

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Емельянов Сергей Геннадьевич

Должность: ректор

Дата подписания: 17.02.2018 20:56:40

Уникальный программный ключ:

9ba7d3e34c012eba476ffd2d064cf2781953be730df2374d16f3c0ce536f0fc6

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра вычислительной техники

УТВЕРЖДАЮ

Проректор по учебной работе

_____ О.Г. Локтионова

« ____ » _____ 2017 г.

СОЗДАНИЕ ПЕРВОГО ПРОЕКТА ДЛЯ КОНТРОЛЛЕРА SIEMENS S7-1200

Методические указания по выполнению лабораторной работы
по дисциплине «Математическое и алгоритмическое обеспечение
автоматизированных технологических процессов»
для аспирантов специальности 09.06.01 «Информатика
и вычислительная техника»

Курск 2017

УДК 681.5, 004.5

Составители: М.В. Бобырь, В.Г. Рубанов, С.А. Кулабухов

Рецензент

Доктор технических наук, профессор *И.В. Зотов*

Создание первого проекта для контроллера SIEMENS S7-1200: методические указания по выполнению лабораторной работы по дисциплине «Математическое и алгоритмическое обеспечение автоматизированных технологических процессов» / Юго-Зап. гос. ун-т; сост.: М.В. Бобырь, В.Г. Рубанов, С.А. Кулабухов. – Курск, 2017. – 19 с.: ил. 16, табл. 2. – Библиогр.: с.19.

Представлено описание программируемого логического контроллера Simatic S7-1200 от фирмы Siemens. Представлены примеры его настройки и программирования. Описаны способы взаимодействия с различными элементами автоматики.

Методические указания соответствуют требованиям программы дисциплины «Математическое и алгоритмическое обеспечение автоматизированных технологических процессов».

Предназначены для аспирантов специальности 09.06.01 «Информатика и вычислительная техника» дневной и заочной форм обучения.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.

Усл.печ. л. 1,1. Уч.-изд. л. 1,0. Тираж экз. Заказ. Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

СОЗДАНИЕ ПЕРВОГО ПРОЕКТА ДЛЯ КОНТРОЛЛЕРА SIEMENS S7-1200

1. Цель работы

Изучить характеристики программируемого логического контроллера (ПЛК) S7-1200 от компании SIEMENS. Понять теоретические основы методов его программирования в среде разработки STEP 7. Изучить механизмы создания нового проекта в среде разработки STEP 7.

2. Общие сведения о ПЛК S7-1200

Программируемый логический контроллер (ПЛК) S7-1200 обеспечивает гибкость и предоставляет достаточную мощность для управления широким кругом устройств, поддерживающих ваши потребности в автоматизации. Компактная конструкция, гибкая конфигурация и мощный набор команд, делают S7-1200 прекрасным решением для управления широким спектром приложений. CPU объединяет в компактном корпусе микропроцессор, встроенный блок питания, входные и выходные цепи, образуя мощный ПЛК. После загрузки вашей программы CPU содержит логику, необходимую для контроля и управления устройствами в вашем приложении. CPU контролирует входы и изменяет выходы в соответствии с логикой вашей пользовательской программы, которая может включать булевы логические операции, счет, отсчет времени, сложные математические операции и связь с другими интеллектуальными устройствами.

Ряд функций обеспечения безопасности помогают защитить доступ как к CPU, так и к управляющей программе:

– Каждый CPU обеспечивает защиту паролем, которая позволяет вам формировать

доступ к CPU в соответствии с вашими потребностями.

– Вы можете использовать "защиту ноу-хау", чтобы скрыть код внутри конкретного блока.

CPU снабжен портом PROFINET для обмена данными через сеть PROFINET. Для обмена данными через сети RS485 или RS232 в вашем распоряжении имеются коммуникационные модули.

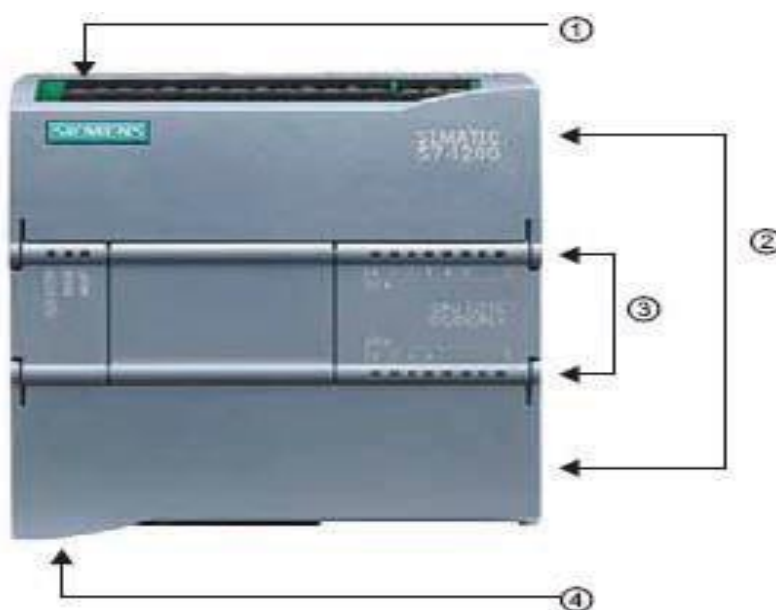


Рис. 1 Внешний вид ПЛК S7-1200 от SIEMENS: 1. Разъем питания; 2. Съёмный клеммный блок для подключения пользователя (за дверцами); 2. Гнездо для карты памяти под верхней дверцей; 3. Светодиоды состояния для встроенных входов/выходов; 4. Разъем PROFINET (на нижней стороне CPU)

