

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. И. Раззакова**

КЫРГЫЗСКИЙ ГЕРМАНСКИЙ ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра «ТЕЛЕМАТИКА»

ТЕЛЕМАТИКА

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

**Для студентов подготовки бакалавров по кредитной технологии
обучения по направлению 690600 «Телематика»**

Бишкек - 2015

«Рассмотрено»
На заседании кафедры
«Телематика»
Прот. № 7 от 17.03.2014г.

«Одобрено»
Методическим советом
КГТИ
Прот. №3 от 11.02.2015г.

Составитель: Кошоева Б.Б.

Телематика: Методические указания к выполнению лабораторных работ для студентов по направлению 690600 «Телематика» / КГТУ им. И. Раззакова; Сост.: Кошоева Б.Б./ - Б.: ИЦ «Текник», 2015. - 32 с.

В работе дается описание последовательности и основных приемов программирования программируемых логических контроллеров (ПЛК) серии SIMATIC S7-300/400 компании **Siemens** с помощью системы программирования SIMATIC STEP 7 V5.4.

Инструментальная система программирования SIMATIC STEP 7 V5.4 компании Siemens решает ряд других задач помимо программирования, который связан с созданием систем автоматизации с применением ПЛК и информационных коммуникационных технологий.

Руководство предназначено для студентов направления 690600 Телематика.

Табл.4, Ил. 37, Библиогр. 5 наименований

Рецензент: проф. Акматбеков Р.А.

Корректор *Эркинбек к. Ж.*
Редактор *Турдукулова А.К.*
Тех.редактор *Кочоров А.Д*

Подписано к печати 20.04.2015 г. Формат бумаги 60x84¹/₁₆.
Бумага офс. Печать офс. Объем 2 п.л. Тираж 50 экз. Заказ 244. Цена 34,2с.
Бишкек, ул. Сухомлинова, 20. ИЦ «Текник» КГТУ им. И.Раззакова, т.: 54-29-43
e-mail: beknur@mail.ru

СОДЕРЖАНИЕ

Лабораторная работа №1	
Знакомство с пакетом программ STEP 7, создание проекта.....	4
Лабораторная работа №2	
Средства диагностики контроллера S7-300.....	17
Лабораторная работа №3	
Программирование работы шифратора.....	18
Лабораторная работа №4	
Одноразовые операции STEP 7.....	20
Лабораторная работа №5	
Операции сдвига и ротации.....	21
Лабораторная работа №6	
Операции с таймерами и с счетчиками.....	23

ЛАБОРАТОРНАЯ РАБОТА №1

ЗНАКОМСТВО С ПАКЕТОМ ПРОГРАММ STEP 7, СОЗДАНИЕ ПРОЕКТА

Цель работы: Ознакомится с пакетом программ STEP7 Simatic Manager, освоить методы составления и редактирования программ, представленных бесконтактным планом. Оборудование: Пакет программ STEP7 Simatic Manager.

Теоретические сведения

В зависимости от степени сложности решаемых задач в контроллерах S7-300 может применяться 8 типов центральных процессоров:

- CPU 312 IFM: компактный центральный процессор с набором встроенных входов и выходов для ввода-вывода дискретных сигналов, который может быть использован в качестве автономной системы автоматизации.

- CPU 313: недорогой центральный процессор для построения небольших модульных систем автоматизации.

- CPU 314 IFM: компактный центральный процессор с набором встроенных входов и выходов для ввода-вывода дискретных и аналоговых сигналов, который может использоваться в качестве автономных систем автоматизации с малым временем реакции и выполнением специальных функций обработки сигналов.

- CPU 314: центральный процессор для построения модульных систем автоматизации с высокой скоростью обработки данных.

- CPU 315/CPU 315-2 DP: центральные процессоры для построения систем автоматизации со средним или большим объемом программы, обслуживающих системы локального и распределенного ввода-вывода, подключаемые по PROFIBUS-DP.

- CPU 316-2 DP: центральный процессор для построения модульных систем автоматизации со сложными алгоритмами обработки информации, использующих системы локального и распределенного ввода-вывода, подключаемые по PROFIBUS-DP.

- CPU 318-2 DP: центральные процессоры для выполнения программ большого объема и обслуживания разветвленных конфигураций распределенного ввода-вывода через сеть PROFIBUS-DP.

Основные технические характеристики центральных процессоров:

- Высокое быстродействие. Время выполнения логической команды в CPU 312IFM и CPU 313 составляет 600нс, в CPU 314IFM ... CPU 316 - 300нс, в CPU 318-2 - 100нс.

- Объемы оперативной памяти, соответствующие классу решаемых задач: от 6 Кбайт в CPU 312IFM до 512 Кбайт в CPU 318-2.

- Гибкие возможности расширения. CPU 312IFM и CPU 313 допускают подключение до 8 (однорядная конфигурация), остальные центральные процессоры - до 32 модулей (4-рядная конфигурация) ввода-вывода.

Для решения задачи автоматизации необходимо использовать языки программирования STEP 7. В них:

- Входные переменные обозначаются I(E). Адресация по байтам (IВ 0...127) – это переменные которые находятся в оперативной памяти контроллера и называются областью отображения входных переменных.
- Выходные переменные Q(A). QВ (0...127) – это область оперативной памяти, в которую записываются значения выходных сигналов периферийных модулей, перед тем как они переносятся в модуль вывода DC.
- Дискретные переменные F(M). FВ (0...255) – находятся во внутренней памяти процессора.

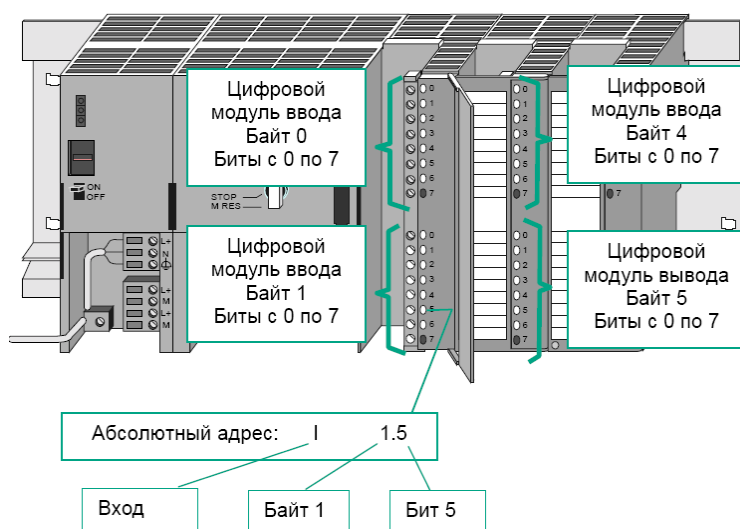


Рис. 1. Адресация в контроллерах S7-300

Для создания программ для контроллеров S7-300 используется пакета STEP 7 Simatic Manager.

Основными утилитами пакета STEP 7 Simatic Manager, которые доступны из папки SIMATIC - STEP 7, являются:

- SIMATIC Manager;
- LAD, STL, FDB - Programming S7;
- Memory Card Parameter Assignment;
- NetPro - Configuring Networks;
- PID Control Parameter Assignment;
- S7 SCL - Programming S7 Blocks;
- S7-GRAPH - Programming Sequential Control System;
- S7-PDIAG - Configuring Process Diagnostic;
- S7-PLCSIM Simulating Modules;
- Setting the PG-PC Interface;

SIMATIC Manager - это графический интерфейс для редактирования объектов S7 (проектов, файлов пользовательских программ, блоков, оборудования станций и инструментов). Основное окно утилиты показано на рисунке 1.1.

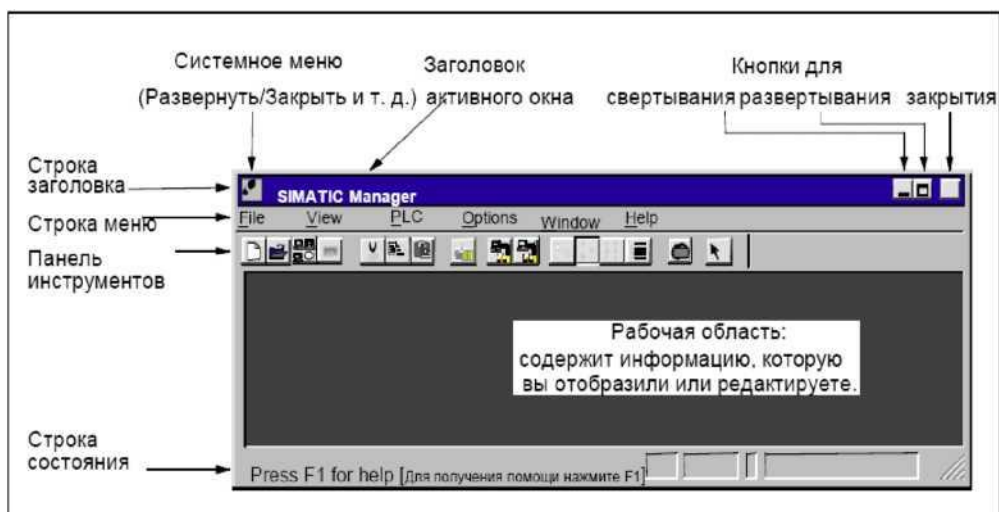
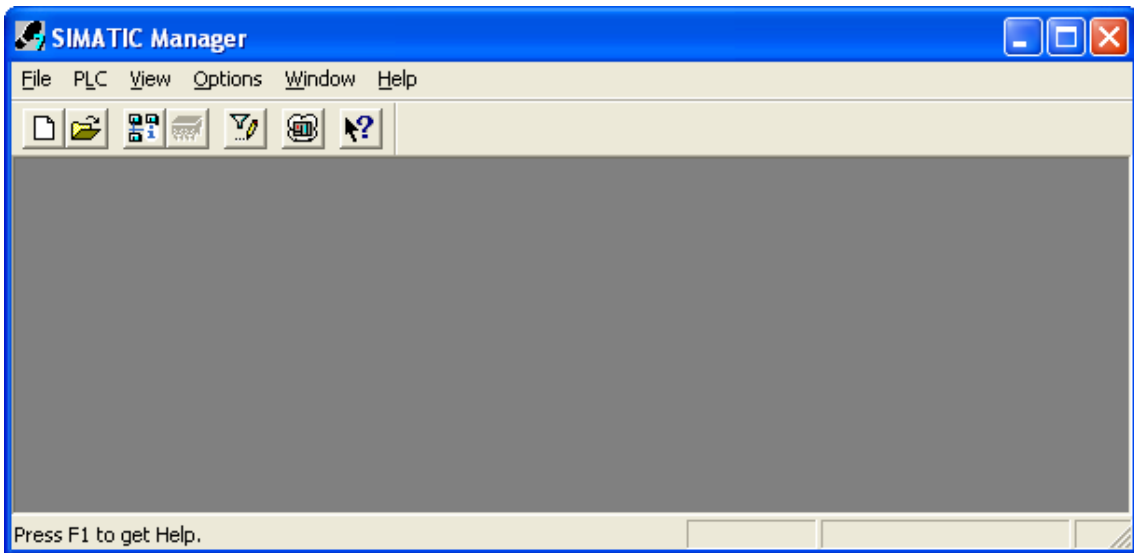


