

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Южно-Уральский государственный университет
Кафедра «Электропривод и автоматизация
промышленных установок»

681.5(07)
Б825

А.М. Борисов, А.С. Нестеров, Н.А. Логинова

ПРОГРАММИРУЕМЫЕ УСТРОЙСТВА АВТОМАТИЗАЦИИ

Учебное пособие

Челябинск
Издательский центр ЮУрГУ
2010

УДК 658.564(075.8) + 681.51(075.8)
Б825

*Одобрено
учебно-методической комиссией энергетического факультета.*

Рецензенты:
В.С. Жабреев, В.М. Попов

Б825 Борисов, А.М.
Программируемые устройства автоматизации: учебное пособие /
А.М. Борисов, А.С. Нестеров, Н.А. Логинова. – Челябинск: Издательский
центр ЮУрГУ, 2010. – 186 с.

Учебное пособие предназначено для студентов специальности 140604 «Электропривод и автоматика промышленных установок и технологических комплексов» при изучении дисциплины «Автоматизация типовых технологических процессов».

В пособии представлены конструкции, технические характеристики, используемые языки и принципы программирования программируемых логических контроллеров и сенсорных мониторов ведущих фирм Германии, Японии и США, нашедших широкое применение на российских предприятиях. Рассмотрено построение, конфигурирование и пользовательское программирование промышленной сети автоматизации PROFIBUS-DP.

Учебное пособие может быть использовано студентами других специальностей, а также инженерно техническими работниками предприятий, занимающихся разработкой систем автоматизации.

УДК 658.564(075.8) + 681.51(075.8)

© Издательский центр ЮУрГУ, 2010

ВВЕДЕНИЕ

Предлагаемое учебное пособие предназначено для освоения современных программируемых средств, используемых для автоматизации промышленного производства. В пособии представлены наиболее широко используемые на российских предприятиях изделия иностранных фирм, таких как Siemens (Германия), Omron (Япония), Automation Direct (США). Пособие основано (полностью) на технической документации фирм-изготовителей этих изделий.

Излагаемый в пособии материал представляется в виде лабораторных работ, используемых при изучении курса «Автоматизация типовых технологических процессов» в Южно-Уральском государственном университете. Однако, по объему представленный материал выходит за пределы материала необходимого для выполнения конкретной лабораторной работы. Например, в лабораторной работе используется только один язык программирования, а излагаются и другие, используемые для данного изделия, языки программирования и т.п.

В пособии по каждому изучаемому изделию дается конструкция, техническая характеристика изделия, принцип его программирования, примеры программирования и варианты задания для индивидуальности обучения студентов.

В пособии представлены зачастую схемы структурные, принципиальные и соединений изучаемых изделий и установок, которые могут оказать существенную помощь студентам при выполнении курсового и дипломного проектирования.

1. РАБОТА №1. ОСНОВЫ ПРОГРАММИРОВАНИЯ ПРОГРАММИРУЕМОГО КОНТРОЛЛЕРА SIMATIC S7-300

1.1. Цель работы

Ознакомиться с техническими характеристиками контроллера фирмы Siemens SIMATIC S7-300. Приобрести навыки программирования на лабораторном стенде с применением программного обеспечения STEP7 v.5.3.

1.2. Содержание работы

а) Изучить назначение, технические характеристики контроллера фирмы Siemens SIMATIC S7-300.

б) Изучить основы системы команд и принципы программирования контроллера.

в) Дома при подготовке к работе:

- выполнить синтез системы автоматизации согласно заданного варианта;
- составить программу на языке лестничных диаграмм.

г) В лаборатории:

- освоить графическую среду программного обеспечения STEP 7 v. 5.3;
- сконфигурировать центральную стойку контроллера;
- набрать на компьютере подготовленную дома программу и откомпилировать ее;
- ввести программу в контроллер и убедиться в правильности её работы;
- запрограммировать управление модулем с аналоговыми входами и выходами. Снять характеристики (вход/выход) одного канала ввода аналогового сигнала и одного канала вывода аналогового сигнала.

1.3. Описание лабораторного стенда

На рис. 1.1 представлена структура лабораторной установки для изучения программирования контроллера фирмы Siemens SIMATIC S7-300.

Лабораторная установка включает в себя:

- 1 – ПЭВМ типа IBM;
 - 2 – адаптер связи ПЭВМ с процессором (CPU) программируемого контроллера для записи и отладки программ;
 - 3 – контроллерный блок;
 - 4 – блок имитации сигналов и команд;
 - 5 – кабель связи между контроллерным блоком и блоком имитации.
- Контроллерный блок включает в себя:
- блок питания типа SITOP;
 - процессорный модуль CPU314 (заказной номер 314-1AE04-0AB0);

- модуль ввода дискретных сигналов SM321 DI32xDC24V (заказной номер 321-1BL00-0AA0) на 32 канала ввода сигналов постоянного тока напряжением 24 В;
- модуль вывода дискретных сигналов SM322 DI32xDO32B/0,5A (заказной номер 322-1BL00-0AA0) на 32 канала вывода сигналов постоянного тока напряжением 24 В с допустимым током нагрузки 0,5 А;
- модуль ввода/вывода аналоговых сигналов SM334 AI4/AO2x8bit (заказной номер 334-0CE01-0AA0) на 4 канала ввода и 2 канала вывода аналоговых сигналов;
- тумблер «Сеть» для питания контроллерного блока напряжением 220 В, 50 Гц;
- разъем для связи контроллерного блока с блоком имитации.

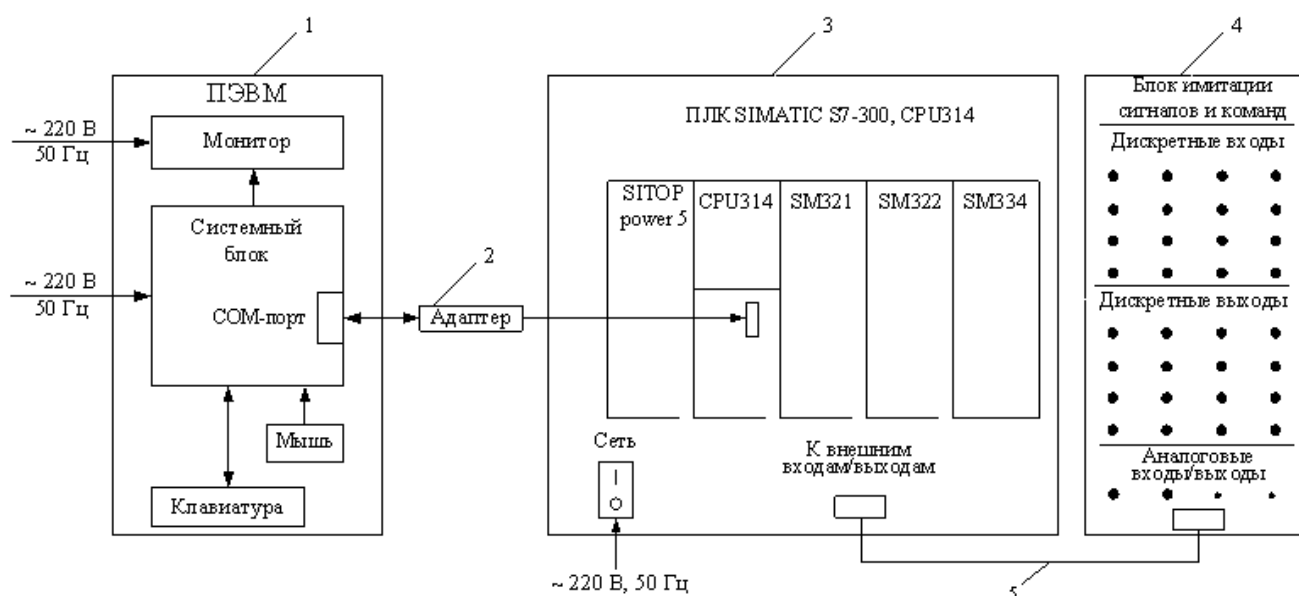


Рис. 1.1. Структурная смена лабораторной установки

Выше для каждого модуля контроллерного блока указан заказной номер, который потребуется для программирования контроллера при его конфигурировании.

Процессорный модуль CPU314 имеет память ОЗУ 48 Кбайт и загрузочную вставную MMC максимум на 8 Мбайт. Языки программирования LAD/FBD/STL в английской аббревиатуре (KOP/FUP/AWL – в немецкой). Работает с модулями, которые могут размещаться в 4-х стойках. Максимальное количество модулей на каждой стойке 8. Максимальное количество дискретных сигналов ввода/вывода 1024, аналоговых сигналов максимум 256, таймеров 256 с диапазоном времени от 10 мс до 9990 с, счетчиков 256 с диапазоном счета от 0 до 999, промежуточные переменные (меркеры) – 256 байт.

В табл. 1.1 представлено распределение памяти контроллера.

Таблица 1.1

Область памяти	Символ	Биты, слова	Пояснения
Дискретные входы	I	I0.0...I127.7	Состояние входных устройств, подключенных к входным цепям
Аналоговые входы	PIW	PIW256...PIW766	Представленная в цифровой форме аналоговая величина
Дискретные выходы	Q	Q0.0...Q127.7	Состояние дискретных выходов
Аналоговые выходы	PQW	PQW256...PQW766	Представленная в аналоговой форме цифровая величина
Таймеры	T	T0...T255	Режим работы: формирователь импульса, формирователь удлиненного импульса, задержка включения, задержка включения с запоминанием, задержка выключения
Счетчики	C	C0...C255	Счетчики
Меркеры	M	M0.0...M255.7	Промежуточные (внутренние) переменные

Адресация сигналов и команд программируемых контроллеров SIMATIC S7-300 представлена детально в Приложении А к данной работе.

Модуль ввода дискретных (цифровых) сигналов SM321 имеет четыре входных байта с адресами I0...I3 (рис. 1.2). В работе используются только входные байты I0 и I1.

Модуль имеет индикацию состояния зелеными светодиодами на каждом канале на лицевой панели модуля. Входной ток канала ввода при сигнале 1 и напряжении 24 В не превышает 7 мА.

Модуль вывода дискретных (цифровых) сигналов имеет четыре выходных байта с адресами Q4...Q7. В работе использованы только выходные байты Q4 и Q5. Как видно из схемы (рис. 1.3) состояние каждого канала выхода индицируется светодиодом, расположенном на лицевой панели модуля. В каждом канале предусмотрен стабилитрон для защиты цепей канала от перенапряжений при коммутации активно-индуктивной нагрузки. Коммутируемые напряжения и токи приведены в обозначении модуля. Диапазон сопротивления нагрузки от 48 Ом до 4 кОм.

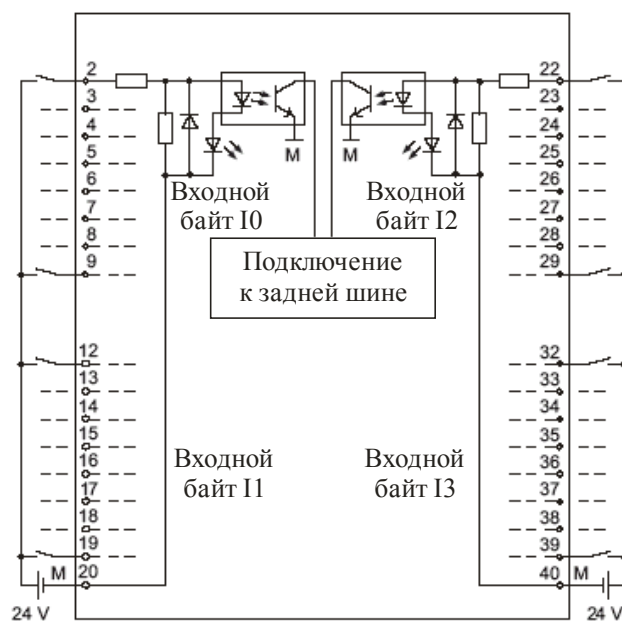


Рис. 1.2. Принципиальная схема цифрового модуля ввода SM321 DI32xDC24V

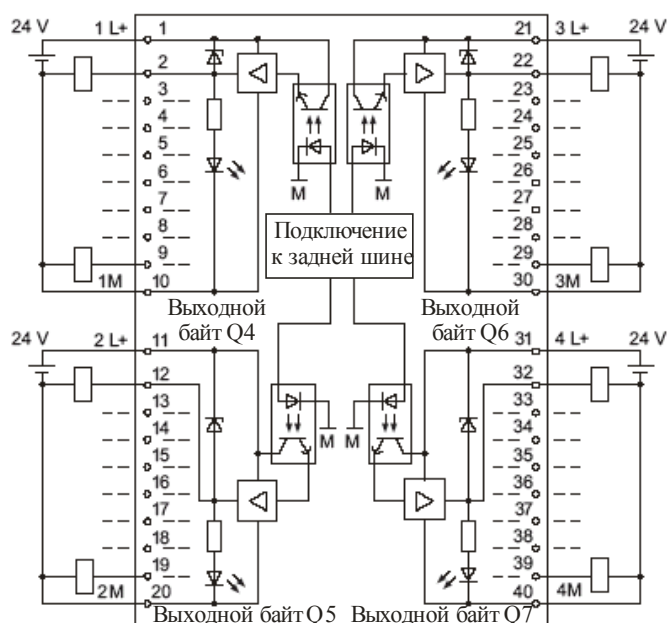


Рис. 1.3. Принципиальная схема цифрового модуля вывода SM322 DO32xDC24V/0,5A

Аналоговый модуль ввода/вывода SM334 (рис. 1.4) не требует параметризации. Установка вида подачи входного сигнала (сигнал напряжения или токовый) и вида вывода обеспечивается путем подключения в соответствии со схемой рис. 1.4.

В лабораторной установке на аналоговые каналы подаются сигналы напряжения с потенциометрических резисторов, установленных на блоке имитации сигналов и команд. Поступающее с выходов напряжение индицируется изменением яркости свечения светодиодов, подключенных к соответствующим каналам

вывода. Указанные светодиоды также установлены на блоке имитации сигналов и команд.

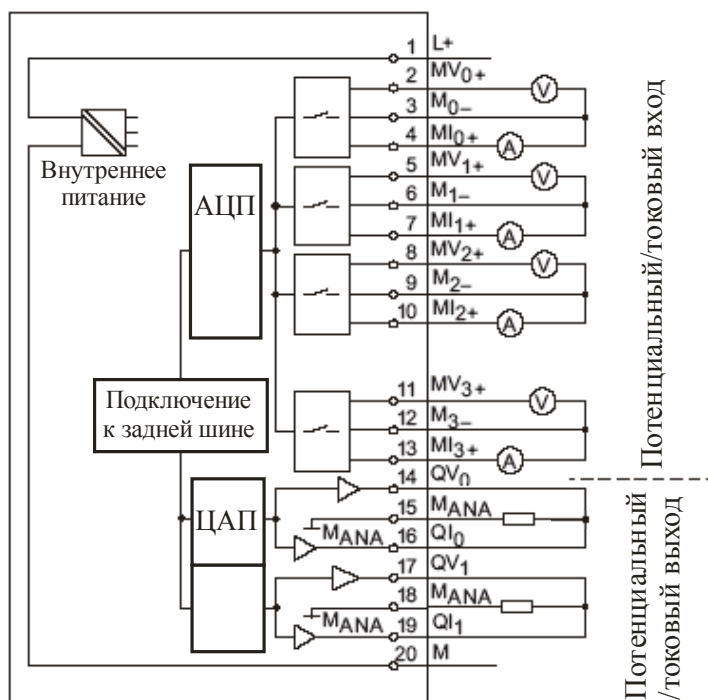


Рис. 1.4. Принципиальная схема аналогового модуля ввода/вывода SM334 AI4/AO2x8bit

Блок имитации сигналов и команд используется для выдачи дискретных и аналоговых сигналов на соответствующие входы контроллера и индикации выходных сигналов с контроллера. Дискретные входные сигналы подаются с 8 кнопок с адресами I0.0...I0.7 и 8 тумблеров с адресами I1.0...I1.7. Выходные дискретные сигналы подаются на 16 светодиодов с адресами Q4.0...Q4.7 и Q5.0...Q5.7. Как уже указывалось выше на блоке индикации установлено два потенциометрических резистора и два светодиода для работы с аналоговыми сигналами.

1.4. Конфигурирование в STEP 7 v 5.3

Для подготовки программ при проведении лабораторных работ служит программа – «SIMATIC Manager». Для запуска «SIMATIC Manager» необходимо на рабочем столе Windows дважды щелкнуть курсором мыши по ярлыку



Откроется окно «SIMATIC Manager».

«SIMATIC Manager [Администратор SIMATIC]» – это центральное окно, которое становится активным при запуске STEP 7. По умолчанию запускается мастер STEP 7 (STEP 7 Wizard) (рис. 1.5), который оказывает помощь при создании проекта STEP 7. Структура проекта используется для надлежащего хранения и

размещения всех данных и программ. В предварительном обзоре (Preview) можно включать и выключать отображение структуры создаваемого проекта. Чтобы перейти к следующему диалоговому окну, щелкните на кнопке «Next [Дальше]».

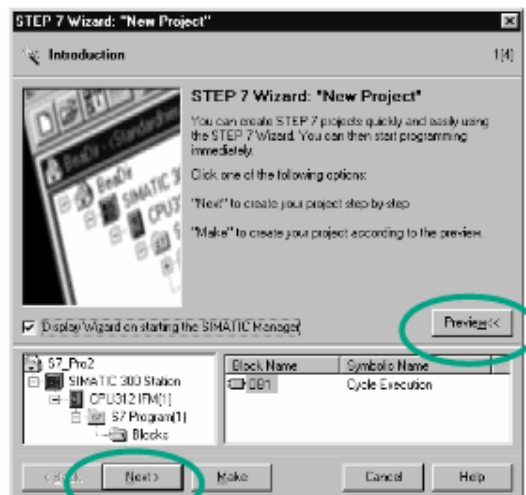


Рис. 1.5. Окно мастера STEP 7 (STEP 7 Wizard)

Выберите CPU (рис. 1.6) в соответствии с тем, который является мастером системы. Установка по умолчанию для адресов MPI равна 2. Щелкните на Next, чтобы подтвердить настройки и перейти к следующему диалоговому окну.

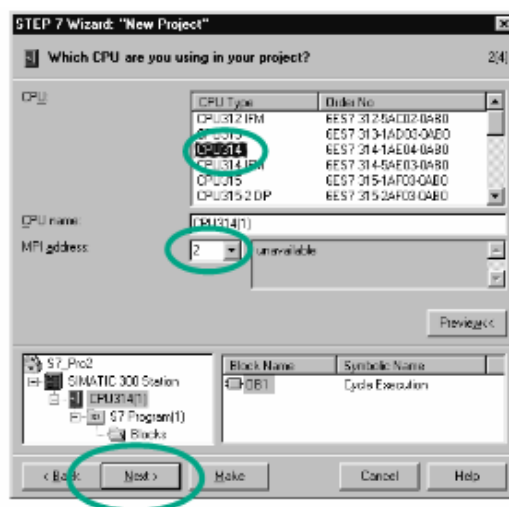


Рис. 1.6. Окно выбора процессора в STEP 7 Wizard

Выберите организационный блок OB1 (рис. 1.7). Выберите один из языков программирования: контактный план (LAD), список операторов (STL) или функциональный план (FBD). Подтвердите настройки кнопкой Next.

Дважды щелкните в поле «Project name [Имя проекта]» (рис. 1.8), чтобы выбрать предлагаемое имя, заменив его, например на «Master S7-300». Щелкните на кнопке «Make [Создать]», чтобы сгенерировать свой новый проект в соответствии с предварительным обзором.