Характеристики контроллеров Siemens

Характеристика	CPU 1211C	CPU 1212C	CPU 1214C
Физический размер (мм)	90 x 100 x 75		110 x 100 x 75
Пользовательская память <ul style="list-style-type: none"> • Рабочая память • Загрузочная память • Сохраняемая память 	<ul style="list-style-type: none"> • 25 Кбайт • 1 Мбайт • 2 Кбайта 		<ul style="list-style-type: none"> • 50 Кбайт • 2 Мбайта • 2 Кбайта
Локальные встроенные входы/выходы <ul style="list-style-type: none"> • цифровые • аналоговые 	<ul style="list-style-type: none"> • 6 входов/4 выхода • 2 входа 	<ul style="list-style-type: none"> • 8 входов/6 выходов • 2 входа 	<ul style="list-style-type: none"> • 14 входов/10 выходов • 2 входа
Величина образа процесса	1024 байта входов (I) и 1024 байта выходов (Q)		
Битовая память (M)	4096 байт		8192 байта
Дополнительные сигнальные модули	Нет	2	8
Сигнальная плата	1		
Коммуникационные модули	3 (левостороннее расширение)		
Скоростные счетчики <ul style="list-style-type: none"> • однофазные • со сдвигом фаз на 90° 	3 <ul style="list-style-type: none"> • 3 на 100 кГц • 3 на 80 кГц 	4 <ul style="list-style-type: none"> • 3 на 100 кГц • 1 на 30 кГц • 3 на 80 кГц • 1 на 20 кГц 	6 <ul style="list-style-type: none"> • 3 на 100 кГц • 3 на 30 кГц • 3 на 80 кГц • 3 на 20 кГц
Импульсные выходы	2		
Карта памяти	Карта памяти SIMATIC (факультативно)		
Длительность сохранения времени для часов реального времени	Тип. 10 дней / 6 дней минимум при 40 градусах С.		
PROFINET	1 коммуникационный порт для связи с Ethernet		
Скорость выполнения арифметических операций	18 мкс/команду		
Скорость выполнения булевых операций	0,1 мкс/ команду		

3. Основы ПЛК. Исполнение программы пользователя

Исполнение программы пользователя CPU поддерживает следующие виды блоков, позволяющие создать эффективную структуру вашей пользовательской программы:

– Организационные блоки (OB) определяют структуру программы. Некоторые OB имеют предопределенное поведение и стартовые события, но вы можете также создавать OB со своими собственными стартовыми событиями.

– Функции (FC) и функциональные блоки (FB) содержат программный код, соответствующий конкретным задачам или

комбинациям параметров. Каждая функция и каждый функциональный блок предоставляет в распоряжение набор входных и выходных параметров для совместного использования данных с вызываемым блоком. FB использует также связанный с ним блок данных (называемый экземплярным DB) для сохранения данных о состоянии во время исполнения, которые могут быть использованы другими блоками в программе.

– Блоки данных (DB) хранят данные, которые могут быть использованы программными блоками.

Исполнение программы пользователя начинается одним или несколькими необязательными организационными блоками (OB), которые после перехода в режим RUN обрабатываются один раз, затем следует один или более OB программного цикла, которые обрабатываются циклически. OB может быть также поставлен в соответствие прерывающему событию, которое может быть стандартным событием или событием-ошибкой; затем он исполняется, когда происходит соответствующее событие.

Функция (FC) или функциональный блок (FB) – это блок с кодом программы, который может быть вызван из OB или из другой функции или другого функционального блока. При этом возможны следующие уровни вложения:

– 16 из циклического OB или OB запуска
– 4 из OB прерываний с задержкой, OB циклических прерываний, OB аппаратных прерываний, OB ошибок по времени или OB диагностируемых ошибок FC не ставятся в соответствие никакому конкретному блоку данных (DB), тогда как FB непосредственно

связаны с DB и используют этот DB для передачи параметров и сохранения промежуточных значений и результатов.

Размер пользовательской программы, данных и конфигурации ограничен имеющейся в распоряжении загрузочной памятью и рабочей памятью в CPU. В рамках свободной рабочей памяти число поддерживаемых блоков не ограничено. Каждый цикл включает в себя запись выходов, чтение входов, исполнение команд программы пользователя и выполнение обслуживания системы или фоновая обработка. Этот цикл называется также циклом сканирования или просто сканированием. Сигнальная плата, сигнальные и коммуникационные модули обнаруживаются и регистрируются только при запуске.

4. Основные типы данных для работы в STEP 7

Типы данных используются для указания размера элемента данных, а также того, как эти данные могут быть интерпретированы. Каждый параметр команды поддерживает, по крайней мере, один тип данных, а некоторые параметры поддерживают несколько типов данных. Подведите указатель мыши к полю параметра команды, чтобы увидеть, какие типы данных поддерживаются для соответствующего параметра. Формальный параметр – это идентификатор на команде, который указывает адрес данных, подлежащих использованию командой (пример: вход IN1 команды ADD). Фактический параметр – это адрес или константа, где содержатся данные, подлежащие использованию командой (пример:

%MD400 "Number_of_Widgets"). Тип данных фактического параметра, указанный вами, должен соответствовать одному из поддерживаемых типов данных формального параметра, определяемого командой. При задании фактического параметра вы должны указать переменную (символ) или абсолютный адрес. Переменные связывают символическое имя (имя переменной) с типом данных, областью памяти, смещением в памяти, и комментарием и могут быть созданы в редакторе переменных ПЛК или в редакторе интерфейса для блока (OB, FC, FB или DB). Если вы вводите абсолютный адрес, не связанный ни с какой переменной, вы должны использовать подходящий размер, соответствующий поддерживаемому типу данных, тогда при вводе создается стандартная переменная. Для многих входных параметров вы можете также вводить постоянное значение. В следующей таблице описаны поддерживаемые элементарные типы данных и даны примеры ввода констант. Все типы данных, кроме типа данных String [строка], доступны как в редакторе переменных ПЛК, так и в редакторах интерфейсов блоков. Тип String имеется только в редакторах интерфейсов блоков. В следующей таблице приведены элементарные типы данных.

