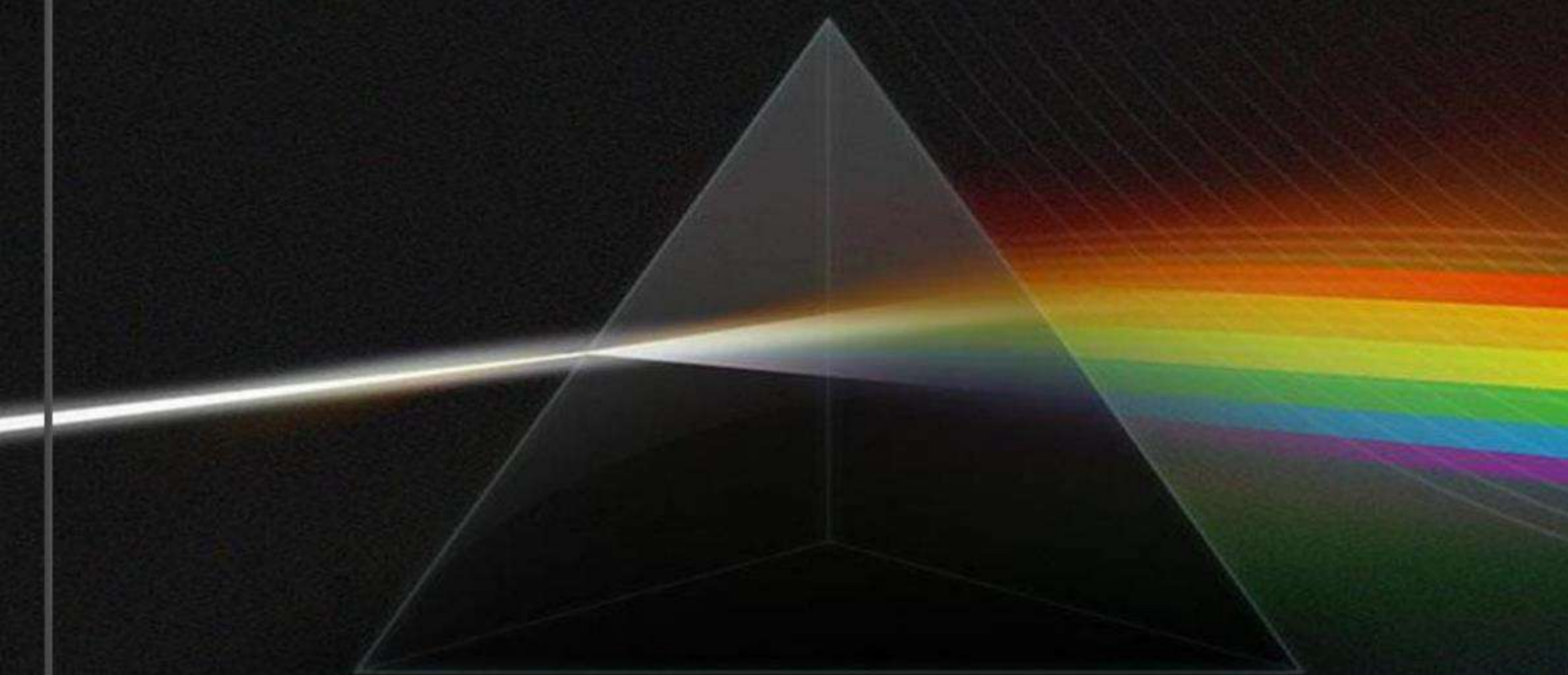


**ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

**М.В. Алешин, А.Б. Бекин**



**ПРОГРАММИРУЕМЫЙ  
ЛОГИЧЕСКИЙ КОНТРОЛЛЕР  
Simatic S7-1200**

**Методические указания**

М.В. Алешин, А.Б. Бекин

**ПРОГРАММИРУЕМЫЙ  
ЛОГИЧЕСКИЙ КОНТРОЛЛЕР  
Simatic S7-1200**

**Методические указания**

Новочеркасск  
ЮРГПУ  
2013

**УДК 681.5: 901.891.53**

**Рецензенты:**

к.т.н., доцент кафедры «Электромеханика» Рожков В.И.,  
к.т.н., доцент каф. «Электропривод и автоматика» Сухенко Н.А.

**Алешин М.В., Бекин А.Б.**

Программируемый логический контроллер Simatic S7-1200.  
Методические указания по курсу «Элементы систем автоматизи́ки».  
- Новочеркасск, ЮРГПУ, 2013. – 24с; ил.

Приведено описание ключевого элемента автоматических систем-  
программируемого логического контроллера Simatic S7-1200 фирмы  
Siemens.

На примерах показаны приёмы его конфигурирования и  
программирования, организации взаимодействия с другими элементами  
автоматики.

Учебное пособие предназначено для изучения дисциплины  
«Микропроцессорные устройства современного электропривода» и  
выполнения выпускных работ магистрантами по направлению  
подготовки «Электроэнергетика и электротехника», а также может  
быть использовано специалистами электротехнического направления.

© Алешин М.В., Бекин А.Б., 2012

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ПЛК	Промышленный логический контроллер
FBD	Язык программирования на основе функциональных блоков
LAD	Язык программирования на основе релейно-контактных элементов
Tag	«Тег» – дескриптор, применяемы как идентификатор для категоризации, описания, поиска данных и задания внутренней структуры
TIA	Программа Totally Integrated Automation

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	6
1 ОПИСАНИЕ СХЕМЫ S7 - 1200.....	7
2 СОЗДАНИЕ ПРОЕКТА .....	8
3 ОБЗОР СРЕДЫ ПРОГРАММИРОВАНИЯ.....	14
3.1 Язык программирования FBD .....	14
3.2 Язык программирования LAD .....	14
4 ПРИМЕР РАБОЧЕЙ ПРОГРАММЫ .....	17
ЗАКЛЮЧЕНИЕ .....	23
СПИСОК ЛИТЕРАТУРЫ .....	24

## ВВЕДЕНИЕ

Нам необходимо изучить промышленный микроконтроллер SIMATIC S7 – 1200 и области его применения. Изучить возможности программирования микроконтроллера для применения в человеко-машинных сетях. А также необходимо наладить связь между персональным компьютером, микроконтроллером и индикаторной панелью.

Эти контроллеры имеют модульную конструкцию и универсальное назначение. Они способны работать в реальном масштабе времени, могут использоваться для построения относительно простых узлов локальной автоматики или узлов комплексных систем автоматического управления, поддерживающих интенсивный коммуникационный обмен данными через сети Industrial Ethernet/PROFINET, а также PtP (Point-to-Point) соединения.

Программируемые контроллеры S7-1200 имеют компактные пластиковые корпуса со степенью защиты IP20, могут монтироваться на стандартную 35 мм профильную шину DIN или на монтажную плату и работают в диапазоне температур от 0 до +50 °C.

Они способны обслуживать от 10 до 284 дискретных и от 2 до 51 аналогового канала ввода-вывода. При одинаковых с S7-200 конфигурациях ввода-вывода контроллер S7-1200 занимает на 35% меньший монтажный объем.

