

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ САРАТОВСКОЙ ОБЛАСТИ  
«САРАТОВСКИЙ КОЛЛЕДЖ ПРОМЫШЛЕННЫХ ТЕХНОЛОГИЙ  
И АВТОМОБИЛЬНОГО СЕРВИСА»

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ  
ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ  
ОБЩЕПРОФЕССИОНАЛЬНОЙ ДИСЦИПЛИНЫ  
ОП.09 ОСНОВЫ МЕХАТРОНИКИ**

программы подготовки специалистов среднего звена  
для специальности

***15.02.09 АДДИТИВНЫЕ ТЕХНОЛОГИИ***

на базе основного общего образования  
с получением среднего общего образования

Разработчик:  
Цыбина Т.В., преподаватель  
ГАПОУ СО «СКПТиАС»

Саратов 2020

## Содержание

Пояснительная записка.....	3
Практическая работа №1.....	4
Практическая работа №2.....	7
Практическая работа №3.....	17
Практическая работа №4.....	48
Практическая работа №5.....	56

## Пояснительная записка

Методические рекомендации по выполнению практических работ предназначены для проведения практических занятий по дисциплине «ОП.09 Основы мехатроники» в соответствии с рабочей программой, разработанной в соответствии с ФГОС СПО по специальности 15.02.09 Аддитивные технологии.

В результате выполнения практических работ обучающийся должен

**обладать профессиональными компетенциями:**

ПК 2.2. Контролировать правильность функционирования установки, регулировать ее элементы, корректировать программируемые параметры.

**уметь:**

- читать и составлять принципиальные схемы электрических, гидравлических и пневматических приводов несложного технологического оборудования;
- составлять управляющие программы для программируемых логических контроллеров в SCADA -системе;
- распознавать, классифицировать и использовать датчики, реле и выключатели в системах управления;
- правильно эксплуатировать мехатронное оборудование.

## Практическая работа №1

**Тема:** Разработка функциональной схемы привода главного движения станка

**Цель работы:**

- уметь читать и составлять принципиальные схемы электрических, гидравлических и пневматических приводов несложного технологического оборудования;
- уметь распознавать, классифицировать и использовать датчики, реле и выключатели в системах управления;
- уметь правильно эксплуатировать мехатронное оборудование.

### Краткие теоретические сведения

*Технологический процесс механической обработки детали на металлорежущем станке*

Приводом движения подачи называется механизм, посредством которого режущий инструмент вводится в соприкосновение с обрабатываемыми участками изделия.

В зависимости от типа станка (токарный, фрезерный, строгальный и т. д.) подача может быть непрерывной или периодической, прямолинейной или круговой. Наиболее сложным является привод движения подачи для токарных станков (рис.1.1).

