME, TIME, DATE; TIME_OF_DAY, CHAR	Элементарные типы данных
-	DATE_AND_TIME, STRING, ARRAY, STRUCT	Комплексные типы данных
Таймеры, Счетчики, Блоки - -	TIMER, COUNTER, BLOCK_FC, BLOCK_FB, BLOCK_DB, BLOCK_SDB, POINTER, ANY	Параметрические типы

Таблица 3–10 Различные форматы для констант в S5 и S7

Форматы в S5	Пример	Форматы в S7	Пример
KB	L KB 10	k8	L B#16# A
KF	L KF 10	k16	L 10
KH	L KH FFFF	16#	L 16# FFFF
KM	LKM 1111111111111111	2#	L 2# 11111111_11111111
KY	L KY 10,12	B#	L B# (10,12)
KT	L KT 10.0	S5TIME# (S5T#)	L S5TIME# 100ms
KC	L KC 30	C#	L C#30
DH	L DH FFFF FFFF	16#	L DW#16# FFFF_FFFF
KS	L KS WW	' xx '	L ' WW '
KG	L KG +234 +09	Floating point	L +2.34 E+08
Способ задания: S5 формат ← Экспонента → ← Мантисса →		Способ задания: Едиственный формат compl. with ANSI/IEEE ← Экспонента → ← Мантисса →	
Экспонента = значение экспоненты SE = символ экспоненты SM = символ мантиссы Диапазон значений: 1.5×10^{-39} до 1.7×10^{38}		Экспонента = фактическая экспонента + B (bias*) (+127) S = символ мантиссы Диапазон значений: приблизительно от 1.18×10^{-38} до $3.4 \times 10^{+38}$	

* Bias: Это коэффициент смещения отдельных экспонент в положительные и отрицательные области. Значение 127 в области экспоненты соответствует значению 0 по абсолютной величине.

Для дальнейшей информации о типах данных см. the *Statement List Programming Manual* /232/.

3.11 Адресные области

3.11.1 Общие сведения

Таблица 3–11 Адресация в S5 и S7

Адресная область	Адресация в S5	Адресация в S7	Замечание
Вводы	I	I	
Выводы	Q	Q	
I/O	P, Q, G	PI для загрузки команд PQ для передачи команд	Глобальная периферия не преобразуется
Область меркерной памяти (меркеры)	F	M	
	S	M	от M 256.0 (конвертер)
	Черновая память меркеров	L	Преобразуется как меркеры
Таймеры	T	T	
Счетчики	C	C	
Область данных	D...	DB...	Преобразование в общие адреса данных
Системные данные	RS, RT, RI, RJ	–	Не преобразуется
Область страниц памяти	C	–	

В S7 есть два регистра блока данных: DB регистр, который в основном используется для глобальных блоков данных и регистра DI, который используется для экземпляров DBs. В связи с этим есть также два типа адресов данных. Адреса DBX, DBB, DBW и DBD являются адресами общих блоков данных; адреса DIX, DIB, DIW, и DID являются адресами экземпляров DBs. При выполнении преобразования, адреса общих блоков данных используются для адресов данных блока D, DB, DW, DD.

Также обратите внимание, как преобразованы блоки данных (см. раздел 3.7.6).

Предупреждение



Знайте, что размер и номер области адреса, номер и размер блоков для S7 зависят от используемого центрального процессора. Критерии производительности CPU и оценка могут быть найдены в разделе 2.2.1.

3.11.2 Новые адреса в S7: Локальные данные

Локальные данные в STEP7	<p>Локальные данные в STEP 7 являются данными, назначенными для логических блоков, которые описаны в соответствующем разделе описания переменных. В зависимости от блока, они состоят из формальных параметров, статических данных и временных данных. К локальным данным обычно обращаются в символьной форме.</p>
Параметры блока	<p>Параметрами блоков у функций (FC) оперируют подобно параметрам блоков в S5: параметры блока представляют указатели, которые указывают на соответствующий фактический параметр.</p> <p>Параметров функциональных блоков (FB) сохранены подобно статическим местным данным в экземпляре блока данных.</p>
Статические локальные данные	<p>Статические локальные данные могут быть использованы в любом функциональном блоке. Они определены в разделе описания и сохранены в экземплярном блоке данных.</p> <p>Статические локальные данные, подобно данным в глобальных блоках данных, сохраняют их, пока они не изменяются программой.</p> <p>Вообще, статические данные обрабатываются только внутри функционального блока. Однако, так как они сохранены в блоке данных, эти локальные данные могут быть доступны пользовательским программам в любое время, так же как это имеет место в глобальных блоках данных.</p>
Временные локальные данные	<p>Временная рабочая область меркерной памяти STEP 5</p> <p>В STEP 5, меркерная память области адреса используется для хранения данных временно в пределах блока. В соответствии с соглашением, меркеры с 200 по 250 зарезервированы для временного хранения. Управление временной рабочей областью памяти меркеров определяется пользователем.</p> <p>Временные локальные данные в STEP 7</p> <p>Временные локальные данные - области хранения для данных, которые являются действительными только в течение обработки блока. Как только блок был обработан, эти временные области снова освобождаются для использования. Каждый класс приоритета имеет свой собственный локальный стек данных. Это препятствует неосторожному затиранию программой промежуточных данных.</p>