Таблица 2

Типы данных, используемые для программирования ПЛК

Тип данных	Размер (в битах)	Диапазон	Примеры ввода констант
Bool	1	от 0 до 1	TRUE, FALSE, 0, 1
Byte	8	от 16#00 до 16#FF	16#12, 16#AB
Word	16	от 16#0000 до 16#FFFF	16#ABCD, 16#0001
DWord	32	от 16#00000000 до 16#FFFFFFFF	16#02468ACE
Char	8	от 16#00 до 16#FF	'A', 't', '@'
Sint	8	от -128 до 127	123, -123
Int	16	от -32768 до 32767	123, -123
Dint	32	от -2147483648 до 2147483647	123, -123
USInt	8	от 0 до 255	123
UInt	16	от 0 до 65,535	123
UDInt	32	0 до 4294967295	123
Real	32	от +/-1,18 x 10 ⁻³⁸ до +/-3,40 x 10 ³⁸	123.456, -3.4, -1.2E+12, 3.4E-3
LReal	64	от +/-2,23 x 10 ⁻³⁰⁸ до +/-1,79 x 10 ³⁰⁸	12345.123456789 -1.2E+40
Time	32	от T#-24d_20h_31m_23s_648ms до T#24d_20h_31m_23s_647ms Хранится как: от -2,147,483,648 мс до +2,147,483,647 мс	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String	переменный	от 0 до 254 символов в размере байта	'ABC'

5. Язык программирования LAD

LAD – это графический язык программирования. Это представление основано на схемах электрических соединений.

Элементы схемы, например, замыкающий и размыкающий контакты, и катушки реле соединены в сети (рис.2).



Рис.2 Реализация кнопки старт

Для создания логики для сложных операций вы можете вставлять разветвления для формирования параллельных цепей. Параллельные цепи открываются вниз или подключаются непосредственно к шине электропитания. Разветвления оканчиваются вверх.

LAD предоставляет команды в виде блоков для ряда функций, например, арифметических операций, таймеров, счетчиков и пересылок.

При создании сетей LAD примите во внимание следующие правила:

- Каждая сеть LAD должна завершаться катушкой или блоковой командой. Не завершайте сеть ни командой сравнения, ни командой распознавания фронта (падающего или нарастающего).

- Вы не можете создавать разветвления, которые могут привести к изменению направления потока сигнала (рис.3).

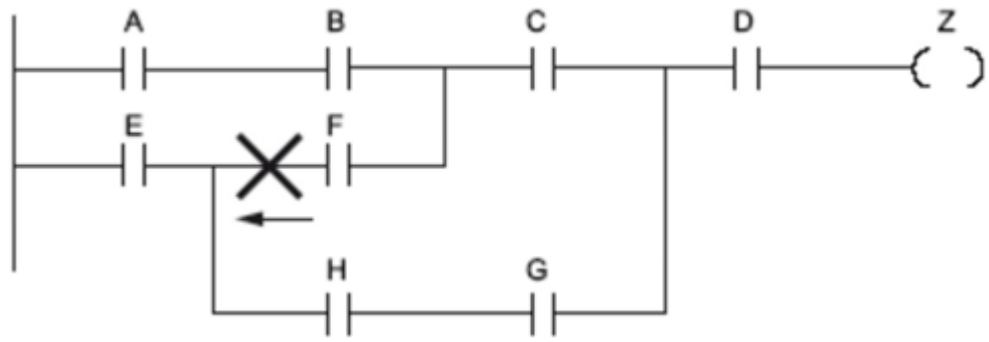


Рис.3 Неправильное соединение

– Вы не можете создать разветвление, вызывающее короткое замыкание (рис.4).

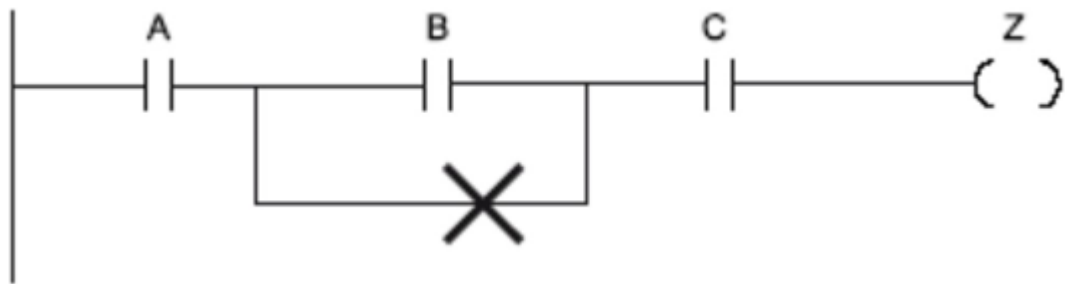


Рис.4 Запрещенное соединение: короткое замыкание

Назначать элементы можно перетаскиванием соответствующих обозначений на входы и выходы vsхеме (рис.5):