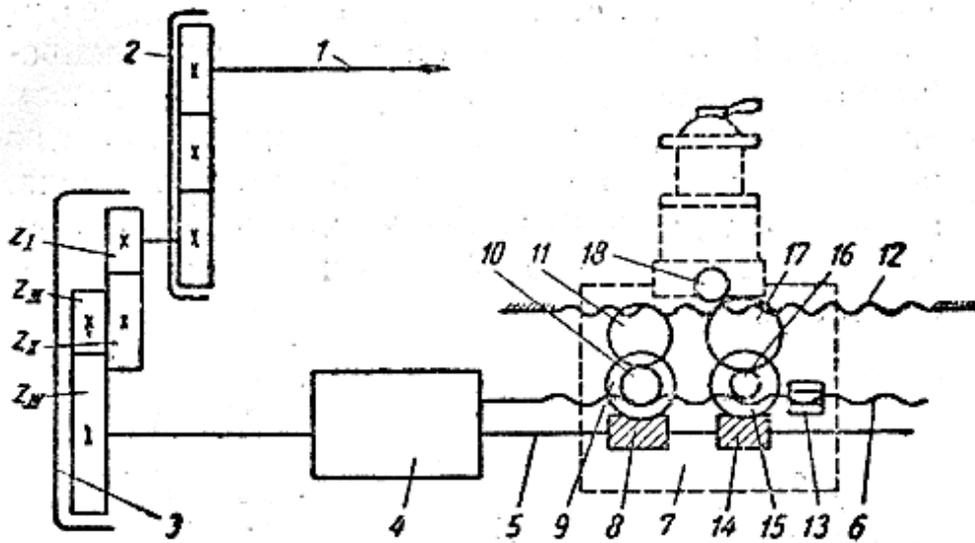


Рисунок 1.1 - Схема привода движения подачи

Движение передается от шпинделя 1 к реверсивному механизму 2, от него через сменные зубчатые колеса 3 и коробку подач 4 к ходовому валу 5 или ходовому винту 6. От ходового вала, или ходового винта, движение передается суппорту 7, а вместе с тем и резцу. В первом случае, т. е. от ходового вала, движение передается червяком 8 к зубчатому колесу 9 и от него через зубчатые колеса 10 и 11 к рейке 12, жестко закрепленной на станине станка. При вращении зубчатого колеса 11, входящего в зацепление с рейкой 12, происходит его перемещение вместе с суппортом вдоль рейки 12 или, что одно и то же, вдоль оси шпинделя.

Если движение от коробки подач передается ходовому винту 6, то в этом случае разъемная гайка 13, жестко связанная с суппортом, перемещаясь по ходовому винту, будет перемещать и супорт (вдоль оси шпинделя). Ходовым валом пользуются при выполнении всех токарных операций за исключением нарезания резьбы; в последнем случае применяется ходовой винт. Такое ограниченное употребление ходового винта вызывается необходимостью предохранять его от излишнего износа, сохраняя точность его размеров, необходимую для работы при нарезании резьб.

Поперечная подача производится от ходового вала. Червяк 14 вращает зубчатое колесо 15, на одной оси с которым находится зубчатое колесо 16, приводящее во вращение зубчатое колесо 17, вращающее зубчатое колесо 18; последнее жестко связано с винтом поперечного самохода супорта. При вращении этого винта перемещается гайка, жестко связанная с верхними салазками супорта, а вместе с ними и резец в направлении, перпендикулярном к оси шпинделя.

Если бы механизмы 3 и 4 имели постоянное передаточное отношение, величина продольной и поперечной подач была бы постоянной. Для получения различных величин подачи пользуются набором сменных зубчатых колес или коробкой подач, или совместно тем и другим механизмом.

Функциональная схема следящей системы привода подачи станка (рис.1.2), в общем случае должна включать: сумматор, электронный усилитель, усилитель мощности, исполнительный двигатель, механическую передачу и датчик обратной связи.

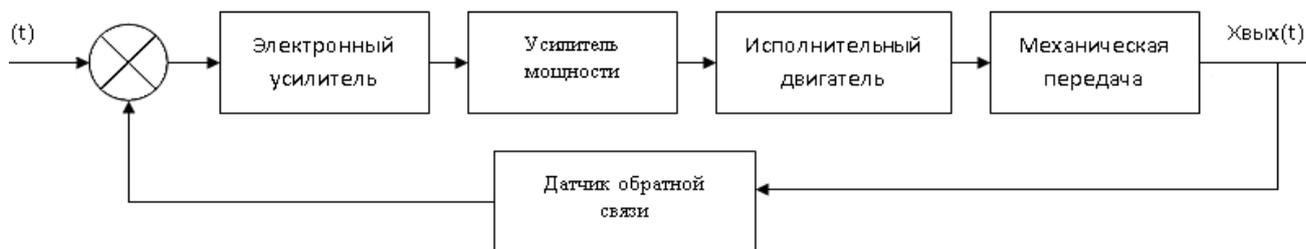


Рисунок 1.2 - Функциональная схема следящей системы привода подачи

### Порядок выполнения практической работы:

1. Изучить краткие теоретические сведения.
2. Изучить функциональную схему следящей системы привода подачи (рис. 1.2).
3. Заменить функциональные блоки схемы следящей системы привода подачи на элементы привода станка.
4. Подготовить отчет.