**Использование
временных
локальных
данных в
STEP 7**

В STEP 7 временные переменные используются в следующих трех случаях:

- Как промежуточное звено в хранении данных пользовательских программ.

Это применение объяснено выше и применяется к функциям (FCs), функциональным блокам (FBs) и организационным блокам (OBs).

- Как использование памяти для передачи операционной системой информации для пользовательских программ.

Информация, поставляемая операционной системой для пользовательской программы имеет специальное название: "стартовая информация". Эта стартовая информация предоставлена только для организационных блоков (OBs) как интерфейс между операционной системой и пользовательской программой.

- Для передачи параметров в FCs.

**Где описаны
временные
локальные
данные**

Вы описываете временные локальные данные в пределах блока. Когда Вы создаете новый блок, Вы вначале описываете символьные имена временных переменных и затем используете их в пределах блока. Каждый класс приоритета имеет 256 байт переменных в S7-300. В S7-400 доступно 16 Кбайт, которые пользователь может делить между приоритетными классами когда назначает параметры для CPU.

3.12 Инструкции

Следующая таблица показывает краткий обзор используемых команд. Кроме того, она также показывает, какие команды могут быть преобразованы. Если команды не могут быть преобразованы, то будут показаны другие варианты преобразования.

Таблица 3–12 Инструкции в S5 и S7

Типы инструкций	S5	S7	Преобразование	Опции преобразования
Инструкции с аккумулятором	TAK, ENT, I, D, ADDBN, ADDKF, ADDDH	TAK, ENT, INC, DEC, +, Новые в S7: CAW, CAD, PUSH, POP, LEAVE	да	-
Инструкции с адресным регистром / Регистровые инструкции	MA1, MBR, ABR, MAS, MAB, MSB, MSA, MBA, MBS; TSG, LRB, LRW, LRD, TRB, TRW, TRD	Новые в S7: LAR1, LAR2, TAR1, TAR2, +AR1, +AR2, CAR	нет	Использование адресного регистра (AR1, AR2)
Инструкции битовой логики	A, AN, O, ON, A(, O(,), O, S, R, RB, RD= TB, TBN, SU, RU	A, AN, O, ON, A(, O(,), O, S, R, = SET; A, SET; AN, SET; S, SET; R Новые в S7: X, XN, X(, XN(, FP, FN, NOT, SET, CLR, SAVE	да	-
Инструкции таймеров	SP, SE, SD, SS/SSU, SF/SFD, FR, SEC	SP, SE, SD, SS, SF, FR, S T	да	-
Инструкции счетчиков	CU/SSU, CD/SFD, FR, SEC	CU, CD, FR, S C	да	-
Инструкции загрузки и передачи	L, LD, LW, LDW, TL PB, L QB, L PW, L QW, T PB, T QB, T PW, T QW	L, LC, T L PIB, L PIW, T PQB, T PQW	да	-
	LY GB / GW / GD / CB / CW / CD, LW GW / GD / CW / CD, TY GB / GW / GD / CB / CW / CD, TW GW / GD / CW / CD		нет	Замена на доступ к области периферии

(продолжение на следующей странице)