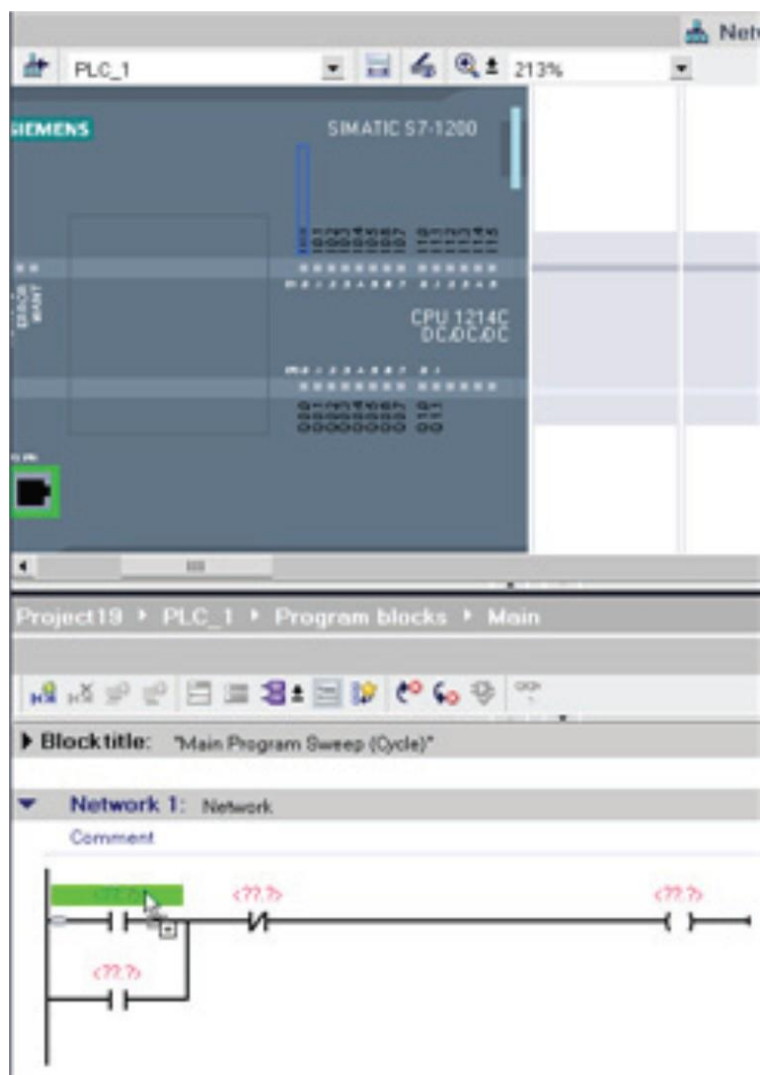


Рис.5 Перетаскивание элементов на входы-выходы ПЛК

Также можно обозначать элементы с клавиатуры, присваивая им имя и адрес:

- входы обозначают от $\%I0.0$, $\%I0.1$ до $\%I0.7$;
- выходы обозначаются от $\%Q0.0$ до $\%Q0.5$;
- промежуточные мнимые реле и ключи, необходимые для работы схемы, обозначают от $\%M0.0$ до $\%M0.7$ или от $\%M1.0$ до $\%M1.7$.

6. Алгоритм создания первого проекта

В работе требуется создать новый проект выполнив следующие шаги:

Шаг 1 - Открываем программу TotallyIntegratedAutomation, появляется окно (рис 6):

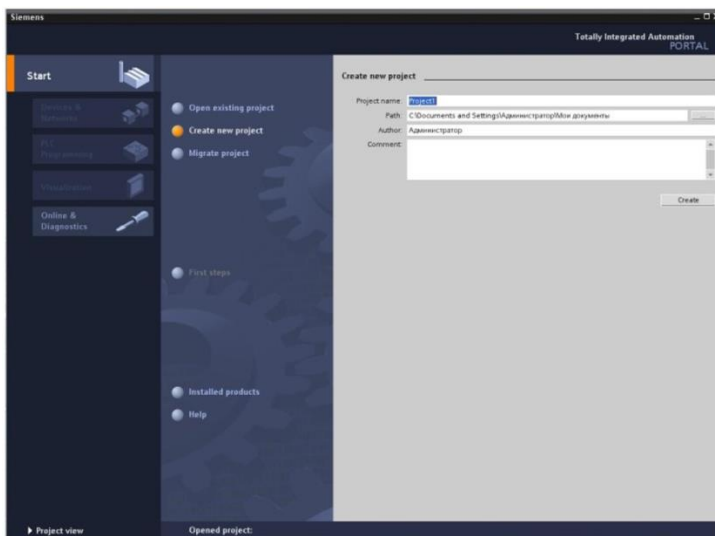


Рис.6 Открытие программы

Шаг 2 – Нажимаем Start => Create new project => Create (рис.7):