### Отчет по работе должен содержать:

1. Название работы.
2. Цель работы.
3. Результаты выполнения практической работы.

### **Вопросы для самопроверки:**

1. Что такое сумматор?
2. Для чего применяют электронный усилитель?
3. Для чего применяют усилитель мощности?
4. Какие двигатели применяют для металлорежущих станков?
5. Какие механические передачи применяют в металлорежущих станках?
6. Какие датчики обратной связи применяют в металлорежущих станках?

### **Средства обучения:**

**1. Оборудование и материалы:** Раздаточный материал, Рабочая тетрадь

**2. Вербальные средства обучения:**

Шишмарёв, В. Ю. Автоматика : учебник для среднего профессионального образования / В. Ю. Шишмарёв. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2017. — 284 с. — (Профессиональное образование). — ISBN 978-5-534-05168-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/409032>

**3. Технические средства обучения (ТСО):** -

## Практическая работа №2

**Тема:** Разработка пневмо-схемы управления прессом

**Цель работы:**

- уметь читать и составлять принципиальные схемы электрических, гидравлических и пневматических приводов несложного технологического оборудования;
- уметь распознавать, классифицировать и использовать датчики, реле и выключатели в системах управления;
- уметь правильно эксплуатировать мехатронное оборудование.

### Краткие теоретические сведения

#### Структура пневматической системы и последовательность прохождения сигнала

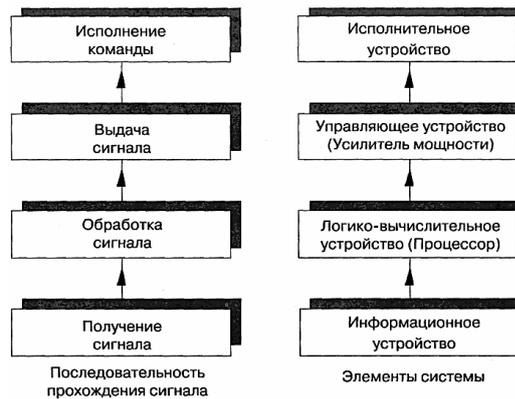


Рисунок 2.1 - Последовательность прохождения сигнала

Элементы системы изображаются с помощью условных графических обозначений, а представление о функциональном назначении элементов дает схема их соединений.

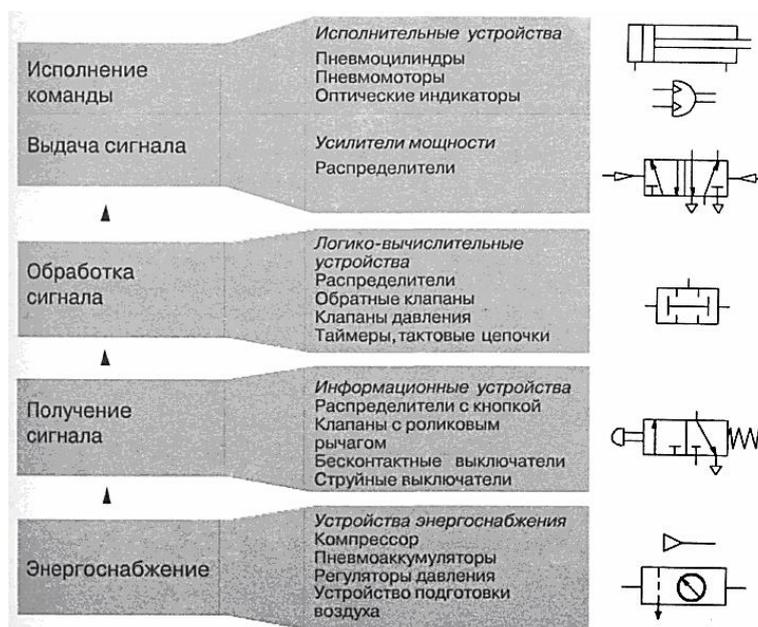


Рисунок 2.2 - Схема пневматической системы управления

Распределители могут применяться как входные элементы, логико-вычислительные элементы или усилители мощности. Пример взаимосвязи в пневматической системе элементов различного функционального назначения представлен на рис.2.3.