Таблица 3–12 Инструкции в S5 и S7, продолжение

Типы инструкций	S5	S7	Преобразование	Опции преобразования
Встроенные математические инструкции	+F, -F, xF, :F, +D, -D	+I, -I, *I, /I, +D, -D, *D, /D Новые в S7: MOD	да	-
Математические инструкции с плавающей точкой	+G, -G, xG, :G	+R, -R, *R, /R	да	-
Инструкции сравнения	!=F, >>F, >F, <F, >=F, <=F, !=D, ><D, D, <D, >=D, <=D, !=G, ><G, >G, <G, >=G, <=G	==I, <>I, >I, <I; >=I, <=I, ==D, <>D, >D, <D, >=D, <=D, ==R, <>R, >R, <R, >=R, <=R	да	-
Инструкции преобразования	CFW, CSW, CSD DEF, DED, DUF, DUD, GFD, FDG	INVI, NEGI, NEGD, BTI, BTD, DTB, ITB, RND, DTR Новые в S7: ITD, RND+, RND-, TRUNC, INVD, NEGR	да	-
Инструкции пословной логики	AW, OW, XOW	AW, OW, XOW Новые в S7: AD, OD, XOD	да	-
Инструкции смещения и циклического сдвига	SLW, SLD, SRW, SRD, SVW, SVD, RLD, RRD	SLW, SLD, SRW, SRD, SSI, SSD, RLD, RRD Новые в S7: RLDA, RRDA	да	-
Инструкции блока данных	G, CX	OPN	да	
	G, GX	SFC22	нет	Замещение вызовом SFC22 CREATE_DB
		Новое в S7: CDB L DBLG, L DBNO, L DILG, L DINO		

(продолжение на следующей странице)

Таблица 3–12 Инструкции в S5 и S7, продолжение

Типы инструкций	S5	S7	Преобразование	Опции преобразования
Инструкции логического управления, переходы	JU, JC, JN, JZ, JP, JM, JO, JOS, JUR	JU, JC, JN, JZ, JP, JM, JO, JOS Новые в S7: JCN, JCB, JNB, JBI, JNBI, JMZ, JPZ, JUO, LOOP, JL	yes	-
Инструкции для работы с блоками управления	JU, JC, DOU, DOC, BE, BEU, BEC	CALL, BE, BEU, BEC	да	-
Команда управления выводами инструкции "Master control relay"	BAS, BAF	Новые в S7: MCRA, MCRD, MCR(,)MCR	нет	Замещение вызовом SFC26, SFC27 или инструкциями контроля управления
Команды останова	STP, STS, STW	SFC46	нет	Замещение вызовом SFC46 STP
Функции обработки	DO <Формальный параметр>	–	нет	Открытие DB / задание номера блока программируется заново
	DO FW, DO DW	Косвенная адресация через память	да	Рекомендации: замещение косвенной адресацией через адресный регистр
	DO RS	Межзонная регистровая косвенная адресация	нет	Замещение должно проводиться с помощью косвенной адресации (см. раздел 3.13.4)
Абсолютная адресация памяти	LIR, TIR, LDI, TDI	-	нет	Замещение должно проводиться с помощью косвенной адресации (см. раздел 3.13.4)
Передача блоков	TNB, TNW, TXB, TXW	SFC20	нет	Замещение вызовом SFC20 BLKMOV
Команды прерывания <i>(продолжение на следующей странице)</i>	LIM, SIM, IAE, RAE, IA, RA	SFC39 to 42	нет	Замещение вызовом SFC39 – 42

Таблица 3–12 Инструкции в S5 и S7, продолжение

Типы инструкций	S5	S7	Преобразование	Опции преобразования
Команды страницы	ACR, TSC, TSG	-	нет	S7 не имеет страниц доступа
Математические функции	-	ABS, COS, SIN, TAN, ACOS, ASIN, ATAN, EXP, LN	-	-
Нулевые инструкции	BLD xxx NOP 0, NOP 1	BLD xxx NOP 0, NOP 1	да	-

3.13 Адресация

3.13.1 Абсолютная адресация

Абсолютная адресация в S5 и S7 идентична с одним исключением:

В S7 адресация данных в блоках данных происходит **в байтах**; таким образом адрес слова в S5 преобразуется в байтовый адрес путем умножения на 2.

Следующая таблица показывает задание области адресации при преобразовании:

S5	S7
DL 0, 1, 2, 3, ...255	DBB 0, 2, 4, 6, ...510
DR 0, 1, 2, 3, ...255	DBB 1, 3, 5, 7, ...511
DW 0, 1, 2, 3, ...255	DBW 0, 2, 4, 6, ...510
DD 0, 1, 2, 3, ...254	DBD 0, 2, 4, 6, ...508
D x.y	DBX 2 x.y для $8 \leq y \leq 15$ DBX (2 x+1).y для $0 \leq y \leq 7$

3.13.2 Символьная адресация

Символьная адресация S5 также используется в S7. Однако, здесь применяются новые опции для создания и использования символов. В программировании нет никаких различий.