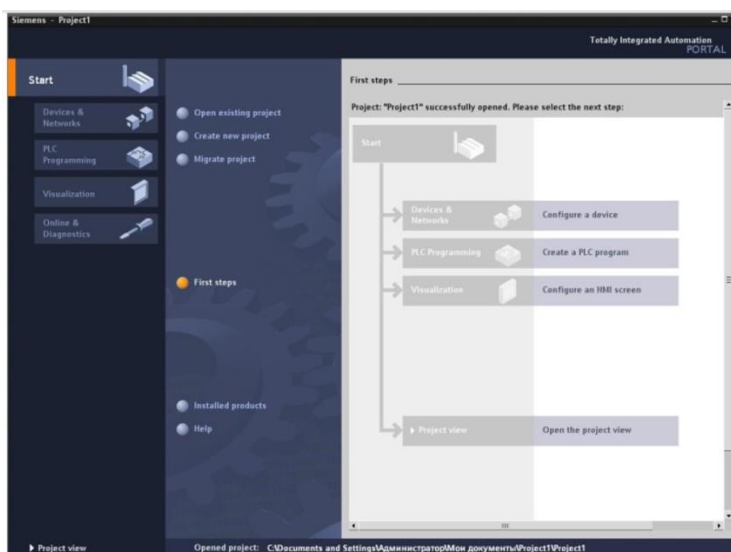
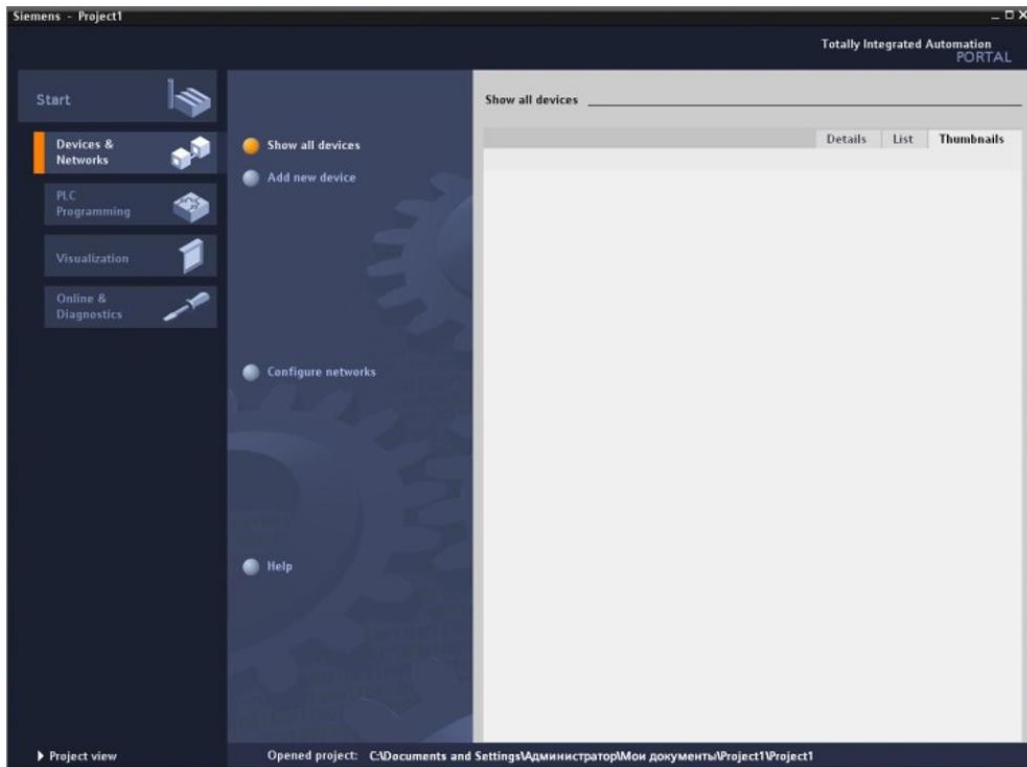
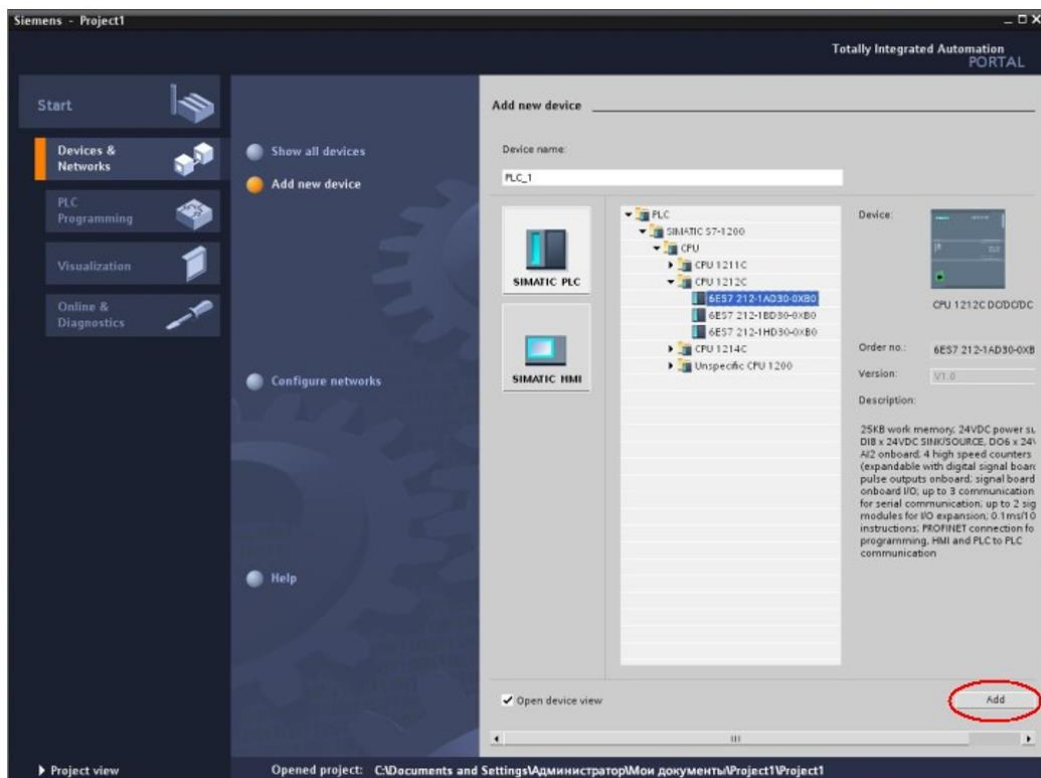