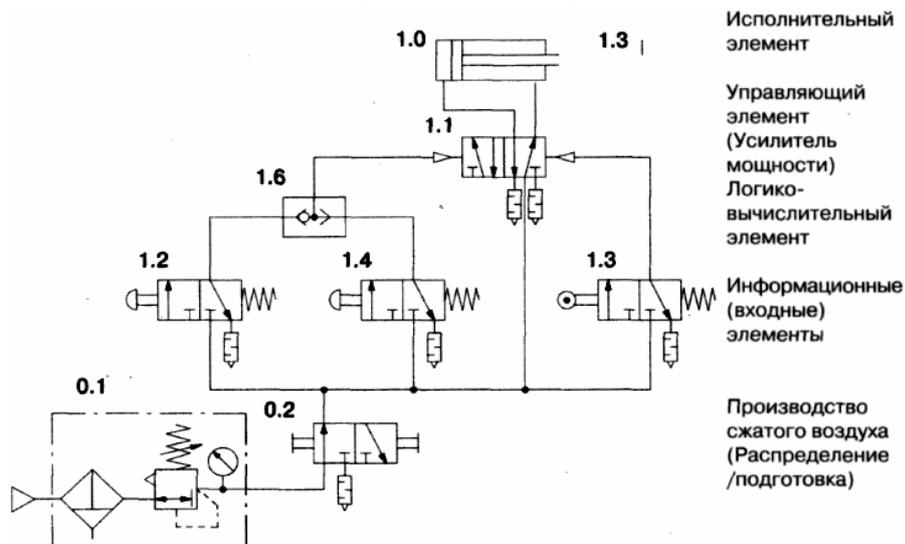


Рисунок 2.3 - Принципиальная схема пневматической системы управления

В общем случае управление цилиндром осуществляется с помощью управляющего распределителя. При этом выбор распределителя (число линий, число положений переключения и способ управления перемещениями) зависит от каждой конкретной задачи.

*Управление цилиндром одностороннего действия*

Пневмосистема состоит из:

- цилиндра одностороннего действия с пружинным возвратом,
- 3/2-распределителя с ручным управлением и пружинным возвратом,
- линии питания, присоединенной к 3/2-распределителю,
- пневмолнии между распределителем и цилиндром.

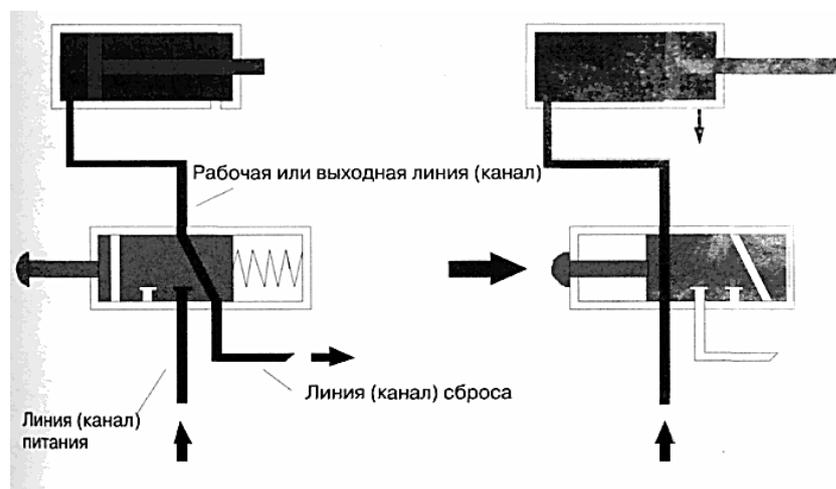


Рисунок 2.4 - Управление цилиндром одностороннего действия

### Управление цилиндром двустороннего действия

Управление цилиндром двустороннего действия осуществляется с помощью 4/2-распределителя с ручным управлением. Распределителем должен выдаваться или сниматься сигнал, если пневмокнопка соответственно нажата или отпущена.