К центральному процессору (CPU) программируемого контроллера S7-1200 могут быть подключены:

- коммуникационные модули (CM);
- сигнальные модули (SM) и сигнальные платы (SB) ввода-вывода дискретных и аналоговых сигналов.

Совместно с ними используются 4-канальный коммутатор Industrial Ethernet (CSM 1277) и модуль блока питания (PM 1207)..

# 1 ОПИСАНИЕ СХЕМЫ S7 - 1200

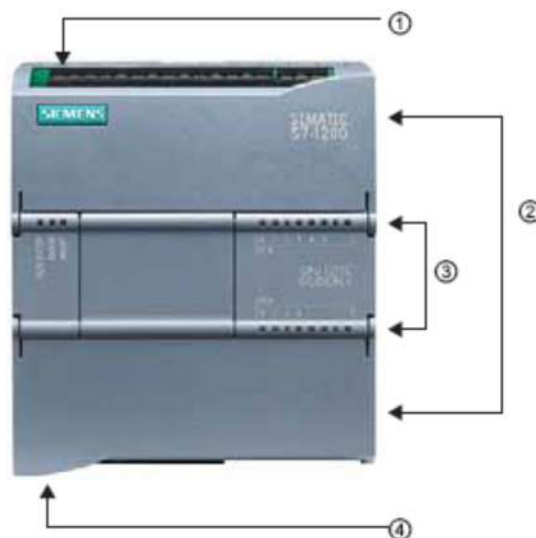


Рис. 1 – Микроконтроллер SIMATIC S7 – 1200.

1. Разъем питания;
2. Съёмный клеммный блок для подключения пользователя;
3. Гнездо для карты памяти под верхней дверцей;
4. Светодиоды состояния для встроенных входов/выходов;
5. Разъем PROFINET (на нижней стороне CPU).

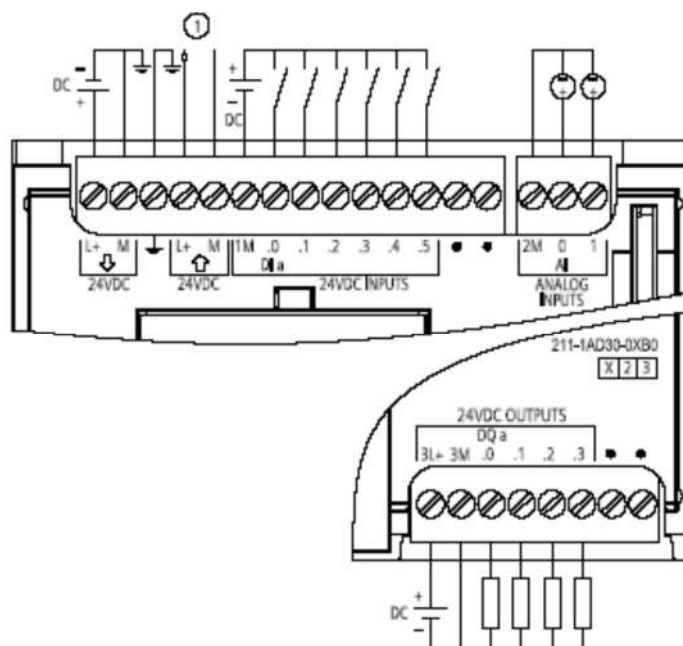


Рис. 2 – Схема подключений CPU 1211C DC/DC/DC (6ES7 211-1AD30-0XB0)

## 2 СОЗДАНИЕ ПРОЕКТА

1 – Открываем программу TIA, появляется окно:

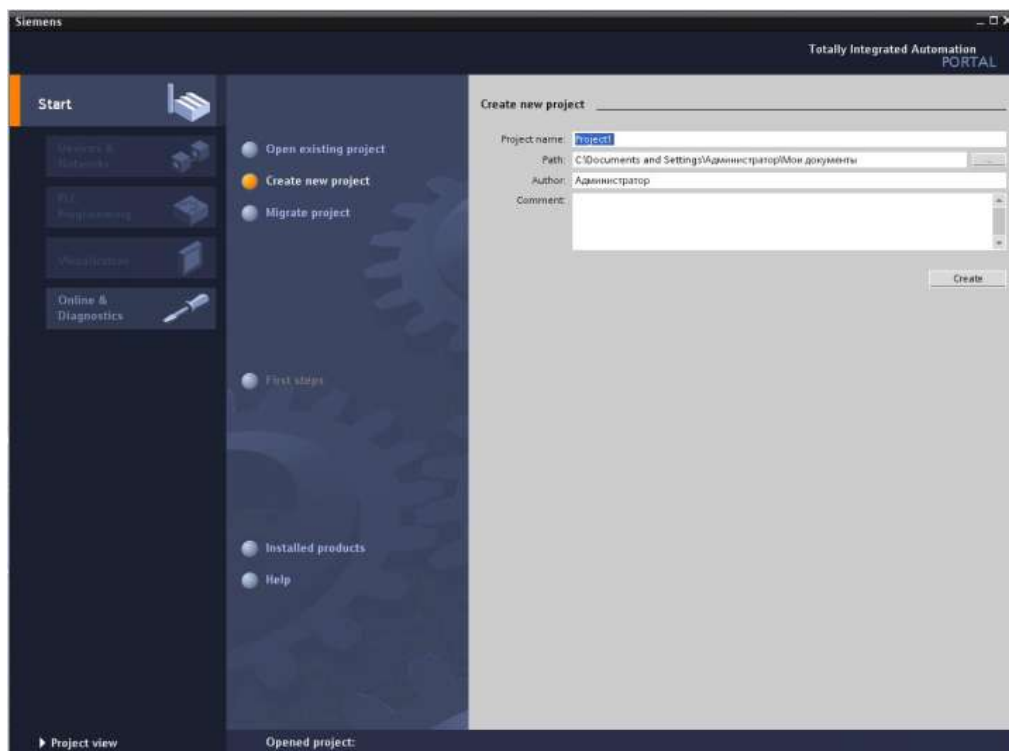


Рис. 3 – Снимок экрана

2 – Жмём Start => Create new project => Create, получаем:

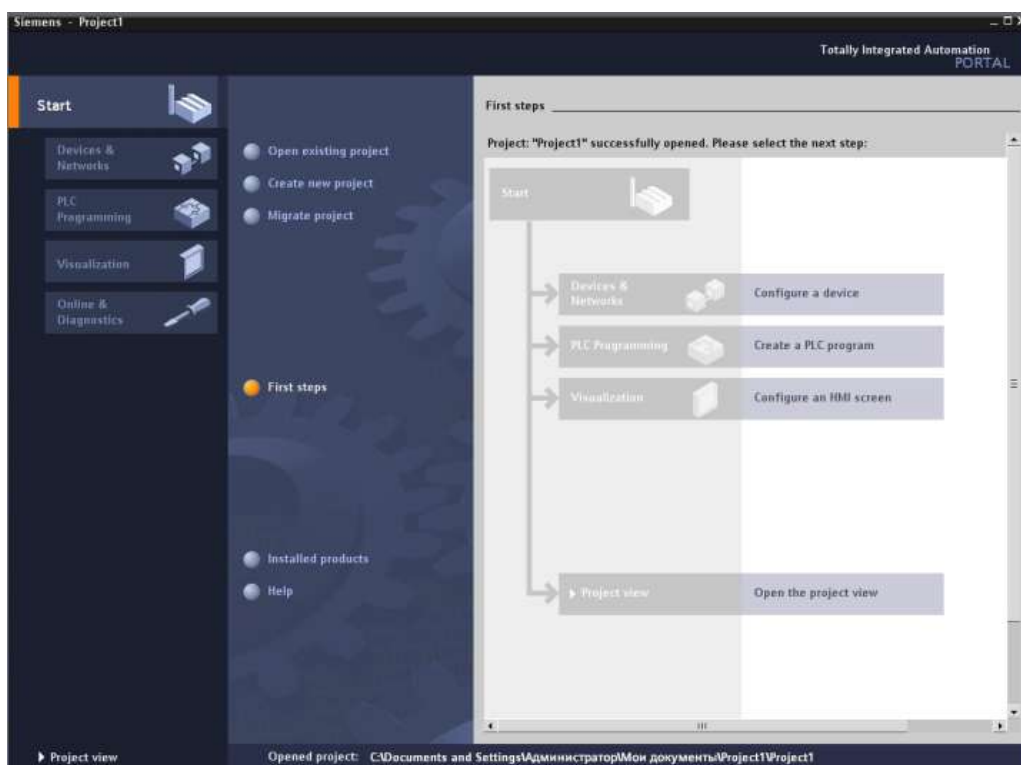


Рис. 4 – Снимок экрана

3 – Выбираем Devices & Networks, получаем:

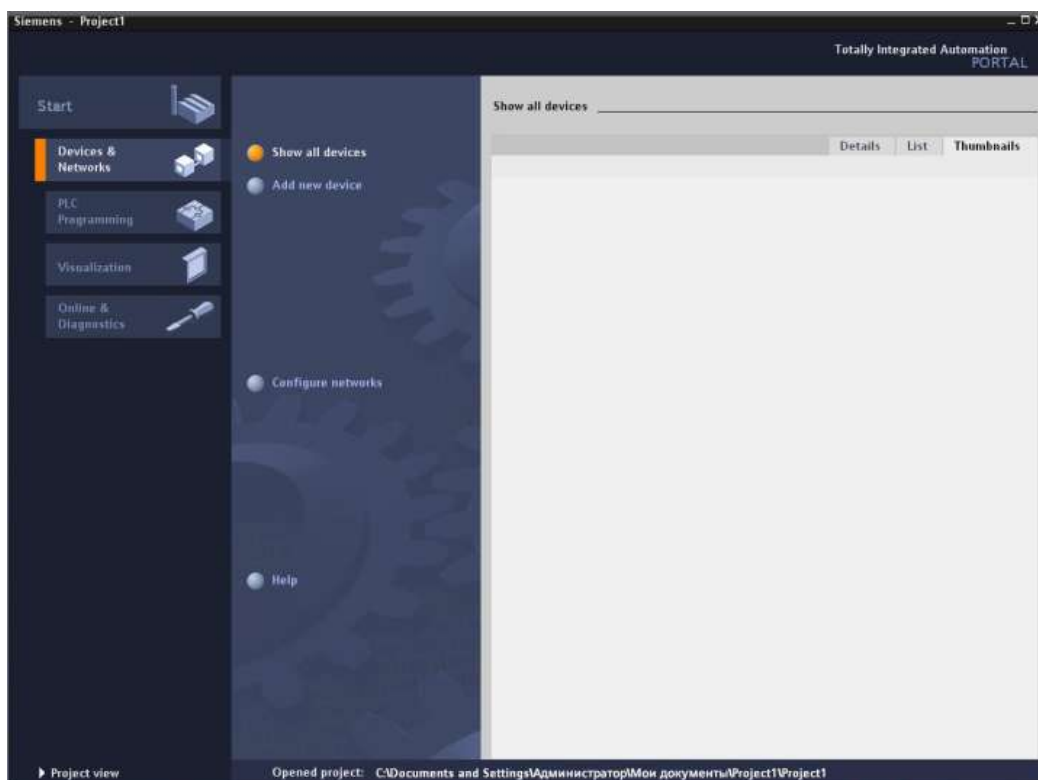


Рис. 5 – Снимок экрана

4 – Выбираем Add new device, затем выбираем тип микроконтроллера и жмём Add:

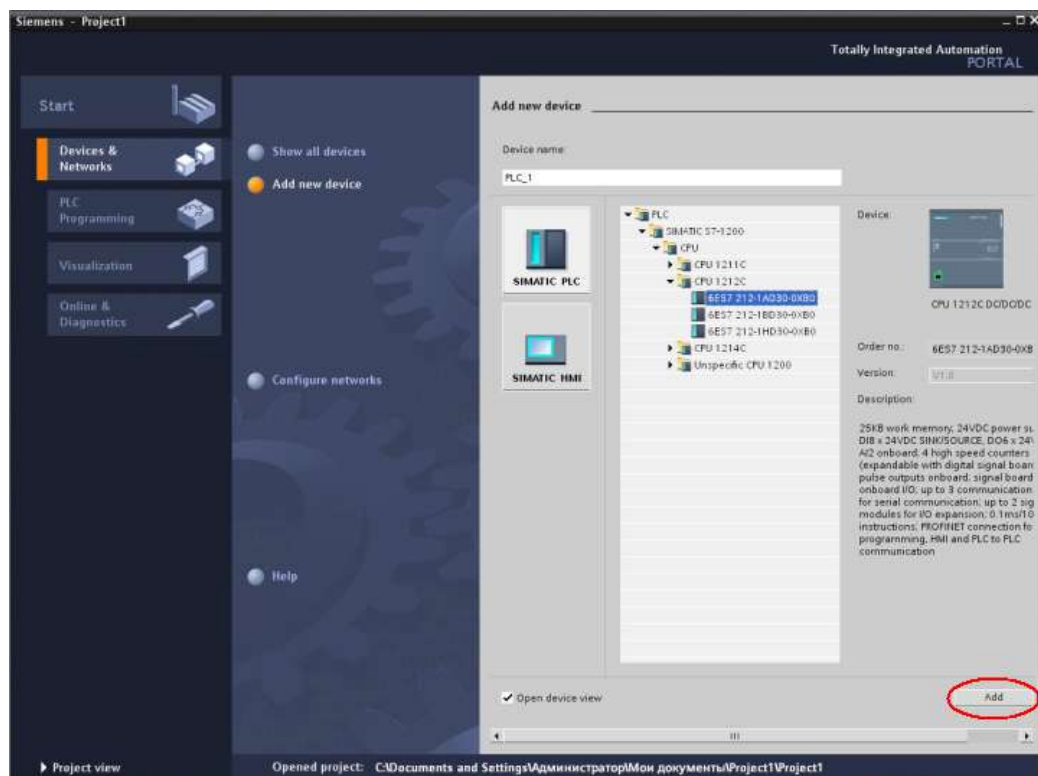


Рис. 6 – Снимок экрана

5 – Во вкладке Catalog выбираем модули расширения (вверху справа) и перетаскиваем нужный модуль, как показано на рис. 7:

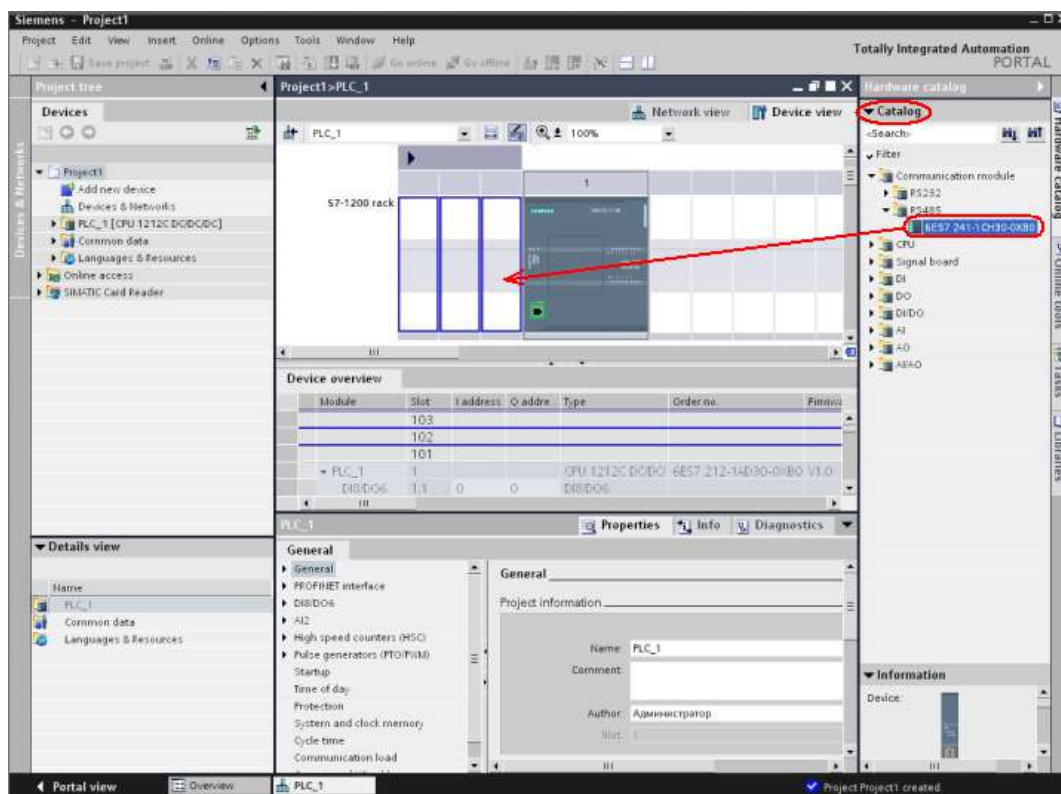


Рис. 7 – Снимок экрана

6 – После чего получаем:

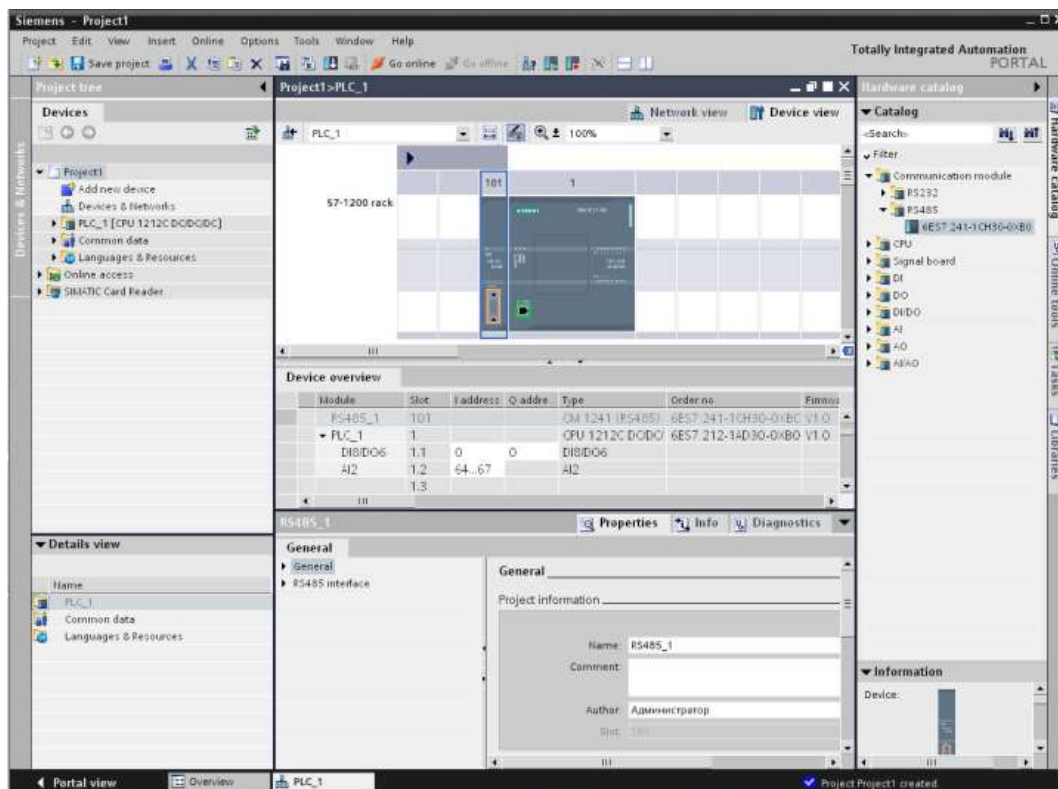


Рис. 8 – Снимок экрана

7 – Во вкладке Project tree открываем Main [OB1]:

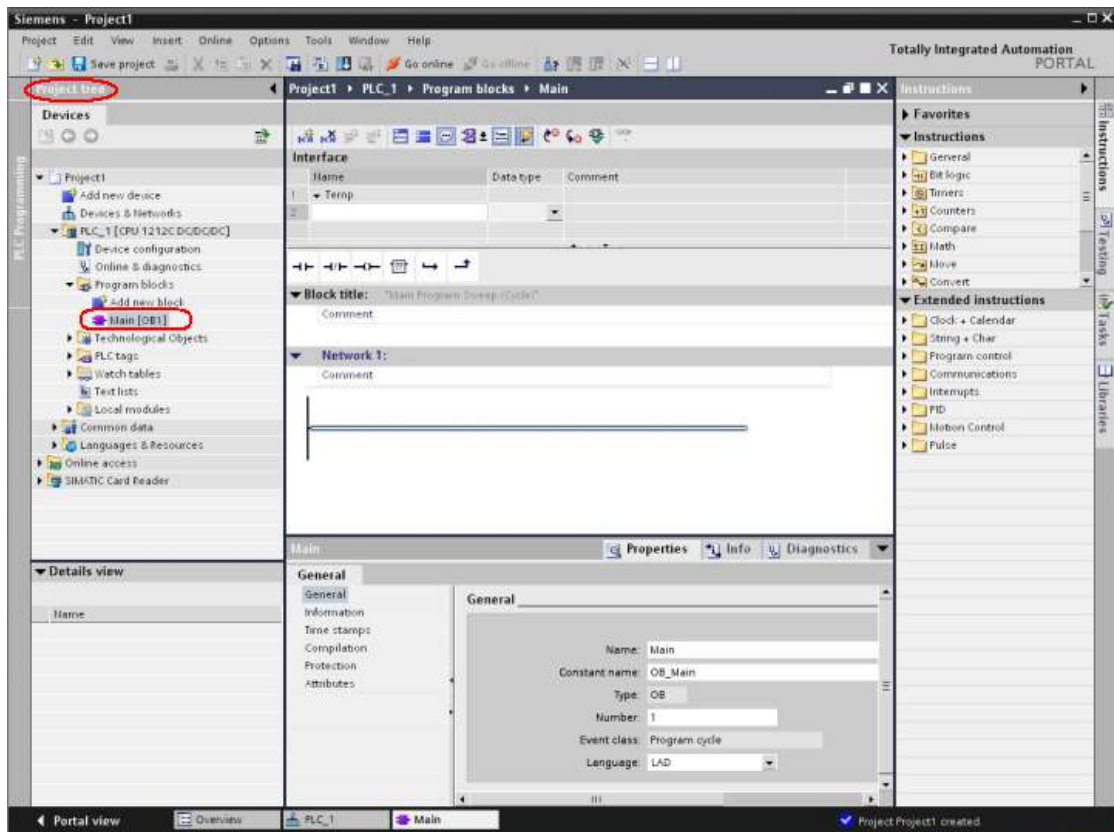


Рис. 9 – Снимок экрана

8 – Составляем релейно-контактную схему, как например:

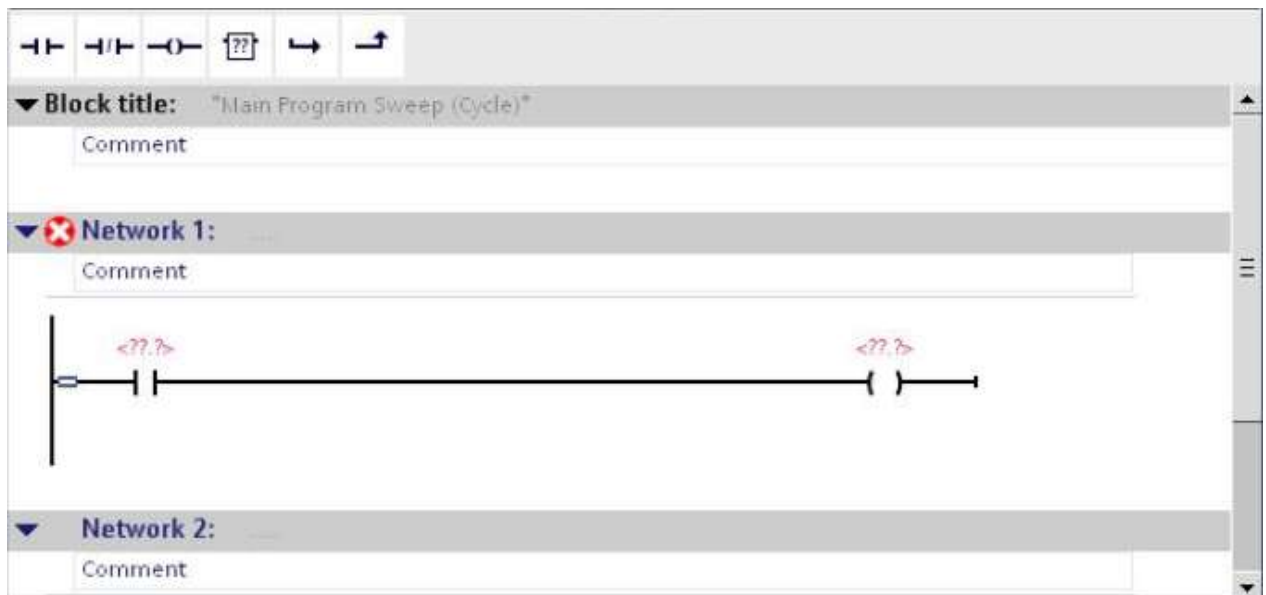


Рис. 10 – Снимок экрана

9 – Вместо вопросов вписываем:

- если вход, то обозначаем его от %I0.0, %I0.1 до %I0.7;

- если выход, то обозначаем его от %Q0.0 до %Q0.5;
- если промежуточное мнимое реле или контакт, то обозначаем от %M0.0 до %M0.7 (или от %M1.0 до %M1.7 и т. п.).

10 – Получаем:

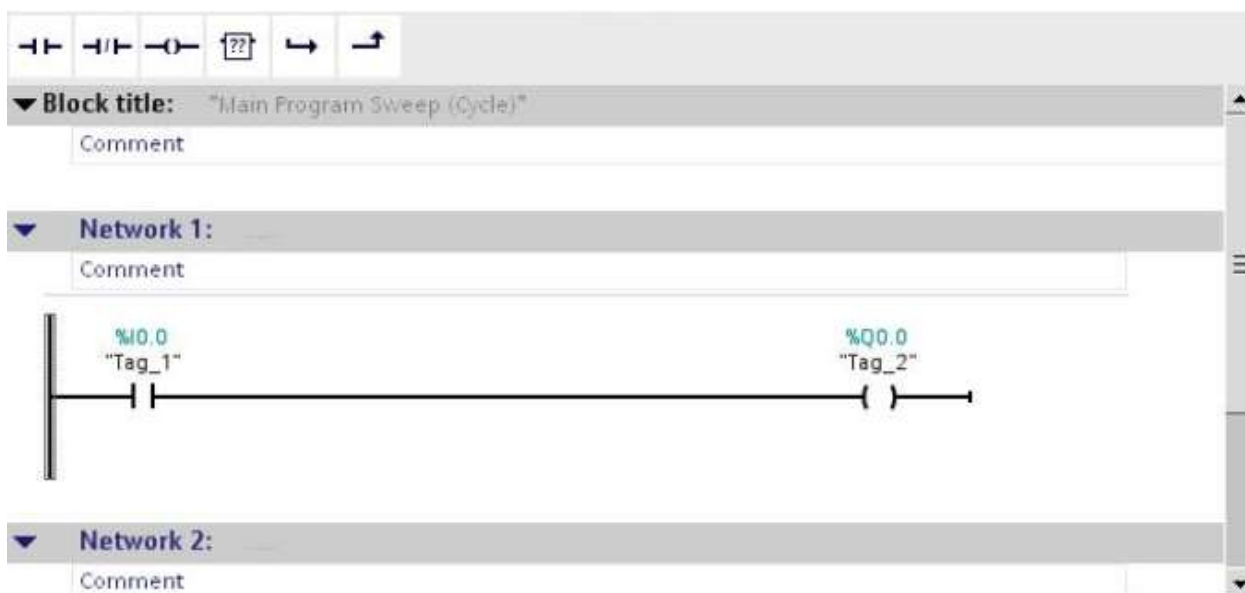


Рис. 11 – Снимок экрана

11 – 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. 12 – Снимок экрана

12 – Сохраняем проект – нажав на Save project; и загружаем его на микроконтроллер – нажав на значок Download to device:

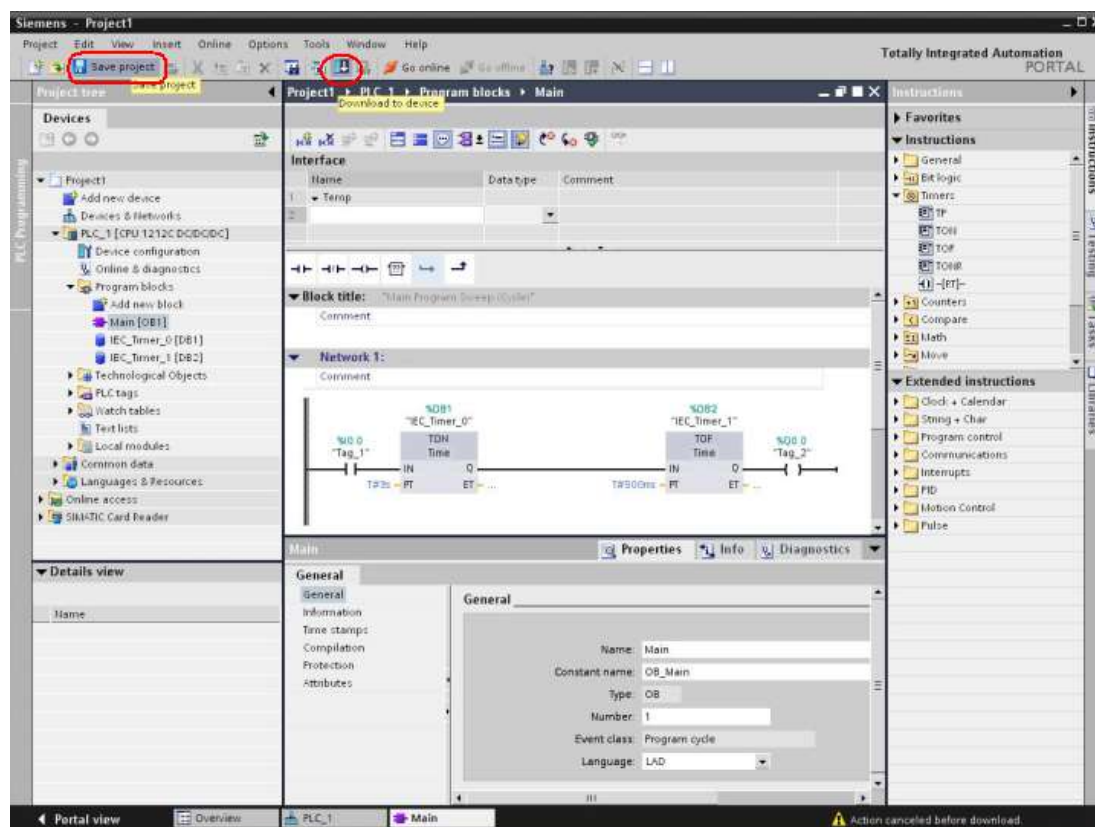


Рис. 13 – Снимок экрана

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

### 3 ОБЗОР СРЕДЫ ПРОГРАММИРОВАНИЯ

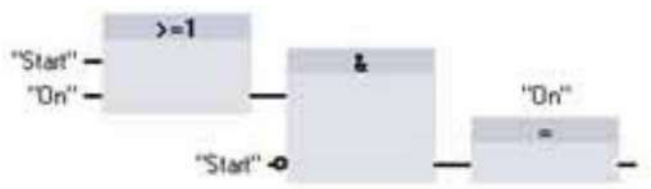
В качестве языка программирования вы можете выбрать цепную логическую схему (ladder logic, LAD), называемую также контактным планом, или функциональную блок-схему (Function Block Diagram, FBD), называемую также функциональным планом.

#### 3.1 Язык программирования FBD

Как и LAD, FBD также является графическим языком программирования.

Представление логики здесь основано на графических логических символах, используемых в булевой алгебре.

Математические и другие сложные функции могут быть представлены непосредственно в соединении с логическими блоками. Чтобы создать логику для сложных операций, вставляйте параллельные ветви между блоками.



*Рис. 14 – Пример функциональной схемы*

#### 3.2 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. 15 – Элементы схемы

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.

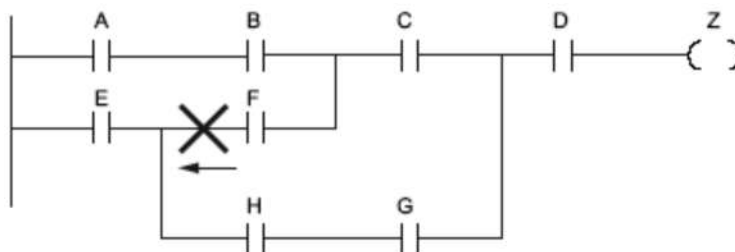


Рис. 16 – Элементы схемы

- Вы не можете создать разветвление, вызывающее короткое замыкание.

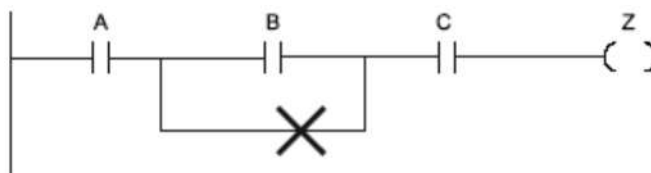


Рис. 17 – Элементы схемы

Назначать элементы можно перетаскиванием соответствующих обозначений на входы и выходы в схеме:

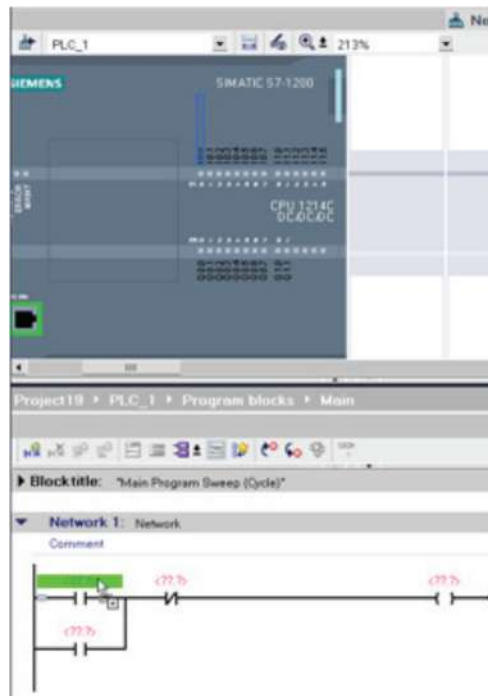


Рис. 18 - Пример

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  $\%M1.7$ .

#### 4 ПРИМЕР РАБОЧЕЙ ПРОГРАММЫ

В бассейн (или резервуар) поступает вода в неконтролируемом объёме. Необходимо поддерживать уровень воды в резервуаре не ниже определенного уровня. При этом нельзя допустить переполнения бассейна – в этом случае необходимо откачивать воду при помощи насоса. Для того чтобы избежать чрезмерной перегрузки оборудования будем использовать три асинхронных короткозамкнутых двигателя, которые будут по очереди откачивать воду из резервуара. В случае не достаточной мощности одного из двигателей, будет подключаться второй двигатель для совместного откачивания воды, а затем и третий, в случае если мощности двух двигателей недостаточно для отвода воды. Чтобы реализовать данную систему потребуется один датчик нижнего уровня воды SL1 и три датчика верхнего уровня воды SL2, SL3, SL4. Датчики верхнего уровня воды необходимо расположить начиная с датчика SL2 – крайний нижний датчик верхнего уровня, при срабатывании которого запускается один насос для отвода воды из резервуара, немного выше датчик SL3 – средний датчик верхнего уровня, при срабатывании которого запускается второй насос, затем ещё выше датчик SL4 – крайний верхний датчик верхнего уровня – он запускает третий насос для отвода воды.