Рис. 1.1. - Меню и панель инструментов SIMATIC Manager

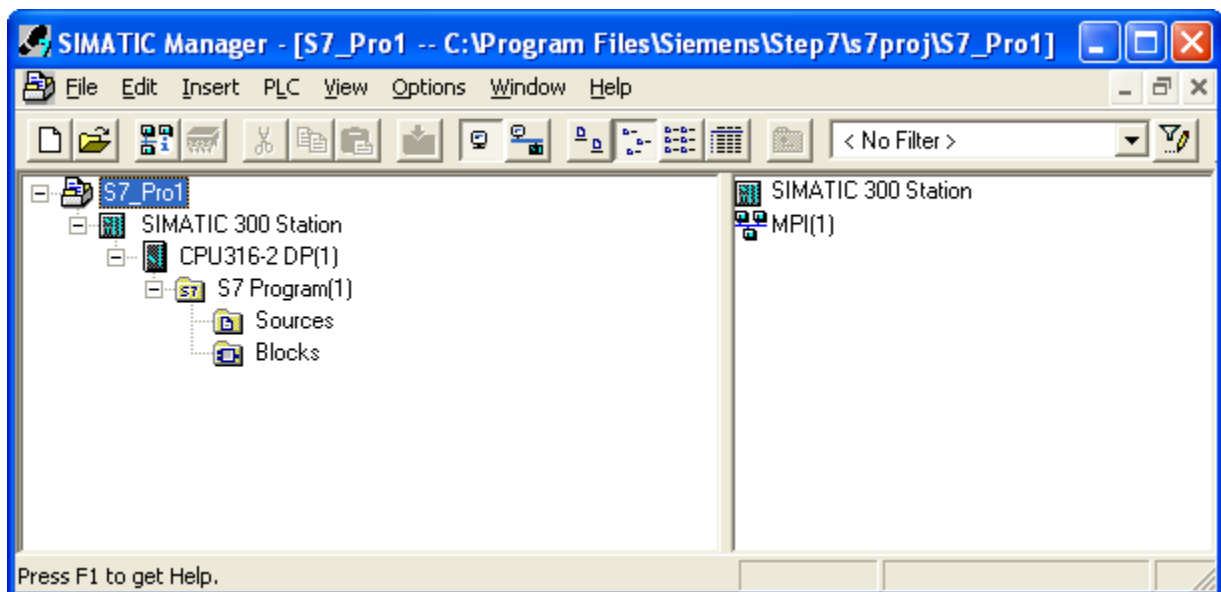


Рис. 1.2. - Структура проекта в SIMATIC Manager

1.1. 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 STEP7. По умолчанию запускается мастер STEP7 (STEP7 Wizard) (рис. 1.3), который оказывает помощь при создании проекта STEP7. 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]».

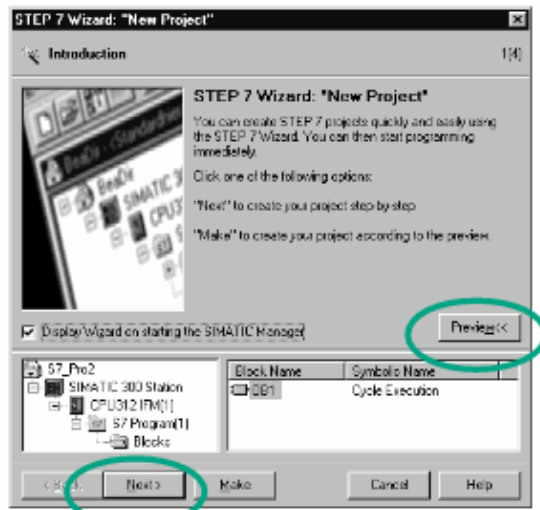


Рис. 1.3. Окно мастера STEP7 (STEP7 Wizard)

Выберите CPU(рис. 1.4) 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.

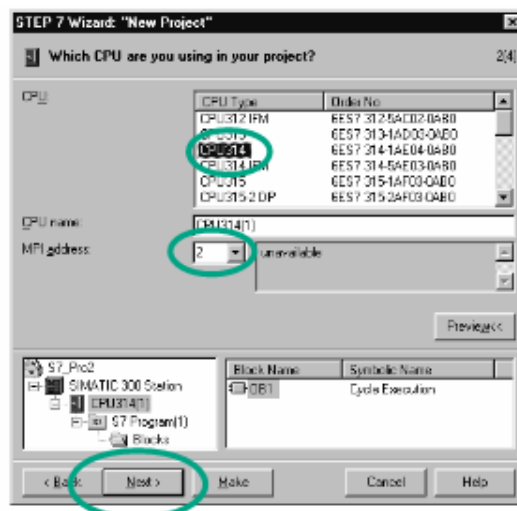


Рис. 1.4. Окно выбора процессора в STEP7 Wizard

Выберите организационный блок OB1 (рис. 1.5). 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. 1.6), 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.

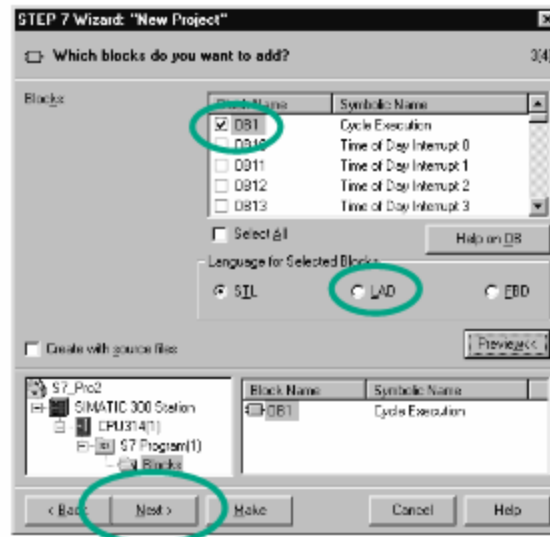


Рис. 1.5. Окно выбора языка программирования

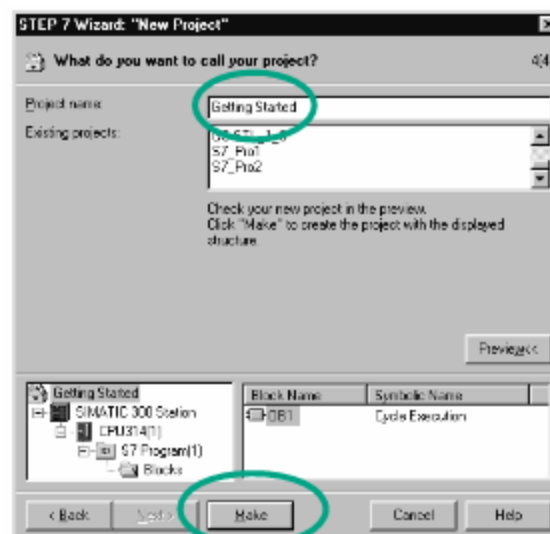


Рис. 1.6. Окно выбора имени проекта

Только что созданный проект отображается с выбранной S7 станцией и CPU(рис. 1.7). Щелкая на значке «+» или «-» можно открыть или закрыть папку.

