

Создание логических блоков

3

В этой главе

Программа пользователя не может существовать без логических блоков. Во многих ситуациях Вы можете использовать логические блоки, встроенные в CPU, или имеющиеся в распоряжении стандартные функциональные блоки. Однако, Вам всегда потребуется создать ряд логических блоков самим. В этой главе описано, как создавать блоки с использованием редактора FUP.

В разделе	Вы найдете	на стр.
3.1	Создание логических блоков. Обзор	3-2
3.2	Логические блоки в редакторе	3-3
3.3	Структура таблицы описания переменных	3-6
3.4	Редактирование таблицы описания переменных. Обзор	3-8
3.5	Описание мультиэкземпляров	3-10
3.6	Присвоение системных атрибутов параметрам	3-11
3.7	Редактирование раздела операторов. Обзор	3-13
3.8	Основные рекомендации по вводу элементов FUP	3-15
3.9	Ввод элементов FUP	3-17
3.10	Редактирование адресов и параметров	3-20
3.11	Символическая адресация	3-21
3.12	Редактирование в режиме замены	3-23
3.13	Ввод заголовков и комментариев	3-24

3.1 Создание логических блоков. Обзор

Логические блоки

Логические блоки (OB, FB, FC) состоят из раздела описания переменных и раздела операторов (кодов). Они также обладают определенными свойствами. При программировании Вы должны редактировать следующие три раздела:

- **Таблица описания переменных:** В таблице описания переменных описываются параметры, системные атрибуты для параметров и локальные переменные блока.
- **Раздел операторов:** В разделе операторов Вы программируете код блока, который должен исполняться программируемым контроллером. Код состоит из одного или нескольких сегментов, содержащих элементы FUP.
- **Свойства блока:** Свойства блока содержат дополнительную информацию, например метку времени и путь, которые вносятся самой системой. Кроме того, Вы можете вводить дополнительные данные об имени, семействе, версии и авторе и назначать системные атрибуты для блоков (см. гл. 5).

Редактирование логического блока

Порядок, в котором редактируются эти три раздела не имеет значения, и Вы, конечно, можете вносить изменения и дополнения.

Когда Вы обращаетесь к символам из таблицы символов, Вы должны обеспечить, чтобы они были полностью описаны, и, если необходимо, добавить отсутствующую информацию.

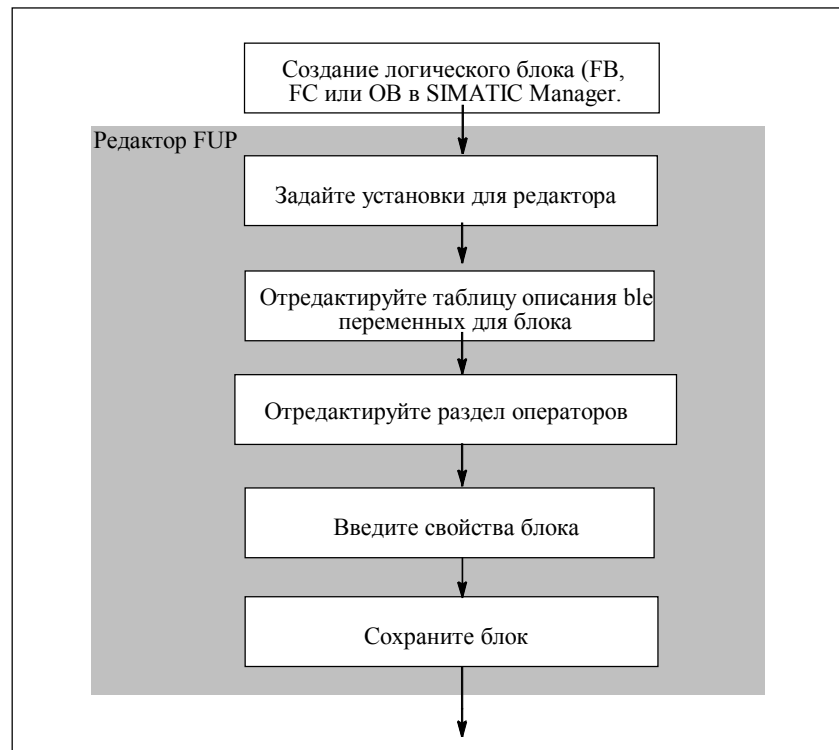


Рис. 3-1. Процедура создания логического блока в FUP

3.2 Логические блоки в редакторе

Обзор

Перед началом программирования в редакторе FUP Вам следует познакомиться с различными способами настройки редактора в соответствии с Вашими предпочтениями и методами работы.

Настройка редактора

С помощью команды меню **Options → Customize** [Параметры → Настроить] Вы можете открыть диалоговое окно с закладками. В закладке "Editor" ["Редактор"] Вы можете выполнить следующие основные настройки для программирования блока:

- Шрифт (тип и размер), используемый в тексте и таблицах.
- Язык программирования по своему выбору (FUP, KOP или AWL). Новый блок откроется в FUP, KOP или AWL в зависимости от выбранного Вами языка программирования. Учитывая определенные ограничения, Вы в дальнейшем можете переключаться из одного языка в другой и просматривать блок.

Отображение символов и комментариев в новом блоке (включить [on] или выключить [off])

Настройки для языка, комментариев и символов могут быть изменены в любое время в процессе редактирования с помощью команд меню **View → ...** [Просмотр →...].

Настройки для FUP

В закладке "LAD/FBD" (при выборе немецкого языка для пакета -"KOP/FUP"), которую Вы также можете отобразить командой меню **Options → Customize** [Параметры → Настроить], Вы можете сделать следующие основные настройки:

- Layout [компоновка]: определяет размер отображения сегментов. Выбранный размер определяет количество элементов FUP, которые Вы можете разместить один за другим в сегменте. Эта настройка влияет также на распечатку блока.
- Width of Address Field[ширина адресного поля]: определяет ширину адресного поля для адресов. Если ширина превышена, то строка обрывается. Большое адресное поле более практично для символической адресации, маленькое поле достаточно для абсолютной адресации.
- Element Representation [представление элемента]: Вы можете решить, будет ли отображение элемента FUP двух- или трехмерным.
- Line/Color [линия/цвет]: для выбранного элемента, контакта, выполнения или невыполнения статуса.