Рис.7 Создание проекта

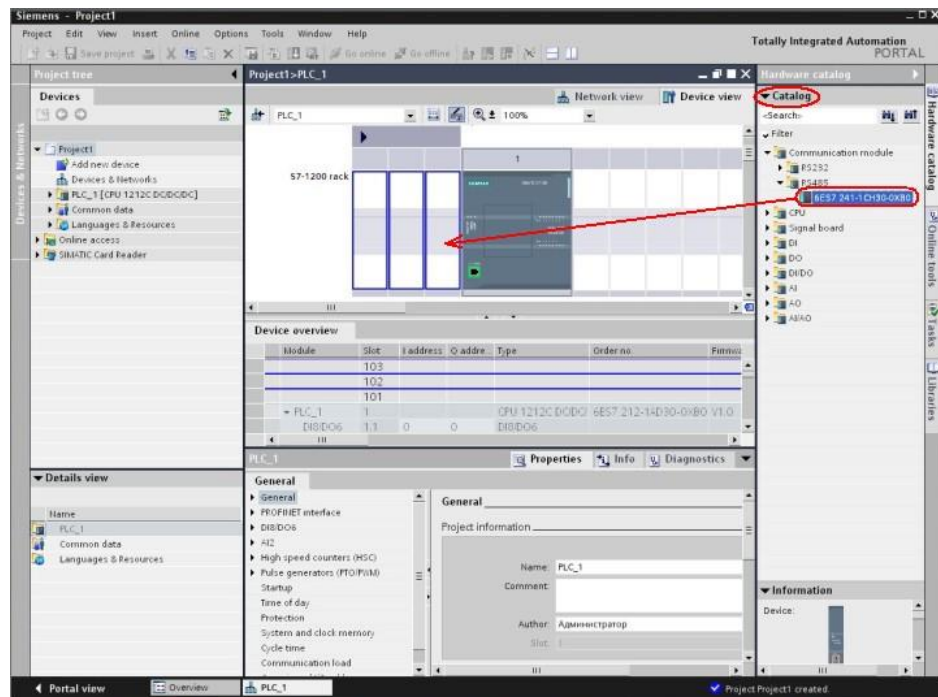
Шаг 3 - Выбираем Devices&Networks (рис.8):



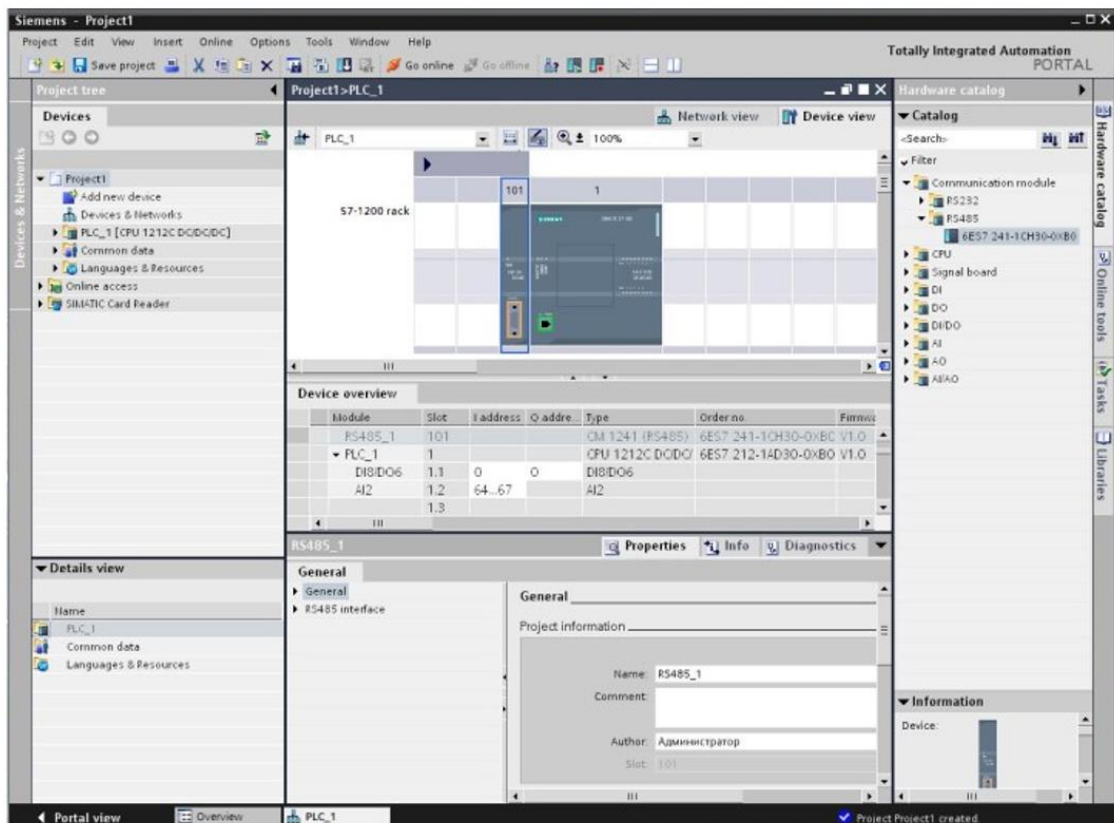
Шаг 4 – Выбираем Addnewdevice, затем выбираем тип микроконтроллера и нажимаем Add (рис.9):