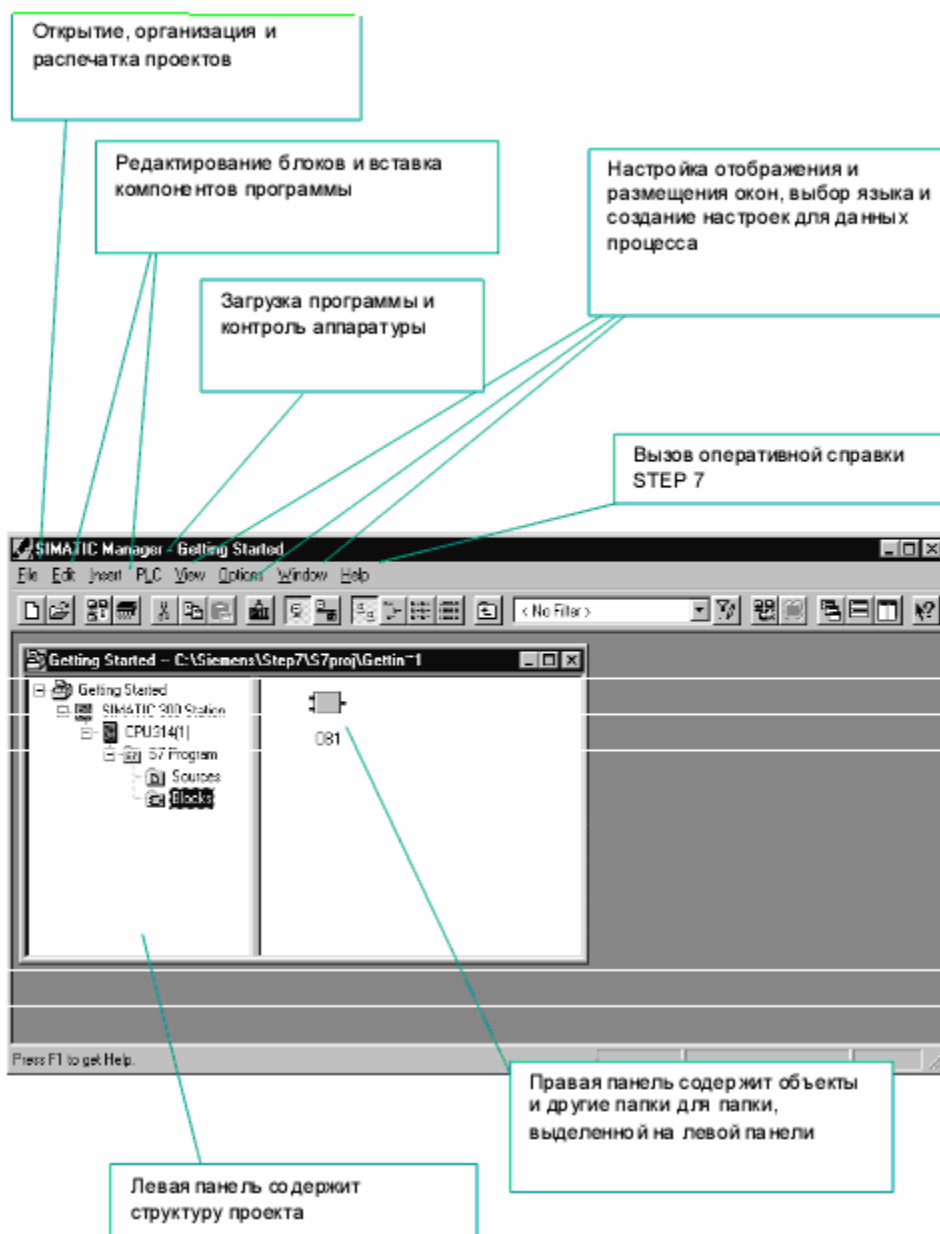


Рис. 1.7. Окно SIMATIC Manager

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. 1.18) 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. 1.8. 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).

STEP7 проверяет, не нарушены ли правила для слотов (CPU S7-300, например, может быть установлен только в слоте 2).

0 - 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-SaveandCom- pile».

1.2. 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 дважды щелкнуть левой кнопкой мыши по знаку CPU..., затем по появившемуся значку S7 Program наконец, по появившемуся значку Blocks. Появляется в правой части пиктограмма OB1. Дважды щелкните на пиктограмме OB1. Открывается окно программирования (рис. 1.9).

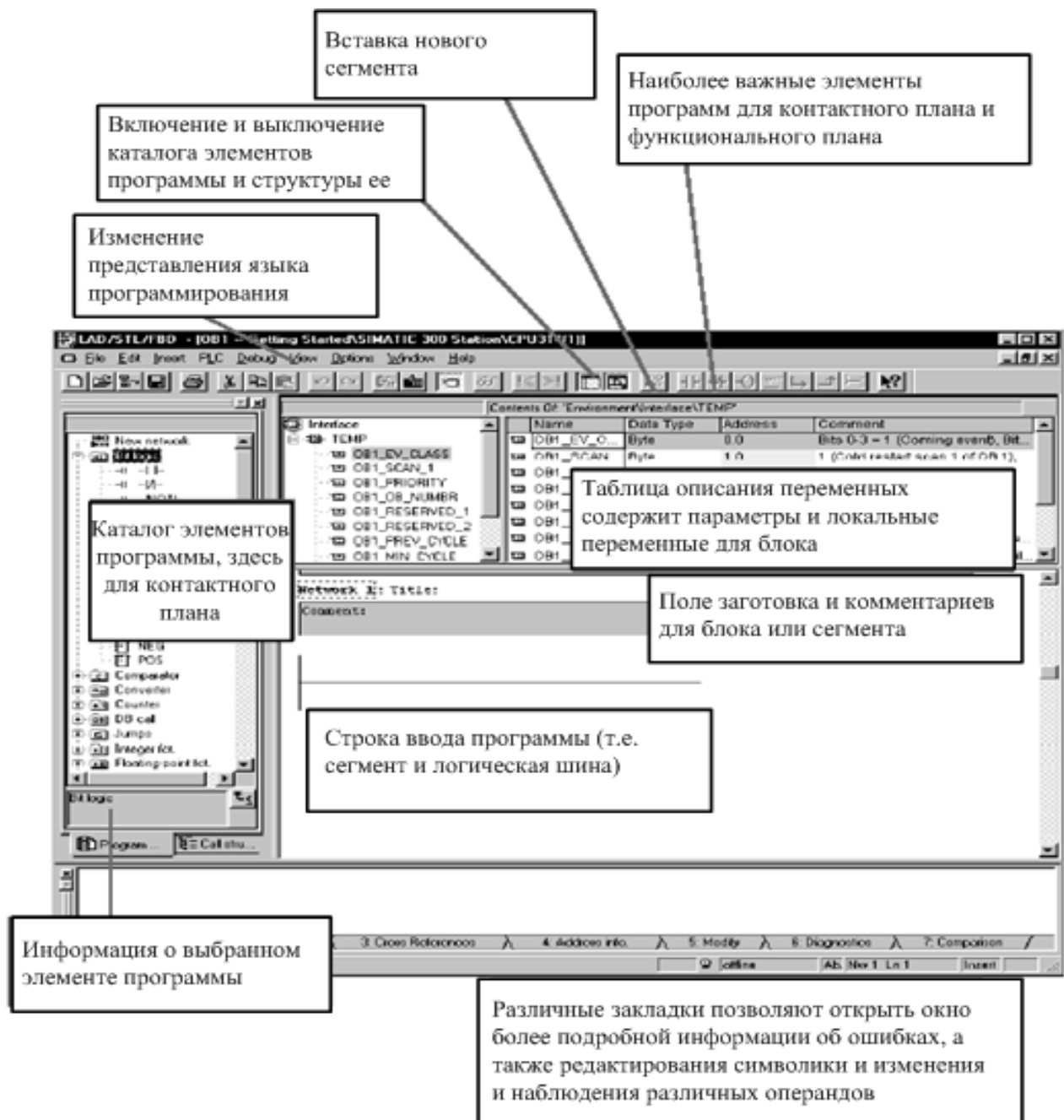



Рис. 1.9. Окно программирования

В меню «View [Вид]» можно выбрать язык программирования LAD/STL/FBD.

Ниже рассматривается программирование на языке LAD.

При программировании лестничными диаграммами (LAD) в виде релейноконтактной 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:

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

1) Поставьте в токовую цепь (сегмент) реле (катушку) (рис. 1.10). Для этого нажмите на кнопку

 - Coil [Катушка]

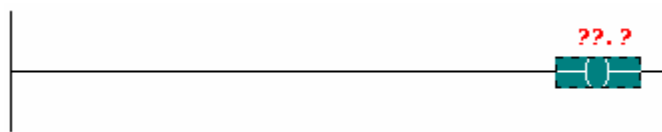


Рис.1.10

2) Щёлкните по тому месту (непосредственно по линии) куда, хотите вставить замыкающий (нормально открытый) контакт (рис. 1.11).



Рис.1.11

3) Вставьте замыкающий контакт, нажав на кнопку


 – Normally Open Contact [Нормально открытый контакт] (рис. 1.1.2)



Рис 1.12

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. 1.13).



Рис.1.13


- 5) Вставьте нормально открытый контакт и завершите параллельную ветвь, нажав на кнопку  - Close Branch [Завершить ветвление] (рис. 1.14).




Рис.1.14

- 6) Выделите место, куда необходимо поставить нормально замкнутый (размыкающий) контакт (рис. 1.15).



Рис.1.15

- 7) Вставьте размыкающий контакт, нажав на кнопку  - Normally Closed Contact [Нормально закрытый контакт] (рис. 1.16).

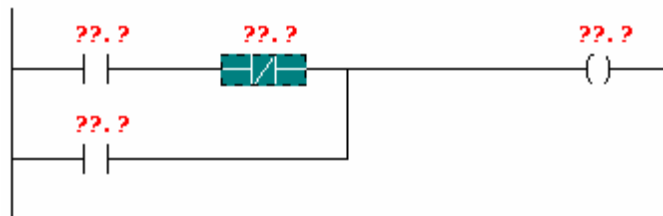


Рис.1.16

- 7) Теперь напишите адреса контактов. Для этого щёлкните по знакам '??.'?' над контактом и введите адрес (ввод адреса завершается нажатием клавиши ENTER) (рис. 1.17).