Основное окно редактора FUP

При открытии логического блока появляется окно, отображающее:

- **таблицу описания переменных** блока в верхней части
- **раздел операторов** в нижней части, в которой редактируется текущий кодовый блок

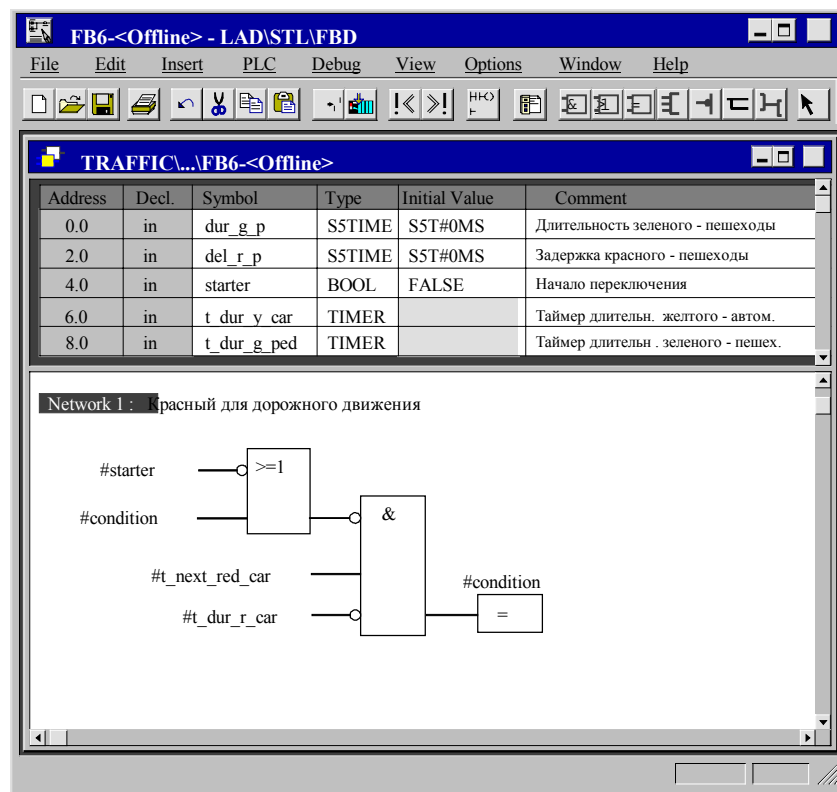


Рис. 3-2. Таблица описания переменных и раздел операторов в FUP

Пояснения к рис. 3-2: File -Файл, Edit - Редактировать, Insert - Вставить, PLC - Контроллер, Debug - Отладка, View - Вид, Options - Параметры, Window - Окно, Help - Помощь; Address - Адрес, Decl. - Описание, Symbol - Символ, Type - Тип, Initial Value - Начальное значение, Comment - Комментарий

Параметры блока редактируются в своем собственном диалоговом окне (см. гл. 5).

Редактор позволяет открывать и редактировать несколько блоков одновременно.

Связь между описанием переменных и разделом операторов

Таблица описания переменных и раздел операторов тесно связаны друг с другом, так как имена из таблицы описания переменных используются в разделе операторов. Это значит, что изменения в таблице описания переменных влияют также на раздел операторов и, следовательно, на весь блок.

Таблица 3–1. Связь между описанием переменных и разделом операторов

Действие в таблице описания переменных	Реакция в разделе операторов
Новый правильный элемент	Если существует неправильный код, ранее недопустимая переменная становится допустимой
Правильное изменение имени без изменения типа	Символ немедленно отображается повсюду с новым именем
Правильное имя заменяется неверным	Код не меняется
Недопустимое имя заменяется правильным	Если имеется недопустимый код, он становится допустимым
Изменение типа	Если имеется недопустимый код, он становится допустимым, а если имеется допустимый код, он становится недопустимым
Удаляется символ, используемый в коде	Допустимый код становится недопустимым
Изменение комментария	Отсутствует
Неправильный ввод новой переменной	Отсутствует
Удаление неиспользуемой переменной	Отсутствует
Изменение начального значения	Отсутствует

3.3 Структура таблицы описания переменных

Обзор

В таблице описания переменных устанавливаются локальные переменные, включая формальные параметры блока и системные атрибуты для параметров. Это оказывает следующие воздействия:

- В результате описания резервируется память в стеке локальных данных или в экземпляре блока данных.
- Устанавливая входные, выходные и проходные параметры, Вы также определяете "интерфейс" для вызова блока в программе.
- Описание переменных в функциональном блоке определяет структуру данных для любого экземпляра блока данных, соответствующего данному функциональному блоку.
- Устанавливая системные атрибуты, Вы присваиваете специальные свойства параметрам для конфигурации сообщений и соединений, функций интерфейса с оператором и конфигурации управления процессом.

Структура таблицы описания переменных

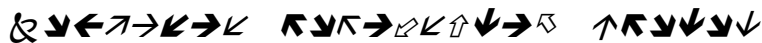
После открытия нового логического блока на экране появляется таблица описания переменных по умолчанию. Она перечисляет все допустимые типы описаний для конкретного блока (in, out, in_out, stat, temp) в правильном порядке.

При создании нового ОВ отображается стандартное описание переменных, в которой Вы можете изменять значения.

Таблица описания переменных содержит поля для адреса, описания, символического имени, типа данных, начального значения и комментария к переменным. Каждая строка представляет описание переменной. Переменные типа массив или структура требуют более одной строки.