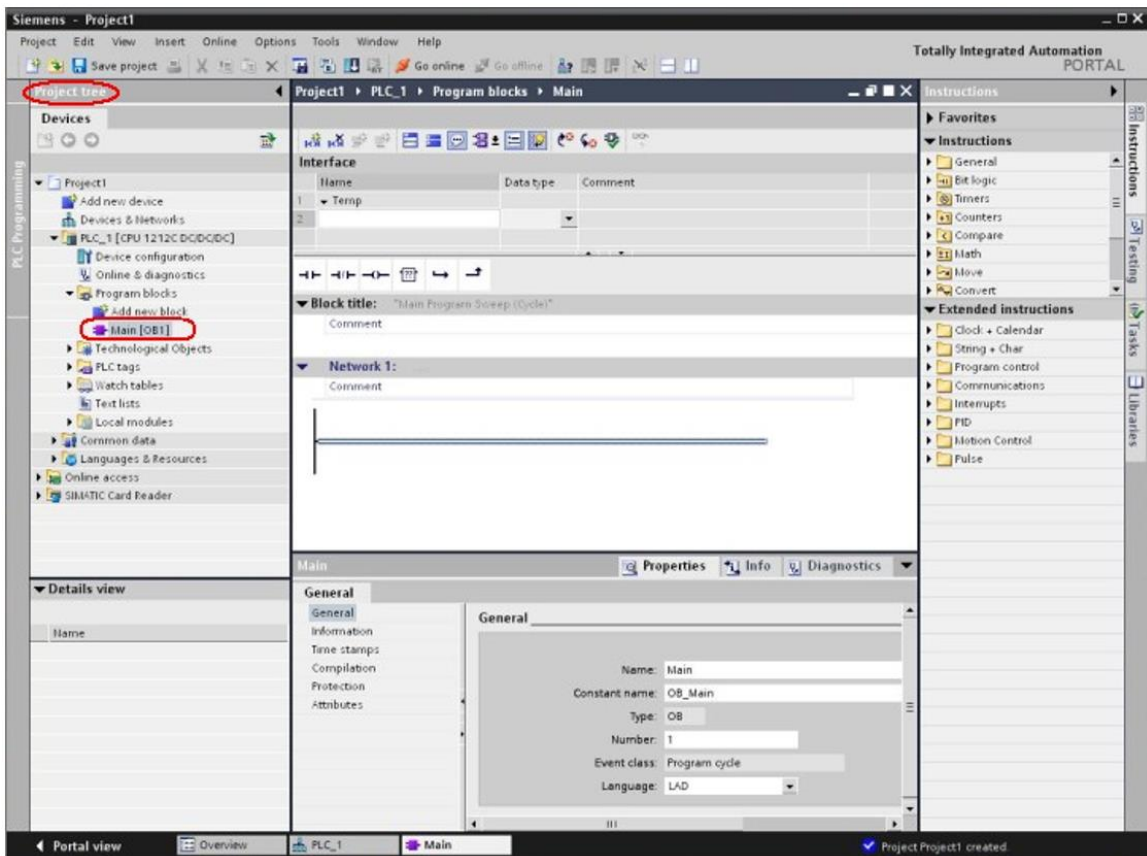
Шаг 5 – На вкладке Catalog определяем требуемы модули расширения (вверху справа) и перетаскиваем нужный модуль, как показано на рисунке ниже (рис.10):



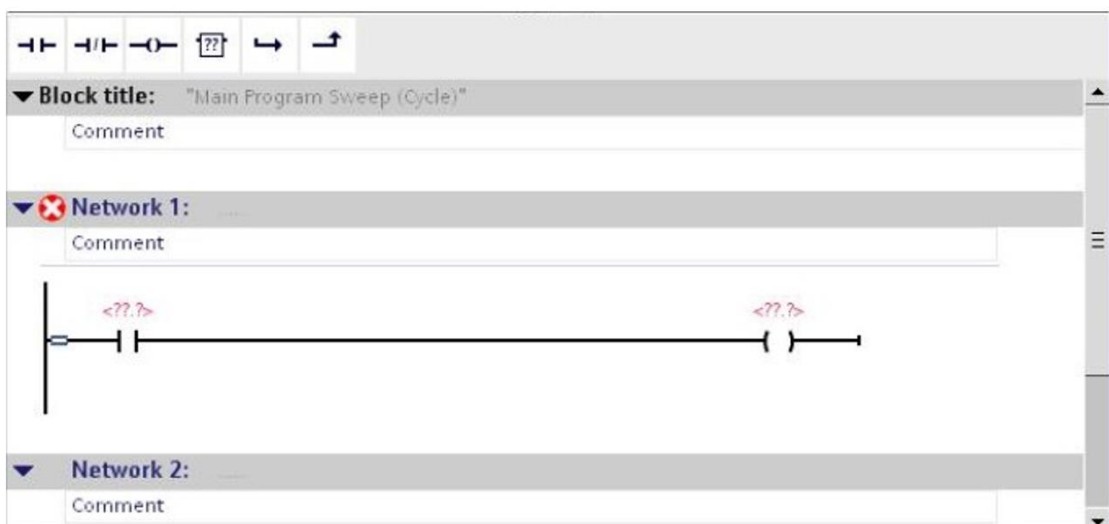
После перетаскивания нужного объекта имеем следующее (рис.11):



Шаг 6 – На вкладке ProjecttreeвыбираемMain[OB1] (рис.12):



Шаг 7 – Составляем произвольную релейно-контактную схему, как представлено на рисунке (рис.13):



Шаг 8 – В поля с вопросительными знаками записываем (рис.14):

- Для входных компонентов используем обозначения $%I0.0$, $%I0.1$ до $%I0.7$;
- для выходных от $%Q0.0$ до $%Q0.5$;
- если промежуточное мнимое реле или контакт (внутренняя переменная), то обозначаем от $%M0.0$ до $%M0.7$ (или от $%M1.0$ до $%M1.7$ и т.п.)