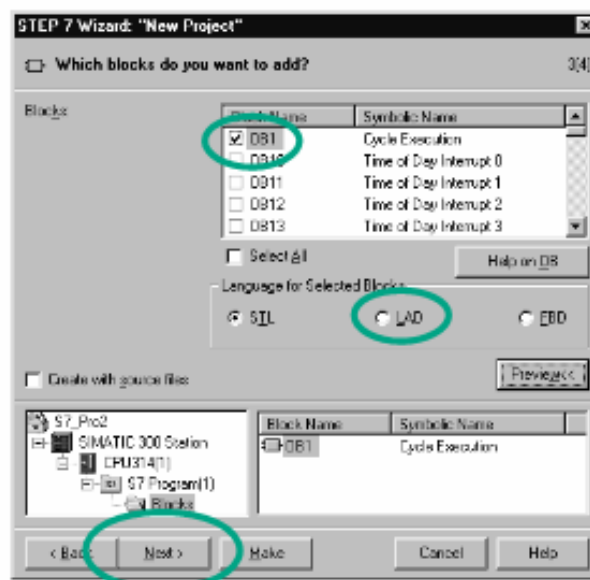


Рис. 1.7. Окно выбора языка программирования

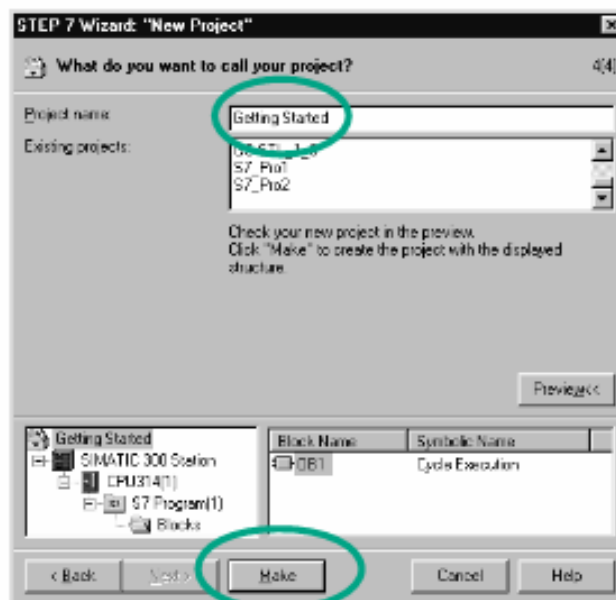
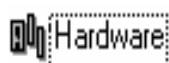


Рис. 1.8. Окно выбора имени проекта

Только что созданный проект отображается с выбранной S7 станцией и CPU (рис. 1.9). Щелкая на значке «+» или «-» открыть или закрыть папку.

Конфигурирование аппаратной части «Master`a S7-300» производится при нажатии на надпись «Master S7-300» в левой части окна. В правой части окна появится значок:



– [Оборудование]. Щелкните по нему, появится окно конфигурирования оборудования – «HW Config».

Добавление модулей производится из каталога «Catalog», содержащий список компонентов оборудования SIMATIC.

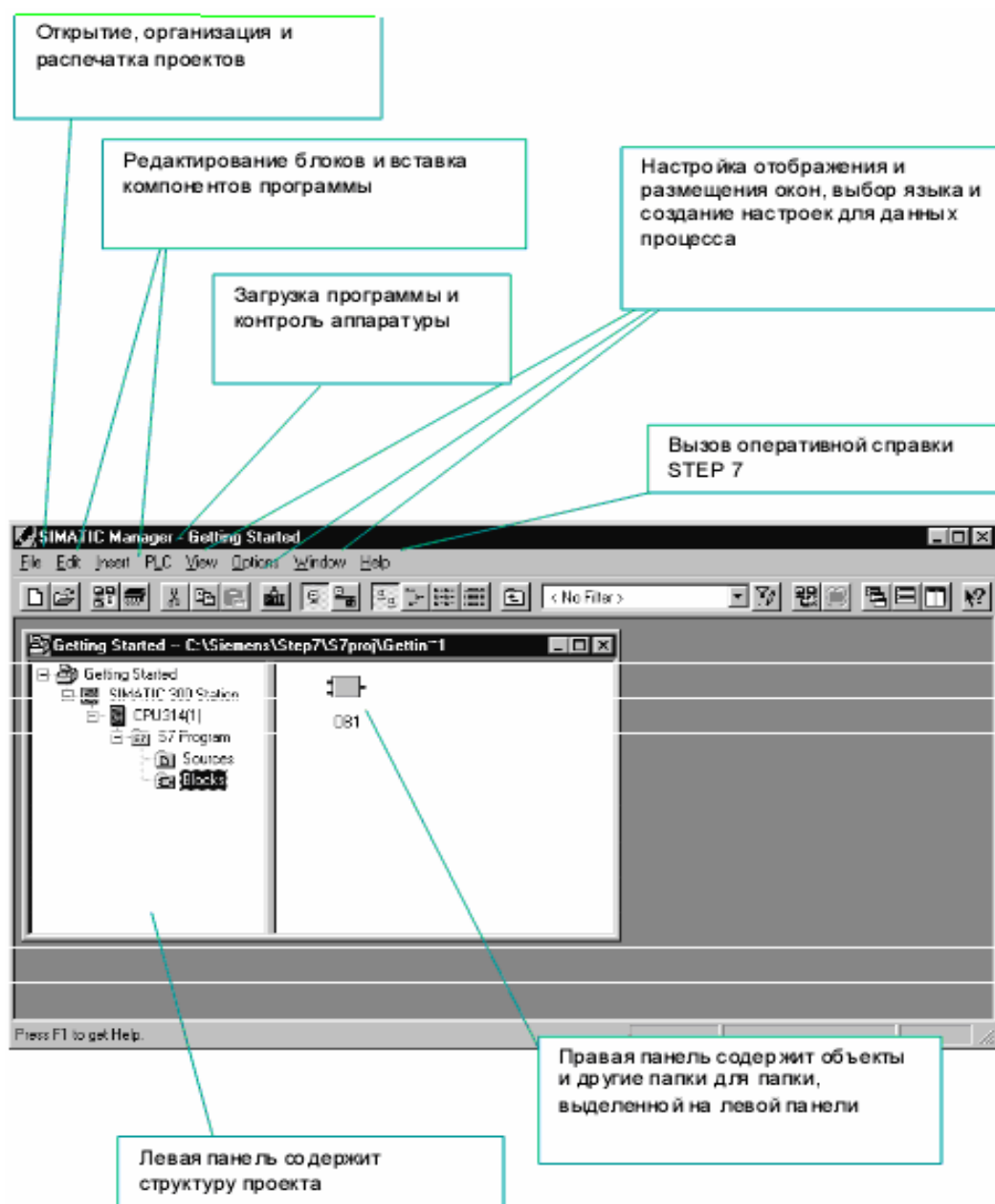


Рис. 1.9. Окно SIMATIC Manager

В каталоге в папке SIMATIC 300/RACK-300 выберите центральную стойку (профильную шину) «Rail». Отбуксируйте стойку, используя Drag&Drop, в окно станции. В нижней части окна появляется подробное представление стойки с дополнительными данными.

В качестве альтернативы можно дважды щелкнуть левой кнопкой мыши на выбранном модуле «Rail» и он появится в конфигурационной таблице.

Основное поле (рис. 1.10) содержит сведения о CPU и модулях системы:

- 1) Slot – позиция модуля в системе (3-я позиция используется только под интерфейсный модуль);
- 2) Module – тип модуля или CPU;
- 3) I address – возможные адреса модуля ввода;

- 4) Q address – возможные адреса модуля вывода;
- 5) Comment – строка для занесения комментариев.

Слот	Модуль	Номер для заказа	Адрес MPI	Комментарий
Slot	Module	Order number	Firmware MPI address I addr... Q addr...	Comment
1	PS 307 10A	6ES7 307-1KA00-0AA0		
2	CPU 314	6ES7 314-1AE01-0AB0	2	
3				
4	DI8xAC120/230V	6ES7 321-1FF10-0AA0		0
5	AI8x12Bit	6ES7 331-7KF02-0AB0		272...287
6	AI8x16Bit	6ES7 331-7NF10-0AB0		288...303
7	AI8xTC/4xRTD, Ex	6ES7 331-7SF00-0AB0		304...319
8	AO2x12Bit	6ES7 332-5HB00-0AB0		320...323
9	AO2x12Bit	6ES7 332-5HB81-0AB0		336...339
10				
11				

Рис. 1.10. Окно HW Config

На стойку устанавливается блок питания PS (при необходимости) и модули (например, CPU) из окна «Hardware Catalog». Слоты, в которые можно установить тот или иной модуль выделяются цветом. Выберите модули по заказным номерам (Order number), которые имеются на лицевой панели каждого модуля. Отбуксируйте модули, используя Drag&Drop, в соответствующие строки стойки (конфигурационная таблица).

СТЕП 7 проверяет, не нарушены ли правила для слотов (CPU S7-300, например, может быть установлен только в слоте 2).

⊘ – Символ нарушения правил для слотов.

Если выделить слот в стойке модулей, можно увидеть список всех возможных для установки модулей, открыв контекстно-зависимое меню (правой кнопкой мыши) «Insert Object [Вставить объект]» или «Replace Object [Заменить объект]». Эта возможность избавляет Вас от необходимости поиска в каталоге аппаратуры. Вы выбираете из всех модулей, которые имеются в текущем открытом профиле каталога.

!!! Помните, что модули в слотах следует располагать в той же последовательности, в какой они расположены физически и с теми заказными номерами, которые указаны на них.

Для проверки правильности проведенного конфигурирования необходимо в меню «Station [Станция]» выбрать и активировать позицию «Consistency Check [Проверить согласованность]». Появится соответствующее сообщение об отсутствии или наличии ошибок, и каких.

Покидание окна конфигурирования производится после сохранения данной станции «Station – Save» или сохранения с компиляцией «Station – Save and Compile».

1.5. Программирование пользовательской задачи

Программирование пользовательской задачи (создание пользовательской программы) реализуется при помощи программного обеспечения STEP7 v5.3. Обучающийся, в соответствии с поставленной задачей, разрабатывает алгоритм автоматизации управления объектом либо в виде логических уравнений, либо в виде схемы алгоритма. Входным, выходным сигналам и внутренним переменным (меркерам) присваиваются проектировщиком адреса в соответствии с конфигурационной таблицей.

Основная программа формируется в организационном блоке OB1. Для доступа к OB1 необходимо в левой панели окна SIMATIC Manager'a (рис. 1.9) дважды щелкнуть левой кнопкой мыши по знаку CPU..., затем по появившемуся значку S7 Program и наконец, по появившемуся значку Bloks. Появляется в правой части пиктограмма OB1. Дважды щелкните на пиктограмме OB1. Открывается окно программирования (рис. 1.11).

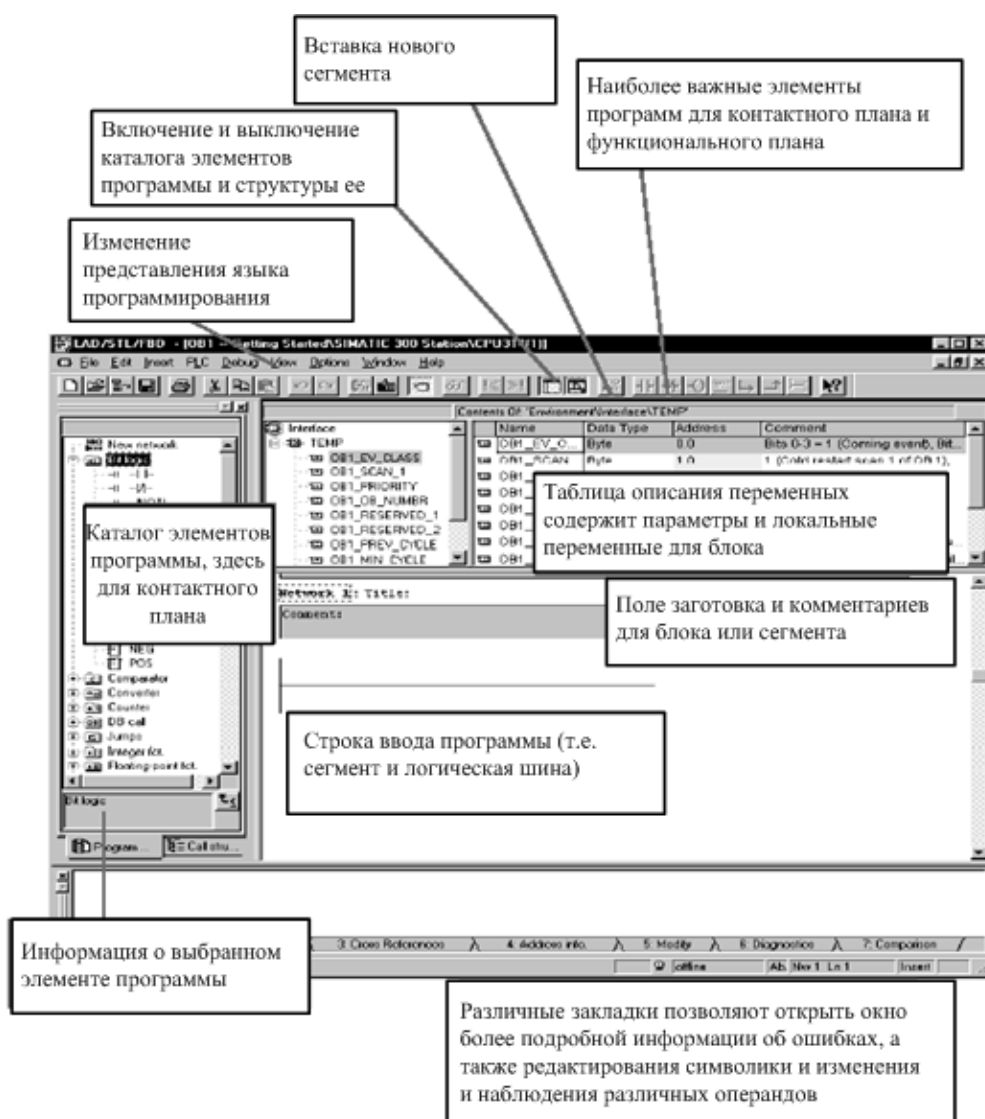


Рис. 1.11. Окно программирования

В меню «View [Вид]» можно выбрать язык программирования LAD/STL/FBD. Язык LAD (язык лестничных диаграмм) удобен для инженеров-электриков, язык STL (язык операторов) ориентирован для пользователей из мира компьютерных технологий (мнемокода) и, наконец, язык FBD (язык функциональных блоков) ориентирован на инженеров-схемотехников. STEP7 v5.3 позволяет в любое время изменить язык программирования.

Ниже рассматривается программирование на языке LAD.

При программировании лестничными диаграммами (LAD) в виде релейно-контактной схемы программа разделяется на сегменты. Каждый сегмент представляет собой отдельную цепь, по которой может протекать ток. Шина питания находится слева (вертикальная линия).

Для создания нового сегмента необходимо нажать на кнопку



– New network.

Рассмотрим ввод программы для реализации простого уравнения:

$$Q4.0 = (I1.1 \cdot \overline{I1.3}) + I1.2$$

1) Поставьте в токовую цепь (сегмент) реле (катушку) (рис. 1.12). Для этого нажмите на кнопку



– Coil [Катушка]



Рис. 1.12.

2) Щёлкните по тому месту (непосредственно по линии) куда, хотите вставить замыкающий (нормально открытый) контакт (рис. 1.13).



Рис. 1.13.

3) Вставьте замыкающий контакт, нажав на кнопку



– Normally Open Contact [Нормально открытый контакт] (рис. 1.14).

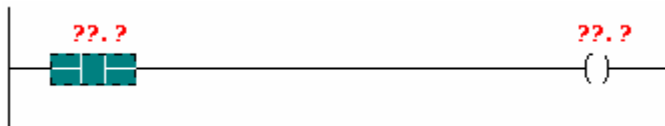


Рис. 1.14.

4) Создайте параллельную ветвь для замыкающего контакта. Для этого щелкните по месту, от которого будет ответвление (в нашем случае – шина питания), нажмите на кнопку



– Open Branch [Создать ветвление] (рис. 1.15).

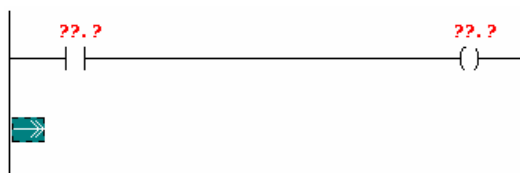


Рис. 1.15.

5) Вставьте нормально открытый контакт и завершите параллельную ветвь, нажав на кнопку



– Close Branch [Завершить ветвление] (рис. 1.16).



Рис. 1.16.

6) Выделите место, куда необходимо поставить нормально замкнутый (размыкающий) контакт (рис. 1.17).



Рис. 1.17.

7) Вставьте размыкающий контакт, нажав на кнопку



– Normally Closed Contact [Нормально закрытый контакт] (рис. 1.18).

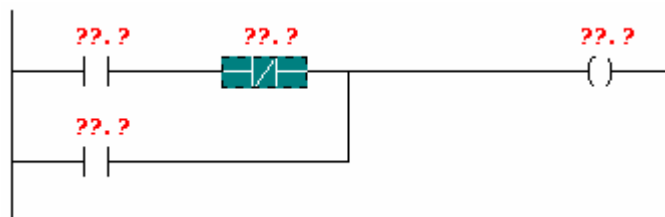


Рис. 1.18.

8) Теперь напишите адреса контактов. Для этого щёлкните по знакам ??? над контактом и введите адрес (ввод адреса завершается нажатием клавиши **ENTER**) (рис. 1.19).

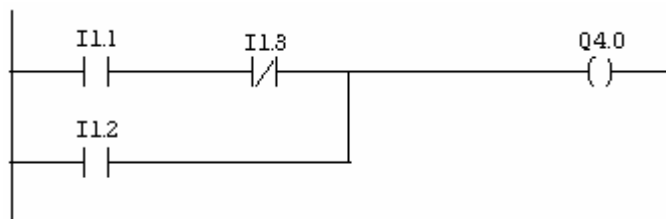


Рис. 1.19.

В среде STEP7 удобно пользоваться символической адресацией, когда любому адресу может быть присвоено удобное для пользователя символическое имя. Присвоим I1.1 – «Key1 [Кнопка 1]», I1.3 – «Key2 [Кнопка 2]», I1.2 – «Switch [Тумблер]», Q4.0 – «Lamp [Лампа]». Для этого необходимо открыть редактор символов, нажав одновременно Ctrl-Alt-T или выбрав Options→Symbol Table (выбрав в главном меню редактора программы Options и затем выбрав из выпадающего меню Symbol Table). Появится окно редактора символов (рис. 1.20).

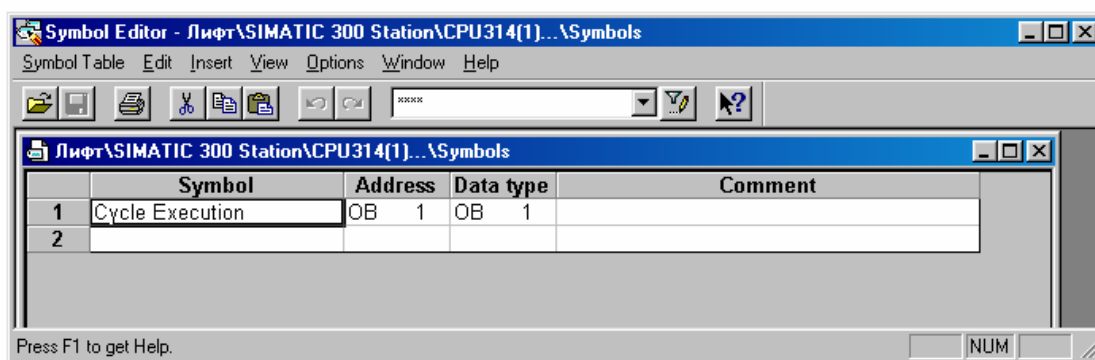


Рис. 1.20. Окно редактора символов

Заполните свободную строку в таблице символов:

- 1) в столбце символов (Symbol) вставьте: «Key1»;
 - 2) в строке адресов (Address) вставьте: «I1.0» (автоматически в столбце типа данных появится BOOL – тип данных «бит»;
 - 3) в столбце для комментариев (Comment) вставьте «Кнопка1».
- Введите символы для остальных адресов (см. рис. 1.21).

Лифт\SIMATIC 300 Station\CPU314(1)...Symbols				
	Symbol	Address	Data type	Comment
1	Cycle Execution	OB 1	OB 1	
2	Key1	I 1.0	BOOL	Кнопка1
3	Key2	I 1.3	BOOL	Кнопка2
4	Switch	I 1.2	BOOL	Тумблер
5	Lamp	Q 4.0	BOOL	Лампа
6				

Рис. 1.21. Окно редактора символов

Сохраните таблицу символов (Symbol Table→Save). Закройте окно редактора символов. Теперь программа будет выглядеть следующим образом (рис. 1.22).