FB40 - <Offline>					
Address	Decl.	Symbol	Data Type	Initial Value	Comment
0.0	in	in	BOOL	FALSE	Свет включен
0.1	in	start	BOOL	FALSE	Выключатель
2.0	out	Motor	BOOL	FALSE	Двигатель
2.1	out	Message	BOOL	FALSE	Сообщение
4.0	in_out	in_outp1	INT	0	
6.0	in_out	in_outp2	INT	0	

Рис. 3-3. Пример таблицы описания переменных



Значение столбцов

Столбцы в таблице описания переменных интерпретируются следующим образом:

Таблица 3–2. Столбцы таблицы описания переменных

Столбец	Значение	Примечания	Редактирование
Address [адрес]	Адрес в формате БАЙТ.БИТ	В случае типов данных, требующих более одного байта, адрес отображает это скачком к следующему байтовому адресу. Ключ: * : Размер элемента массива в байтах + : Адрес относительно начала структуры = : Общая потребность структуры в памяти	Системный ввод: адрес назначается и отображается системой по завершении ввода описания.
Decl. [описание]	Тип описания, “назначение” переменной	В зависимости от типа блока возможно следующее: Входные параметры “in” Выходные параметры “out” Проходные параметры “in_out” Статические переменные “stat” Временные переменные “temp”	Устанавливаются по умолчанию в соответствии с типом блока
Символ [символ]	Символическое имя переменной	Имя должно начинаться буквой. Резервированные ключевые слова не допускаются.	Обязательно
Data Type [тип данных]	Тип данных переменной (BOOL, INT, WORD, ARRAY и т.д.)	Основные типы могут быть выбраны в меню, отображаемом правой кнопкой мыши.	Обязательно
Initial Value [начальное значение]	Начальное значение, если программа не должна принимать значение по умолчанию	Должно быть совместимым с типом данных. если не было выбрано конкретное фактическое значение, то начальное значение используется в качестве фактического значения переменной при первом сохранении DB.	Необязательно
Comment [комментарий]	Комментарий к документации		Необязательно

Значение “флага”

Если Вы назначили переменной системные атрибуты (см. раздел 3.6), то в столбце “Symbol” появляется символ, напоминающий флаг (см. рис. 3–3). Дважды щелкните на этом флаге, чтобы открыть диалоговое окно “System Attributes” [“Системные атрибуты”]

Изменение ширины столбца

Вы можете изменять ширину столбцов, поместив курсор мыши между двумя столбцами и перемещая мышью горизонтально при нажатой левой клавише мыши. Ширину столбца можно изменить также командой меню **View → Column Width** [Вид → Ширина столбца], предварительно выбрав таблицу. Это позволяет минимизировать необязательны столбцы с комментариями и начальными значениями и сосредоточиться только на оставшихся столбцах.

3.4 Редактирование таблицы описания переменных. Обзор

Процедура

После ввода требуемого типа нового описания введите имя переменной, тип данных, начальное значение (необязательно) и комментарий (необязательно). Переходить к следующему полю можно с помощью клавиши TAB. В конце строки переменной автоматически присваивается адрес.

После редактирования каждого поля таблицы проверяется его синтаксис и любая ошибка отображается красным цветом. Здесь Вы можете продолжить редактирование, отложив исправление ошибок на более поздний этап.

Функции редактирования

При редактировании таблицы Вам доступны все обычные функции в меню **Edit** [Редактировать]. Использование контекстно чувствительной правой кнопки мыши облегчает редактирование.

Меню, отображаемое при нажатии правой кнопки мыши, поможет Вам также при вводе типа данных. Меню "Data Type" ["Тип данных"] включает в себя все элементарные типы данных.

Вы можете выбирать отдельные строки, щелкнув на защищенной от записи ячейке с адресом. Вы можете выбрать также несколько строк с одинаковым типом описания, удерживая в нажатом состоянии клавишу SHIFT. Выбранные строки появляются на черном фоне.

Изменение типа описания

Столбцы "Decl." доступны только для чтения. Тип описания определяется положением описания внутри таблицы. Это гарантирует, что переменные могут введены только в правильном порядке их типов описания. Если Вы хотите изменить тип описания, сначала вырежьте описание, а затем вставьте его под новым типом описания.

Ввод структур

Если Вы хотите ввести в качестве переменной структуру, введите ее имя в столбце "Symbol" и ключевое слово STRUCT в столбце с типом данных. Нажмите клавишу TAB или RETURN, чтобы вставить пустую строку и заключительную строку (END_STRUCT) для структуры. В пустой строке введите элемент структуры, указав его имя, тип данных и начальное значение (необязательно). Вы можете создать несколько строк и ввести дальнейшие элементы, используя команды меню или нажимая RETURN.

Если Вы хотите выбрать структуру, щелкните на адресе в первой или последней строке структуры (содержащей ключевое слово STRUCT или END_STRUCT). Вы можете выбирать отдельные описания внутри структуры, щелкая на ячейке с адресом в соответствующей строке.

Если Вы хотите ввести структуру внутри другой структуры, то иерархия отображается смещенными именами переменных.

Ввод массивов

Чтобы ввести в качестве типа данных массив, введите имя в столбце "Symbol" и ключевое слово ARRAY в ячейке для типа данных вместе с размером массива, например, ARRAY [1..20, 3..24] для двухмерного массива. Нажмите клавишу TAB (если необходимо, несколько раз) для вставки строки, в которой Вы можете ввести тип данных массива.

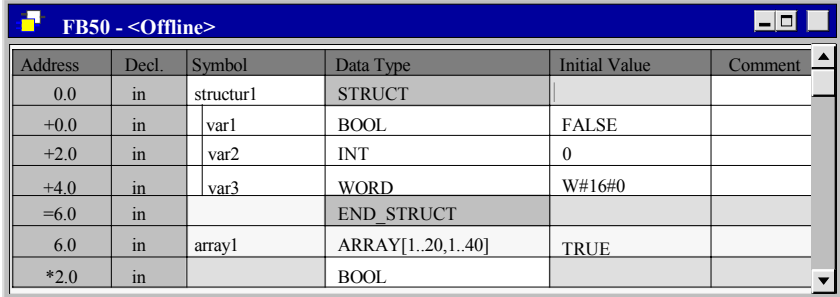
Если Вы хотите выбрать массив, щелкните на ячейке с адресом в соответствующей строке.

Начальные значения для каждого элемента массива можно вводить по одному или с коэффициентом повторения (см. рис. 3–4):

- Индивидуальный ввод: Вы присваиваете каждому элементу его собственное начальное значение. Значения отделяются запятыми.
- Коэффициент повторения: Вы назначаете одно и то же начальное значение нескольким элементам. Само значение показывается в круглых скобках, которым предшествует коэффициент повторения, определяющий количество элементов.