Символы в S5

Символы для программ STEP 5 описаны при помощи редактора символов. Этот редактор создает список назначений, который позволяет Вам использовать символы вместо абсолютной адресации.

Символы в S7

В S7 символы могут быть размером до 24 знаков.

Глобальные символы

В STEP 7 также есть редактор символов, но список присвоений (ZULI) теперь известен как "таблица символов". В нем описаны все глобальные символы, такие как вводы, выводы, меркерная память (меркеры) и блоки.

Когда Вы назначаете символы с помощью редактора символов, они действительны в программах CPU.

Локальные символы	<p>Кроме возможности описывать символы с помощью редактора символов, STEP 7 также дает Вам опцию определения локальных символов для адресов данных и для локальной области данных при программировании блоков.</p> <p>Если Вы назначаете символы в пределах блока вместо назначения их с помощью редактора символов, этот символ будет приемлем только для рассматриваемого блока. В этом случае символ - "локальный для блока".</p>
Когда символы описываются?	<p>В STEP 7 точно не оговорено, когда Вы можете задавать символы. Для выполнения этого Вы можете использовать следующие два варианта:</p> <ul style="list-style-type: none">• Задайте их перед началом программирования (Это необходимо, если пользовательская программа вводится в инкрементном редакторе, то есть если синтаксис программы проверяется после ввода каждой строки.)• Задайте их после создания пользовательской программы, но перед компилированием (Это требуется при входе в программу в свободном режиме редактирования; то есть если программа создана как файл ASCII (исходный файл))
Импортирование таблицы символов	<p>В S7 для создания и редактирования таблицы символов используется редактор.</p> <p>Вы можете импортировать таблицы, созданные Вами с помощью других инструментов, в вашу таблицу символов и затем редактировать их далее. Например, импорт функций может быть использован при создании списка присвоений в STEP 5/ST после преобразования.</p> <p>Доступны следующие форматы данных: *.SDF, *.ASC, *.DIF, and *.SEQ.</p> <p>Для импорта таблицы символов выполните следующее:</p> <ol style="list-style-type: none">1. Откройте программную папку S7, содержащую в проектном окне таблицу символов.2. Дважды кликните на папке "Symbols" для открытия таблицы символов.3. Выберите команду меню Symbol Table ► Import в окне папки таблицы символов. Будет отображено диалоговое окно.4. В диалоговом окне выберите таблицу символов, которую Вы хотите импортировать и затем нажмите на кнопку "Open".5. Проверьте записи данных в таблице символов и сделайте необходимые исправления.6. Сохраните и закройте таблицу символов.

Замечание

Таблица символов в *.SEQ формате файла, которая была преобразована от S5 к S7, не может быть снова импортирована в S5. Формат файла *.DIF рекомендуется для обмена таблицами символов между S5 и S7.

Для дальнейшей информации относительно таблиц символов, см. руководство пользователя /231/.

3.13.3 Новая возможность: Полноценный доступ к адресам данных

Полноценный доступ означает, что блок данных определен одновременно с адресом данных. Это было невозможно в S5.

Полноценный доступ может задаваться как абсолютно, так и символично. Совместное использование абсолютной и символической адресации в пределах одного операнда не возможно.

Пример

L DB100.DBW6

L DB_MOTOR.SPEED

DB_MOTOR является символическим именем блока данных DB100 и определен в таблице символов. MOTOR.SPEED является адресом данных, который был описан в блоке данных. Это означает, что символическое имя для адреса данных (DB_MOTOR.SPEED) является столь же уникальным, как и абсолютный адрес (DB100.DBW6).

Полноценный доступ к адресу данных возможен только при открытии соответствующего глобального блока данных, открытого с помощью регистра DB (DB-register). При выполнении полноценного доступа STL редактор выполняет следующие инструкции:

1. Открыть блок данных через DB регистр (OPN DB100)
2. Обращение к адресу данных (L DBW 6)

Возможное использование операций полноценного доступа к данным

Вы можете использовать опцию полноценного доступа для всех инструкций, что позволяет определять адрес данных для всех типов данных.