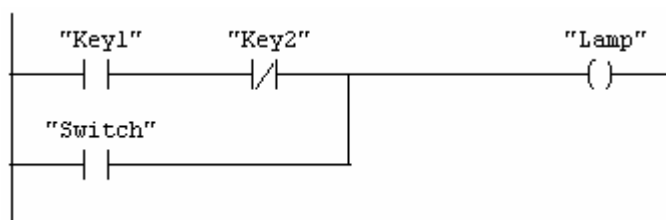


Рис. 1.22.

Для удаления какого-либо элемента программы необходимо его выделить и выбрать: Edit→Delete. Для отмены предыдущего действия необходимо выбрать: Edit→Undo.

В случае некорректного ввода адреса или символического имени программа высвечивает некорректное обозначение красным цветом.

Рассмотрим вставку функции на примере создания следующей программы: если кратковременно нажать на кнопку Key1, то зажжётся лампа Lamp (адрес Q4.0), если кратковременно нажать на кнопку Key2, лампа Lamp (адрес Q4.2) погаснет. Для этого воспользуемся функцией SR триггера:

- 1) создайте новый сегмент нажав на кнопку ;
- 2) внесите в таблицу символов символ Lamp с адресом Q4.2;
- 3) нажмите в редакторе программы на  кнопку Program Elements – появится окно списка функций;
- 4) в окне списка функций выбрать: Bit logic→ SR (SR триггер) – щёлкнуть по этому значку быстро два раза;
- 5) над функцией SR триггера (рис. 1.23) вместо символов "???" вставьте символическое имя выхода;

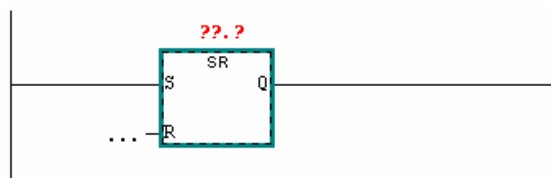


Рис. 1.23.

6) на входную линию функции S поставьте нормально открытый контакт Key1, а на место входа R – Key2 (рис. 1.24).

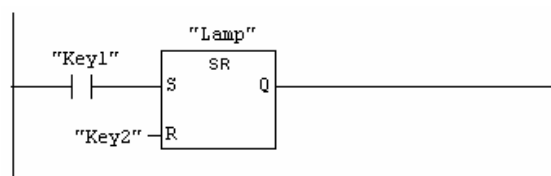


Рис. 1.24.

Для удаления функции необходимо её выделить, подвести к ней курсор, нажать на правую клавишу мыши и из выпадающего меню выбрать Delete.

1.6. Операции с таймерами

Таймеры имеют собственную область памяти в CPU. Для каждого из 256 таймеров резервируется 16-битное слово. Биты с 0 по 11 в таймерном слове выделены для записи установки времени, в битах 12 и 13 записывается база времени (код дискретности генератора импульсов), биты 14 и 15 не используются.

Уставка времени таймера может загружаться в двоичном, шестнадцатеричном или двоично-десятичном (BCD) коде.

Уставку времени можно задать в BCD коде с использованием следующего синтаксиса:

S5T#aH_bbM_ccS_dddMS,

где a=часы, bb=минуты, cc=секунды и ddd= миллисекунды.

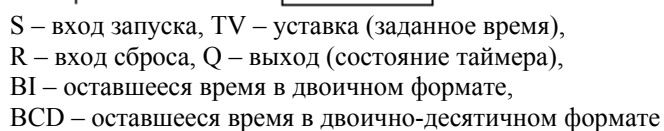
Диапазон уставки времени от 0 до 9990 с, что соответствует 2H_46M_30S.

База времени устанавливается автоматически и значение округляется до ближайшего меньшего с этой базой времени. База времени определяет дискрету времени используемого генератора. Минимальная база времени равна 10 мс, максимальная 10 с. (см. табл. 1.3).

Таблица 1.3

База времени	Двоичный код для базы времени
10 мс	00
100 мс	01
1 с	10
10 с	11

В STEP 7 реализуется 5 разновидностей таймеров. В лестничных диаграммах блоки всех разновидностей обозначаются одинаково и различаются только обозначением типа таймера (рис. 1.25).



Блок таймера имеет вход R. Появление сигнала «1» на входе R во время работы таймера, заданного уставкой, сбрасывает таймер и выход отключается.

Имеется второй упрощенный вариант программирования таймеров, в котором отсутствуют вход R и выходы BI и BCD.

Ниже на рис. 1.26 – 30 приведены для каждого типа таймера оба варианта программирования и временные диаграммы их работы, из которых без дополнительных комментариев ясны особенности их работы.

Во втором варианте программирования таймеров возможен сброс таймеров подачей команды R (RESET) по адресу таймера (рис. 1.31).



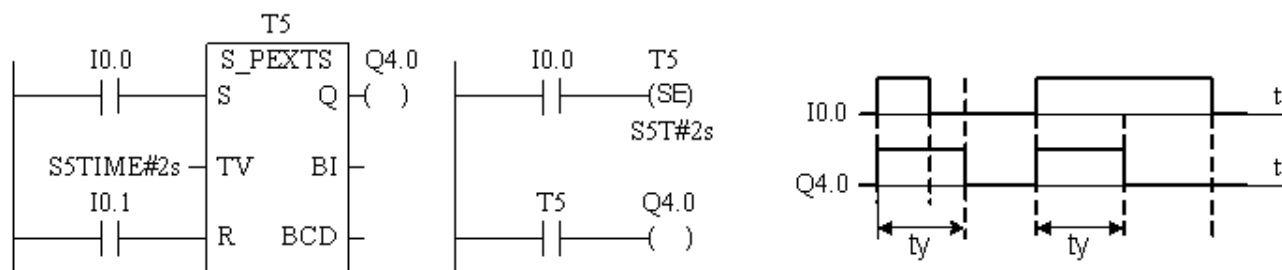


Рис. 1.27. Таймер формирователь удлиненного импульса

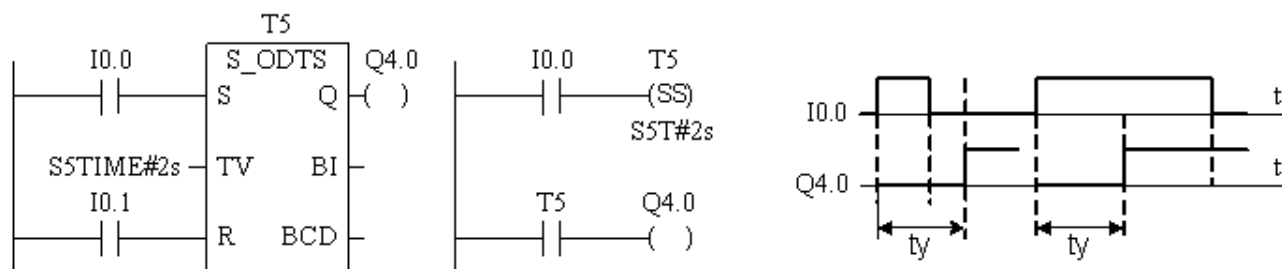


Рис. 1.28. Таймер формирователь задержки включения с запоминанием

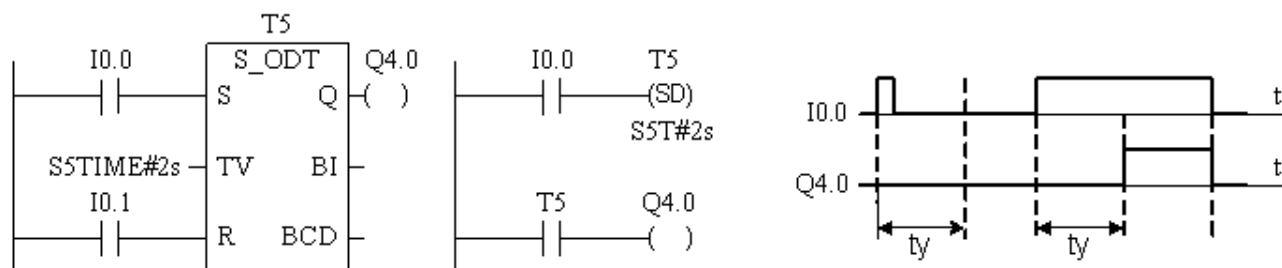


Рис. 1.29. Таймер формирователь задержки включения

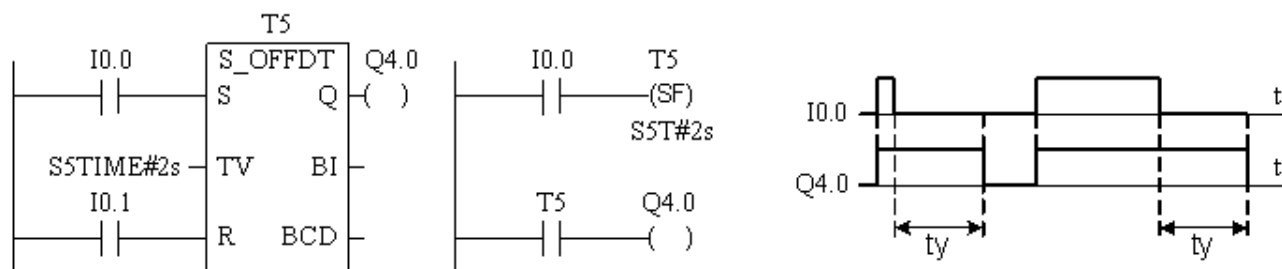


Рис. 1.30. Таймер формирователь задержки выключения

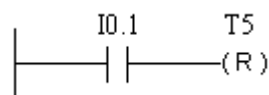


Рис. 1.31. Сброс таймера с адресом T5

1.7. Операции счета

Счетчики имеют собственную область памяти в CPU. Для каждого из 256 счетчиков резервируется слово длиной 16 бит, но для счета используется только биты с 0 по 11. Поэтому Счет возможен до 999, то есть уставка счетчика находит-

ся в интервале 0...999. Задание уставки, например, числа 127 осуществляется в следующем формате: C#127, где C# означает BCD код.

В контроллере реализованы 3 вида счетчиков:

- на сложение (прямой счет);
- на вычитание (обратный счет);
- реверсивный (прямой/обратный счет).

На рис. 1.32 представлены блоки всех 3-х видов счетчиков.

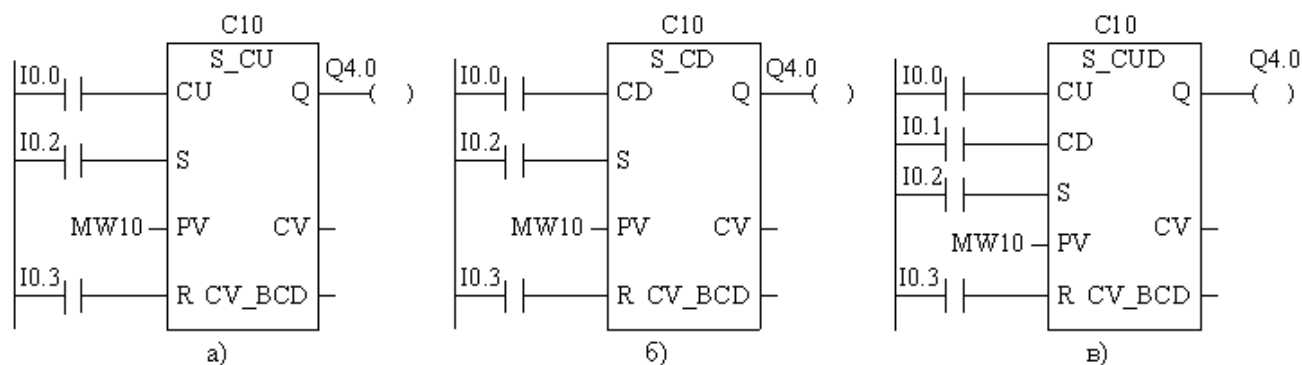


Рис. 1.32. Счетчики: а) на сложение, б) на вычитание, в) реверсивный

На блоках счетчиков рис. 1.32 указаны следующие обозначения: C10 – номер счетчика, S_CU, C_CD, S_CUD – тип счетчика, CU – вход на сложение, CD – вход на вычитание, S – вход предустановки счетчика, PV – задание (уставка) счетчика, R – вход сброса, Q – выход (состояние) счетчика, CV – текущее значение счетчика в двоичном формате, CV_BCD – текущее значение счетчика в формате BCD.

Уставка счетчика задается либо константой в формате C#<значение>, либо из области памяти (I, Q, M, D). Нарастающим фронтом сигнала на входе S в счетчик заносится уставка с входа PV. При нарастающем сигнале на входе CU значение счетчика увеличивается на «1», если значение счетчика меньше, чем 999. При нарастающем фронте сигнала на входе CD значение счетчика уменьшается на «1», если значение счетчика больше, чем «0». При нарастающем фронте сигнала на входе R счетчик сбрасывается (в счетчик записывается значение «0»).

Для реверсивного счетчика, если оба счетных входа имеют нарастающий фронт, то значение счетчика остается неизменным.

Для всех типов счетчиков выход Q равен «0», если значение счетчика равно «0», и устанавливается в «1», когда значение счетчика отлично от «0».

1.8. Операции с аналоговыми сигналами

CPU может обрабатывать аналоговые сигналы только в двоичном коде. Аналоговые входные модули преобразуют аналоговые сигналы в этот код.

Аналоговый модуль SM334, используемый в лабораторной работе, имеет более низкую разрешающую способность, чем другие аналоговые модули, и не имеет области отрицательных значений.

Входы и выходы модуля адресуются, начиная со стартового адреса модуля. Адрес канала получается из начального адреса модуля и адресного смещения.

Входам соответствуют следующие адреса каналов:

Таблица 1.4

Канал	Адрес
0	Начальный адрес модуля
1	Начальный адрес модуля + 2 байта адресного смещения
2	Начальный адрес модуля + 4 байта адресного смещения
3	Начальный адрес модуля + 6 байта адресного смещения

Выходам соответствуют следующие адреса каналов:

Таблица 1.5

Канал	Адрес
0	Начальный адрес модуля
1	Начальный адрес модуля + 2 байта адресного смещения

При выполнении лабораторной работы требуется снять характеристики вход/выход одного канала ввода и одного канала вывода аналогового сигнала. Для этого предлагается ввести программу, представленную на рис. 1.33.

Команда MOV в Network 1 обеспечивает считывание кода напряжения, поданного на канал ввода аналогового сигнала с адресом PIW288 и выдачу этого кода для светодиодной индикации модулем вывода дискретных сигналов по адресу QW5. Для снятия характеристики канала ввода аналогового сигнала мультиметром измеряется напряжение между клеммами – и + модуля SM334 и считывается по светодиодам код числа с байта QW5 модуля SM322 при изменении положения движка потенциометрического резистора, подключенного к каналу ввода.

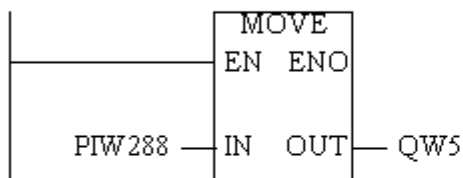
Команда MOV в Network 2 обеспечивает передачу кода набранного на входе IB1 (тумблерами на блоке имитации сигналов команд) на вход цифроаналогового преобразователя (ЦАП) канала вывода аналогового сигнала с адресом PQW288. Для построения характеристики вход/выход канала аналогового вывода считывается код, установленный на входе IB1, и мультиметром измеряется напряжение на аналоговом выходе.

OB1:

Снятие характеристик каналов ввода и вывода модуля SM334

Network: 1

Индикация кода преобразованного аналогового сигнала светодиодами модуля вывода дискретных сигналов по адресу QW5



Network: 2

Передача кода набранного на входах байта IB1 на канал вывода с адресом PQW288 для формирования соответствующего напряжения вывода

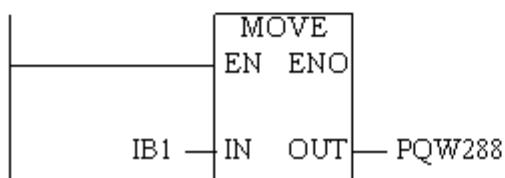


Рис.1.33. Программа для снятия характеристик каналов аналогового ввода и вывода

1.9. Программирование функциональных блоков и функций

Язык STEP7 предусматривает возможность упростить программирование, если возникает необходимость в программировании управления однотипными технологическими операциями. Это, например, рассматриваемые ниже испытания двигателей бензинового и дизельного. Испытания в принципе одинаковые, только аппараты (кнопки, датчики и устройства индикации) свои. Тогда централизованно программируется функциональный блок FB, который может многократно вызываться в организационном блоке OB1. Все формальные параметры и статические данные FB сохраняются в отдельном блоке (DB), назначаемом функциональному блоку.

Например, будем программировать функциональный блок FB1, которому дадим символическое имя «Двигатель». При кратковременном появлении команды «Bkl [Включить]» двигатель должен включиться и сформировать сигнал «Bkl_on [Включен]».

При достижении фактической скорости Fakt заданного значения Zадание формируется сигнал Est. Двигатель отключается по сигналу «Otkl [Отключить]» или по сигналу «Neispr [Неисправность]». Здесь используется английский шрифт, потому что при программировании FB кириллица не воспринимается.

Прежде всего, необходимо определиться с символьным представлением сигналов и команд (табл. 1.6). Для организационного блока OB1 в табл. 1.6 имеется

ряд сигналов, которые будут использованы позднее при рассмотрении программирования функций.

Таблица 1.6

Описание переменных организационного блока OB1

Simbol	Address	Data type	Comment
BD_Bkl	I0.0	BOOL	Включить бензиновый
BD_Otkl	I0.1	BOOL	Отключить бензиновый
BD_Neisprr	I0.2	BOOL	Неисправность бензинового
BD_Fakt	MW0	INT	Фактическая скорость бензинового
BD_Bkl_on	Q4.0	BOOL	Бензиновый включен
BD_Est	Q4.1	BOOL	Заданная скорость достигнута
BD_Bent_on	Q4.2	BOOL	Вентилятор бензинового включен
BD_Timer_Bent	T1	TIMER	Таймер вентилятора бензинового
DD_Bkl	I1.0	BOOL	Включить дизель
DD_Otkl	I1.1	BOOL	Отключить дизель
DD_Neisprr	I1.2	BOOL	Неисправность дизеля
DD_Fakt	MW2	INT	Фактическая скорость дизеля
DD_Bkl_on	Q5.0	BOOL	Дизель включен
DD_Est	Q5.1	BOOL	Заданная скорость достигнута
DD_Bent_on	Q5.2	BOOL	Вентилятор дизеля включен
DD_Timer_Bent	T2	TIMER	Таймер вентилятора дизеля

Для создания FB1 в левой части окна проекта следует вызвать папку Blocks, правой кнопкой мыши вызвать всплывающее меню, в нем выбрать Insert New Object > Function Block и присвоить FB1 имя «Двигатель». В правой части окна проекта возникнет папка FB1.

Дважды щелкните левой кнопкой мыши на FB1. В появившемся диалоговом окне «Properties-Function Block» выберите язык, который хотите использовать, активизируйте триггерную кнопку «Multiple instance FB [Мультиэкземплярный FB]» и подтвердите остальные параметры настройки, щелкнув на ОК. FB1 вставлен в папку блоков Blocks.

Для программирования FB1 необходимо дважды щелкнуть на FB1. Открывается окно для программирования. Следует заполнить таблицу описания переменных (рис. 1.34). Тип переменной (Bool, Int и т.п.) выбирается с помощью всплывающего меню «Elementary Types [Элементарные типы]», используя правую кнопку мыши.

☐ Interface ☐ IN • • • •	Name	Data Type	Address	Initial Value	Comment
	Bkl	Bool	0.0	False	Включить двигатель
	Otkl	Bool	0.1	False	Отключить двигатель
	Neispr	Bool	0.2	False	Неисправность
	Fakt	Int	2.0	0	Текущая скорость
☐ Interface • ☐ Out • • •	Name	Data Type	Address	Initial Value	Comment
	Bkl_on	Bool	4.0	False	Двигатель включен
	Est	Bool	4.1	False	Есть заданная скорость
☐ Interface • • • ☐ STAT •	Name	Data Type	Address	Initial Value	Comment
	Zadanie	Int	6.0	1500	Заданная скорость

Рис. 1.34. Описание переменных функционального блока FB1

Затем программируется включение и отключение двигателя, как было описано выше. Выделите вопросительные знаки и введите соответствующие имена из таблицы описания переменных (знак «#» назначается автоматически). Помеченные знаком «#» локальные переменные действительны только в этом блоке.