Шаг 9 – Во вкладке Instructions можем выбрать различные элементы для реализации технологического процесса, как например элементы TON и TOF.

TON после подачи сигнала на вход (замыкания контакта Tag_1) подаст сигнал на выход (реле Tag_2) через установленное время (в нашем случае 3 сек).

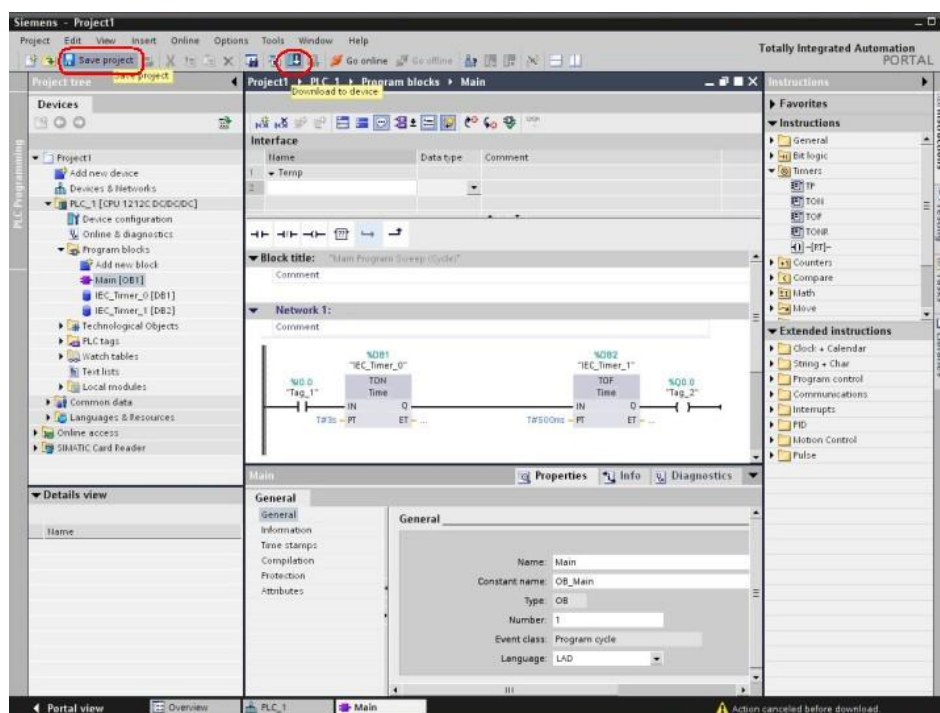
TOF после снятия сигнала (размыкания контакта Tag_1) снимет сигнал с выхода (отключит реле Tag_2) через установленное время (в нашем случае 0,5 сек).

Пример (рис.15):



Рис. 15 Пример работы с таймерами

Шаг 10 – Сохраняем проект – нажав на Saveproject; и загружаем его на микроконтроллер – нажав на значок Downloadto device (рис.16):



Шаг 11 – В появившемся окне жмём Refresh, затем выбираем устройство и жмём Load, далее Load и Finish.

7. Задание для выполнения

Необходимо разработать простую схему с использованием не менее 5 элементов в среде разработки TotallyIntegratedAutomation с применением языка программирования LAD.

8. Контрольные вопросы

1. Расскажите из чего состоит CPU ПЛК?

2. За реализацию каких функций отвечает CPU?
3. Расскажите какие разъёмы имеются в ПЛК S&-1200 от компании Siemens и для чего они необходимы?
4. Расскажите,какие виды блоков поддерживает CPU и что это дает?
5. Какие основные типы данных применяются для работы сS7-1200?
6. Что такое LAD? Какие правила программирования применяются в нем?
7. Какие переменны используются в языке программирования LAD?
8. Расскажите о способах обозначения переменных в LAD?

9. Содержание отчёта

Отчёт должен содержать:

- 1) титульный лист;
- 2) наименование работы и цель исследований;
- 3) описание хода выполнения задания;
- 4) результаты построенного программного кода на языкеLADв виде схем с описанием их работы.

10. Библиографический список

1. Корятин А. М., Петров Н. К., Радимов С. Н., Шапарев Н. К. Автоматизация типовых технологических процессов и установок: Учебник для ВУЗов – 2-ое изд., перераб. и доп. – М.: Энергоатомиздат, 1988. – 432 с.
2. Кравчик А. Э., Шлаф М. М., Афонин В. И., Соболенская Е. А. Асинхронные двигатели серии 4А: Справочник – М.: Энергоатомиздат, 1982. – 504 с.
3. Программируемый логический контроллер S7-1200. Руководство по эксплуатации. www.automation.siemens.com.
4. Сименс Эйджи Системное руководство программируемый контроллер S7-1200, 2015. – 1296 с.
5. Парр Э. Программируемые контроллеры: руководство для инженера / Э. Парр ; пер. 3-го англ. изд. Б. И. Копылова. - 3-е изд. - Москва : Бинوم. Лаб. знаний, 2007. 516 с.