Адрес данных с полноценным доступом может также быть определен как параметр блока. Это настоятельно рекомендуется, так как активирует нужный блок данных, открываемый при вызове программного блока. Полная адресация гарантирует, что правильный адрес данных будет передан из нужного блока данных.

**Опасности
"Частичной
адресации"**

В принципе, возможно обратиться к адресам данных тем же самым способом как в STEP 5 ("частичная адресация").

Пример:

L DBW 6
L SPEED

В STEP 7 это может вызвать проблему, так как STEP 7 при выполнении различных операций изменяет регистры для S7-300/S7-400 CPU. В некоторых случаях номер DB в регистре DB может быть перезаписан.

Регистр DB может быть перезаписан в следующих ситуациях, на которые следует обратить особое внимание:

- DB регистр перезаписывается при использовании доступа к данным с полноценным доступом.
- При вызове функционального блока (FB) регистр блока данных для вызываемого блока перезаписывается.
- После того, как сделан вызов функции (FC), в котором используются параметры комплексных типов данных (STRING, DATE_AND_TIME, ARRAY, STRUCT, или UDT), содержание DB регистра для вызываемого блока будет перезаписано.
- После того, как Вы назначили фактический параметр для FC, со ссылкой на глобальный блок данных (DB100.DBX0.1), STEP7 открывает блок данных (DB100), и перезаписывает содержание регистра DB
- После обращения к параметрам in/out FB с комплексным типом данных STRING, DATE_AND_TIME, ARRAY, STRUCT или UDT, STEP 7 использует DB регистр для доступа к данным
- . Этот шаг перезаписывает содержимое DB регистра.
- После обращения к параметрам FC input, output или in/out с комплексным типом данных STRING, DATE_AND_TIME, ARRAY, STRUCT, или UDT STEP 7 использует DB регистр для доступа к данным. Этот шаг перезаписывает содержимое DB регистра.

3.13.4 Косвенная адресация

Косвенная адресация, использующая функцию “DO” S5, была заменена в S7 новой косвенной адресацией через память и косвенной адресацией через адресный регистр.

Формат указателя в STEP 5

В S5 указатель для обозначения выполняемой операции занимает одно слово. Структура указателя описана на рис. 3–7:

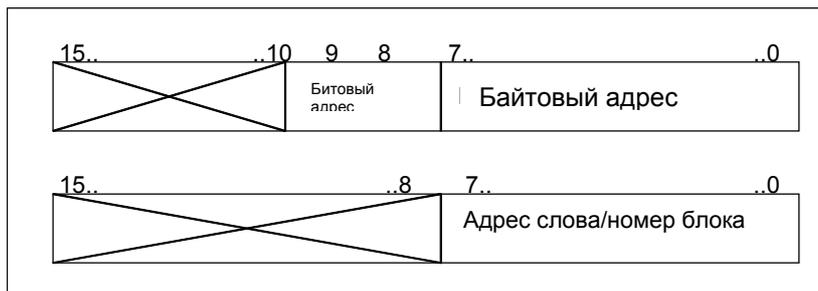


Рис.3-7 Структура указателя S5

Формат указателя в STEP 7

В S7 есть два возможных формата указателя: слово и двойное слово.

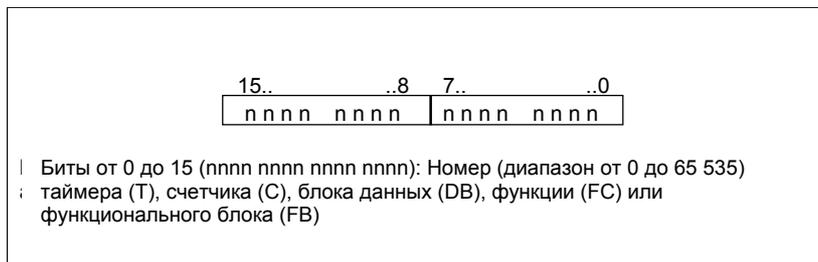


Рис.3-8 Указатель в формате слова для косвенной адресации через память

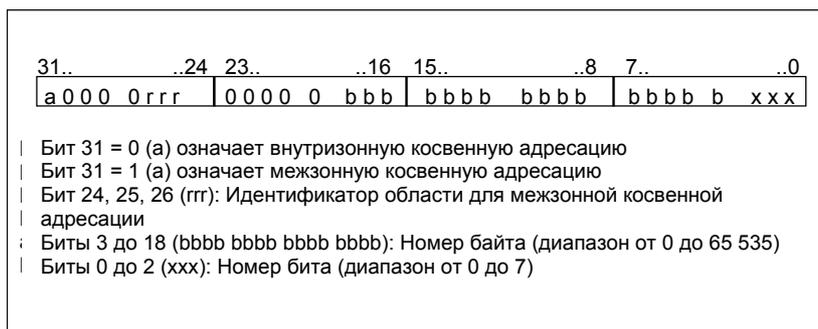


Рис.3-9 Указатель в формате двойного слова для косвенной адресации через память и через адресный регистр