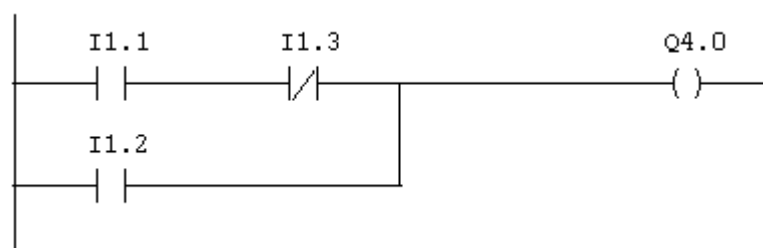


Рис.1.17

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-Тили выбрав Options-SymbolTable(выбрав в главном меню редактора программы Optionsи затем выбрав из выпадающего меню Symbol Table). Появится окно редактора символов (рис. 1.18).

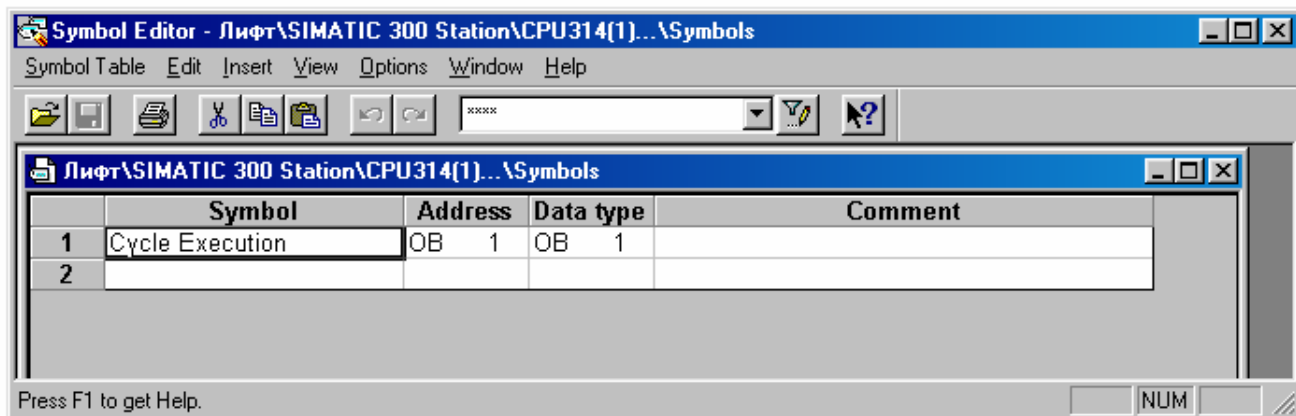


Рис. 1.18. Окно редактора символов

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 (рис. 1.19)

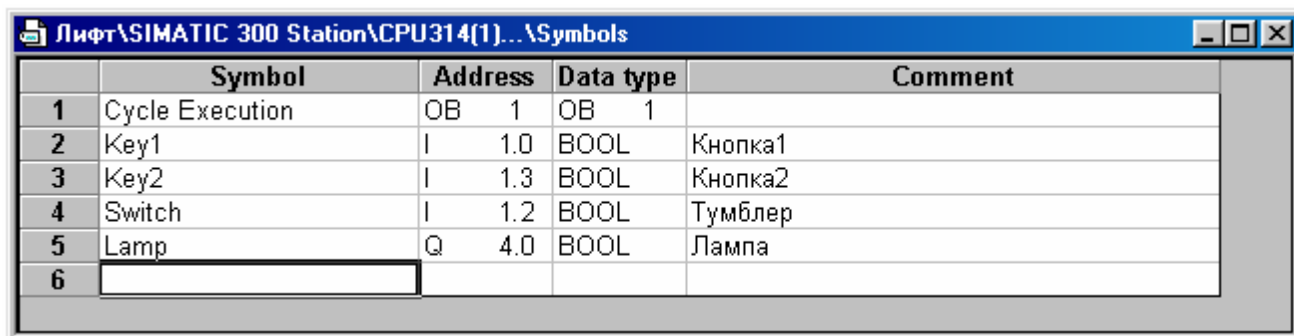


Рис. 1.19. Окно редактора символов

Сохраните таблицу символов (SymbolTablesSave). Закройте окно редактора символов. Теперь программа будет выглядеть следующим образом (рис. 1.20).

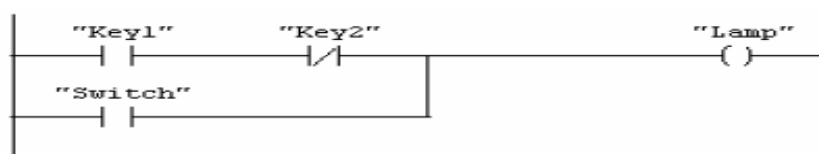





Рис.1.20

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 зажётся 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 →  SR logic (SR триггер) - щёлкнуть по этому значку быстро два раза;
- 5) над функцией SR триггера (рис. 1.21) вместо символов "???" вставьте символическое имя выхода;

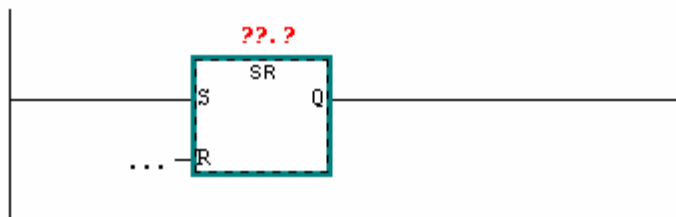


Рис.1.21

- 6) на входную линию функции S поставьте нормально открытый контакт Key1, а на место входа R - Key2 (рис. 1.22).

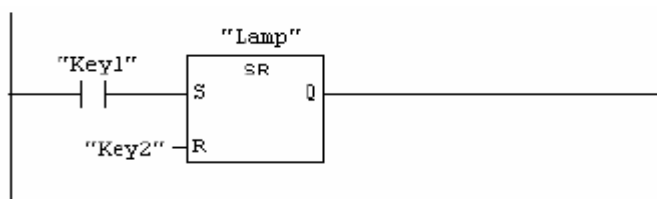


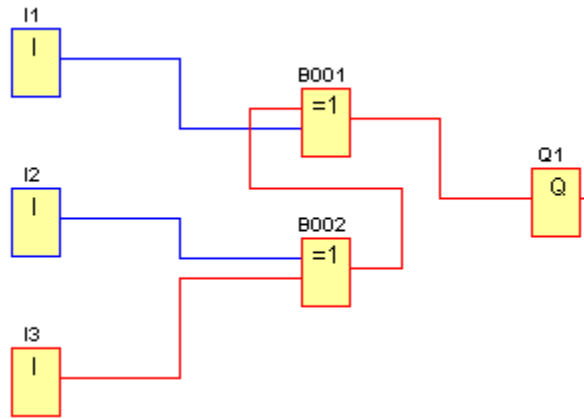
Рис.1.22

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

1. Порядок выполнения работы:

1.1. Загрузить программу SIMATIC Manager, создать проект для CPU315-2 DP с одним блоком OB1 и ознакомиться с основными элементами окна программы.

1.2. Используя средства программы ввести следующую программу:

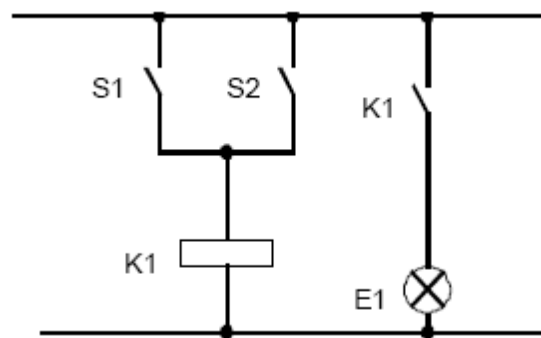


Нажав кнопку симуляция, проверить работу программы и составить таблицу истинности для схемы.

Заменить элементы B001 и B002 на элементы И.

Нажав кнопку симуляция, проверить работу программы и составить таблицу истинности для схемы.

1.3.Используя средства программы составить программу на бесконтактных элементах, реализующие следующие алгоритмы работы:



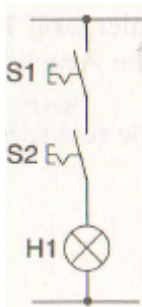
1.3.1 Лампа E1 загорается при замыкании любого из выключателей S1, S2

1.4 Используя средства программы составить программу, реализующую следующую функцию:

$$Y = [(a+d) * b + z] * c$$

Проверить работу программы в режиме эмуляции

1.5 Функция “И”



Список назначений		
Символьное обозначение	Абсолютная адресация	Комментарий
S1	I 124.0	Коммутационный контакт
S2	I 124.1	Коммутационный контакт
H1	Q 124.0	Индикатор

Рис.1.23 Принципиальная схема

На рис. 1.23 показано, принципиальная электрическая схема логического элемента “И”. Н1 световой индикатор включается только тогда, когда все переключатели замкнуты. При открытии выключатель в цепь разрывается и Н1 индикатор не горит.

2. Отчет должен содержать

- 2.1. Название работы
- 2.2. Цель работы
- 2.3. Перечень оборудования.
- 2.4. Элементы окна программы и их назначение.
- 2.5. Схемы и программы, реализующие их функцию.
- 2.6. Таблицы истинности для пункта 1.2
- 2.7. Вывод по работе.

3. Контрольные вопросы и задания.