Всего используется четыре двигателя: три из них участвуют в производственном цикле, а четвертый используется в качестве резервного. Очередность работы двигателей и порядок работы контролируется промышленным контроллером Simatic S7 1200, в котором содержится программа функционирования системы, а также подключены датчики уровней. Исправность двигателей также контролируется микроконтроллером, который подключает вместо неисправного двигателя резервный.

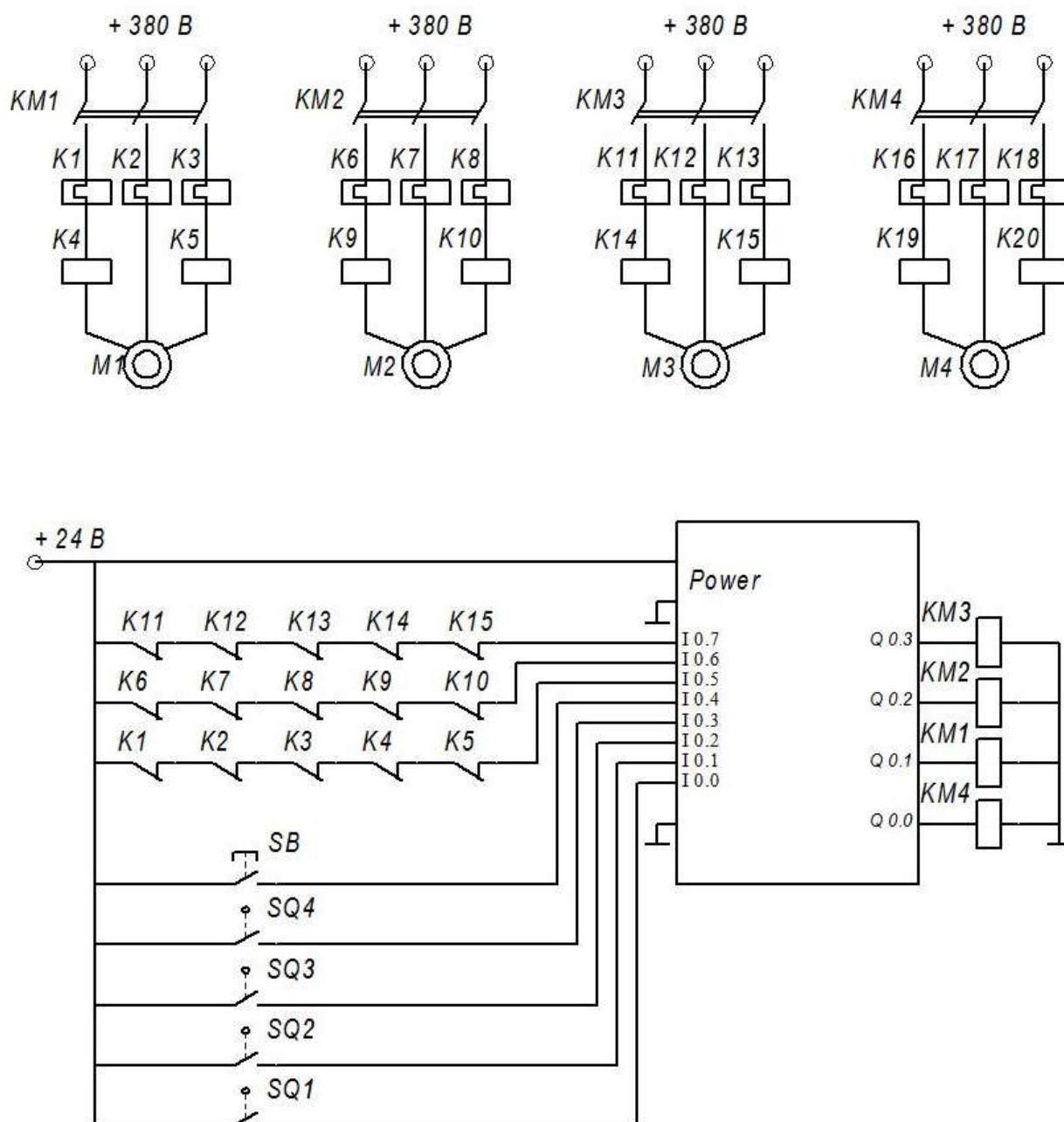


Рис. 19. Принципиальная электрическая схема системы отвода воды.

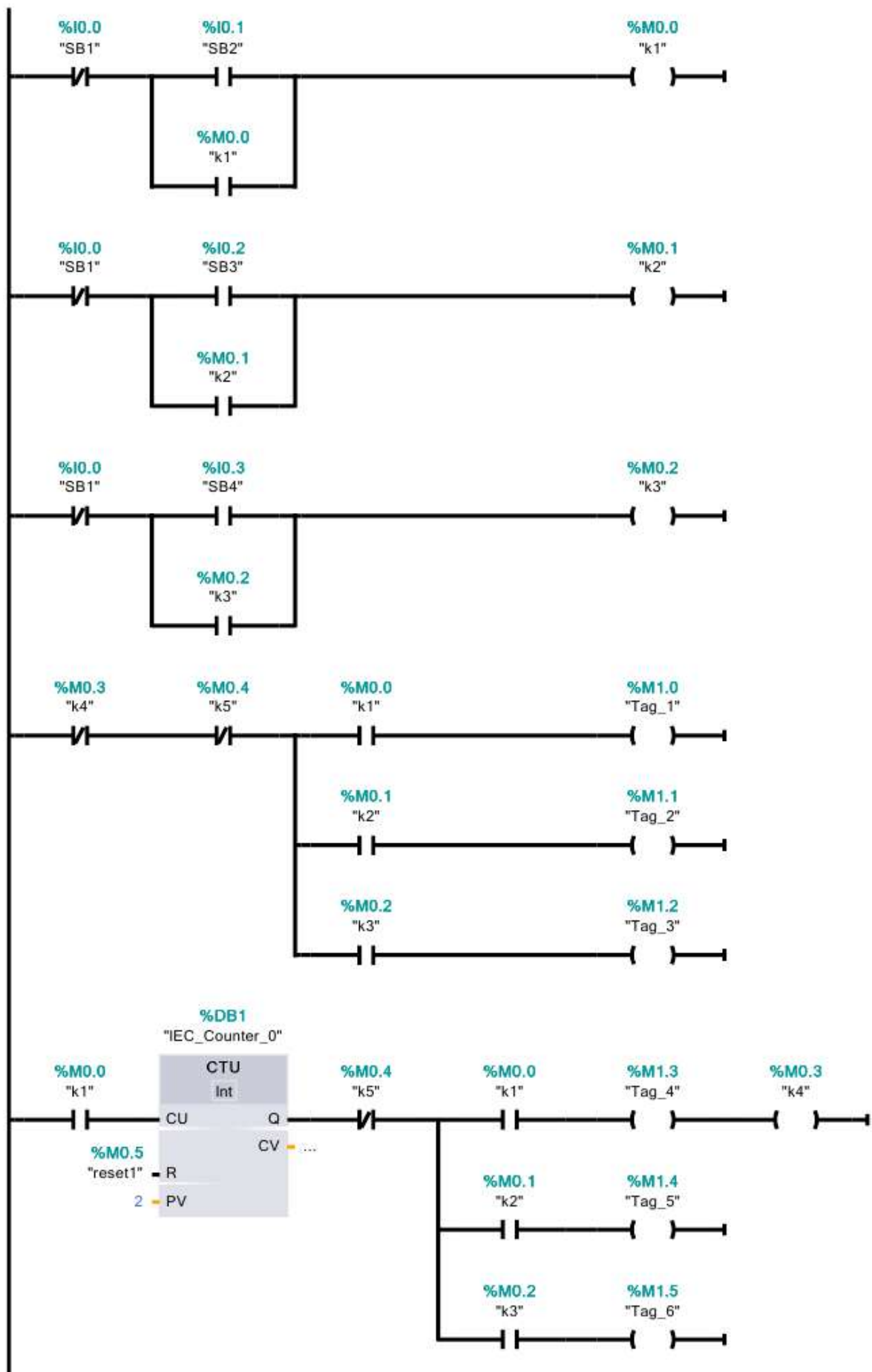
Бассейн постепенно заполняется водой до срабатывания датчика SL2, после чего включается первый двигатель и начинает откачивать воду из резервуара, если его мощности недостаточно, то вода будет продолжать прибывать, по срабатыванию датчика SL3 включается второй двигатель и два двигателя откачивают воду, при срабатывании датчика SL4 включается третий двигатель – вода откачивается из бассейна тремя двигателями.