Схема системы состоит из:

- цилиндра двустороннего действия,
- 4/2-распределителя с ручным управлением и пружинным возвратом,
- линии питания, присоединенной к 4/2-распределителю,
- пневмолиний между распределителем и цилиндром.

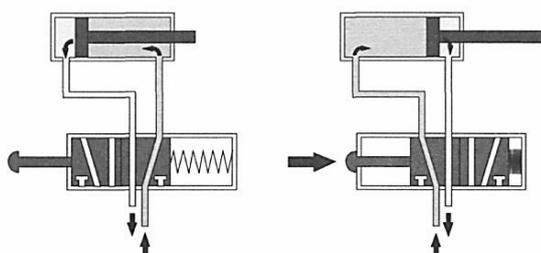


Рисунок 2.5 - Управление цилиндром двустороннего действия

### Условные обозначения и стандарты в области пневмоавтоматики

#### Энергоснабжение сжатым воздухом

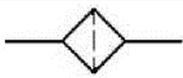
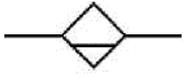
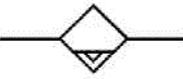
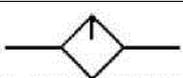
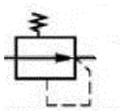
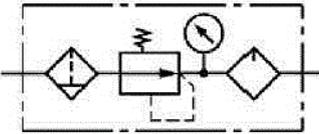
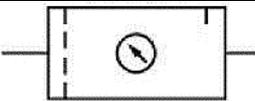
Условные обозначения подсистемы снабжения сжатым воздухом могут представлять как отдельные элементы, так и комбинации нескольких элементов (табл. 2.1). Если применяется присоединение линии питания общее для всех пневмоэлементов, то тогда возможно обозначение источника сжатого воздуха в виде упрощенного символа.

Устанавливает условные графические обозначения кондиционеров рабочей жидкости и рабочего газа, гидравлических и пневматических емкостей в схемах и чертежах всех отраслей промышленности ГОСТ 2.780-96 ЕСКД Обозначения условные графические. Кондиционеры рабочей среды, емкости гидравлические и пневматические, который соответствует ИСО 1219-91 Гидропривод, пневмопривод и устройства. Условные графические обозначения и схемы.

Таблица 2.1 - Условное обозначение элементов подсистемы энергоснабжения

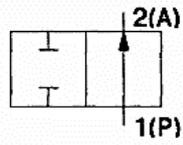
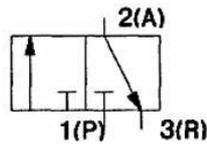
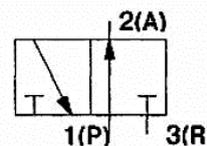
Питание	
Компрессор с постоянным рабочим объемом	
Аккумулятор пневматический (изображается только вертикально)	
Источник давления	

Продолжение табл. 2.1

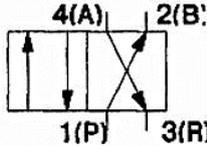
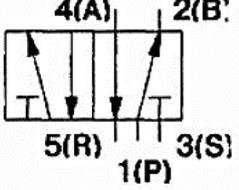
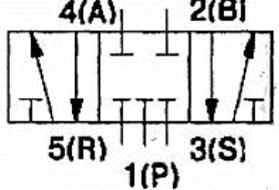
<b>Подготовка воздуха</b>	
Фильтр. Сепарация и фильтрация твёрдых частиц	
Влагоотделитель с ручным отводом конденсата	
Влагоотделитель с автоматическим отводом конденсата	
Маслораспылитель. Дозированное количество масла примешивается к распылителю	
Регулятор давления с разгрузочным вентиляционным каналом, регулируемый	
<b>Условные обозначения комбинации элементов</b>	
Блок подготовки воздуха, состоящий из воздушного фильтра, регулятора давления, манометра и маслораспылителя	
Упрощённое обозначение блока подготовки воздуха	

Для того, чтобы избежать ошибок при монтаже, все входы и выходы распределителя необходимо маркировать (табл. 2.2).