Пример

На рис. 3–4 показан пример таблицы описания переменных:



Address	Decl.	Symbol	Data Type	Initial Value	Comment
0.0	in	structur1	STRUCT		
+0.0	in	var1	BOOL	FALSE	
+2.0	in	var2	INT	0	
+4.0	in	var3	WORD	W#16#0	
=6.0	in		END_STRUCT		
6.0	in	array1	ARRAY[1..20,1..40]	TRUE	
*2.0	in		BOOL		

Рис. 3-4. Структуры и массивы в таблице описания переменных

Замечание

Если Вы вносите изменения в описание переменных блоков, вызовы которых Вы уже запрограммировали в программе, может возникнуть конфликт меток времени.

Поэтому Вы должны сначала запрограммировать все блоки, подлежащие вызову, а затем запрограммировать блоки, которые их вызывают. В случае функциональных блоков должны быть также заново созданы экземпляры блоков данных.

При внесении изменений в UDT, которые были внесены в качестве типа данных в описание переменных, проверьте описание переменных блока и снова его сохраните.

3.5 Описание мультиэкземпляров

Мультиэкземпляры

Мультиэкземпляры возникают из описания статической переменной того же типа данных, что и функциональный блок (FB). В разделе операторов этот экземпляр вызывается как элемент FUP.

За более подробной информацией о мультиэкземплярах обращайтесь к *Руководству по программированию /234/*. Часть 2 этого руководства объясняет синтаксис вызова мультиэкземпляров.

Правила

Запомните следующие правила при описании мультиэкземпляров:

- Описание мультиэкземпляров возможно только в функциональных блоках.
- Функциональные блоки, в которых были описаны мультиэкземпляры, должны иметь также связанные с ними экземпляр DB.
- Мультиэкземпляр может быть описан только как статическая переменная (тип описания "stat").

Ввод мультиэкземпляров

Чтобы описать мультиэкземпляр, введите имя переменной в столбце "Symbol" после типа описания "stat". В столбце "Data Type" введите функциональный блок. Это может быть сделано вводом абсолютного или символического имени FB. Вы можете также добавить необязательный комментарий.

Address	Decl.	Symbol	Data Type	Initial Value	Comment
0.0	in	varin	BYTE	B#16#0	
2.0	out	varout	BYTE	B#16#0	
4.0	in_out	varinout	BYTE	B#16#0	
	stat	locinst	FB6		локальн. экземпляр
0.0	temp	tempo	REAL		

Рис. 3-5. Описание мультиэкземпляра (пример)

3.6 Присвоение системных атрибутов параметрам

Системные атрибуты

Системные атрибуты можно назначить блокам и параметрам. Они влияют на конфигурацию сообщений и связей, функции связи с оператором и конфигурацию управления процессом.

С помощью стандартного пакета программного обеспечения STEP 7 можно назначить следующие атрибуты для параметров в таблице описания переменных.

Ввод системных атрибутов для параметров

Чтобы ввести системные атрибуты для параметров, выберите имя параметра в таблице описания переменных и выберите **Edit** → **Object Properties** [Редактировать → Свойства объекта] в строке меню, чтобы отобразить диалоговое окно “Properties” [“Свойства”]. Выберите закладку “System Attributes” [“Системные атрибуты”] и введите требуемый атрибут и его значение.

В таблице 3–3 показано, какие системные атрибуты можно ввести для параметров в таблице описания переменных.

Таблица 3–3. Системные атрибуты для параметров

Атрибут	Значение	Когда назначать атрибут	Допустимый тип описания
S7_server	connection, alarm_archiv	Если параметр соответствует конфигурации связей или сообщений. Этот параметр содержит номер связи или сообщения.	IN
S7_a_type	alarm, alarm_8, alarm_8p, alarm_s, notify, ar_send	Если параметр будет определять тип блока сообщений в блоке сообщений (возможен только, если атрибут S7_server установлен на alarm_archiv).	IN, только с блоками типа FB и SFB
S7_co	pbkl, pbk, ptpl, obkl, fdl, iso, pbks, obkv	Если параметр будет указывать на тип связи в конфигурации связей (возможен только, если атрибут S7_server установлен на connection).	IN
S7_m_c	true, false	Если параметр будет модифицироваться или управляться с панели оператора.	IN/OUT / IN_OUT, только с блоками типа FB и SFB
S7_shortcut	Любые 2 символа, например, W, Y	Этим параметром назначается краткая форма для оценки аналоговых значений.	IN/OUT / IN_OUT, только с блоками типа FB и SFB
S7_unit	Единица, например, литры	Этим параметром назначается единица для оценки аналоговых значений.	IN/OUT / IN_OUT, только с блоками типа FB и SFB
S7_string_0	Любые 16 символов, например, OPEN	Этим параметром назначается текст для оценки двоичных значений.	IN/OUT / IN_OUT, только с блоками типа FB, SFB, FC и SFC
S7_string_1	Любые 16 символов, например, CLOSE	Этим параметром назначается текст для оценки двоичных значений.	IN/OUT / IN_OUT, только с блоками типа FB, SFB, FC и SFC
S7_visible	true, false	Если Вы не хотите, чтобы параметр отображался в CFC.	IN/OUT / IN_OUT, только с блоками типа FB, SFB, FC и SFC
S7_link	true, false	Если Вы не хотите, чтобы параметр связывался в CFC.	IN/OUT / IN_OUT, только с блоками типа FB, SFB, FC и SFC
S7_dynamic	true, false	Если Вы хотите, чтобы параметр обладал динамическими свойствами при тестировании в CFC.	IN/OUT / IN_OUT, только с блоками типа FB, SFB, FC и SFC
S7_param	true, false	Если Вы хотите, чтобы параметр был защищен от неправильного присвоения значений в CFC.	IN/OUT / IN_OUT, только с блоками типа FB, SFB, FC и SFC

3.7 Редактирование раздела операторов. Обзор

Раздел операторов

В разделе операторов Вы описываете последовательность выполнения программы в Вашем логическом блоке. Чтобы сделать это, Вы формируете сегменты с элементами FUP. В большинстве случаев раздел операторов логического блока составляется из нескольких сегментов. После ввода элемента FUP редактор проводит синтаксический контроль и показывает, были ли вводы неверными (ошибки отображаются красным цветом). Неправильно расположенные элементы отвергаются с сообщением об ошибке.