- 3.1. Перечислите основные параметры ПЛК S7-300
- 3.2. Какова область применения ПЛК S7-300
- 3.3. Какие основные элементы окна программы вы знаете, каково их назначение.

ЛАБОРАТОРНАЯ РАБОТА №2

СРЕДСТВА ДИАГНОСТИКИ КОНТРОЛЛЕРА S7-300

Задание. Диагностика контроллера S7-300.

1. Кнопки и лампы соединены с блоком входа/ выхода контроллера в произвольном порядке (не последовательно!).
2. Для поиска адресов элементов входа/ выхода использовать инструмент диагностики контроллера: STATUS CHART.
3. Нажимая на любую кнопку, загорается лампа с соответствующим номером, а также лампа h6.
4. При нажатии одновременно 2-х и более кнопок загорается лишь лампа h6.

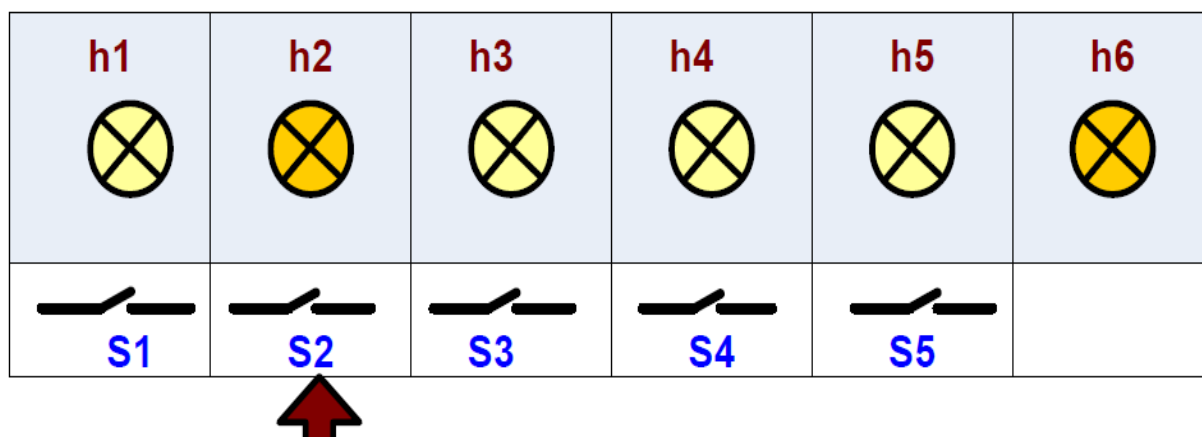


Рис. 2.1. Схема устройства.

Источники информации: S1 - S5 кнопки пульта

Исполнительные устройства: h1 – h6 лампы пульта

Ход работы:

1. Поиск адресов.

1. Заполнить таблицу STATUS CHART: для этого набрать первый адрес входа I0.0. Далее нажимать ENTER – таблица будет заполнена автоматически. Те же действия для выходов.

2. Активизировать режим ON-LINE:

3. Нажимать поочередно кнопки и контролировать изменение их состояния в таблице STATUS CHART. Записать адреса кнопок в тетрадь.

2. Программа.

Закончить аналитическое описание и составить программу управления.

H1 =1, если S1 and no S2 and no S3 and no S4 and no S5

H2 =1, если ...

H3 =1, если ...

H4 =1, если ...

H5 =1, если ...

H6 =1, если S1 or S2 or S3 or S4 or S5

3. Трансляция программы в контроллер.

4. Оформить отчет.

ЛАБОРАТОРНАЯ РАБОТА №3

ПРОГРАММИРОВАНИЕ РАБОТЫ ШИФРАТОРА.

Задание. Десятично-двоичный шифратор.

Устройство, с помощью которого можно автоматически преобразовывать числа из одного числового кода в двоичный код, называется шифратором (кодером).

Условием работы такого шифратора является то, что одновременно на входе устройства может быть активным только один вход. Это означает, что в таблице истинности не может быть комбинаций входных величин.

Таблица 1. Таблица истинности десятично-двоичного шифратора.

0	1	2	3	4	5	6	7	8	9	Десятичное число / Двоичное число			
Входы										Выходы			
	S1	S2	S3	S4	S5	S6	S7	S8		H4	H3	H2	H1
	1												1
		1										1	0
			1									1	1
				1							1	0	0
					1						1	0	1
						1					1	1	0
							1				1	1	1
								1		1	0	0	0
									1	1	0	0	1

Рис 3.1

* пустые клетки для выходных переменных соответствуют нулевому состоянию.

Источники информации: кнопки S1 - S8 соответствуют десятичные числа от 1 до 8. Контроллер S7 – 300 имеет лишь 8 физических входов.

Исполнительные устройства: лампы H1, H2, H3, H4 соответствуют разрядам двоичного числа: 0, 1, 2, 3.

Ход работы:

1. Найдите функцию в СНДФ для каждого выхода, используя таблицу истинности дешифратора.

$$H1 = S1 + S3 + S5 + S7 + (S9)$$

$$H2 = \dots$$

$$H3 = \dots$$

$$H4 = \dots$$

2. Составьте программу в FBD.

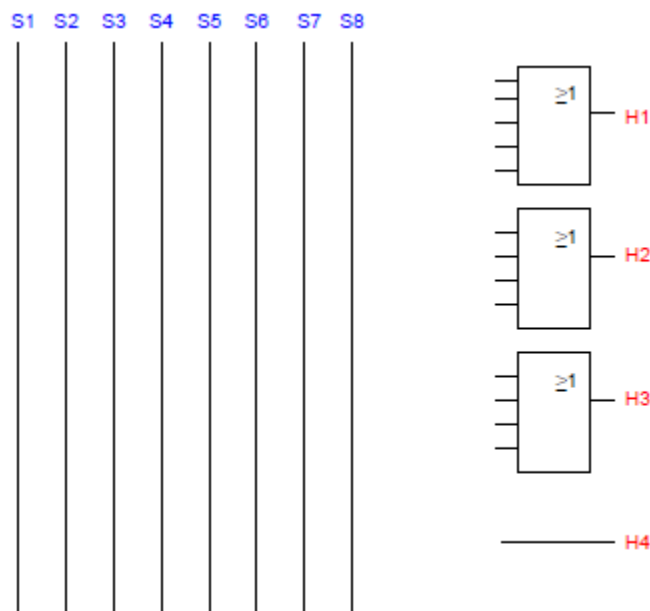


Рис. 3.2 Закончить программу в FBD

3. Составить таблицу символов

4. Составить программу на FBD, LAD, STL

5. Загрузить программы управления.

-Создать проект управления.

-Проверить синтаксис программы. Для этого компилировать проект

Если ошибок нет, то транслировать проект в контроллер. Микропереключатель режима работы должен быть включен режим ТЕРМ или RUN.

5. Сетевые настройки системы.

Если загрузка проекта не удалась, то необходимо проверить наличия сетевого соединения в окне COMMUNICATION.

В случае отсутствия сети проверить:

1. Наличие питания на самом контроллере
 2. Программные установки на соответствие реальному подключению кабеля PPI (COM1, COM2 или USB). Убедиться в том, что в программе выбран соответствующий порт подключения и правильно выбрана коммуникационная карта. Также необходимо снять птичку с опции *Multiple master network*, если в сети нет нескольких контроллеров.
 3. Скорость обмена данными на 9,6 кбит/сек (микрорелеключатели на кабеле PPI).
- 6. Оформить отчет.**

ЛАБОРАТОРНАЯ РАБОТА №4

ОДНОБИТОВЫЕ ОПЕРАЦИИ СТЕП 7.

Задание. Указатель свободных мест на стоянке.

1. № лампы табло указывает на количество свободных (пустых) мест. Наличие автомобиля на площадке определяется датчиками В.
2. Для контроля адресов элементов входа/ выхода использовать инструмент диагностики контроллера: STATUS CHART.

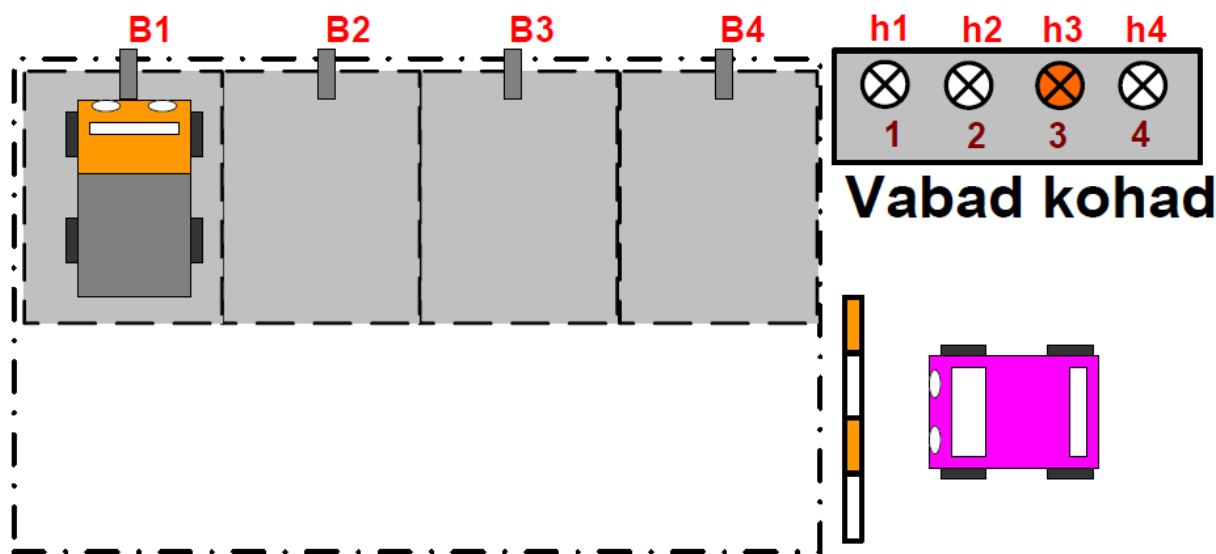


Рис. 4. Схема стоянки

Источники информации: В1 – В4 датчики, реагирующие на наличие автомобиля.