Таблица 2.2 - Линии (каналы) присоединения и позиции переключения распределителей

<p>Число линий присоединения</p> <p>Число позиций переключения</p> <p>2/2 – распределитель нормально открытый</p>	
<p>3/2 – распределитель нормально закрытый</p>	
<p>3/2 – распределитель нормально открытый</p>	

Продолжение табл. 2.2

4/2 – распределитель с протоками от 1(P) к 2(B) и от 4(A) к 3(R)	
5/2 – распределитель с протоками от 1(P) к 2(B) и от 4(A) к 5(R)	
5/3 – распределитель, перекрытый в средней позиции	

Линии присоединения распределителей могут обозначаться буквами или цифрами в соответствии с DIN ISO 5599-3. Обе возможности представлены в табл.2.3.

Таблица 2.3 - Рабочие и управляющие пневмолинии

DIN ISO 5599-3	Буквенная система	Линии (каналы) присоединения
1	P	Линии (каналы) питания сжатым воздухом
2,4	A, B	Рабочие (выходные) линии (каналы)
3,5	R,S	Линии (каналы) выхлопа
10	Y	Подаваемый сигнал закрывает проход от линии 1 к линии 2
12	Y,Z	Подаваемый сигнал соединяет линию 1 с линией 2
14	Z	Подаваемый сигнал соединяет линию 1 с линией 4
81,91	Pz	Пневматическое сервоуправление

В зависимости от системных требований к виду управления пневматическими распределителями применяют:

- мускульное управление,
- механическое управление,
- пневматическое управление,
- электрическое управление,
- комбинированное управление.

Условные обозначения способов управления должны соответствовать DIN ISO 1219 (табл. 2.4). Основные способы управления распределителем и возврата его в исходное состояние должны изображаться вместе с распределителем. Обычно условные обозначения способов управления изображаются с обеих сторон позиций переключения. Дополнительные способы управления, как, например, ручное управление, показываются отдельно.

Таблица 2.4 - Способы управления распределителями

Мускульное управление		Пневматическое управление	
Общее обозначение		Прямое направление, путём подачи давления	
С помощью кнопки		Непрямое направление (с предварительным усилием), путём подачи давления	
С помощью рычага			
С помощью рычага с фиксацией			
С помощью педали			
Механическое управление		Электрическое управление	
С помощью толкателя		С помощью электромагнита	
С помощью роликового рычага		С помощью двух электромагнитов	
С помощью ломающегося ролика			
С помощью пружины			
Комбинированное управление			
Непрямое электрическое и вспомогательное ручное управление с пневматическим усилием			

**Обратные клапаны** (табл. 2.5) служат в качестве базового элемента в целом ряде устройств. Известны обратные клапаны как с возвратной пружиной, так и без нее. Для того, чтобы поток воздуха мог проходить в прямом направлении в обратном клапане с пружиной, сила от давления газа должна быть больше, чем сила от пружины.

**Дроссели** (табл. 2.5) являются регулируемыми с возможностью дросселирования в обоих направлениях. Если дроссель изображен со стрелкой, то это означает, что дроссель является регулируемым. Стрелка не указывает на направление прохождения воздуха. В дросселе с обратным клапаном последний включается параллельно дросселю. В этом случае дросселирование осуществляется только в одном направлении.

Таблица 2.5 - Условное обозначение клапанов и дросселей

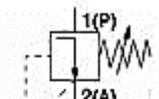
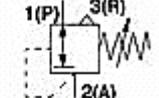
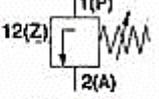
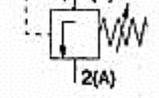
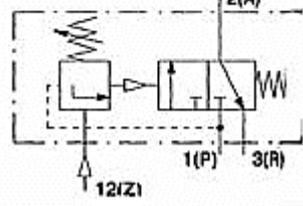
Обратный клапан и клапаны, построенные на его основе	Дроссели		
Обратный клапан		Дроссель	
Перекидной клапан (ИЛИ-элемент)		Дроссель с обратным клапаном	
Клапан двух давлений (И-элемент)			
Клапан быстрого выхлопа			

**Клапаны давления** предназначены для поддержания давления в пневматической системе или в отдельных ее частях. Клапаны давления являются обычно регулируемы за счет изменения усилия предварительного сжатия пружины. В зависимости от назначения различают следующие типы клапанов:

- редукционный клапан без разгрузки,
- редукционный клапан с разгрузкой,
- предохранительные клапаны.