Набранная программа на языке лестничных диаграмм имеет вид, представленный на рис. 1.35.

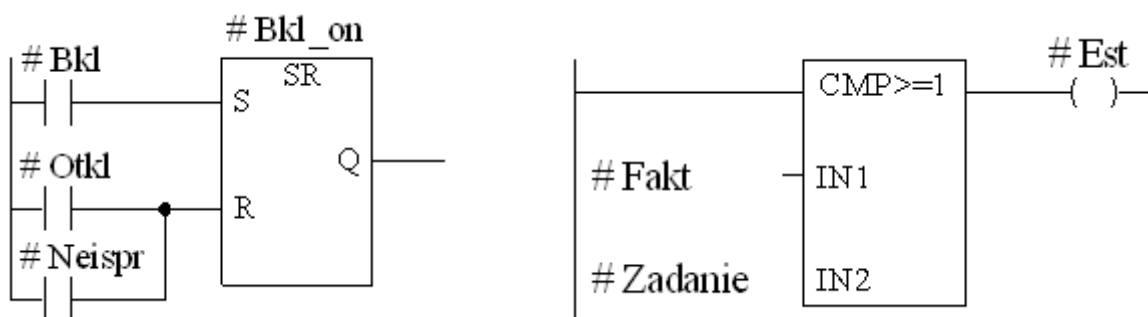


Рис. 1.35. Программа функционального блока FB1

Чтобы получить возможность вызова организованного блока FB1 в OB1 необходимо сгенерировать соответствующий блок данных. Экземплярный блок данных (DB) всегда ставится в соответствие функциональному блоку.

Рассматриваемый FB1 должен управлять и контролировать работу бензинового и дизельного двигателей. Различные заданные скорости хранятся в отдельных блоках данных.

В левой части окна проекта необходимо переместиться в папку Blocks и в правой половине окна щелкнуть правой кнопкой мыши. Используя всплывающее меню, выбрать Insert New Object > Data Block. Задайте имя блоку данных DB1 «Бензиновый», выберите тип блока «Instance DB [Блок экземпляра]» и поставьте ему в соответствие блок FB1. Примите все параметры настройки, отображаемые в диалоговом окне «Properties [Свойства]», щелкните на ОК. Блок данных DB1 добавляется к проекту. Дважды щелкните на значке DB1, чтобы открыть его. Подтвердите назначение блока экземпляра для FB1 с помощью ОК.

Затем введите значение 1500 для бензинового двигателя в столбец «Actual Value [Фактическое значение]» в строке Zадание. Теперь определена максимальная скорость для этого двигателя. Сохранить DB1 и закройте блок.

Так же, как и в случае с DB1, сгенерируйте еще один блок данных DB2 для FB1. Теперь введите значение 1200 для дизельного двигателя.

Следующий шаг – это запрограммировать вызов функционального блока в OB1. Это демонстрируется на рис. 1.36.

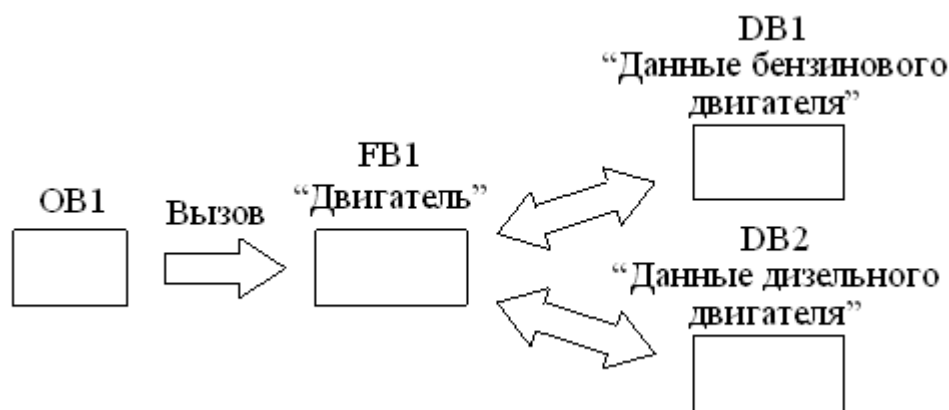


Рис. 1.36. Вызов блока FB1 в OB1

Необходимо открыть SIMATIC Manager с вашим проектом. Переместиться к папке Blocks и открыть OB1. Вставить сегмент программы. Затем перемещаетесь в каталог элементов программы, пока не достигнете папки функциональных блоков. Выделите FB1 и вставьте его в OB1.

Вставьте контакты перед каждым из следующих входов Bkl, Otkl и Neispr. Щелкните на знаке «???» над блоком «Двигатель», а затем, удерживая курсор в том же положении, щелкните правой кнопкой мыши, выберите во всплывающем меню Insert Symbol. Щелкните на блоке данных «Бензиновый». Этот блок затем автоматически вводится в рамку ввода в кавычках.

Выделяя вопросительные знаки, назначьте адреса всем остальным параметрам функционального блока, используя соответствующие символьные имена из табл. 1.4.

Вызовите новый сегмент и в нем вызовите функциональный блок «Двигатель» (FB1) с блоком данных «Дизель» и используйте для него соответствующие символьные адреса.

Набранные программы будут иметь вид, представленный на рис. 1.37.

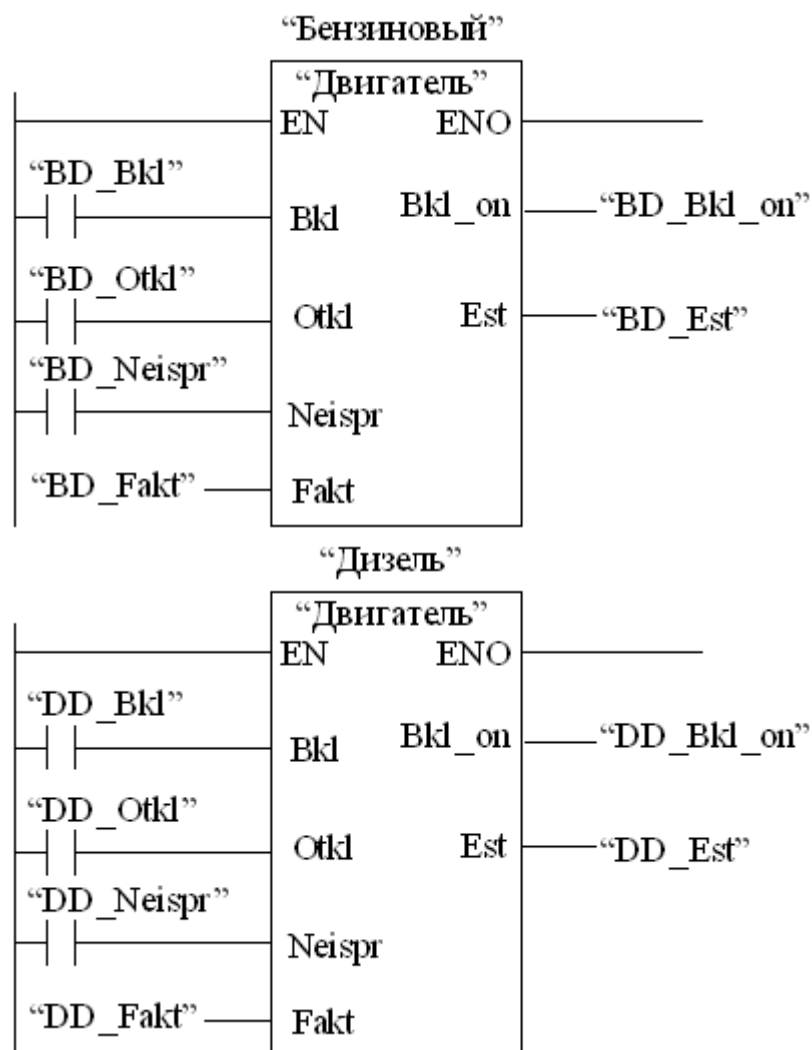


Рис. 1.37. Программирование вызова функционального блока FB1 в OB1

Сохраните свою программу и закройте блок OB1.

Функции, как и функциональные блоки, расположены в иерархии программы ниже организационного блока. Однако в отличие от функционального блока, функции не нужен блок данных. У функций параметры также перечисляются в таблице описания переменных, но статические данные не разрешаются.

Для создания функции в левой части окна проекта, переместитесь к папке Blocks и откройте ее. Щелкните правой кнопкой мыши в правой половине окна. Во всплывающем меню выбрать Insert New Object > Function. В диалоговом окне «Properties – Function [Свойства – Функция]» задайте имя FC1, в нашем случае «Вентилятор», и выберите нужный язык программирования. Подтвердите остальные параметры настройки с помощью ОК. Функция FC1 добавляется в папку Blocks.

В рассматриваемой задаче предусматривается, что каждый двигатель как бензиновый, так и дизель имеет свой вентилятор. Вентилятор включается при ра-

боте двигателя и продолжает еще работать 4 с после отключения двигателя. Для управления этими двигателями программируется функция FC1.

Дважды щелкните на папке FC1, чтобы открыть ее. Открывается окно для программирования. Введите описание переменных (рис. 1.38).

☐ Interface ☐ IN	Name	Data Type	Comment
	Bkl_on	Bool	Двигатель включен
	Timer_Bent	Timer	Задержка на отключение
☐ Interface . ☐ Out . . .	Name	Data Type	Comment
	Bent_on	Bool	Вентилятор включен

Рис. 1.38. Описание переменных функций

Затем выделите шину для ввода команд лестничных диаграмм и запрограммируйте управление вентилятором. Выделяйте вопросительные знаки и вводите соответствующие имена из таблицы описания переменных (знак # назначается автоматически). Набранная программа на языке лестничных диаграмм имеет вид, представленный на рис. 1.39.

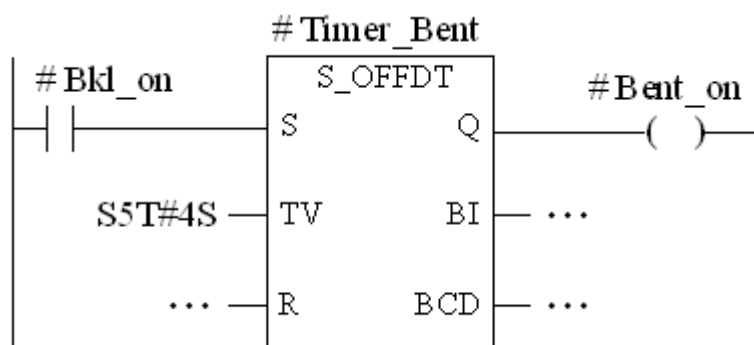


Рис. 1.39. Программирование функции FC1

Вызов функции FC1 в OB1 выполняется аналогично вызову функционального блока. Все переменные функции снабжаются в OB1 соответствующими адресами бензинового или дизельного двигателя, которые уже отражены в табл. 1.4.

Откройте свой проект SIMATIC Manager. В левой части окна проекта переместитесь к папке Blocks и откройте в правой части окна проекта OB1.

Вставьте новый сегмент программы. Затем перемещайтесь в каталоге элементов программы, пока не достигнете папки функций FC Blocks. Выделите FC1

и вставьте ее в OB1. Вставьте замыкающий контакт перед «Bkl_on». Выделив вопросительные знаки при вызове FC1, введите символьные имена из табл. 1.4 для бензинового двигателя.

Вызовите очередной сегмент программы и запрограммируйте вызов функции FC1, используя адреса для дизельного двигателя.

Набранные программы будут иметь вид, представленный на рис. 1.40.

Сохраните блок и закройте окно.

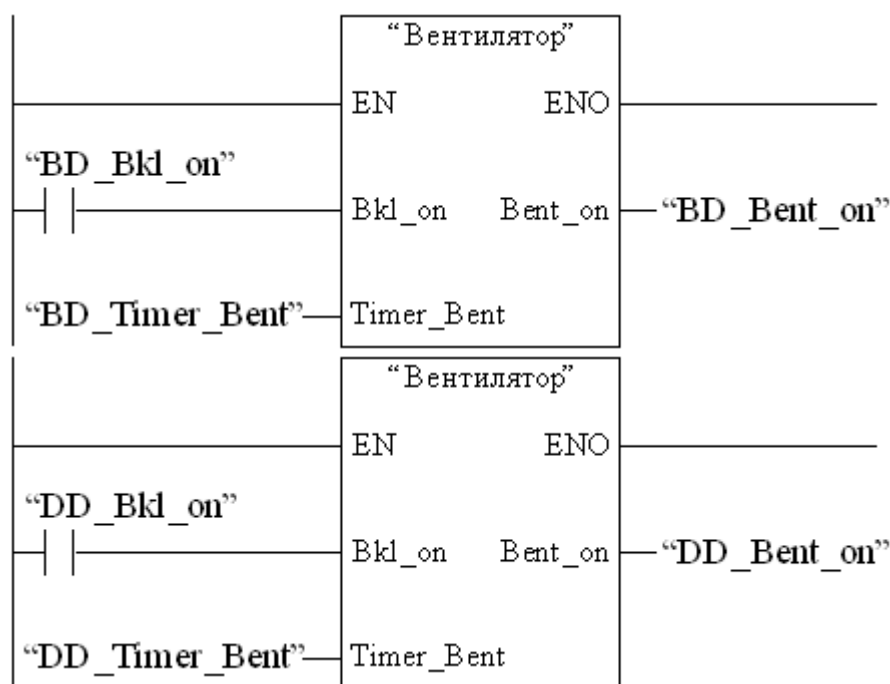


Рис. 1.40. Программирование вызова функции FC1 в OB1

1.10. Пример программирования контроллера SIMATIC S7-300

В качестве примера рассмотрим управление контроллером SIMATIC S7-300 движением тележки.

Тележка перемещается по прямолинейному пути. Имеется два фиксированных положения П0 и П1. Исходное положение тележки – П0.

При кратковременном нажатии на кнопку «Пуск» тележка движется вперед до положения П1, стоит в положении П1 5 с и возвращается назад в положение П0. В положении П0 немедленно возникает реверс привода тележки. Тележка вновь движется в положение П1, стоит в нем 5 с и возвращается в П0. В положении П0 вновь возникает команда на движение до положения П1 и т.д. после 10 циклов перемещения в положение П1 при возвращении тележки в положение П0 возникает сигнал «Конец цикла».

Для управления движением тележки следует предусмотреть счетчик (СЧ) со счетом до 10, задержку времени (таймер) для формирования задержанного сигнала $П1^{\uparrow}$ и память P_{Π} о нажатии на кнопку «Пуск».

$$P_{\Pi} = (Пуск + P_{\Pi}) \cdot \overline{КЦ}$$

Команда на перемещение тележки вперед (В) возникает в положении П0, если отсутствует сигнал счетчика СЧ, и сохраняется при наличии сигнала памяти пуска P_{Π} до достижения положения П1. Этому описанию соответствует логическое уравнение

$$B = (П0 \cdot \overline{СЧ} + B) \cdot \overline{П1} \cdot P_{\Pi}$$

Команда на перемещение тележки назад (Н) возникает через 5 с нахождения в положении П1 и сохраняется при наличии переменной P_{Π} до достижения положения П0. Тогда логическое уравнение для команды Н имеет вид

$$H = (П1^{\uparrow} + H) \cdot \overline{П0} \cdot P_{\Pi}$$

Сигнал конца цикла (КЦ) возникает в положении П0 при наличии сигнала счетчика СЧ о завершении 10 циклов перемещения в положение П1 и сигнала P_{Π} . Сигнал КЦ сохраняется до очередного нажатия на кнопку «Пуск». Тогда логическое уравнение для рассматриваемого сигнала имеет вид

$$КЦ = (П0 \cdot СЧ \cdot P_{\Pi} + КЦ) \cdot \overline{Пуск}$$

В ниже приведенной программе (рис. 1.41) использован счетчик на вычитание. При нажатии на кнопку «Пуск» в счетчик вводится уставка 10. Счетчик считает число циклов перемещения в положение П1, при достижении этого положения число в счетчике уменьшается на единицу. Счетчик сбрасывается по сигналу конца цикла КЦ. Признаком окончания цикла является число в счетчике равное нулю, при котором на выходе счетчика «0». Поэтому в уравнениях команд В и КЦ необходимо использовать инверсию сигнала СЧ по отношению к ранее записанным уравнениям

$$B = (П0 \cdot СЧ + B) \cdot \overline{П1} \cdot P_{\Pi}$$

$$КЦ = (П0 \cdot \overline{СЧ} \cdot P_{\Pi} + КЦ) \cdot \overline{Пуск}$$

Сброс счетчика производится сигналом конца цикла КЦ.

В табл. 1.7 представлена адресация команд и сигналов для контроллера SI-MATIC S7-300.

Таблица 1.7

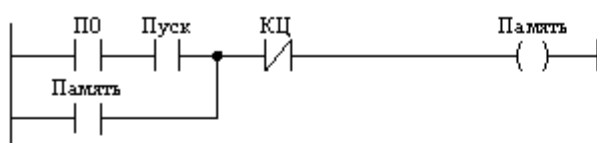
Команда, сигнал	П0	П1	Пуск	В	Н	КЦ	P_{Π}	СЧ	$П1^{\uparrow}$
Адрес	I0.0	I0.1	I0.2	Q4.0	Q4.1	Q4.2	M0.0	M0.1	M0.2
Символ	П0	П1	Пуск	Вперед	Назад	КЦ	Память	СЧ	Задержка

ОВ1:

Управление движением тележки

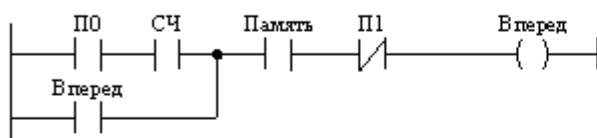
Network: 1

Память пуска



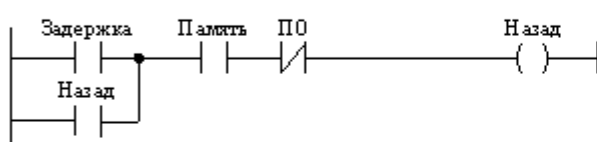
Network: 2

Движение вперед



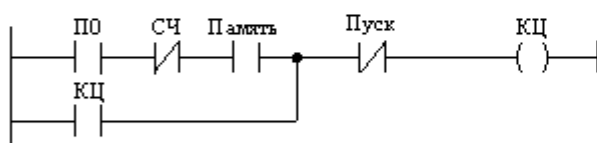
Network: 3

Движение назад



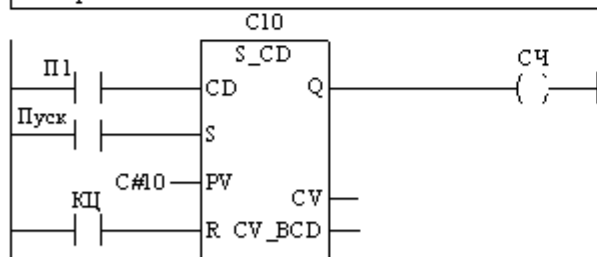
Network: 4

Конец цикла



Network: 5

Управление счетчиком



Network: 6

Управление таймером с задержкой на включение

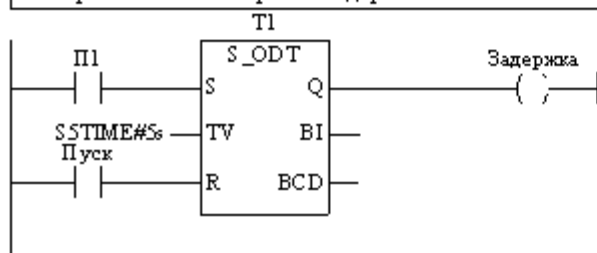


Рис. 1.41. Программа управления движением тележки

1.11. Загрузка программы в контроллер и запуск программы

Перед загрузкой программы необходимо сбросить память в CPU. Это выполняется в следующей последовательности:

- поверните переключатель режимов работы в положение MRES. Удерживайте переключатель в этом положении, пока светодиод STOP не загорится во второй раз и затем останется гореть (это занимает 3 с);
- в течение 3 с необходимо повернуть переключатель обратно в положение MRES. Светодиод STOP начинает быстро мигать и CPU производит сброс памяти. Когда светодиод STOP снова начинает гореть непрерывно, это означает, что CPU завершил сброс памяти.