Редактируемые части раздела операторов

В разделе операторов можно редактировать заголовок блока, названия сегментов, комментарии к блоку, комментарии к сегментам и, конечно, команды внутри сегментов.

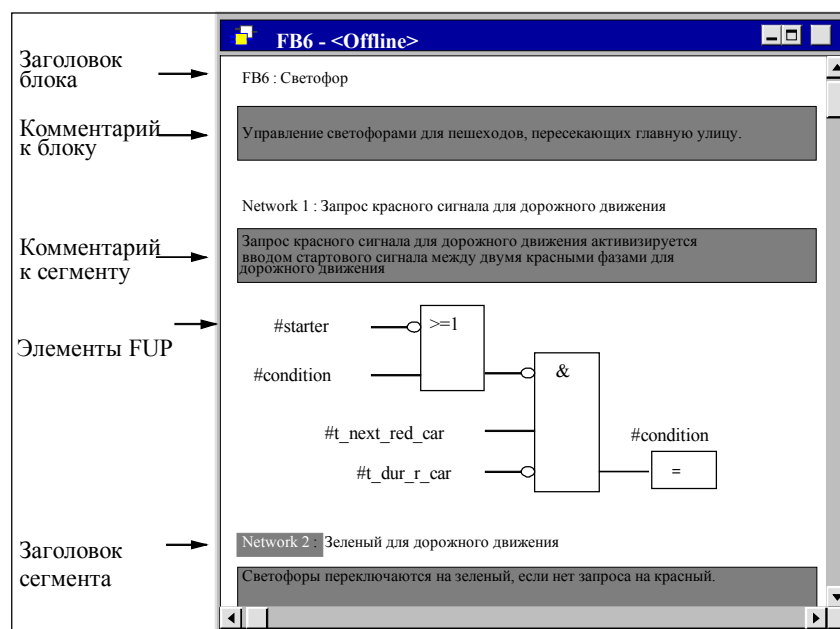


Рис. 3-6. Структура раздела операторов

Ввод новых блоков

Последовательность выполнения каждого из следующих шагов не фиксирована. При программировании раздела операторов нового блока мы рекомендуем Вам действовать следующим образом:

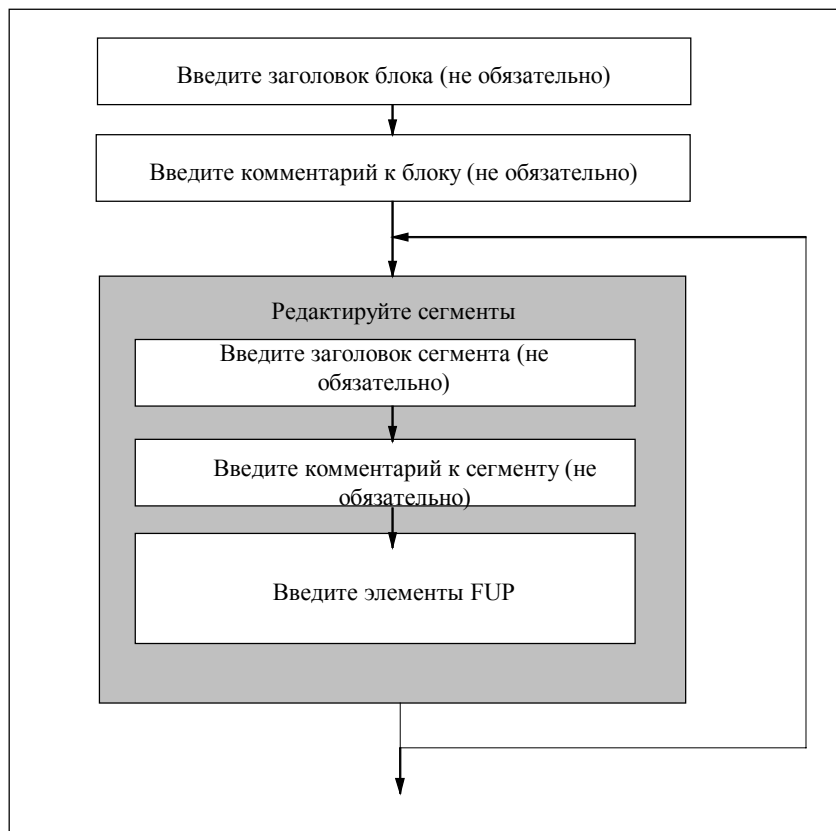


Рис. 3-7. Редактирование раздела операторов

Изменения можно вводить в режиме вставки или в режиме замены. Переключение между этими режимами производится клавишей INSERT.

3.8 Основные рекомендации по вводу элементов FUP

Обзор

Сегмент FUP может содержать несколько элементов в различных ветвях. Все элементы должны быть соединены между собой (IEC 1131-3).

При программировании в FUP Вы должны соблюдать некоторые правила. Все ошибки сопровождаются соответствующим сообщением.

Размещение блоков

Стандартные блоки (триггеры, счетчики, таймеры, математические операции и т.д.) могут быть присоединены к двоичным логическим операциям (&, >=1, XOR). Блоки сравнения являются исключением из этого правила.

Отдельные логические операции не могут программироваться с отдельными выходами в одном сегменте. Однако, используя Т-образное ответвление, Вы можете сделать несколько присваиваний в одной последовательности логических операций. Пример показан на рис. 3–8 .