Условное обозначение (табл. 2.6) представляет собой клапан давления с проточной частью, которая в исходном состоянии является или открытой, или закрытой. У редукционных клапанов проточная часть всегда открыта. У предохранительных клапанов проточная часть остается закрытой так долго, пока сила давления не достигнет настроенного значения силы предварительного натяжения пружины.

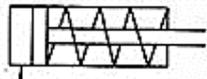
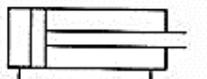
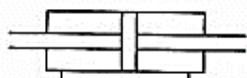
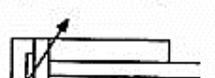
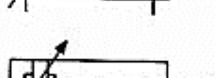
Таблица 2.6 – Условное обозначение клапанов давления

Регулируемый редукционный клапан без разгрузки	
Регулируемый редукционный клапан с разгрузкой	
Предохранительный клапан с внешней линией управления	
Предохранительный клапан с внутренней линией управления	
Клапан последовательности (Реле давления регулируемое)	

**Линейные исполнительные устройства** или цилиндры (табл.2.7) изображаются в виде, близком к их конструктивной форме.

Цилиндры одностороннего действия, двустороннего действия и бесштоковые служат основой других вариантов конструкций. Применение демпфирования в конце хода способствует увеличению долговечности, что достигается за счет снижения нагрузки в конечном положении путем уменьшения скорости движения поршня. Если условное обозначение цилиндра включает стрелку, то это означает, что демпфирование в конце хода является регулируемым.

Таблица 2.7 - Линейные исполнительные устройства

Цилиндр одностороннего действия	
Цилиндр двустороннего действия	
Цилиндр двустороннего действия с проходным штоком	
Цилиндр двустороннего действия с простым нерегулируемым демпфированием	
Цилиндр двустороннего действия с простым регулируемым демпфированием	
Цилиндр двустороннего действия с двусторонним регулируемым демпфированием	
Бесштоковый цилиндр с магнитной муфтой	

Различают **вращательные приводы** (табл. 2.8) в виде моторов с неограниченным углом поворота и поворотные приводы с ограниченным углом поворота.

Пневмомоторы вращаются обычно с очень большим числом оборотов, которое может быть постоянным или регулируемым. Поворотные цилиндры имеют или нерегулируемый, или регулируемый угол поворота, а также могут иметь демпфирование в зависимости от нагрузки или скорости вращения.

Таблица 2.8 - Исполнительные устройства вращательного действия

Пневмомотор с постоянным рабочим объемом и одним направлением течения воздуха	
Пневмомотор с регулируемым рабочим объемом и одним направлением течения	
Пневмомотор с регулируемым рабочим объемом и двумя направлениями течения воздуха	
Пневматический поворотный цилиндр	

**Задание:**

Управление прессом (рис. 2.6). осуществляется с помощью цилиндра двухстороннего действия. После нажатия пневмокнопки шток цилиндра выдвигается, обратный ход происходит сразу после достижения штоком конечного положения.