Более подробно о стирании памяти, в частности, о стирании по запросу со стороны CPU и форматировании его сменной платы микропамяти SIMATIC (MMC) представлено в п.2.6 лабораторной работы №2 «Изучение промышленной сети PROFIBUS-DP».

Загрузка программы в контроллер осуществляется в следующей последовательности:

- переместитесь через SIMATIC 300 Station CPU314 к S7 Program. В SIMATIC Manager щелкните на символе «Blocks [Блоки]» в правой части окна. Ключ режима контроллера поставьте в положение STOP;
- из меню PLC (ПЛК) выберите команду «Download [Загрузить]», чтобы передать программу и конфигурацию аппаратуры в CPU. Щелкаете на Yes во всех появляющихся диалоговых окнах.

Для запуска программы необходимо перевести ключ режима работы контроллера в положение «RUN».

Во время работы программы можно следить за ходом ее выполнения. Переведя переключатель режима работы контроллера в положение «RUN» и нажав на кнопку с изображением очков.

При этом будут отображаться состояния контактов и катушек, а также значения счетчиков, таймеров в реальном времени.

1.12. Выполнение лабораторной работы

При подготовке к лабораторной работе студент должен в соответствии с табл. 1.8 выбрать свой вариант задания.

Таблица 1.8

Номер бригады	1 и 5			2 и 6			3 и 7			4 и 8		
Член бригады	1	2	3	1	2	3	1	2	3	1	2	3
Номер варианта	1	2	3	4	5	6	7	8	9	10	11	12

Для заданного варианта необходимо разработать алгоритм ее решения. В соответствии с заданием определиться с адресацией входных и выходных переменных. Затем определиться с адресацией вводимых промежуточных переменных и с адресацией используемых таймеров и счетчиков. Составить программу для контроллера SIMATIC S7-300.

Запустить программу в работу и убедиться в правильности ее работы. При наличии ошибок в работе устранить их и продемонстрировать преподавателю работу правильно функционирующей программы.

Программу для снятия характеристик аналоговых каналов ввода/вывода можно вписать в программу своего варианта задания. Для этого следует учесть, что для индикации кода преобразованного напряжения канала аналогового ввода необходимо предусмотреть свободный байт модуля вывода дискретных сигналов, например, QW5.

Вариант 1. Перекладыватель служит для перемещения листов со стола на постоянно вращающийся рольганг. Перекладыватель имеет подъемник листов, установленный на тележке. Подъемник представляет собой поперечину, перемещающуюся по вертикальным стойкам сверху вниз и обратно. Поперечина имеет на конце электромагнит для притягивания листов.

В исходном положении перекладыватель стоит в положении П0 над столом, а его подъемник в крайнем верхнем положении КВ. При поступлении листа на стол подается сигнал наличия листа, подъемник включается для движения вниз. При касании листа подъемник останавливается и включается электромагнит. Через 2 с включается подъемник на подъем, достигает положения КВ и останавливается. Включается тележка и перемещается в положение П1 над рольгангом, где останавливается и электромагнит отключается. Лист падает на рольганг. Через 2 с тележка движется в положение П0. Цикл повторяется при поступлении очередного сигнала наличия листа. Включение системы в работу – нажатием кнопки «Пуск».

Вариант 2. В контроллере программным путем необходимо реализовать генератор импульсов. Время наличия импульса 1 с, время его отсутствия 2 с. При нажатии на кнопку «Пуск» начинается счет импульсов. При прохождении 10 импульсов загорается лампа Л1, при прохождении следующих 10 импульсов – лампа Л2, аналогично включаются лампы Л3 и Л4. После загорания лампы Л4 (т.е. после прохождения 40 импульсов) счет прекращается и все лампы горят до нажатия на кнопку «Стоп». Генератор импульсов реализовать программным путем.

Вариант 3. Автоматизировать перемещение пунсона штампованного прессы. Рабочий заправляет металлическую ленту (заготовку) и нажимает кнопку «Пуск». Пуансон из крайнего верхнего положения П0 движется вниз, при своем движении до положения П1 входит в матрицу, выбивая из ленты деталь необходимой фор-

мы, и возвращается в положение П0. Чисто механическим устройством лента перемещается на один шаг и через 2 с пуансон совершает очередное движение. Лента рассчитана на изготовление 10 деталей, поэтому через 10 ходов пуансона формируется сигнал «Конец цикла».

При подаче питания на систему автоматизации пуансон из любой точки должен прийти в положение П0.

Вариант 4. На программируемом контроллере реализовать систему управления толкателем с кривошипно-шатунным механизмом с реверсивным электродвигателем.

Толкатель имеет два фиксированных положения П0 и П1. При подаче питания на систему толкатель из любого положения движется вперед медленно до положения П0. При появлении кратковременного сигнала «Пуск» толкатель из положения П0 медленно движется до положения П1, стоит там в течение 2 с, а затем электродвигатель включается для движения назад быстро и возвращается в положение П0, где формируется сигнал «Конец цикла». Цикл повторяется при нажатии кнопки «Пуск».

При нажатии кнопки «Стоп» толкатель из любого положения на быстрой скорости возвращается в положение П0.

Вариант 5. Автоматизировать управление крышкой котла (типа большой кастрюли).

При кратковременном нажатии кнопки «Открыть» крышка поднимается до крайнего верхнего положения, в котором поворачивается, открывая доступ к содержимому котла.

При кратковременном нажатии на кнопку «Заккрыть» крышка поворачивается в положение точно над котлом и затем опускается, закрывая котел.

Для исключения ударов предусмотреть переход на пониженную скорость перед касанием крышкой котла при его закрывании.

Вариант 6. Автоматизировать управление дверью (воротами) въезда/выезда гаража.

Нормальное (исходное) положение двери закрытое. При кратковременном нажатии кнопки «Открыть» включается звонок и через 5 с включается привод двери на открывание. При полном открытии двери привод двери отключается сразу, а звонок звенит еще 2 с.

Закрывание дверей – аналогично: при кратковременном нажатии кнопки «Заккрыть» включается звонок, через 5 с включается привод двери на закрывание. При полном закрытии двери привод двери отключается, а звонок звенит еще 2 с.

Вариант 7. Печь сопротивлений содержит три нагревательных элемента (НЭ), каждый из которых питается от собственного тиристорного преобразователя соответственно ТП1...ТП3. Сигнал «1» на входе ТП соответствует максималь-

но допустимому току НЭ, сигнал «0» – минимально допустимому току НЭ. Синтезировать схему, обеспечивающую алгоритм работы группы ТП, представленный в табл. 1.9.

Таблица 1.9

Номер импульса генератора	-	1	2	3	4	5	6	7	8	9	...
Состояние ТП1	0	1	0	0	1	1	0	1	0	0	...
Состояние ТП2	0	0	1	0	0	1	1	0	1	0	
Состояние ТП3	0	0	0	1	1	0	1	0	0	1	

Начало работы осуществляется после нажатия на кнопку «Пуск».

Цикл управления печью сопротивления состоит из 6 тактов. Переход с одного такта на другой осуществляется при поступлении очередного импульса от генератора импульсов. Циклы управления непрерывно следуют друг за другом.

При поступлении команды «Стоп» прекращается подача импульсов и все ТП отключаются. Для возобновления работы необходимо нажать на кнопку «Пуск». Генератор импульсов реализуется программным путем.

Вариант 8. Манипулятор (рука со схватом) служит для подачи заготовок из накопителя в пресс. Рука манипулятора может перемещаться вперед/назад из крайнего заднего КЗ в крайнее переднее КП положение и назад, а также поворачиваться из крайнего правого положения КПр в крайнее левое положение КЛ и обратно. Схват имеет электромагнитный механизм зажима/разжима заготовок.

Исходное положение манипулятора перед накопителем (в положениях КЗ и КПр, схват под действием пружин зажат). При поступлении запроса со станка и наличии заготовки в накопителе включается электромагнит разжима схвата. Рука идет в крайнее переднее положение КП, через 2 с схват зажимает заготовку (электромагнит схвата отключается) и рука возвращается в положение КЗ. Рука поворачивается в положение КЛ и выдвигается вперед до КП. Схват разжимается (электромагнит схвата включается) и через 2 с рука идет в положение КЗ, затем поворачивается в положение КПр (исходное положение руки). Если есть запрос со станка, а накопитель пуст, то включается сигнальная лампа «Нет детали» и кратковременный звуковой сигнал (в течение 5 с). Световой сигнал снимается кнопкой «Сброс». Включение системы в работу – нажатием кнопки «Пуск».

Вариант 9. Автоматизировать процесс безупорной остановки длинномерного металла на рольганге Р1.

На рольганг Р1 металл передается с предыдущего рольганга Р0, наличие металла на котором контролируется датчиком Д0. Вдоль рольганга Р1 расположены датчики, контролирующие наличие металла последовательно в положениях П1, П2 и П3.

Нормальное положение рольганга Р1 – неподвижное. При наличии сигнала с датчика Д0 и отсутствии металла на рольганге Р1, рольганг Р1 включается и ме-

талл перемещается по рольгангу Р1. При достижении передним концом металла положения П1 происходит снижение скорости Р1 до «ползучей» скорости, а в положении П2 привод рольганга Р1 отключается. Если по каким-либо причинам металл достиг положения ПЗ, то привод рольганга Р1 реверсируется и работает на пониженной скорости, пока металл не уйдет из положения ПЗ. При остановке переднего конца металла между положениями П2 и ПЗ формируется сигнал разрешения работы механизмов уборки металла с Р1.

Включение системы в работу – при нажатии на кнопку «Пуск».

Вариант 10. Автоматизировать работу толкателя с кривошипно-шатунным механизмом с нереверсивным электродвигателем.

При нажатии кнопки «Пуск» кривошип из положения $\alpha = 0^\circ$ движется до положения $\alpha = 120^\circ$. В этом положении происходит переход на пониженную (малую) скорость, кривошип идет до положения $\alpha = 180^\circ$ и останавливается. Через 3с электродвигатель автоматически вновь включается и на повышенной скорости шатун идет в положение $\alpha = 0^\circ$. Стоит в нем 3 с, включается и движется к положению $\alpha = 120^\circ$ и на пониженной скорости до $\alpha = 180^\circ$ и т.д. После 5 толканий толкатель останавливается в исходном положении (кривошип - в положении $\alpha = 0^\circ$) и формируется сигнал «Конец цикла».

Очередной цикл из пяти толканий – после очередного нажатия на кнопку «Пуск».

При нажатии на кнопку «Стоп» толкатель из любого положения на повышенной скорости движется в положение $\alpha = 0^\circ$ и останавливается.

Вариант 11. Тележка движется по кольцевому пути, на котором есть 3 фиксированных положения П0, П1 и П2. Привод тележки – от нереверсивного электродвигателя.

При подаче питания из любого положения тележка должна идти в положение П0. При нажатии на кнопку «Пуск» тележка без задержки движется в положение П1, стоит там в течение 3 с, затем движется в П2, стоит там в течение 4 с, движется в положение П0, стоит там в течение 5 с, движется в П1, стоит там в течение 3 с и т.д. То есть тележка непрерывно движется по кольцевому пути с остановками в фиксированных положениях.

При нажатии на кнопку «Стоп» тележка должна без остановки идти в П0 и там ожидать очередного нажатия кнопки «Пуск».

Вариант 12. Автоматизировать линию сортировки изделий. По конвейеру движутся низкие изделия, но редко могут встречаться и высокие. По ходу конвейера установлены две фотоголовки на расстоянии друг от друга, равном половине ширины изделия. Первая по ходу конвейера фотоголовка настроена на низкие изделия (нижняя фотоголовка), вторая – на высокие изделия (верхняя фотоголовка).

При прохождении низкой детали перекрывается только нижняя фотоголовка, а при высоких – вначале нижняя, а затем верхняя фотоголовки. На выходе кон-

вейера детали ударяются о сортировочную планку и высокие изделия поступают в правый накопитель (левое положение планки), низкие – в левый (правое положение планки). Планка перебрасывается пневмоприводом. Для переброски планки влево или вправо достаточно хотя бы кратковременно подать команду на ее перемещение влево или вправо. Включение системы в работу – нажатием кнопки «Пуск».

1.13. Требования к отчёту

Отчет должен содержать:

- а) цель работы;
- б) условия задачи по варианту, принятые обозначения переменных, логические функции для выходных и промежуточных переменных, при необходимости циклограмму работы оборудования;
- в) логические функции в адресах программируемого контроллера;
- г) программу для реализации системы управления;
- д) методику экспериментальной проверки функционирования реализованной системы управления и результаты проверки;
- е) характеристики вход-выход каналов ввода и вывода аналогового модуля SM 334;
- ж) выводы по работе.

1.14. Контрольные вопросы

1. Что означает термин «конфигурирование контроллера»?
2. Что означает термин «заказной номер модуля контроллера», где и как он используется?
3. В какой последовательности располагаются модули в слотах конфигурационной таблицы?
4. Что необходимо сделать при покидании окна конфигурирования?
5. Что такое таблица символов и как она используется при программировании контроллера?
6. Когда и как осуществляется сброс памяти CPU?
7. Можно ли в SIMATIC S7-300 реализовать таймер с уставкой времени 0,07 с?
8. При каких условиях на выходе счетчиков контроллера формируется сигнал «0» и сигнал «1»?
9. Как снять характеристику вход-выход аналогового канала ввода?
10. Как снять характеристику вход-выход аналогового канала вывода?
11. Что означает создание функционального блока (FB) при программировании контроллера?
12. Что означает создание и открытие функции (FC) при программировании контроллера?

2. РАБОТА №2. ИЗУЧЕНИЕ ПРОМЫШЛЕННОЙ СЕТИ PROFIBUS-DP

2.1. Цель работы

Ознакомиться с принципом построения промышленной сети PROFIBUS-DP и приобрести навыки программирования обмена информацией между компонентами сети.

2.2. Содержание работы

а) Изучить состав представленной на лабораторном стенде промышленной сети PROFIBUS-DP, назначение и технические характеристики компонентов сети.

б) Изучить конфигурирование сети, программирование обмена информацией между компонентами сети и принципы программирования задач управления технологическими процессами.

в) Дома при подготовке к работе:

- выполнить для заданного варианта синтез системы автоматизации объектом согласно заданного варианта, сигналы управления которым поступают с разных компонентов сети, а информация о его работе также передается с объекта на разные компоненты сети;

- составить программу управления контроллерами Master'ом и Slave'ом.

г) В лаборатории:

- сконфигурировать контроллеры Master и Slave, записав для себя выставляемые при конфигурировании адреса компонентов сети и их модулей;

- учитывая установленные при конфигурировании адреса, откорректировать подготовленные дома программы, записать их в менеджере проектов «Simatic Manager» и запомнить;

- загрузить программы в программируемые контроллеры и запустить сеть в работу;

- используя кнопки и тумблеры «Блока имитационных сигналов» лабораторного стенда, подавать (имитировать) сигналы управляющих команд и датчиков и, наблюдая по светодиодам указанного блока ответную реакцию имитируемого объекта автоматизации, оценить правильность его функционирования.

2.3. Описание лабораторного стенда

На рис. 2.1 представлена структура лабораторного стенда.

Лабораторный стенд позволяет производить изучение системы автоматизации, построенной на промышленной сети PROFIBUS-DP.

Лабораторный стенд включает в себя:

- модуль «Блок питания» на базе фирменного источника питания PS 307 на ток 5 А;

– модуль «Контроллер SIMATIC S7-300 (Master S7-300)», включающий в себя CPU 315-2DP (заказной номер 315-2G10-OABO), модуль дискретного ввода/вывода SM323 DI8/DO8xDC24V (заказной номер 323-1BH01-OAAO) и модуль дискретного вывода SM322 DO8xDC24V/2A (заказной номер 322-1BF01-OAAO);

– модуль «Контроллер SIMATIC S7-300 (Slave S7-300)», включающий в себя CPU 315-2DP (заказной номер 315-2G10-OABO), модуль дискретного ввода DI16xDC24V (заказной номер 321-1BH50-OAAO) и модуль дискретного вывода SM322 DO16xDC24V/0,5A (заказной номер 322-1BH10-OAAO);

– модуль «Децентрализованная периферия ET200M (Slave ET200M)», включающий в себя интерфейсный модуль IM153-2F O (заказной номер 153-2BBOO-OXBO) и модуль ввода/вывода SM323 DI16/DO16xDC24V (заказной номер 323-1BL00-OAAO);

– модуль «Оптический шинный терминал ОВТ»;

– модуль «Блок имитационных сигналов»;

– ПЭВМ типа IBM (системный блок, монитор, клавиатура, мышь);

– адаптер для связи ПЭВМ с промышленной сетью;

– соединительные кабели.

Для полноты представления сети в лабораторной установке используются как проводные кабели типа витой пары интерфейса RS485, так и пластиково-оптический кабель. Оптический сетевой терминал ОВТ обеспечивает переход от проводного к пластиково-оптическому кабелю. На схеме рис. 2.1 показано:

– сплошными линиями – кабели низковольтного питания постоянным напряжением 24 В;

– штриховыми линиями кабели проводные (витая пара) сети PROFIBUS-DP;

– сплошной двойной толстой линией – пластиково-оптический кабель;

– штрихпунктирной линией – связь программируемых контроллеров и ET200M с блоком имитационных сигналов.

Программирование программируемых контроллеров осуществляется по сети через разъемы DP процессорных модулей CPU, поэтому адаптер связи с компьютером подключен непосредственно к сети.

2.4. Конфигурирование сети в STEP7 v 5.3

Работа с программным проектом осуществляется в менеджере проектов – «SIMATIC Manager». Для запуска «SIMATIC Manager» необходимо на рабочем столе Windows дважды щелкнуть курсором мыши по ярлыку:



–откроется окно «SIMATIC Manager» (рис. 2.2).

!!! Вам будет предложено воспользоваться мастером по созданию проекта одной системы (упрощенный вариант), необходимо отказаться, нажав «Cancel [отменить]».

Для создания нового проекта необходимо в главном меню выбрать «File [Файл]» «New...». Задайте имя проекта и покиньте окно. Откроется окно нового проекта.



Рис. 2.2. Окно SIMATIC Manager

Далее необходимо создать объекты: выделив проект в правой части окна и открыв контекстное меню с помощью правой клавиши мыши, выбрать «Insert new object [Вставить новый объект]». Вставляем в проект станцию «SIMATIC 300 Station». Вновь вставленный объект появится в правой половине. Здесь, так же, как у всех других объектов можно присваивать специфическое имя объекту. Присваиваем, например, «Master S7-300». Вставляем еще одну станцию «SIMATIC 300 Station» и сеть «PROFIBUS», даем имена «Slave S7-300» и «Profibus (1)» соответственно.

Конфигурирование аппаратной части «Master`а S7-300» производится при нажатии на надпись «Master S7-300» в левой части окна. В правой части окна появится значок:

 Hardware – [Оборудование]. Щелкните по нему, появится окно конфигурирования оборудования – «HW Config».

В правой части окна располагается окно «Hardware Catalog [Каталог аппаратуры]». Добавление модулей производится из этого каталога, содержащий список компонентов оборудования SIMATIC.

В каталоге выберите для вашей структуры в папке SIMATIC 300/RACK-300 центральную стойку «Rail [профильную шину]».

Отбуксируйте стойку, используя Drag&Drop, в окно станции. Стойка появляется в виде небольшой конфигурационной таблицы в верхней части окна станции. В нижней части окна станции появляется подробное представление стойки с дополнительными данными.

В качестве альтернативы можно дважды щелкнуть левой кнопкой мыши на выбранном модуле, и он появится в конфигурационной таблице.

Основное поле (рис. 2.3) содержит сведения о CPU и модулях системы:

- 1) Slot – позиция модуля в системы (3-я позиция используется только под интерфейсный модуль);
- 2) Module – тип модуля или CPU;
- 3) I address – возможные адреса модуля ввода или CPU;
- 4) Q address – возможные адреса модуля вывода или CPU;
- 5) Comment – строка для занесения комментариев.

На стойку (рис. 2.3) устанавливается блок питания PS (при необходимости) и модули (например, CPU) из окна «Hardware Catalog». Слоты, в которые можно установить тот или иной модуль выделяются цветом. Отбуксируйте модули, используя Drag&Drop или двойным щелчком левой кнопкой мыши, в соответствующие строки стойки (конфигурационная таблица).

!!! Когда левой кнопкой мыши выделяете модуль в каталоге, в нижней части каталога появляется заказной номер модуля, который нанесен на корпусе модуля.