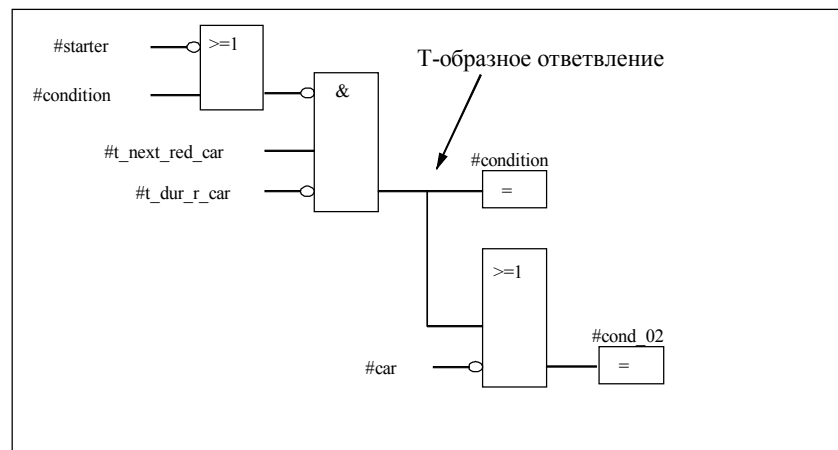


Рис. 3-8. Сегмент с двумя присваиваниями

Следующие блоки могут быть расположены только у правого края строки логических операций, где они завершают строку:

- Установка значения счетчика
- Прямой счет, Обратный счет
- Импульсный таймер, Таймер с продленным импульсом
- Задержка включения с запоминанием/Задержка выключения

Имеются блоки, которые требуют булевых логических операций, и блоки, которые не допускают булевых логических операций.

- **Блоки, требующие булевых логических операций:**

Выход, Установить выход, Сбросить выход –[R]

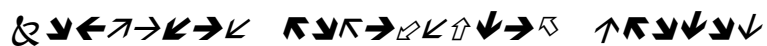
Промежуточный выход (коннектор) –[#]–, Положительный фронт –[P]–, Отрицательный фронт –[N]–

Все счетчики и таймеры

Перейти–если–не –[JMPN]

Включить главное управляющее реле –[MCR<]

Сохранить результат логической операции в BR-регистр –[SAVE]



Возврат –[RET]

- **Блоки, не допускающие логических операций:**

Активизировать главное управляющее реле [MCRA]

Деактивировать главное управляющее реле [MCRD]

Открыть блок данных [OPN]

Выключить главное управляющее реле [MCR>]

Все остальные блоки могут или иметь булевы логические операции, или не иметь их в зависимости от ситуации.

Разрешить вход/ Разрешить выход

Подключение деблокировки входа EN или деблокировки выхода ENO возможно, но не является абсолютно необходимым.

Удаление и модификация блоков

При удалении блока удаляются также все ветви, подключенные к булевым входам блока, за исключением главной ветви.

Самый простой способ замены элементов одного типа состоит в использовании режима замены (см. раздел 3.12).

3.9 Ввод элементов FUP

Операции над сегментами

Чтобы выбрать сегмент для ввода элементов FUP, щелкните на имени сегмента (например, "Network 1"). Когда сегмент выбран, Вы можете его вырезать, вставить его снова или скопировать.

Для создания нового сегмента выберите команду меню **Insert → Network** [Вставить → Сегмент] или щелкните на соответствующей кнопке на панели инструментов. Новый сегмент вставляется под выбранным сегментом.

Если вводится больше элементов, чем может быть отображено на экране, то сегмент сдвигается влево. Для получения лучшего обзора Вы можете настроить изображение командой меню **View → Zoom In/Zoom Out/Zoom Factor** [Вид → Распахнуть/Сжать/Коэффициент масштабирования].

Выбор объектов в сегменте

Переход в сегмент осуществляется щелчком мыши на элементе FUP в этом сегменте. Внутри сегмента в принципе можно выбрать щелчком мыши три области:

- Элементы FUP, например, блок AND, или стандартный блок, например, счетчик
- Входы и выходы блока
- Пустые элементы (соединительные линии)

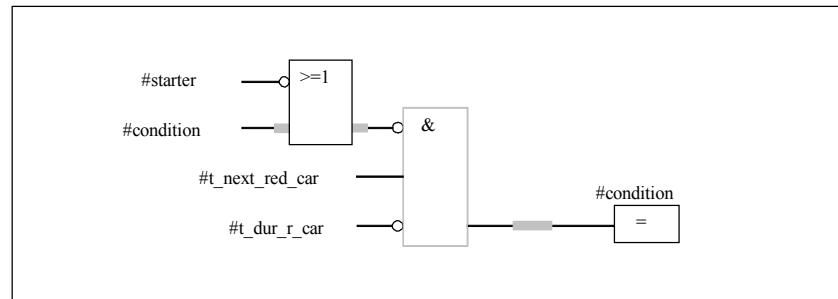


Рис. 3-9. Возможности выбора в сегменте FUP

Вы можете изменить цвет выбираемых объектов, перейдя с помощью команды меню **Options → Customize** [Параметры → Настроить] во вкладку "LAD/FBD" (в немецкоязычном варианте "KOP/FUP").

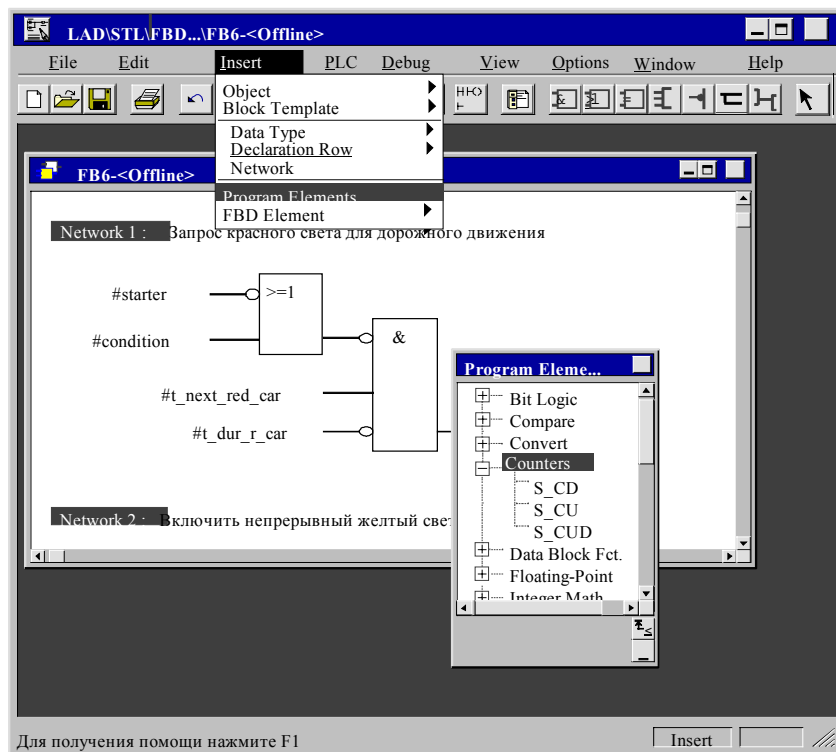
Разрыв и повторное соединение строки логических операций

Вы можете разорвать строку логических операций в выбранной точке ввода, нажав клавишу DEL, включить в разрыв новые элементы FUP и вновь соединить строку с помощью команды **Insert → FBD Element → Connection** [Вставить → Элемент FUP → Соединение]. Это возможно только в том случае, если Вы выбрали двоичный вход и двоичный выход. Если необходимо, элементы затем графически переупорядочиваются.