Исполнительные устройства: h1 – h4 лампы табло.

Ход работы.

1. **Логическая функция** представляет собой одноканальный автомат (выход незамедлительно реагирует на изменение входных величин). Логическую функцию можно составить, используя СДФ. Если количество входов 4, то возможных комбинаций 16.

2. Таблица истинности 2. Закончить.

№	B1	B2	B3	B4	H1	H2	H3	H4
1	0	0	0	0				
2				1				
3			1					
4			1	1				
5		1						
6		1		1				
7		1	1					
8		1	1	1	1			
9	1							
10	1			1				
11	1		1					
12	1		1	1	1			
13	1	1						
14	1	1		1	1			
15	1	1	1		1			
16	1	1	1	1				

3. Закончить СНДФ.

$$H1 = \text{no}B1 \cdot B2 \cdot B3 \cdot B4 + B1 \cdot \text{no}B2 \cdot B3 \cdot B4 + B1 \cdot B2 \cdot \text{no}B3 \cdot B4 + B1 \cdot B2 \cdot B3 \cdot \text{no}B4$$

H2=

H3=

H4=

4. Написать программу и компилировать ее.

5. Трансляция программы в контроллер.

6. Оформить отчет.

ЛАБОРАТОРНАЯ РАБОТА №5

ОПЕРАЦИИ СДВИГА И РОТАЦИИ.

Задание: Указатель свободных мест на стоянке.

1. № лампы табло указывает на количество свободных (пустых) мест. Наличие автомобиля на площадке определяется датчиками В.

2. Для контроля адресов элементов входа/ выхода использовать инструмент диагностики контроллера: STATUS CHART

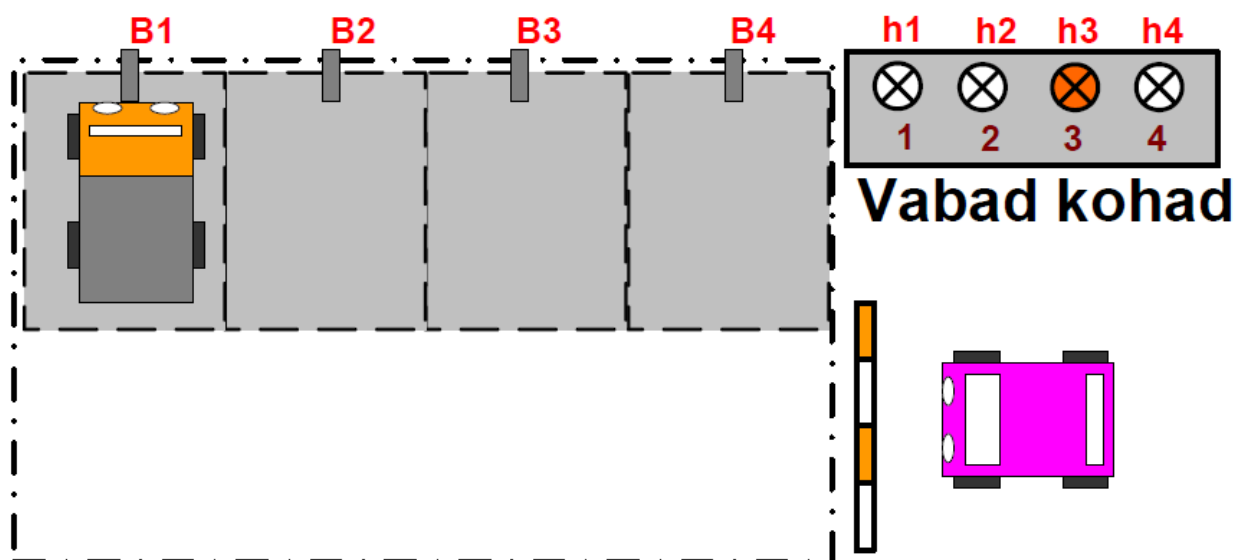


Рис. 5. Схема стоянки

Источники информации: B1 – B4 датчики, реагирующие на наличие автомобиля.

Исполнительные устройства: h1 – h4 лампы табло.

Ход работы:

1. Подключить лампы в соответствии с адресами, указанными в таблице.

Принцип работы: управлять лампами возможно с использованием многобитовых операций загрузки и сдвига битов операнда QB0.

Таблица 3. Загрузка и пошаговый сдвиг.

Адреса выхода	Q0.3	Q0.2	Q0.1	Q0.0
Лампы	h1	h2	h3	h4
Загрузить «1» → QB0, SM0.1→MOV	0	0	0	1
Сдвиг влево (прибытие авто), SHL, ROL +1	0	0	1	0
+2	0	1	0	0
+3	1	0	0	0
Сдвиг вправо (выезд авто), SHR, ROR -1	0	1	0	0

2. Первичную загрузку «1» в первый бит QB0 произвести при включении режима RUN с помощью стартового бита SM0.1, который активен только первый кадр сканирования в режиме RUN. В результате этого активным станет бит QB0.0, к которому подключена лампа h4.

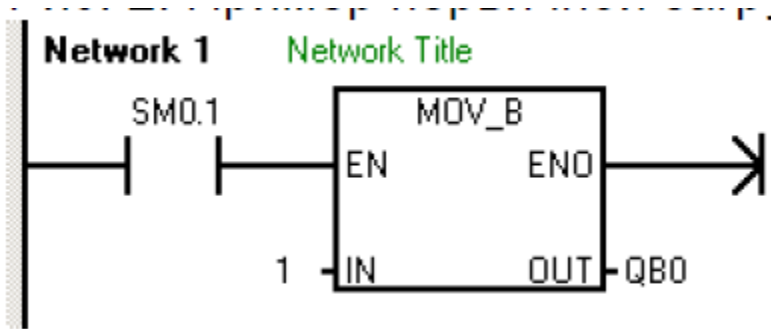


Рис.5.1. Пример первичной загрузки

3. При прибытии авто (положительный фронт датчиков В1 или В2 или В3 или В4) производится сдвиг (ротация) битов операнда QV0 **влево**. Соответственно начнет гореть следующая лампа. При этом горит только одна лампа, т.к. активен только 1 бит операнда QV0.

При отъезде авто (негативный фронт датчиков В1 или В2 или В3 или В4) производится сдвиг (ротация) битов операнда QV0 **вправо**.

Программа реагирует на изменение значений датчиков (импульсная логика):

положительный фронт: |P| ; негативный фронт: |N| .

- появление автомобиля: $V_n \rightarrow |P| \rightarrow \text{ROL}$.

- убытие автомобиля: $V_n \rightarrow |N| \rightarrow \text{ROR}$.

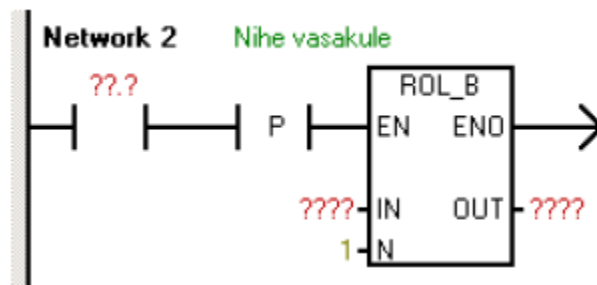


Рис. 5.2. Пример операции сдвига

4. Написать программу и компилировать ее.

5. Трансляция программы в контроллер.

6. Оформить отчет.

ЛАБОРАТОРНАЯ РАБОТА №6

ОПЕРАЦИИ С ТАЙМЕРАМИ И С СЧЕТЧИКАМИ

Цель работы: Освоить методы составления и редактирования программ с использованием временных задержек (таймеров) и счетчиков.

Основные понятия

Таймеры (Timers) – это ячейки памяти, используемые для реализации интервалов ожидания и мониторинга.

Таймеры позволяют программно реализовать последовательности синхронизации, такие как интервалы ожидания и наблюдения, измерение интервалов или генерирование импульсов.

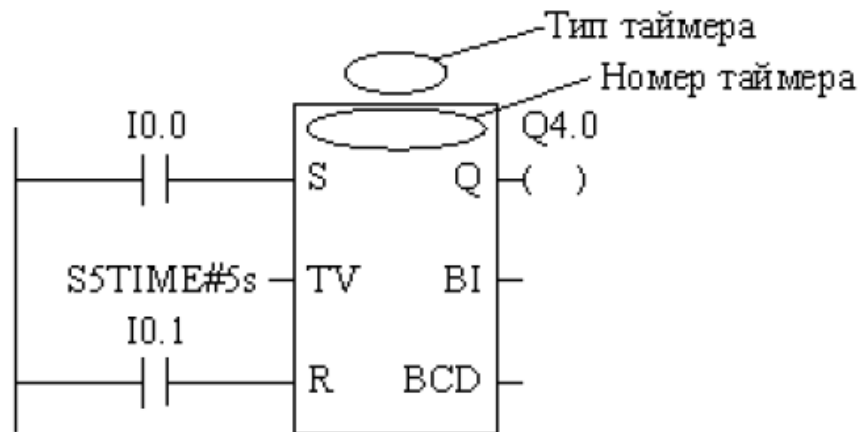
Существуют следующие *типы таймеров*:

- 1) Импульсные таймеры (Pulsetimers);
- 2) Импульсные таймеры с памятью (Extendedpulsetimers);
- 3) Таймеры задержки включения (On-delaytimers);
- 4) Таймеры задержки включения с запоминанием (Retentiveon-delaytimers);
- 5) Таймеры задержки выключения (Off-delaytimers).