Уровень воды в резервуаре уменьшается и при срабатывании датчика SL1 двигатели выключаются. При повторном заполнении бассейна при срабатывании датчика SL2 включится второй двигатель, SL3 – третий, SL4 – первый. В третьем цикле при заполнении бассейна при срабатывании датчика SL2 включится третий двигатель, SL3 – первый, SL4 – второй. При выходе из строя одного из двигателей в работу включается четвертый (резервный) двигатель. В схеме также предусмотрен перезапуск работы контроллера – он осуществляется кнопкой SB, при нажатии которой контроллер сбрасывает своё состояние и возвращается к начальному состоянию.

Три тепловых реле и два реле максимального тока на каждом двигателе контролируют исправную работу. В случае исправного двигателя на вход, за которым закреплен двигатель, поступает сигнал 1. Если какой-либо двигатель неисправен, то срабатывает одно из реле, а оно, в свою очередь, размыкает свой контакт и на входе контроллера соответствующего этому двигателю формируется сигнал о неисправности – сигнал 0. В этом случае контроллер вводит резервный двигатель, до тех пор, пока не будет устранена неисправность.

Все циклы работы двигателей запрограммированы и защиты в промышленный контроллер.

Программа, реализующая работу такой системы, приведена на рис. 20.





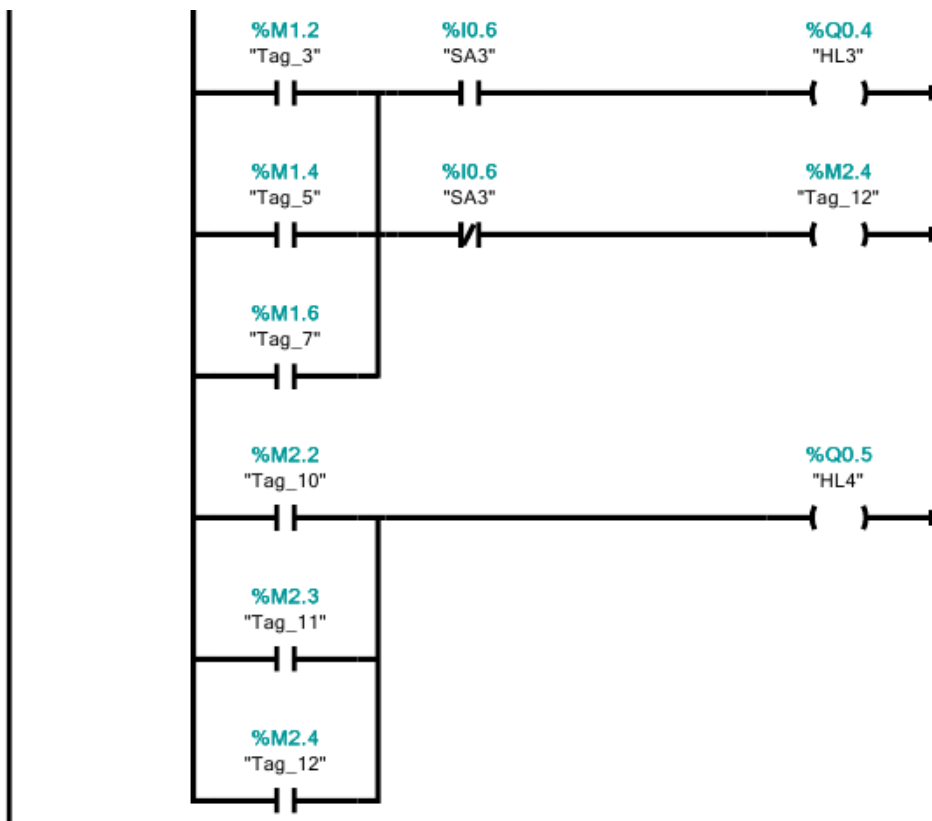


Рис. 20. Программа управления на языке LAD.

## ЗАКЛЮЧЕНИЕ

STEP 7 Basic является программным продуктом единой среды разработки, позволяющей:

- использовать однородную среду разработки для решения любых задач автоматического управления;
- обеспечивать поддержку всех фаз жизненного цикла систем автоматизации;
- использовать единый набор сервисных служб для поддержки единой концепции оперативного управления и мониторинга, конфигурирования аппаратуры, организации промышленной связи, диагностики и т.д.

К основным достоинствам пакета можно отнести:

- Поддержку всеобъемлющей концепции использования библиотек для многократного использования любых компонентов проекта.
- Поддержку интеллектуальных механизмов Drag & Drop для передачи данных между различными редакторами для программируемых контроллеров и приборов человеко-машинного интерфейса.
- Наличие единой базы данных проекта с однородным набором символьных имен.
- Быстрый доступ к любым задачам автоматизации, включая интерактивную работу с системой автоматизации и ее диагностики.
- Простое графическое конфигурирование аппаратуры и сетевых структур в среде одного редактора.
- Наличие простого и интуитивно понятного интерфейса пользователя для обеспечения доступа к различным вариантам отображения информации и редакторам.
- Наличие высокоэффективного редактора для разработки программ контроллеров.

## СПИСОК ЛИТЕРАТУРЫ

1. Кобытин 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](http://www.automation.siemens.com).