Косвенная адресация через память

Косвенная адресация через память соответствует косвенной адресации в S5. При выполнении косвенной адресации через память, указанная область памяти задает значение адреса, который обрабатывается в результирующей команде. Адрес состоит из следующих частей:

- Идентификатор адреса, такой как “IB” для “входного байта”, и
- Слово, содержащее номер таймера (Т), счетчика (С), блока данных (DB), функции (FC) или функционального блока (FB), или
- Двойное слово, которое определяет точный адрес в пределах области памяти, заданной идентификатором адреса.

Адрес содержит указатель для косвенного задания адреса значения или номера. Это слово или двойное слово может быть определено в одной из следующих областей:

- Меркерная память (M)
- Блок данных (DB)
- Экземпляр блока данных (DI)
- Локальные данные (L)

Преимущество косвенной адресации через память в том, что Вы можете динамически модифицировать адреса операций без редактирования программы.

Примеры

Следующие примеры показывают, как Вы можете работать с указателем в формате слова:

STL S5	STL S7	Объяснение
L KB 5 T FW 2 DO FW 2 L T 0	L +5 T MW 2 L T [MW 2]]	Загрузите значение 5 как целое число в ACCU 1. Переместите содержание из ACCU 1 в меркерное слово MW2. Загрузите значение времени таймера Т 5.

Следующие два примера показывают, как Вы можете с указателем в формате двойного слова.

STL S5	STL S7	Объяснение
L KB 8 T FY 3 L KB 7 T FY 2 DO FW 2 A I 0.0 DO FW 2 = Q 0.0	L P#8.7 T MD 2 A I [MD 2] = Q [MD 2]	Загрузите 2#0000 0000 0000 0000 0000 0000 0100 0111 (двоичное значение) в ACCU 1 (S7). Сохраните адрес 8.7 в слове памяти FW 2 (S5) / двойном слове памяти MD 2 (S7). Контроллер делает запрос ввода I 8.7 и назначает его состояние выходу Q 8.7.

STL S5	STL S7	Объяснение
L KB 8 DO FW 2 DO FW 2 L IB 0 DO FW 2 T FW 0	L P#8.0 T MD2 L IB [MD2] T MW [MD2]	Загрузите 2#0000 0000 0000 0000 0000 0000 0100 0000 (двоичное значение) в ACCU 1 (S7). Сохраните адрес 8 в меркерном слове FW 2 (S5) / меркерном двойном слове MD 2 (S7). Контроллер загружает входной байт IB 8 и передает содержание меркерному слову FW 8 (MW 8 в STEP 7).

Использование правильной последовательности синтаксиса

При работе с косвенной адресацией через память с использованием в качестве области памяти блока данных, Вы должны сначала открыть блок данных с использованием инструкции "Open data block" (OPN). После этого Вы можете использовать слово данных или двойное слово данных как адрес в инструкциях косвенной адресации через память, как показано в следующем примере:

```
OPN          DB10
L            IB [DBD 20]
```

При доступе к байту, слову или двойному слову необходимо обеспечить битовый адрес указателя - "0".

Косвенная адресация через адресный регистр

В STEP 7, регистры адреса AR1 и AR2 могут использоваться для косвенной адресации.

При косвенной адресации, адрес определяет расположение в памяти и начальный адрес, с которого начнется обработка инструкции. Адрес состоит из следующих двух частей:

- Идентификатор адреса
- Адресный регистр и указатель для задания смещения, которое добавляется к содержанию адресного регистра, для задания точного адреса, который будет обрабатываться в инструкции.
Пример указателя: P#Byte.Bit.

Адрес косвенно указывает на адрес значения. Это сделано с использованием адресного регистра плюс постоянное смещение.

Инструкции с использованием внутризонной косвенной адресации через адресный регистр не изменяют значение в адресном регистре.

Для дальнейшей информации, смотри инструкцию по программированию *Statement List Programming Manual I232I*.