Ввод элементов FUP

Для вставки элементов FUP имеются в распоряжении следующие возможности:

- Ввести блок AND, блок OR или выход с помощью функциональных клавиш F2, F3 или F4.
- Щелкнуть на кнопке для элемента FUP на панели инструментов.
- Выбрать элемент из меню **Insert → FBD Element** [Вставить → Элемент FUP].
- Выбрать элементы из окна списка в диалоге **Program Elements** [Элементы программы] (см. рис. 3-10). Чтобы отобразить это диалоговое окно, используйте команду меню **Insert → Program Elements** [Вставить → Элементы программы], кнопку на панели инструментов или функциональную клавишу F11.



Пояснения к рис. 3-10:

В меню Program Elements [Элементы программы]: Bit Logic - битовая логика, Compare - сравнение, Convert - преобразование, Counters - счетчики, Data Block Fct. - функции блоков данных, Floating-Point - операции с плавающей точкой, Integer Math - математика целых чисел.

Элементы FUP вставляются позади выделенного в данный момент элемента.

Указание

Если в окне списка "Program Elements" Вы выберете группу "Libraries"

["Библиотеки"], то будут перечислены стандартные библиотеки STEP 7 и любые библиотеки, которые Вы создали.

Таким способом Вы можете очень быстро включать в свой сегмент целые блоки и программировать вызовы для других блоков.

Вызов мультиэкземпляров

Вы можете также вызывать мультиэкземпляры как элементы FUP, если Вы определили их в таблице описания переменных. Для этого выберите команду меню **Insert → Program Elements** [Вставить → Элемент программы]. В окне списка элементов FUP Вы найдете группу "Multiple Instances" ["Мультиэкземпляры"], где перечислены все описанные мультиэкземпляры.

3.10 Редактирование адресов и параметров

Использование

Редактор блоков использует строку символов ???.? как трафарет для адресов и параметров при вставке элемента FUP. В разделе операторов все адреса и параметры должны быть правильно и полностью введены. Исключение составляют FB и SFB или таймеры и счетчики, которые не нуждаются в назначении всех параметров. Адреса и параметры могут вводиться в абсолютной или символической форме.

Последовательность действий

Для редактирования адресов или параметров откройте соответствующее текстовое окно, щелкнув на пустой строке ???.?. По завершении ввода проверяется ее синтаксис. При обнаружении ошибки адрес или параметр отображается красным цветом, а в строке состояния появляется сообщение об ошибке. Если синтаксис верен, открывается следующее текстовое окно, которое еще не редактировалось.

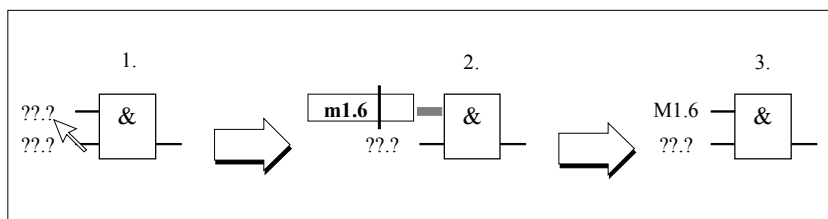


Рис. 3-11. Ввода адресов для элементов FUP

Познакомившись с инструментами редактирования FUP, Вы можете сначала ввести все элементы в сегменте, а затем назначить адрес или параметр каждому элементу.

Отладка

Ошибки легко распознаются, так как они помечены красным цветом. Чтобы дать Вам возможность более легко находить ошибки, расположенные за пределами видимого поля экрана, у редактора есть две функции поиска: **Edit → Go To → Previous Error/Next Error** [Редактировать → Перейти → Предыдущая ошибка/Следующая ошибка], которые могут быть также активизированы кнопками на панели инструментов.

Поиск ошибок распространяется за пределы сегмента. Это значит, что ошибки ищутся по всему разделу операторов, а не только внутри одного сегмента или видимого в данный момент участка программы. Если активизировать строку состояний командой меню **View → Status Bar** [Вид → Строка состояний], то информация об ошибках будет отображаться в этой строке.

Вы можете исправлять ошибки и производить изменения в режиме замены (см. раздел 3.12).

3.11 Символическая адресация

Использование символической адресации

В языке программирования FUP можно вводить абсолютные адреса, параметры и имена блоков или использовать символы. С помощью команды меню **View → Symbolic Representation** [Вид → Символическое представление] Вы можете переключаться между абсолютной и символической адресацией.

Определение символов

Для использования глобальных (разделяемых) символов Вы должны внести их в таблицу символов следующим образом:

- Откройте таблицу символов командой меню **Options → Symbol Table** [Параметры → Таблица символов].
- Используя команду меню **Options → Edit Symbols** [Параметры → Редактировать символы], откройте диалоговое окно, в котором Вы можете определять и модифицировать отдельные символы.

За дальнейшей информацией о редактировании символов обращайтесь, пожалуйста, к *Руководству пользователя STEP 7 /231/*.

Представление

В большинстве случаев нет необходимости определять, является ли символ локальным или глобальным. Однако, в тех случаях, когда может возникнуть путаница, например, если один и тот же символ используется как в таблице символов, так и в таблице описания переменных, Вы можете различить эти символы следующим образом:

- Символы из **таблицы символов** отображаются в кавычках `”..”`.
- Символам из **таблицы описания переменных** блока предшествует знак `”#”`.

Вам нет необходимости самим вводить эти различительные знаки. Если символические адреса содержатся в таблице описания переменных или в таблице символов, их идентификация выполняется после синтаксического контроля.