Необходимо быть внимательным, чтобы заказной номер модуля, устанавливаемого в конфигурационную таблицу, соответствовал модулю, установленному реально на стенде.

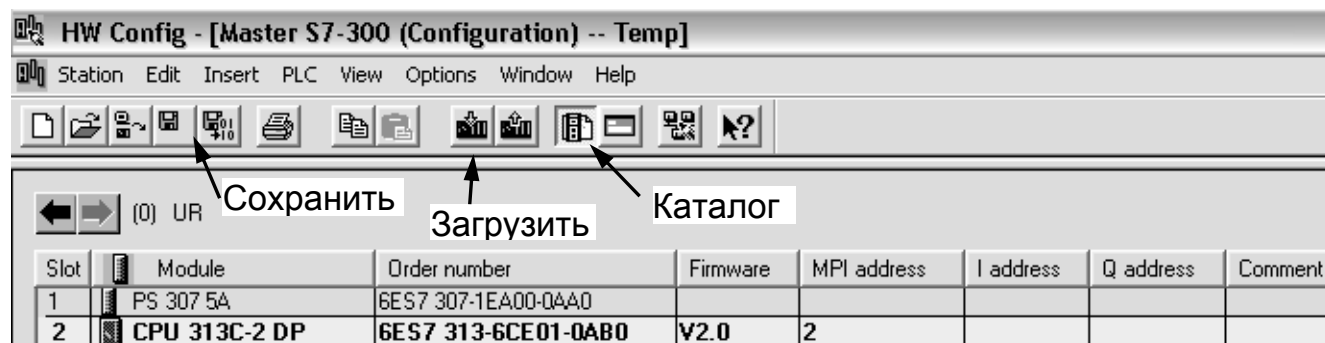


Рис. 2.3. Окно HW Config

STEP 7 проверяет, не нарушены ли правила для слотов, например CPU S7-300 может быть установлен только в слоте 2.

⊘ – Символ нарушения правил для слотов.

!!! В качестве альтернативы Вы можете также выделить соответствующую строку или несколько строк в конфигурационной таблице и дважды щелкнуть в окне каталога аппаратуры «Hardware Catalog» на желаемом модуле. Если выделено несколько строк, то все выделенные строки будут оснащены этим модулем сразу.

!!! Если выделить слот в стойке модулей, можно увидеть список всех возможных для установки модулей, открыв контекстно-зависимое меню (правой кнопкой мыши) «Insert Object [Вставить объект]» или «Replace Object [Заменить объект]». Эта возможность избавляет Вас от необходимости поиска нужного модуля в каталоге аппаратуры. Вы выбираете из всех модулей, которые имеются в текущем открытом профиле каталога.

!!! Помните, что модули в слотах следует располагать в той же последовательности, в какой они расположены физически и с теми номерами, которые указаны на них.

При установке CPU Master'a следует выбрать подключение к сети «Profibus (1)». Для этого левой кнопкой мыши выбрать строку DP процессорного модуля и в появившемся окне «Properties – PROFIBUS interface DP (RO/S2.1)» выделить строку «Profibus (1)» и нажмите ОК. В верхнем окне появится изображение шины (в просторечии «вешалка» или «якорь»), к которой будут подключаться компоненты сети.

При установке в конфигурационную таблицу модулей ввода/вывода STEP7 автоматически присваивает им адреса. Эти адреса необходимо учитывать в подготовленных дома программах.

Затем устанавливается на шину Profibus (1) Slave ET200M. Для этого в разделе PROFIBUS-DP каталога аппаратуры найдите раздел ET200M и из него методом Drag&Drop перетащите на изображение шины сам коммуникационный процессор

IM 153-2 FD (заказной номер 153-2BB00-0XB0) и затем установить на него DI/DO-300 “SM323” (блок ввода-вывода дискретных сигналов).

Для проверки правильности проведенного конфигурирования необходимо в меню «Station [Станция]» выбрать и активизировать позицию «Consistency Check [Проверить согласованность]». Появляется соответствующее сообщение об отсутствии или наличии ошибок, и каких.

Покидание окна конфигурирования производится после сохранения данной станции «Station – Save» или сохранение с компиляцией «Station – Save and Compile».

«Slave S7-300» реализуется на программируемом контроллере SIMATIC S7-300 и поэтому является интеллектуальным Slave'ом (I-Slave'ом). Перед подключением его к системе DP-Master он должен быть вначале создан внутри проекта. Особенность его конфигурации заключается в том, что в нем должна быть предусмотрена промежуточная память для обмена информацией с Master'ом (рис. 2.4). CPU Slave'a берет на себя обмен данными с Master-устройством DP.

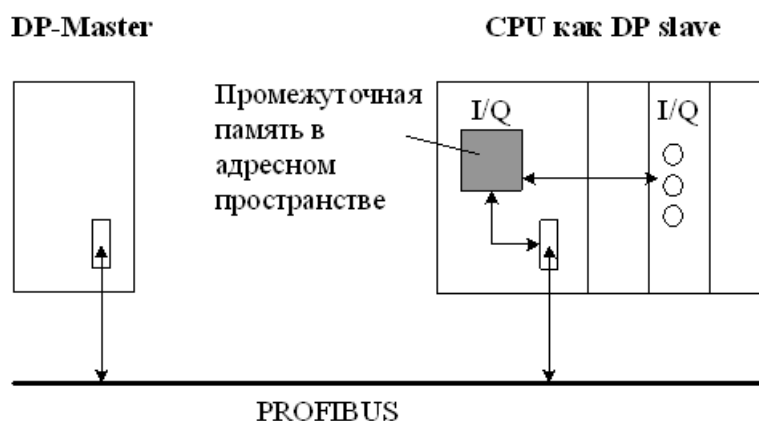


Рис. 2.4. Промежуточная память в CPU 315-DP, используемая как I-Slave

В промежуточной памяти проектируются адресные области входов (I) и выходов (Q). STEP7 для сетей PROFIBUS предусматривает возможность:

- запроектировать до 32 адресных областей входов и выходов;
- каждая из этих адресных областей может иметь величину (длину) до 32 байт;
- всего можно запроектировать максимум 244 байта входов и 244 байта выходов.

Следующая таблица (табл. 2.1) показывает принцип проектирования адресных областей. Для простоты для I и Q приняты одинаковые адреса.

Таблица 2.1

	Адрес Master-устройства	Адрес Slave-устройства	Длина	Единица
1	I100	Q100	2	Байт
2	Q100	I100	2	Байт

Указанные в табл. 2.1 адресные области CPU как Master-устройства DP и CPU как I-Slave-устройства DP задаются проектировщиком. Нельзя использовать адреса I и Q, используемые физическими каналами модулей ввода и вывода. Следует помнить, что в принципе для CPU315-2DP адресная область входов/выходов составляет 2048 байтов/2048 байтов.

Конфигурирование S7-300/CPU 315-2DP как Slave производится идентично изложенному ранее. Различие будет в указании работы DP интерфейса, то есть перевести из «DP-Master'a» (стоит по умолчанию) в «DP-Slave». Для этого выделите «DP» у CPU315-2DP. Откроется окно «Properties-DP Master». Откройте закладку «Operating Mode [Режим работы]» и установите режим работы «DP-Slave». После этого название «DP-Master» изменится на «DP-Slave». Далее перейдите на закладку «Configuration [Конфигурация]». Нажмите на кнопку New и в окне «Properties-DP – (Ro/S2.1) – Configuration – Row1» установите требуемый адрес промежуточной памяти для принимаемой информации, например, Input 100, Length: 2, Unit: Byte. После нажатия на кнопку ОК возвращаетесь в предыдущее окно, вновь нажимаете кнопку New и теперь устанавливаете адрес промежуточной памяти для информации выдаваемой, например, Output 100, Length: 2, Unit: Byte.

После нажатия на кнопку ОК возвращаетесь в окно HW Config.

Перед выходом из «HW Config» вновь используйте команду меню «Station», далее «Check Consistency», чтобы убедиться, что в конфигурации станции нет ошибок. Затем через команды «Save» или «Save and Compile» возвращаетесь в окно проекта SIMATIC Manager.

Теперь созданный «Slave S7-300» необходимо перенести на сеть «Profibus (1)». Для этого нужно перейти в окно «HW Config» для станции «Master' a S7-300». Откройте в каталоге оборудования папку «PROFIBUS-DP», а в ней подкаталог «Configured Stations», и подключите объект «CPU31x-2DP» к шине («вешалке») DP-Master'a методом Drag&Drop. При этом появляется окно «DP slave properties» с открытой закладкой «Connection», где показан перечень соответствующих Slave'ов. Выделите нужный и нажмите кнопку «Connect». Выбранный Slave исчезает из списка. Перейдите на закладку «Configuration» и внесите туда через функцию Edit адреса и длину промежуточной памяти для Master' a как показано на рис. 2.5.

Row	Mode	Partner DP a...	Partner addr	Local addr	Length	Consiste ...
1	MS	2	Q100	I100	2 Byte	Unit
2	MS	2	I100	Q100	2 Byte	Unit

Рис. 2.5. Информация на закладке «Configuration», «DP slave properties», «HW-Config»

Приведенные здесь адреса свободно выбраны для проекта-примера. Вы можете задавать другие области и адреса. Нужно только обратить внимание на то,

чтобы выходная область DP-Master' а всегда соответствовала входной области DP-Slave и наоборот.

Вернитесь в главное окно «HW Config [Master (Configuration)]» с помощью кнопки ОК. Slave в этом окне получит PROFIBUS-адрес. Свой адрес имеет каждый компонент подключенный к сети. Этот адрес присваивается STEP7 компонентам автоматически, но проектировщик его может изменить по своему усмотрению.

Компоненты распределенной периферии на лицевой панели имеют наборное поле для установки его адреса в сети. Необходимо следить, чтобы установленный адрес соответствовал адресу компонента в конфигурационной таблице.

Сохраняем проекты конфигурирования «Master S7-300» и «Slave S7-300» с компиляцией. На этом процесс конфигурирования закончен.

2.5. Программирование контроллеров

Для проверки работоспособности сконфигурированной сети необходимо загрузить в контроллеры проверочную программу. Как осуществить набор программы рассматривается в работе 1 «Программирование контроллера Simatic S7-300» этого пособия.

В лабораторной установке модули контроллеров Master'a и Slave'a выполнены на одинаковых CPU и отличаются только модулями ввода/вывода. В «Блоке имитационных сигналов» для них предусмотрены идентичные кнопки и тумблеры для имитации входных сигналов и команд и светодиоды для индикации выходных сигналов и команд. При конфигурировании контроллеров у обоих из них будут присутствовать байт входов IB0 и байт выходов QB4.

Ниже приводится два варианта программ, которые предусматривают при подаче сигналов на модуль ввода с адресом IB0 Master'a загораются индикаторы выходных сигналов адреса QB4 Slave'a и наоборот.

На рис. 2.6 представлены пользовательские программы в организационных блоках OB1 для Master'a (рис. 2.6а) и для Slave'a (рис. 2.6б) с использованием команды MOVE.

При передаче между Master'ом и Slave'ом больше четырех последовательных байт данных (консистентных данных) используются функции SFC14 и SFC15 (SFC – System Function Call).

Функция SFC14 применяется для чтения последовательных принимаемых данных (приемник) (рис. 2.7).

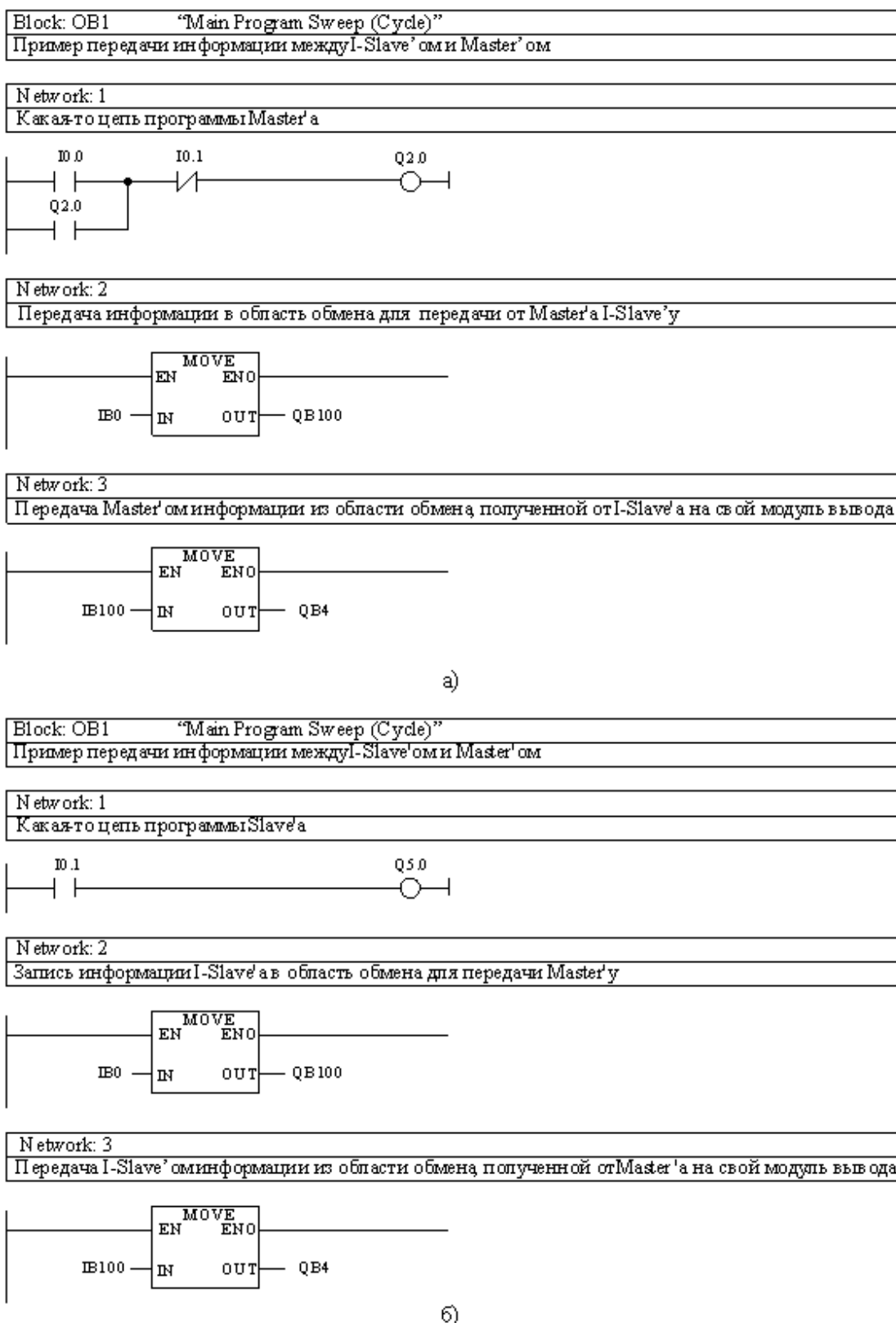


Рис. 2.6. Пример передачи информации между контроллерами с использованием команд MOVE: а) – программа в OB1 Master'a, б) – программа в OB1 I-Slave'a

Параметр LADDER (рис. 2.7) в шестнадцатеричном формате указывает начальный адрес области (промежуточной памяти) в которую поступают принимаемые данные, например w#16#64.

Параметр RECORD (рис. 2.7) указывает область, в которую направляется полученная информация, например P#M40.0 BYTE8.

Параметр RET_VAL (рис. 2.7) указывает адрес памяти, в котором будет код ошибки при сбое передачи информации, например MW60.

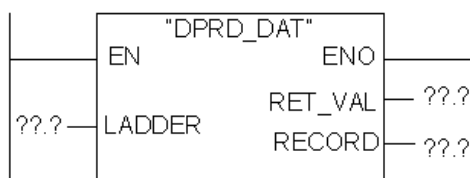


Рис. 2.7. SFC14 на языке лестничных диаграмм

Функция SFC15 применяется при передаче данных (передатчик) (рис. 2.8).

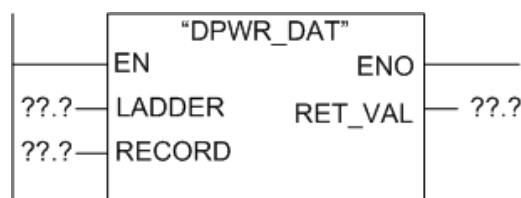


Рис. 2.8. SFC15 на языке лестничных диаграмм

Параметр LADDER (см. рис. 2.8) в шестнадцатеричном формате указывает начальный адрес области (промежуточной памяти) куда записать данные для отправки, например, w#16#64.

Параметр RECORD (см. рис. 2.8) указывает, откуда взять информацию, которую нужно передать, например P#M80.0 BYTE8.

Параметр RET_VAL (см. рис. 2.8) указывает адрес памяти, в котором будет записан код ошибки при сбое передачи информации, например MW70.

На рис. 2.9 схематично представлен обмен данными между Master'ом и Slave'ом с использованием функций SFC14 и SFC15.

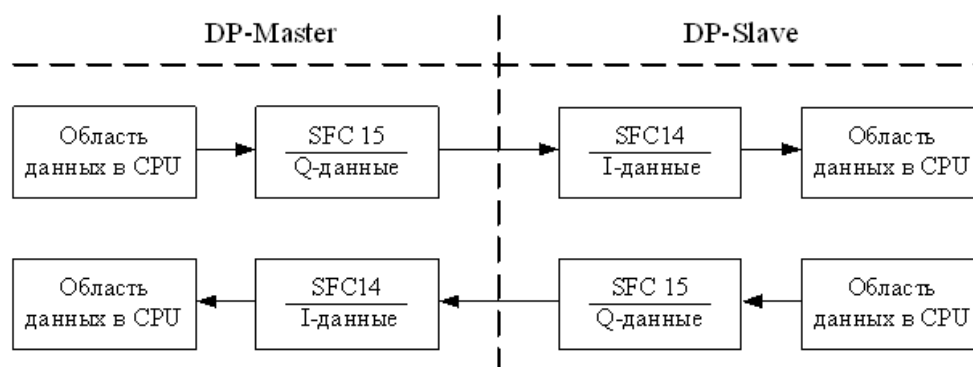


Рис. 2.9. Обмен входными/выходными данными через функции SFC14 и SFC15

На рис. 2.10 представлены программы для контроллеров те же самые, что и на рис. 2.6, но с использованием функций SFC14 и SFC15.

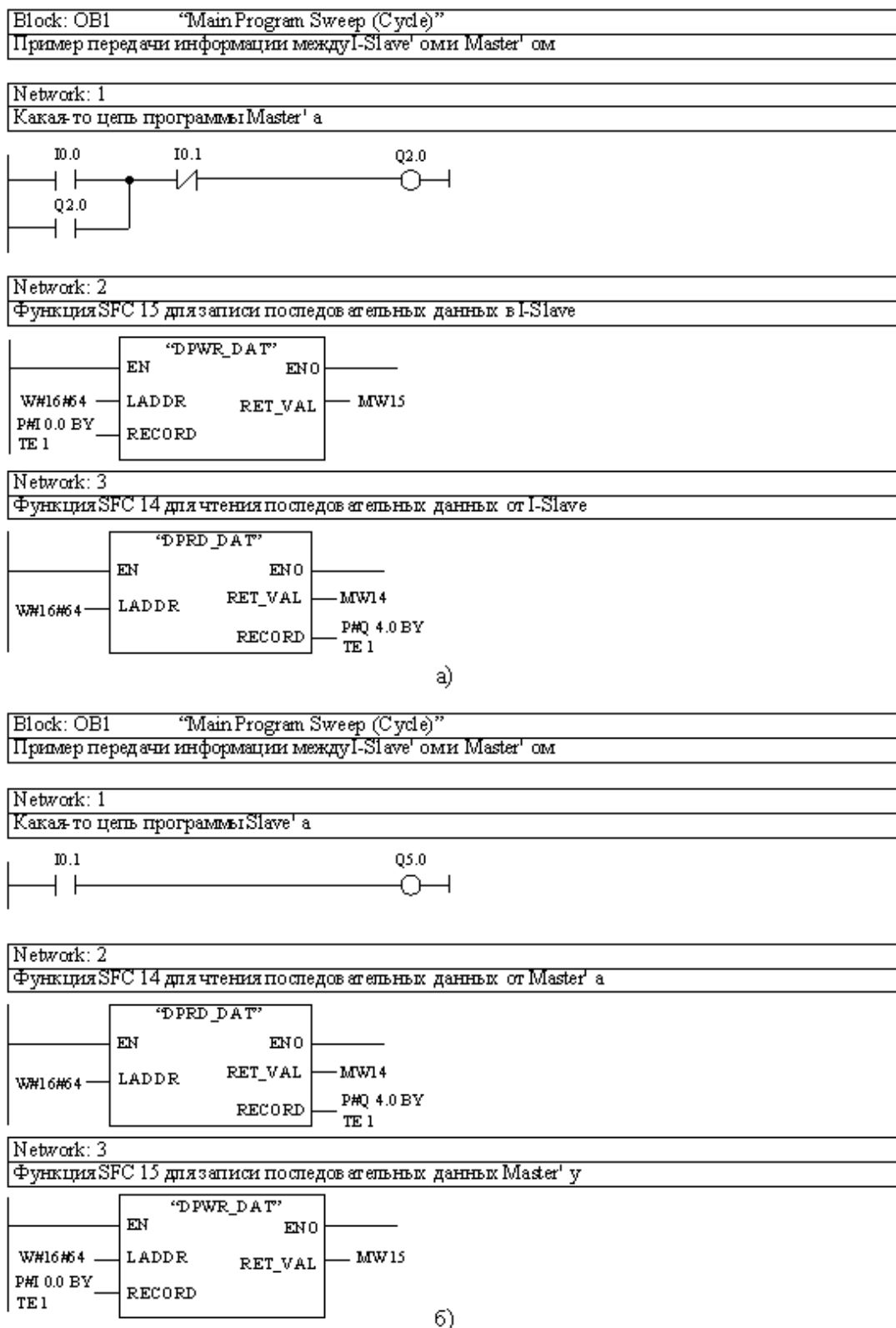


Рис. 2.10. Пример передачи информации между контроллерами с использованием функций SFC14 и SFC15:

а) – программа в OB1 Master'a, б) – программа в OB1 I-Slave'a

Программы демонстрируют, что функциями SFC14 и SFC15 можно передавать информацию с числом байт меньше четырех.