Таблица 4

<i>Таймер</i>	<i>Описание</i>
S_PULSE Таймер «Импульс»	Максимальное время в течение которого выходной сигнал остается равным 1, совпадает с запрограммированным временем T. Выход сбрасывается раньше, если входной сигнал меняет состояние на 0.
S_PEXT Таймер «Импульс с памятью»	Выходной сигнал остается равным 1 в течение запрограммированного времени независимо от того, как долго остается равным 1 входной сигнал.
S_ODT Таймер «Задержка включения»	Выходной сигнал устанавливается в 1 только по истечении запрограммированного времени, при этом входной сигнал все еще должен быть равен 1.
S_ODTS Таймер «Задержка включения с памятью»	Выходной сигнал устанавливается в 1 только по истечении запрограммированного времени независимо от того, как долго остается равным 1 входной сигнал.
S_OFFDT Таймер «Задержка выключения»	Выходной сигнал устанавливается в 1, когда устанавливается в 1 входной сигнал, и остается равным 1, пока таймер работает. Отсчет времени начинается, когда входной сигнал меняется с 1 на 0.

Далее рассмотрим только *S_PULSE* и *S_PEXT*, которые используются при выполнении лабораторной работы.



S – вход запуска, TV – уставка (заданное время),
 R – вход сброса, Q – выход (состояние таймера),
 BI – оставшееся время в двоичном формате,
 BCD – оставшееся время в двоично-десятичном формате

Рис.6 Таймер формирователь импульса

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. (6.1 и 6.2) 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.

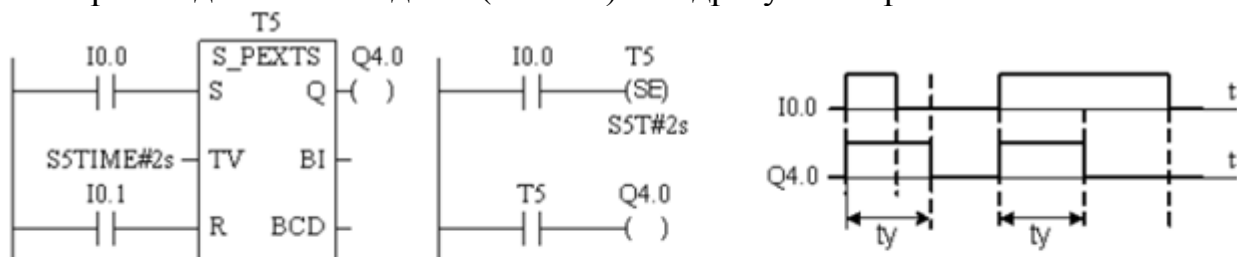


Рис. 6.1 Таймер формирователь удлиненного импульса

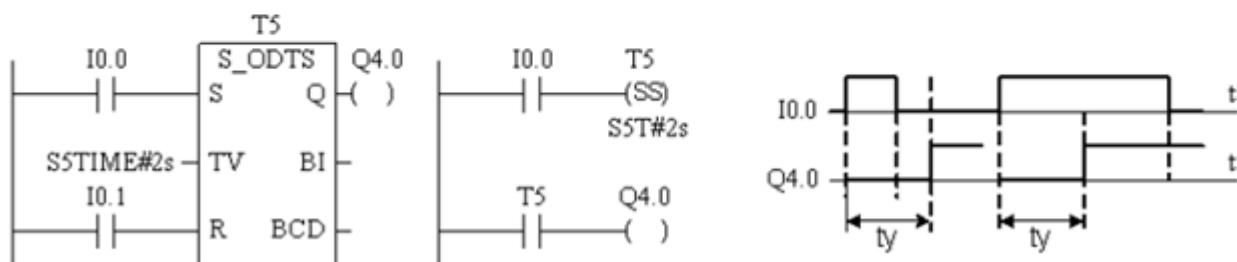


Рис. 6.2 Таймер формирователь задержки включения с запоминанием

Загрузить значение времени можно с использованием следующего синтаксиса:

S5T#aH_bM_cS_dMS,

где: a = часы,
b = минуты,
c = секунды,
d = миллисекунды.

Максимальное время, которое можно ввести, составляет 9 990 секунд или 2H_46M_30S.

S5TIME#4S = 4 секунды

s5t#2h_15m = 2 часа и 15 минут

S5T#1H_12M_18S = 1 час, 12 минут и 18 секунд

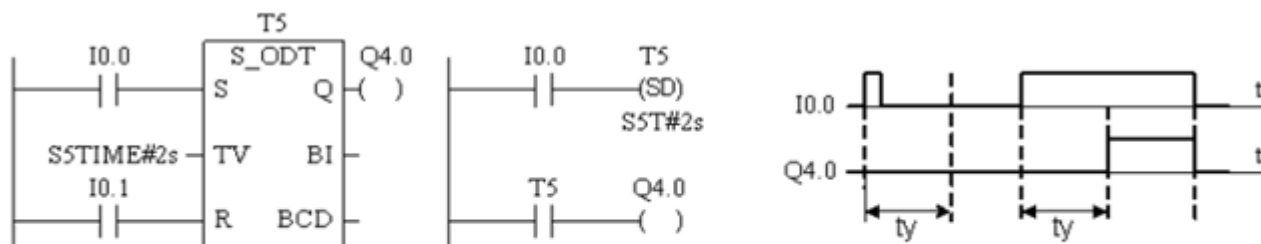


Рис. 6.3 Таймер формирователь задержки включения

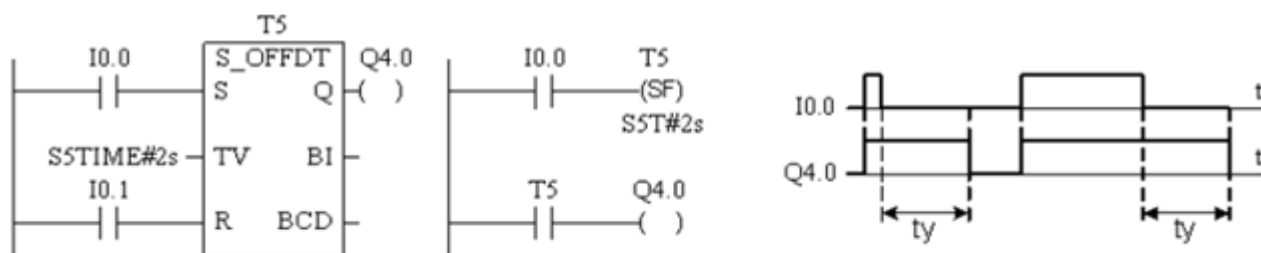


Рис. 6.4 Таймер формирователь задержки выключения

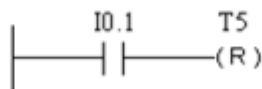


Рис. 6.5 Сброс таймера с адресом T5

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 находится 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. 6.6. представлены блоки всех 3-х видов счетчиков.

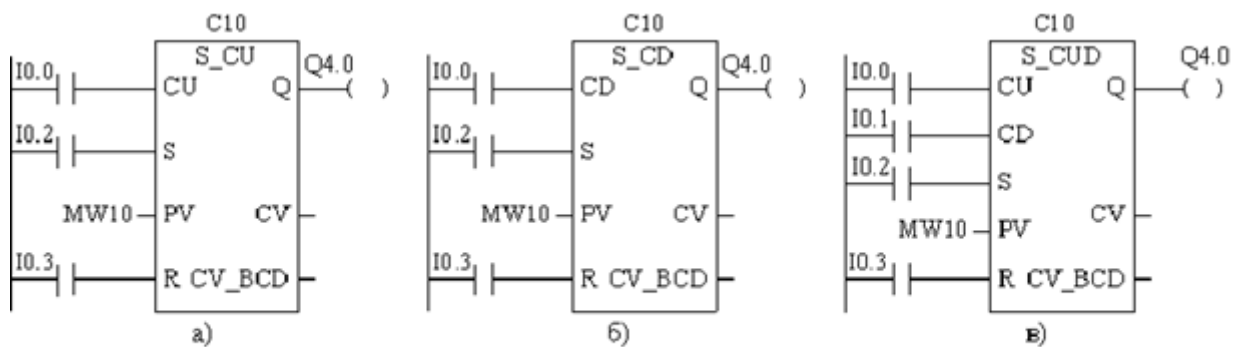


Рис. 6.6 Счетчики: а) на сложение, б) на вычитание, в) реверсивный

На блоках счетчиков рис. 6.6. 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».

Задание: Запрограммировать работу двух светофоров пешеходного перехода: светофор 1- для автомобилей, а светофор 2 - для пешеходов.

Элементы управления.

1. Входы. S1 – кнопки запуска; S2 – ночной режим.

2. Выходы.

Лампы светофора: Н1 – Н5.