Легкость получения информации о символах

Для облегчения программирования с помощью символической адресации Вы можете отобразить абсолютный адрес и комментарий для символа. Эта информация отображается с помощью команды меню **View → Symbol Information** [Вид → Информация о символах]. Если Вы выбрали этот параметр, то после каждого сегмента отображается текстовое окно. В этом окне нельзя редактировать. Любые требуемые Вам изменения должны быть выполнены в таблице символов или в таблице описания переменных.



Рис. 3-12. Информация о символах в FUP

При выводе блока на печать распечатка имеет вид текущего экрана и сопровождается соответствующей инструкцией и комментариями к символам.

Замечание

При загрузке программы в CPU таблица символов не загружается. Это значит, что при редактировании программы пользователя, оригинал которой отсутствует в устройстве программирования или PC, первоначальные символы становятся недоступными.

3.12 Редактирование в режиме замены

Замена адресов и параметров

В редакторе блоков удобно изменять адреса или параметры, используя режим замены. Для переключения между режимами вставки и замены используется клавиша INSERT. В режиме замены Вы пишете поверх своих записей в текстовых окнах для адресов или параметров.

Замена элементов FUP

Режим замены позволяет заменять элементы FUP одного и того же типа. Все булевы логические сопряжения и параметры сохраняются.

Это имеет то преимущество, что Вам нет необходимости заново вводить адреса и параметры. Элемент FUP, который Вы хотите переписать, может быть заменен только элементом FUP того же самого типа. Например, Вы можете заменить AND, OR, R/S-триггер триггером S/R или заменить один таймер другим.

Чтобы заменить существующий элемент FUP, выделите его и переключитесь в режим замены, нажав клавишу INSERT. Элемент FUP будет переписан, как только Вы вставите другой элемент FUP того же типа.

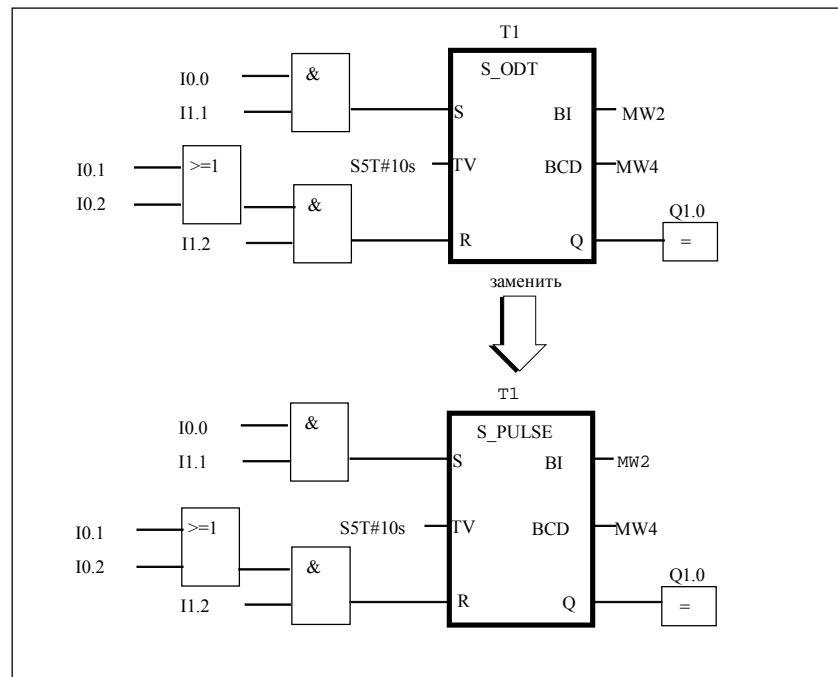


Рис. 3-13. Замена блоков

Указание

Вы можете исправлять комментарии и заголовки в режиме замены.

3.13 Ввод заголовков и комментариев

Обзор

В разделе операторов Вы можете вводить такую информацию, как заголовки блока и сегментов и комментарии к блоку и к сегментам. Эти записи необязательны и не существенны для выполнения программы.

Ввод заголовков блоков и сегментов

Для ввода заголовка блока или сегмента поместите курсор на три вопросительных знака справа от имени блока или сегмента (например, network 1 : ???). Открывается текстовое окно, в котором Вы можете ввести заголовок. Его длина может быть до 64 символов.

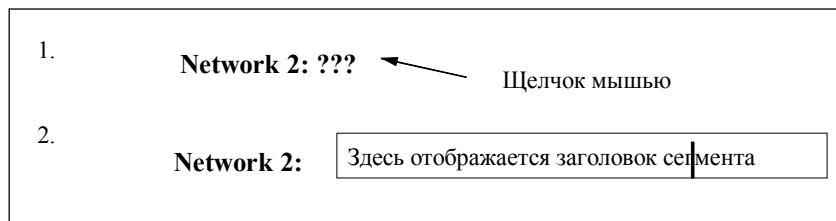


Рис. 3-14. Ввод заголовков

Ввод комментариев

Используя команду меню **View → Comment** [Вид → Комментарий], Вы можете отобразить или спрятать серое поле для комментариев. При двойном щелчке на этом поле появляется текстовое окно, которое Вы можете использовать для ввода комментариев. В Вашем распоряжении имеется 64 Кбайта на блок для ввода комментариев к блоку и сегментам.

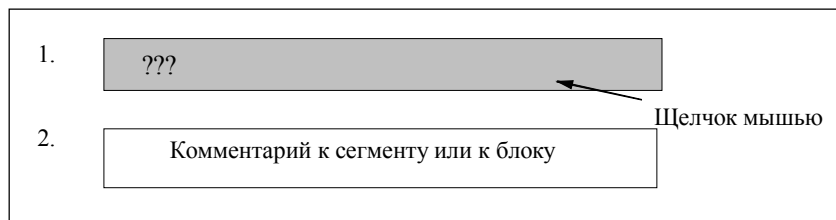


Рис. 3-15. Ввод комментариев

Замечание

При загрузке блока в CPU комментарии не загружаются. При выгрузке из CPU блока, оригинал которого отсутствует на Вашем устройстве программирования или PC. Вы не сможете просматривать или редактировать исходные комментарии.