Slave'ами распределенной периферии, например, типа ET200M Master распоряжается как «собственными» модулями ввода/вывода, используя ту адресацию, которая определена при конфигурации сети. При этом не обращается внимание на промежуточные усилительно-повторительные элементы, проводную или оптоволоконную среду передачи. Если же необходимо использовать информацию с модулей распределенной периферии I-Slave'ом на программируемом контроллере, в нашем случае на SIMATIC S7-300/CPU 315-2DP, то необходимо ее получать или передавать через Master'a.

При вводе в сеть I-Slave'a на программируемом контроллере всегда как в программе Master'a, так и в программе Slave'a предусматривайте введение организационных блоков OB82 (диагностические сигналы) и OB86 (выход из строя носителя модулей). **Без этих блоков обмен между Master'ом и Slave'ом невозможен.**

2.6. Загрузка программ в программируемые контроллеры и включение в работу

Перед загрузкой программы в любой контроллер необходимо произвести стирание памяти его CPU. Для этого необходимо осуществить осмысленную последовательность операций с его переключателем режимов работы на лицевой панели CPU:

- перевести переключатель в положение STOP;
- перевести переключатель в положение MRES. Удерживайте переключатель в этом положении, пока светодиод STOP не загорится во второй раз. После этого отпустите переключатель. Он вернется в положение STOP;
- в течение 3 с необходимо снова перевести переключатель в положение MRES и держать так до тех пор, пока светодиод STOP мигает быстро (с частотой 2 Гц). Теперь переключатель можно отпустить. Когда CPU совершит общее стирание, светодиод STOP перестает мигать и горит ровным светом.

Указанные операции с переключателем режимов необходимы только тогда, когда пользователь сам хочет выполнить общее стирание CPU без запроса на эту операцию со стороны CPU (медленное мигание светодиода STOP). Если CPU сам запрашивает общее стирание, то достаточно кратковременно переключатель режимов работы перевести в положение MRES, чтобы запустить процесс общего стирания CPU. Если CPU после его успешного стирания вновь требует общего стирания, то в определенных случаях необходимо форматирование его сменной платы микропамяти SIMATIC (MMC).

Для форматирования платы MMC необходимо:

- переключатель режимов работы перевести в положение MRES и удерживать его в этом положении (около 9 с), пока светодиод STOP не загорится ровным светом;

– в течение следующих 3 с необходимо отпустить переключатель и вновь перевести его в положение MRES. Теперь светодиод STOP мигает только во время форматирования.

Всегда выполняйте эту последовательность операций в течение указанного времени, так как в противном случае MMC не форматируется, а возвращается в состояние общего стирания.

В сети у каждого контроллера будь то Master или Slave есть своя пользовательская программа.

В лабораторной установке используется контроллер Master и один I-Slave. Поэтому в левой части окна SIMATIC Manager, которая содержит структуру проекта, имеется два проекта «DP-Master» и «DP-Slave». Они должны загружаться каждый в свой контроллер.

Запись программ в контроллеры, как уже было указано выше, осуществляется по сети.

Для записи необходимо, чтобы:

- компьютер через адаптер был подключен к разъему DP CPU «DP-Master'a»;
- все компоненты сети должны иметь разные сетевые адреса и действительная конфигурация должна соответствовать реальной структуре;
- конфигурация может быть загружена только при согласованности и отсутствии ошибок.

Последовательность записи следующая:

- в меню SIMATIC Manager выбрать функцию «Option», затем «Set PG/PC Interface», а в ней режим «PC Adapter (Auto)»;
- переместитесь через «Master S7-300», CPU 315-2DP, S7 Program (1) и щелкните на символе «Blocks [Блоки]». Ключ режима контроллера поставьте в положение STOP;
- из меню «PLC [ПЛК]» выберите команду «Download [Загрузить]» для передачи программы и конфигурации аппаратуры в CPU. Щелкните на Yes во всех появляющихся окнах. Запись конфигурации и пользовательской программы Master'a записана в соответствующий контроллер;
- аналогично следует поступить с проектом «Slave S7-300», то есть через «Slave S7-300», CPU 315-2DP, S7 Program (2) и так далее.

Для запуска сети необходимо перевести переключатели режимов работы контроллеров в положение «Run».

Во время работы программы можно на экране монитора компьютера следить за ходом ее выполнения. Для этого при нахождении переключателей режимов работы контроллеров в положениях «Run» следует нажать на панели инструментов монитора кнопку с изображением очков «Monitor (on/off) [Монитор (включение/выключение)]». При этом изменением цвета будут отображаться состояния контактов и катушек, а также значения счетчиков и таймеров в реальном времени.

2.7. Выполнение лабораторной работы

Схема лабораторной установки перед началом работы полностью собрана и никаких изменений в процессе работы не предусматривается. Программы предыдущих бригад студентов удалены из компьютера.

В начале работы необходимо:

- ознакомиться с составом лабораторной установки, конструктивным исполнением компонентов сети и соединениями между ними. Обратить особое внимание на подключение кабелей проводных (витая пара) RS485 и оптического;
- включить компьютер, запустить программу «SIMATIC Manager [Администратор SIMATIC]»;
- провести конфигурирование сети, убеждаясь в ее правильности, используя команды меню «Station > Consistency Check», перед выходом из окна «HW Config» с использованием команд меню «Save and Compile [Сохранить и компилировать]»;
- в организационные блоки OB1 Master'a S7-300 и Slave'a S7-300 ввести соответствующие программы обмена, представленные на рис. 2.6 или рис. 2.10 (по выбору студентов) для обмена информацией между указанными компонентами сети. Не забыть вставить в программы организационные блоки **OB82 и OB86**;
- включить блок питания сети и убедиться в наличии питания всех компонентов сети;
- осуществить стирание памяти CPU контроллеров;
- записать подготовленные дома программы с конфигурацией в соответствующие контроллеры Master'a S7-300 и Slave'a S7-300;
- проверить правильность функционирования сети: при включении/выключении входов с адресом байта I0 контроллера Master'a S7-300 соответственно загораются/гаснут выходы байта Q4 Slave'a S7-300 и наоборот;
- если проверка прошла успешно, то можно приступить к выполнению индивидуальных заданий.

Вариант задания выбирается в соответствии с табл. 2.2.

Таблица 2.2

Номер бригады	1 и 5			2 и 6			3 и 7			4 и 8		
Член бригады	1	2	3	1	2	3	1	2	3	1	2	3
Вариант задания	1	2	3	4	5	6	7	8	9	10	11	12

В каждом варианте предусматривается ручное и автоматическое управление объектом. Объектом управляет контроллер DP-Slave' а 1, команды задания выбора режима работы (ручной/автоматический) и пуска/остановка автоматического режима работы подаются с контроллера DP-Master' а, а команды ручного управления – со Slave'a ET200M.

Ниже рассмотрен пример программирования управления тележкой, перемещающейся по прямолинейному пути от положения П1 к положению П2 и обратно (рис. 2.11).

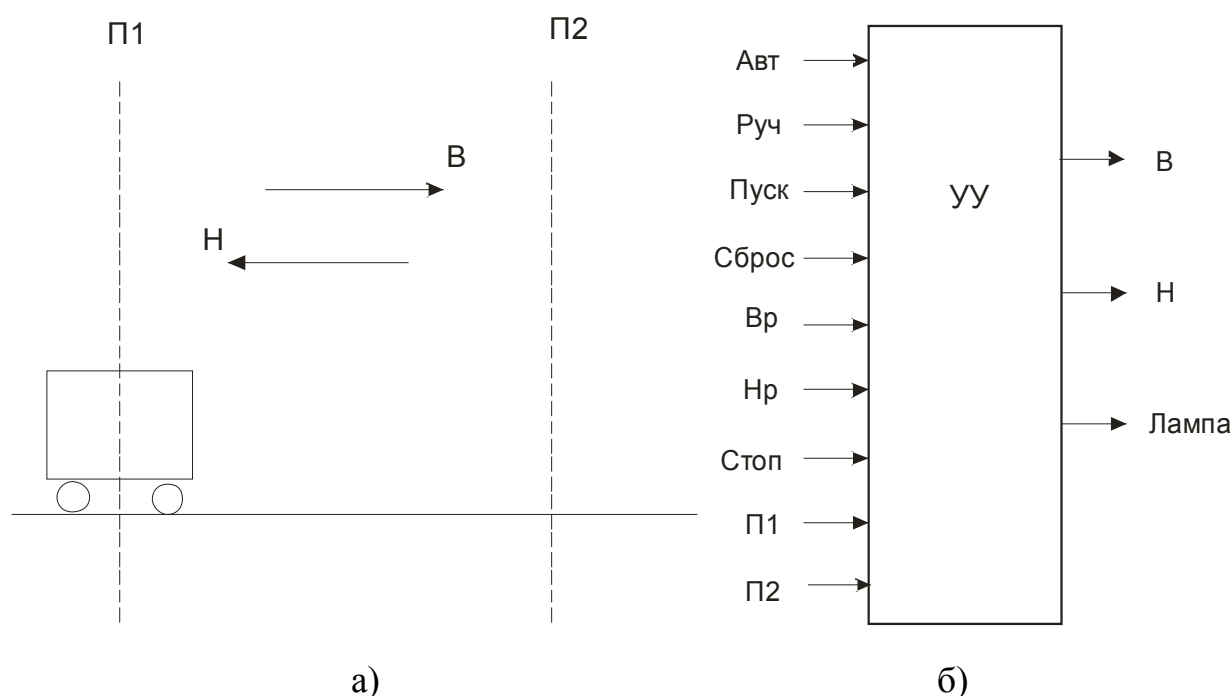


Рис. 2.11. К условию задачи: а) конструктивная схема механизма; б) входные и выходные сигналы/команды устройства управления УУ

Исходное положение тележки П1. В автоматическом режиме работы «Авт» при нажатии кнопки «Пуск» возникает команда на движение тележки вперед «В». Тележка доходит до положения П2, команда «В» отключается. Возникает команда на движение назад «Н». Тележка доходит до положения П1, команда «Н» отключается. Возникает команда «В», тележка движется до П2. Включается команда «Н» и т.д. Тележка непрерывно перемещается от положения П1 к П2 и обратно. При работе в автоматическом режиме включена световая индикация «Лампа».

При нажатии кнопки «Сброс», независимо от того где находилась тележка, она немедленно должна вернуться в положение П1. «Лампа» гаснет.

В ручном режиме работы «Руч» команды «В» и «Н» формируются при нажатии кнопок ручного управления соответственно «Вр» и «Нр». Останов тележки в ручном режиме – при нажатии кнопки «Стоп».

Для управления тележкой вводится память о включении автоматического режима « p_a ». Она возникает в положении П1 при нажатии кнопки «Пуск» и сохраняется до появления сигнала памяти « p_c » о нажатии кнопки «Сброс». Этому соответствует логическое уравнение:

$$p_a = Авт \cdot (П1 \cdot Пуск + p_a) \cdot \overline{p_c}$$

Память « p_c » возникает в автоматическом режиме при любом положении тележки при нажатии кнопки «Сброс» и сохраняется до нажатия кнопки «Пуск»:

$$p_c = Авт \cdot (Сброс + p_c) \cdot \overline{Пуск}$$

Команда на движение вперед «В» возникает в автоматическом режиме в положении П1 и сохраняется до достижения тележкой положения П2. В ручном режиме она возникает в любом положении тележки при нажатии кнопки «Вр» и сохраняется до нажатия кнопки «Стоп» или до достижения положения П2. Этому соответствует логическое уравнение:

$$B = \left[p_a \cdot (П1 + B) + p_a \cdot \overline{H} + Руч \cdot (Вр + B) \cdot \overline{Стон} \right] \cdot \overline{П2}$$

Команда на движение тележки назад «Н» возникает в автоматическом режиме в положении П2 и сохраняется до достижения тележкой положения П1. Она возникает также в любом положении тележки при наличии сигнала памяти сброса и отсутствии движения вперед. В ручном режиме команда «Н» возникает в любом положении тележки при нажатии кнопки «Нр» и сохраняется до нажатия кнопки «Стоп» или до достижения положения П1. Этому соответствует логическое уравнение:

$$H = \left[p_a \cdot (П2 + H) + p_a \cdot \overline{B} + Руч \cdot (Нр + H) \cdot \overline{Стон} \right] \cdot \overline{П1}$$

Для исключения одновременного появления команд «В» и «Н» в уравнения этих команд вводится взаимная блокировка. Тогда:

$$B = \left[p_a \cdot (П1 + B) + Руч \cdot (Вр + B) \cdot \overline{Стон} \right] \cdot \overline{П2} \cdot \overline{H}$$

$$H = \left[p_a \cdot (П2 + H) + p_a \cdot \overline{B} + Руч \cdot (Нр + H) \cdot \overline{Стон} \right] \cdot \overline{П1} \cdot \overline{B}$$

«Лампа» включается сигналом « p_a », т.е. по логическому уравнению:

$$Лампа = p_a$$

По условию задачи команды выбора режима «Авт», «Руч» и пуска/останова автоматического режима («Пуск», «Сброс») подаются с контроллера Master S7-300, а команды ручного управления «Вр», «Нр» и «Стоп» – со Slave'a ET200M. Механизмом управляет Slave S7-300 (рис. 2.12).

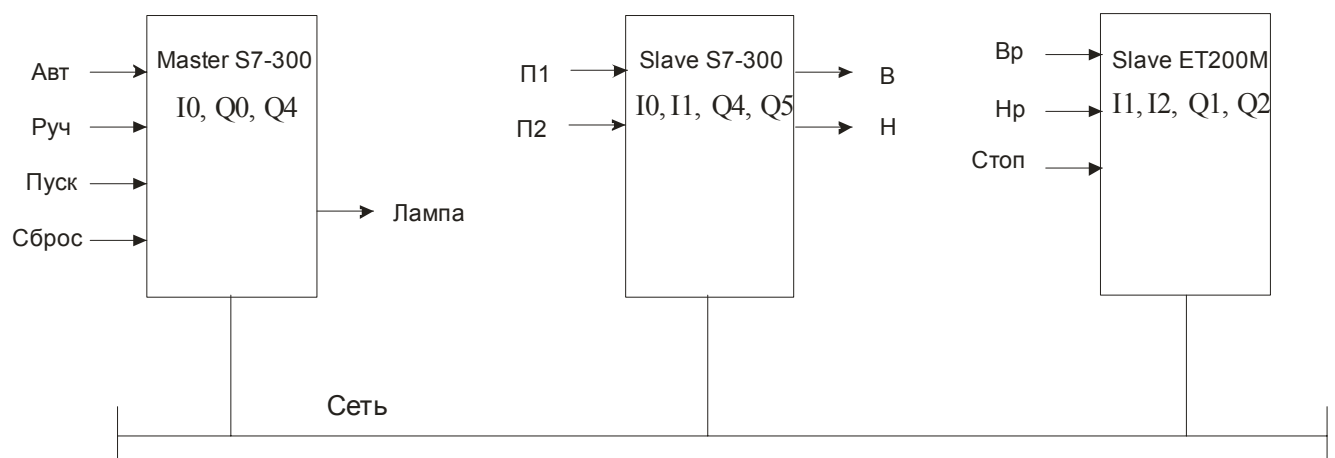


Рис. 2.12. Подключение входных/выходных сигналов к компонентам сети

После конфигурирования рассматриваемой сети автоматически были присвоены адреса байтов входов I и выходов Q модулям компонентов сети. Эти адреса указаны внутри корпусов компонентов сети на рис. 2.12. Обратите внимание, что входами/выходами децентрализованной периферии Slave'a ET200M контроллер Master S7-300 распоряжается как с входами/выходами своей стойки, но расположенными на удалении от этой стойки (через кабель с витой парой, оптический шинный терминал, оптоволоконный кабель и IM153-2 станции ET200M).

При программировании пользовательской программы следует всем входам и выходам рассматриваемого примера присвоить физические адреса. В примере принято, что у Master'a все входы подключены к байту I0, а выход к байту Q4. У Slave'a ET200M все выходы ручного управления подключены к байту I1. У Slave'a S7-300 сигналы с датчиков положения П1, П2 подключены к байту I0, а выходные команды В = Н – к байту Q4 выходных сигналов.

Воспользуемся возможностью символьного представления адресов контроллеров Simatic. Символы сигналам и командам присвоим те, которые использованы в логических уравнениях. В табл. 2.3 представлена адресация сигналов и команд Master'a S7-300.

Таблица 2.3

Адресация переменных Master'a S7-300

Символ сигнала, команды	Авт	Руч	Пуск	Сброс	Вр	Нр	Стоп	Лампа
Адрес физический	I0.0	I0.1	I0.2	I0.3	I1.0	I1.1	I1.2	Q4.0

Так как объектом управляет Slave S7-300, а обеспечивает Master S7-300, то в Slave S7-300 выделяется область памяти для обмена информацией с Master'ом S7-300. Воспользуемся адресной областью, рассмотренной в табл. 2.1, в которой для Master'a и Slave'a присвоено по два бита с адресов IB100 и QB100.

Тогда программа для Master'a S7-300 принимает вид, представленный на рис. 2.13.

Команды управления, поступающие с Master'a S7-300, будут размещаться в байтах соответственно IB100 и IB101 Slave'a S7-300. Выдаваемый со Slave'a S7-300 сигнал «Лампа» будет передаваться в составе байта QB100 Slave'a S7-300 на вход IB100 Master'a S7-300.

С учетом этого у Slave'a S7-300 адресация сигналов и команд принимается следующая (табл. 2.4).