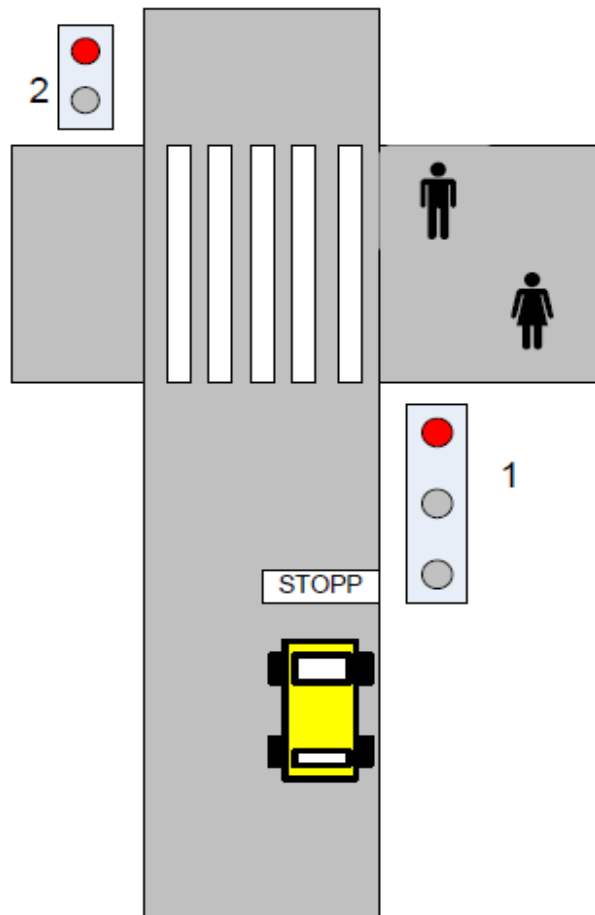


Рис.6.7 Схема устройства

Режим светофора запускается однократно кнопкой S1.

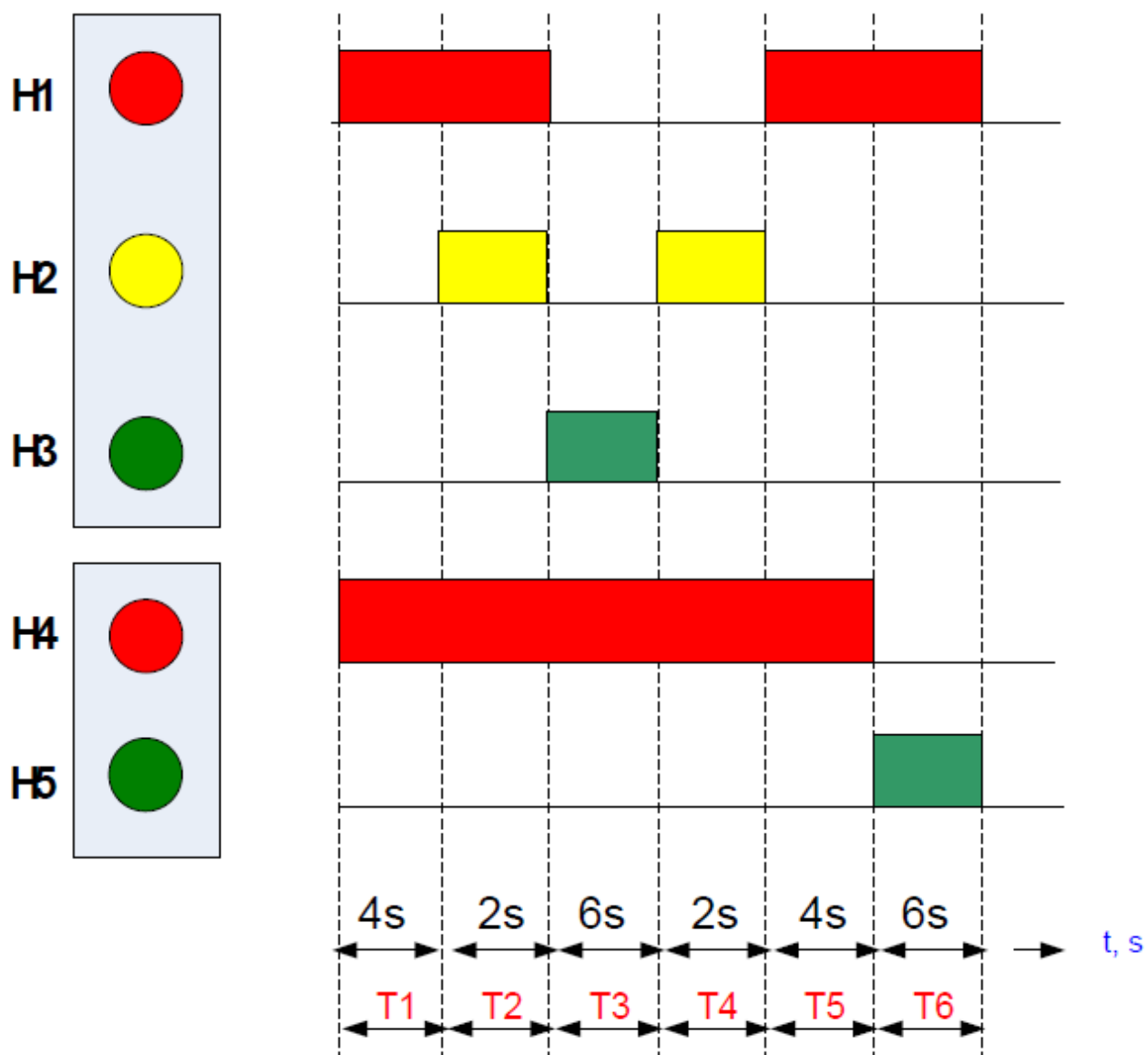


Рис.6.8. Диаграмма работы светофора

Задание имеет несколько решений. Одно из них является поочередное переключение меркеров M1 – M6 с использованием функций таймеров.

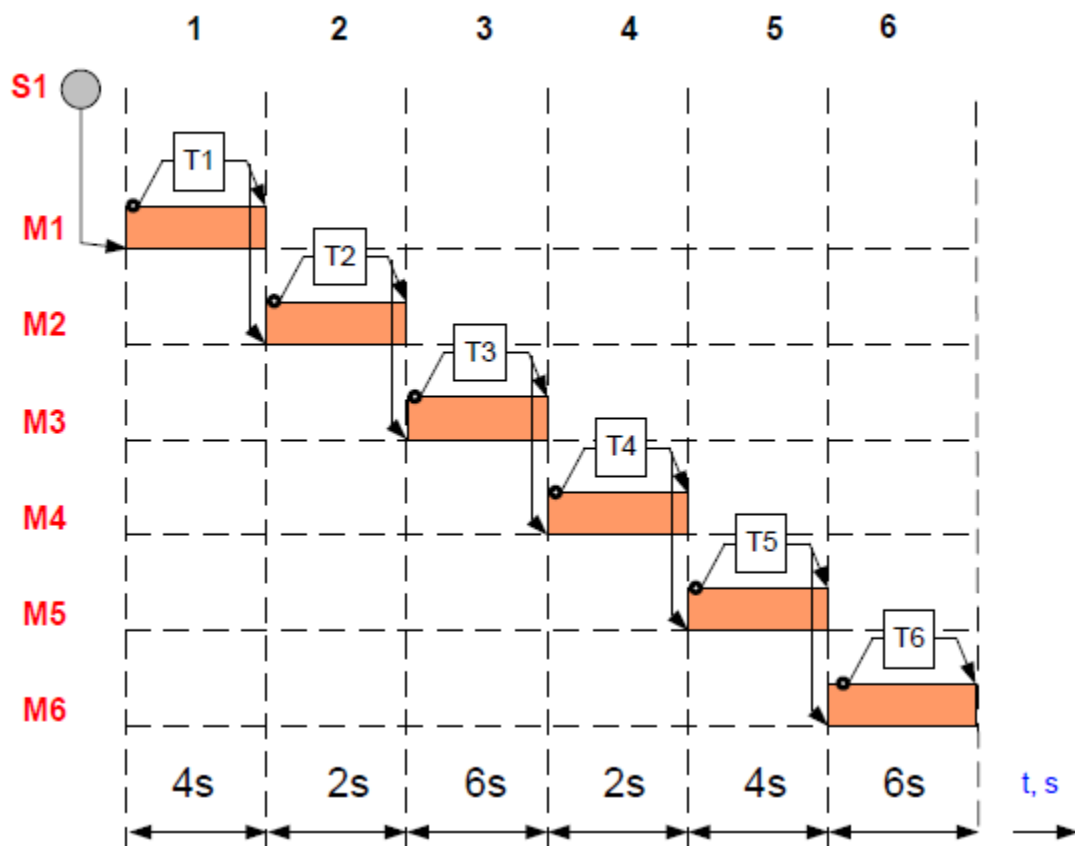


Рис. 6.9. Диаграмма переключения меркеров

Закончить аналитическое описание (на базе TON).

M1

Set M1: S1 and no M1 and no M2 and no M3 and no M4 and no M5

Reset M1: T1

M2

Set M2: T1

Reset M2: T2

...

Лампы

H1 = M1 or M2 or ...

H2 = M2 or ...

Написать программу и загрузить в контроллер.

Оформить отчет

ЛИТЕРАТУРА

1. Бергер Ганс: Автоматизация с помощью программ STEP7 LAD и FBD. Издание 2-е переработанное, – Siemens AG. 2001. – 605 с.
2. Бергер Ганс: Автоматизация посредством STEP 7 с использованием STL и SCL и программируемых контроллеров SIMATIC S7-300/400. – Siemens AG. 2001. – 776 с
3. Акматбеков Р.А. Руководство для программирования управляющих станций SIMATIC S7-300/400
4. www.siemens.com/automation
5. <http://www.automation-drives.ru/>