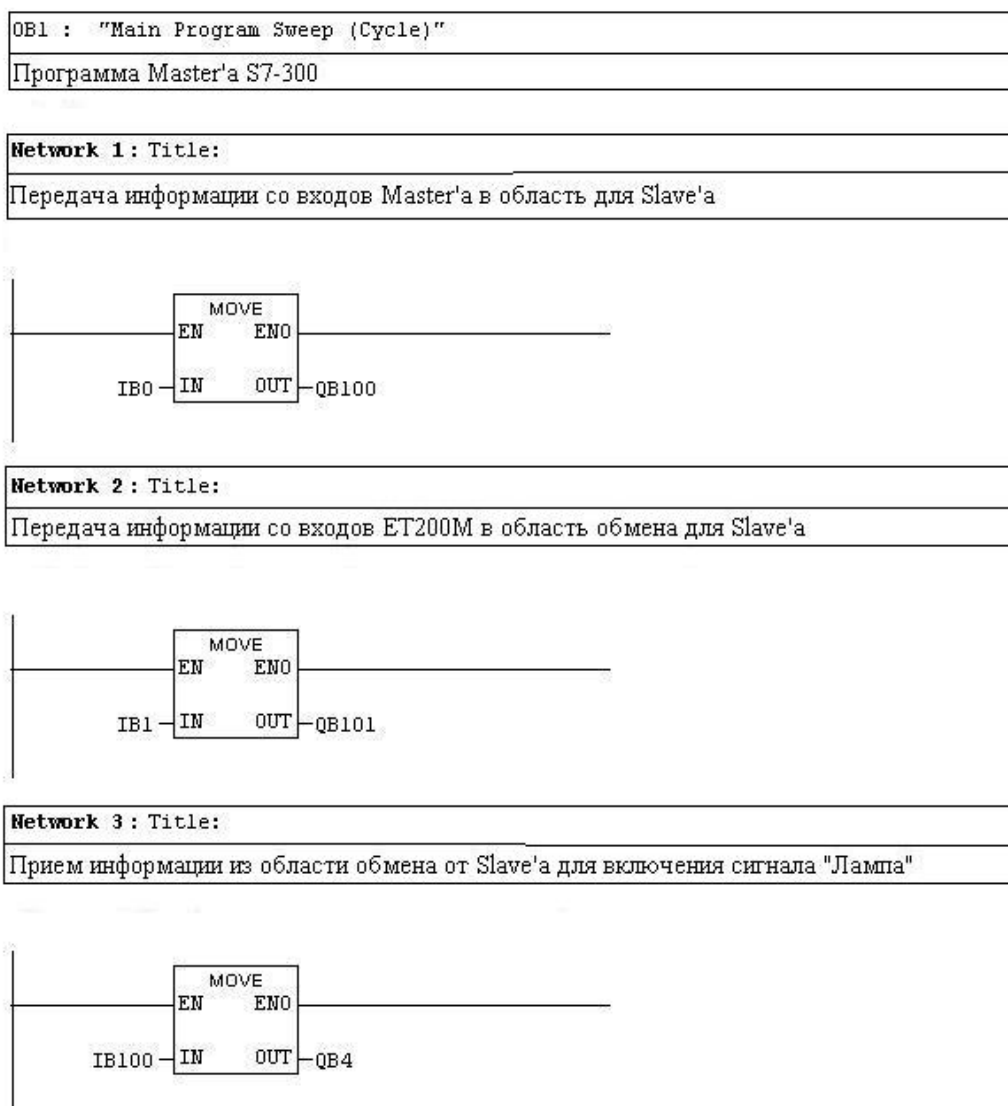


Рис. 2.13. Программа для Master'a S7-300

Таблица 2.4

Адресация переменных Slave'a S7-300

Символ сигнала, команды	Авт	Руч	Пуск	Сброс	Вр	Нр	Стоп
Адрес физический	M0.0	M0.1	M0.2	M0.3	M1.0	M1.1	M1.2

Символ сигнала, команды	Лампа	П1	П2	В	Н	р _а	р _с
Адрес физический	M2.0	I0.0	I0.1	Q4.0	Q4.1	M3.0	M3.1

Из табл. 2.4 видно, что в программе Slave'a S7-300 принимаемые от Master'a S7-300 команды должны заноситься в байты меркеров соответственно MB0 и MB1, а сигнал на включение лампы записывается в байт меркеров MB2 для передачи Master'у.

На рис. 2.14 представлена программа для Slave'a S7-300 на языке лестничных диаграмм.

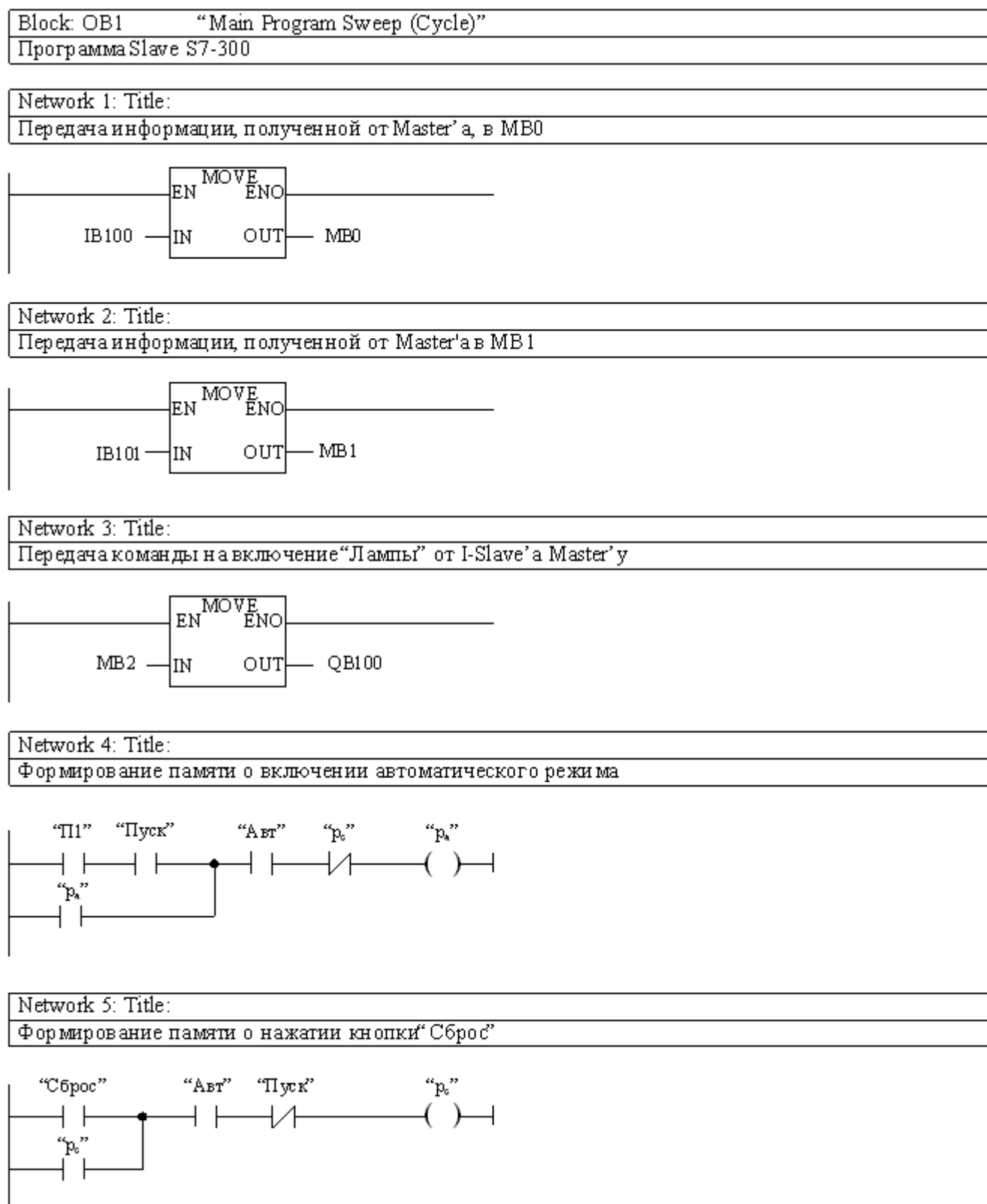


Рис. 2.14. Программа для Slave'a S7-300

Вариант 1. Автоматизировать управление дверью типа двери кабины лифта. Нормальное положение двери закрытое. Есть кнопка «Открыть», датчики открытого и закрытого положения двери соответственно До и Дз, а также датчик наличия препятствия Пр закрывания двери.

При нажатии кнопки «Открыть» происходит открытие двери. Через 5 с формируется команда на закрытие двери. Если при закрытии двери появился сигнал препятствия, закрывание двери отключается и дверь открывается. Через 5 с вновь формируется команда на закрывание двери. Если вновь появился сигнал препятствия, то дверь снова открывается на 5 с и так далее. После 3 таких закрываний/открываний, вызванных препятствием, дверь открывается и остается открытой, а на диспетчерском пункте возникает сигнал «Препятствие».

Для устранения неисправностей предусмотрена кнопка «Сброс» и кнопки ручного управления «Открыть» и «Заккрыть».

Вариант 2. Реализовать управление формированием выходной команды Х, зависящей от входных сигналов А, В и С.

В автоматическом режиме при комбинации $A=1$ и $B=1$ выходной сигнал принимает значение 1. После исчезновения сигнала А или В выходной сигнал Х принимает значение 0. При последующих появлениях комбинации $A=1$ и $B=1$ сигнал Х не должен принимать значение 1 до тех пор, пока хотя бы кратковременно не появится сигнал С. То есть сигнал С возвращает схему в исходное состояние.

При ручном управлении любая комбинация $A=1$ и $B=1$ вызывает появление команды Х.

Вариант 3. Реализовать управление светофором. Горение ламп в следующей последовательности: Красный, желтый, зеленый, желтый, красный, желтый и так далее. Время горения каждой лампы условно 5 с.

В ручном режиме работы (настроечном) каждая лампа управляется своим тумблером или кнопкой и можно включать их в любой комбинации.

Вариант 4. Автоматизировать процесс передачи детали с рольганга (роликового транспортера) 1 на транспортер 2 (рис. 2.15).

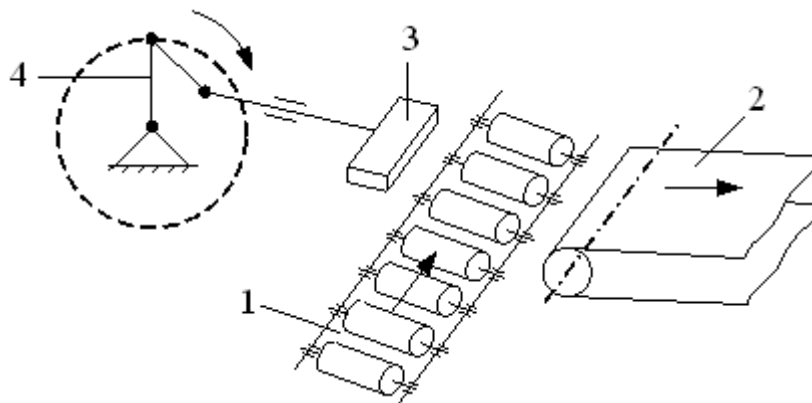


Рис. 2.15. Конструктивная схема автоматизируемого объекта

Рольганг 1 движется до появления детали перед толкателем 3 с кривошипно-шатунным механизмом 4. Рольганг останавливается, включается привод кривошипа толкателя, и толкатель сталкивает деталь на транспортер 2, который затем включается на 5 с, а кривошип толкателя, сделав полный оборот, возвращается в исходное положение. Через 2 с после прихода толкателя в исходное положение вновь включается рольганг 1.

При ручном управлении механизмы управляются при нажатии на соответствующие кнопки включено/отключено.

Вариант 5. Автоматизировать процесс распределения заготовок.

Машина М1 (рис. 2.16) производит изделия, которые необходимо по 5 штук поочередно передавать то к рабочему месту 1, то к рабочему месту 2. Подача изделий осуществляется рольгангами (роликовыми транспортерами) 1, 2 и 3.

Имеется направляющая планка 4, которая может занимать только 2 положения для направления заготовок к соответствующему рабочему месту. Положение планки зависит от числа переданных изделий. После передачи 5 изделий к одному рабочему месту она перебрасывается для подачи изделий к другому рабочему месту и наоборот.

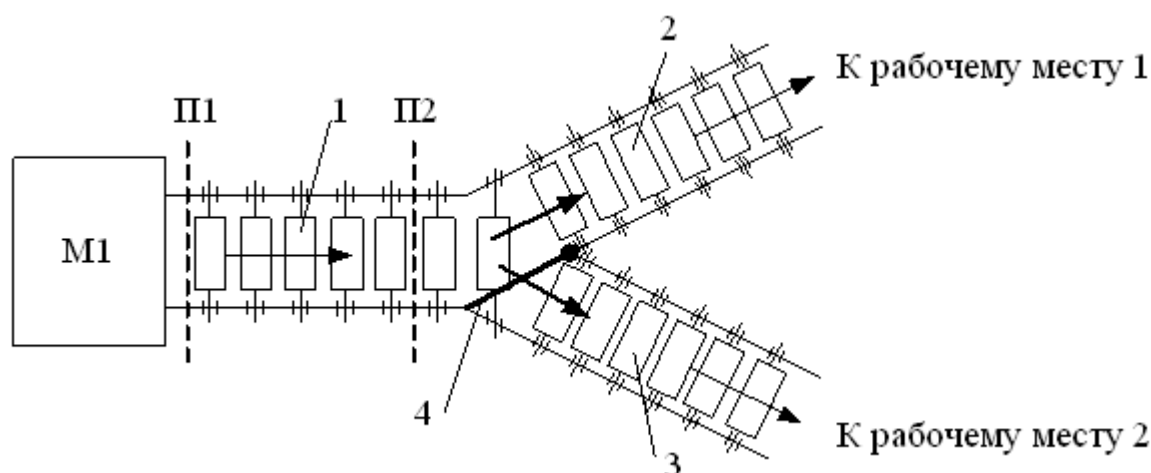


Рис. 2.16. Конструктивная схема распределения изделий

При подаче питания планка 4 должна занимать положение для направления изделий к рабочему месту 1. При выходе изделий из машины М1 по сигналу о положении П1 включается рольганг 1. При достижении положения П2 включается на 10 с, в зависимости от положения планки, рольганг 2 или рольганг 3. Рольганг 1 отключается через 3 с после достижения изделием положения П2. Исключить возможность удара изделия направляющей планкой при ее перебрасывании.

При ручном управлении механизмы управляются соответствующими кнопками.

Вариант 6. Автоматизировать процесс сортировки деталей.

Машина М1 (см. рис. 2.16) выдает на рольганг 1 вперемешку то низкие, то высокие детали. Высокие детали необходимо отправлять рольгангом 2 к рабочему месту 1, низкие – рольгангом 3 к рабочему месту 2. Для этого имеется направляющая планка 4, которая может занимать только два положения для направления деталей к соответствующему рабочему месту.

При выдаче детали из машины М1 по сигналу положения П1 включается рольганг 1. При движении детали от положения П1 к положению П2 организовать контроль высоты детали и организовать переброс планки 4 в соответствующее положение в зависимости от высоты детали. В положении П2 включается соответствующий, в зависимости от высоты детали, рольганг 2 или 3. Рольганг 2 включается на 5с, рольганг 3 – соответственно на 7 с. Рольганг 1 отключается через 3 с после достижения деталью положения П2.

Организовать счет высоких и низких деталей. Через каждые 3 штуки выданных высоких деталей включается на 3 с световой сигнал и счет начинается снова. Аналогично формируется световой сигнал на 3 с после прохождения 3 штук низких деталей.

При ручном управлении механизмы управляются соответствующими кнопками включено/отключено.

Вариант 7. Кривошипно-шатунный толкатель имеет одно контролируемое положение П0. В автоматическом режиме при кратковременном нажатии на кнопку «Пуск» привод кривошипа немедленно включается и из положения П0 движется «Вперед». Кривошип совершает полный круг, возвращается в положение П0 и останавливается. По истечении 3 с привод вновь включается, кривошип совершает полный круг, возвращается в положение П0, вновь стоит 3 с, включается и так далее. После 5 толканий кривошип останавливается в положении П0 и формируется сигнал «Конец цикла».

В ручном режиме работы предусмотреть возможность движения кривошипного толкателя «Вперед» и «Назад».

Вариант 8. Тележка перемещается по кольцевому пути. Исходное положение П0, промежуточное положение П1. При кратковременном нажатии на кнопку «Пуск цикла» тележка немедленно движется «Вперед». В положении П1 останавливается на 3 с, вновь включается и движется в положение П0. Стоит в П0 3 с, включается «Назад», в положении П1 стоит 3 с и затем продолжает двигаться «Назад» до положения П0. В П0 формируется сигнал «Конец цикла».

В ручном режиме управления предусмотреть движение тележки «Вперед» и «Назад».

Вариант 9. Тележка перемещается по кольцевому пути. Имеются 3 фиксированных положения П0...П2. При кратковременном нажатии на кнопку «Пуск цикла» привод тележки немедленно включается «Вперед» к положению П1. В П1

стоит 3 с и продолжает движение «Вперед» до положения П2. В положении П2 тележка стоит 3 с и включается «Назад» до положения П1. В положении П1 тележка стоит 3 с, включается «Вперед», останавливается в положении П2 на 3 с и затем продолжает движение «Вперед» до положения П0, где формируется сигнал «Конец цикла».

В ручном режиме управления предусмотреть движение тележки «Вперед» и «Назад».

Вариант 10. У шлагбаума при въезде на территорию гаража имеется датчик контроля препятствия в створе шлагбаума (машина, человек). У оператора для управления в автоматическом режиме есть кнопка «Открыть», для управления в ручном режиме кнопки «Открыть» и «Заккрыть».

Нормальное положение шлагбаума закрытое. В автоматическом режиме при появлении автомобиля оператор нажимает кнопку «Открыть». Шлагбаум поднимается в крайнее верхнее открытое положение. Через 5 с он включается на закрытие, если нет сигнала датчика препятствия. Если при закрывании шлагбаума появится сигнал препятствия, то шлагбаум вновь откроется и вновь сделает попытку на закрывание через 5 с при отсутствии сигнала препятствия. Если сделано 5 таких попыток на закрытие, то шлагбаум остается открытым и автоматический режим сбрасывается.

При ручном управлении команды на открывание/закрывание шлагбаума осуществляется оператором, на которого возлагается контроль наличия препятствия.

Вариант 11. У двери кабинета офисного здания 2 замка: один механический, второй электромагнитный. Механический замок закрывается ключом на ночь, а днем этот замок открыт. Днем работает электромагнитный замок. При включенном электромагните замок открыт. Для управления замком на двери со стороны коридора стоит тумблер (включен/отключен) и светодиод, с внутренней стороны – одна кнопка «Открыть».

Нормальное положение замка под действием пружин на ригель закрытое. Электромагнит отключен. Для открытия замка со стороны коридора используется осмысленная последовательность действий с тумблером аналогичная последовательности при сбросе памяти CPU контроллера SIMATIC S7-300. Тумблер следует поставить в верхнее (включенное) положение. Начинается мигание светодиода. Когда он включится во второй раз нужно немедленно тумблер выключить. Электромагнит включается, открывая замок на 10 с, после которых он выключается. Отключенный замок не препятствует захлопыванию двери. С внутренней стороны замок на 10 с открывается при кратковременном нажатии на кнопку «Открыть».

Вариант 12. Автоматизировать тактовый стол для подачи заготовок к станку. Стол круглый с центральной осью вращения по типу стола капитал-шоу «Поле Чудес». Стол разбит на секторы, на которые устанавливаются заготовки. Есть

датчик, который реагирует на воздействующие элементы, закрепленные на каждом секторе. Когда робот снял очередную заготовку для ее установки на станке, он вырабатывает кратковременный импульс. По этому импульсу включается привод поворота стола, и движение продолжается до воздействия на датчик элемента очередного сектора.

При ручном режиме работы при нажатии кнопки «Вращение» стол вращается, не реагируя на сигналы датчика. При отпускании этой кнопки стол должен остановиться по сигналу датчика.

2.8. Требования к отчёту

Отчет должен содержать:

- а) цель работы;
- б) описание технологической задачи по варианту, конфигурацию или технологическую схему автоматизированного объекта, обозначение переменных, содержательное описание формирования логических функций, логические функции, соответствующие работе объекта с учетом ручного и автоматического режимов работы;
- в) физические и символические адреса переменных, отредактированные и проверенные программы для программируемых контроллеров Master S7-300 и Slave S7-300 с комментариями;
- г) описание методики и экспериментальной проверки работоспособности системы автоматизации в автоматическом и ручном режимах;
- д) выводы по работе.

2.9. Контрольные вопросы

1. Что означает термин PROFIBUS?
2. Какой «Slave» называется интеллектуальным слэйвом «I-Slave»?
3. В какой последовательности осуществляется конфигурирование сети PROFIBUS?
4. Как осуществляется проверка правильности проведенной конфигурации сети PROFIBUS?
5. Как осуществляется запись программ в контроллеры Master' а и Slave' а по сети PROFIBUS?
6. Как информацию с распределенной периферии передать в I-Slave?
7. Как в лабораторной работе проверить правильность работы сети PROFIBUS?
8. Как используется функция SFC14?
9. Как используется функция SFC15?