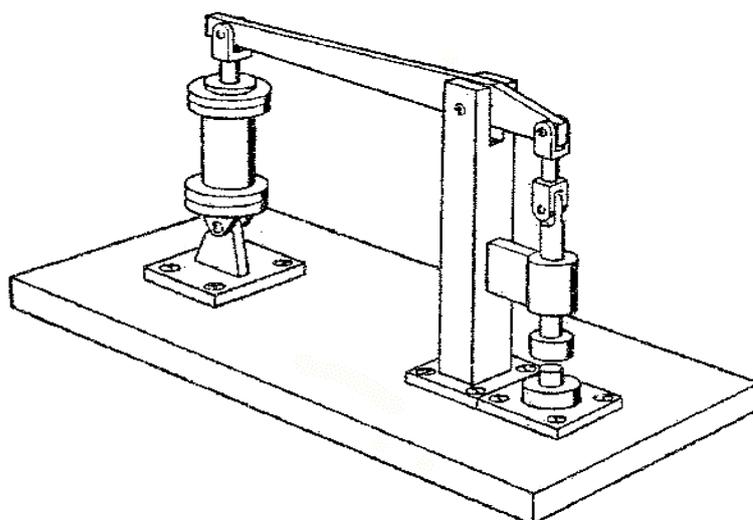


Рисунок 2.6 – Пресс

Таблица 2.9 – Приборы, входящие в состав пресса

Приборы	Наименование
0Z1	Блок подготовки сжатого воздуха
1A1	Цилиндр двустороннего действия
1V1	5/2-пневмораспределитель импульсный
1S1	3/2-пневмораспределитель с пневмокнопкой, нормально-закрытый
1S2	3/2-пневмораспределитель с роликовым рычагом, нормально-закрытый

**Порядок выполнения практической работы:**

1. Изучить краткие теоретические сведения.
2. Изучить принцип работы пресса (рис. 2.6) и приборы, входящие в состав пресса (табл. 2.9).
3. Разработать пневмосхему управления прессом.
4. Подготовить отчет.

**Отчет по работе должен содержать:**

1. Название работы.
2. Цель работы.
3. Результаты выполнения практической работы.

### **Вопросы для самопроверки:**

1. Перечислите этапы прохождения сигнала через элементы системы.
2. Объясните взаимосвязи в пневматической системе элементов различного функционального назначения.
3. Какие применяют виды управления пневматическими распределителями?
4. Для чего служат обратные клапаны? Что такое дроссель?
5. Для чего предназначены клапаны давления?
6. Опишите линейные исполнительные устройства и устройства вращательного действия.

### **Средства обучения:**

**1. Оборудование и материалы:** Раздаточный материал, Рабочая тетрадь

**2. Вербальные средства обучения:**

Шишмарёв, В. Ю. Автоматика : учебник для среднего профессионального образования / В. Ю. Шишмарёв. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2017. — 284 с. — (Профессиональное образование). — ISBN 978-5-534-05168-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/409032>

**3. Технические средства обучения (ТСО):** -

### Практическая работа № 3

**Тема:** Разработка и отладка релейной системы управления бойлером

**Цель работы:**

- уметь читать и составлять принципиальные схемы электрических, гидравлических и пневматических приводов несложного технологического оборудования;
- уметь составлять управляющие программы для программируемых логических контроллеров в SCADA – системе;
- уметь распознавать, классифицировать и использовать датчики, реле и выключатели в системах управления;
- уметь правильно эксплуатировать мехатронное оборудование.

#### Краткие теоретические сведения

##### 1. Структура системы управления SIMATIC S7

**SIMATIC S7-300** - это модульный программируемый контроллер, предназначенный для построения систем автоматизации низкой и средней степени сложности.

##### Основные характеристики

Контроллеры SIMATIC S7-300 имеют модульную конструкцию и могут включать в свой состав:

Модуль центрального процессора (CPU). В зависимости от степени сложности решаемой задачи в контроллерах могут быть использованы различные типы центральных процессоров (20 типов), отличающихся производительностью, объемом памяти, наличием или отсутствием встроенных входов-выходов и специальных функций, количеством и видом встроенных коммуникационных интерфейсов и т.д.

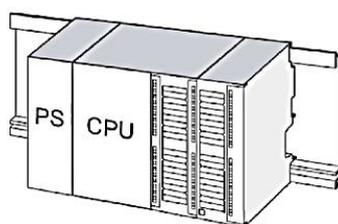
Модули блоков питания (PS), обеспечивающие возможность питания контроллера от сети переменного тока напряжением 120/230В или от источника постоянного тока напряжением 24/48/60/110В.

Сигнальные модули (SM), предназначенные для ввода-вывода дискретных и аналоговых сигналов с различными электрическими и временными параметрами.

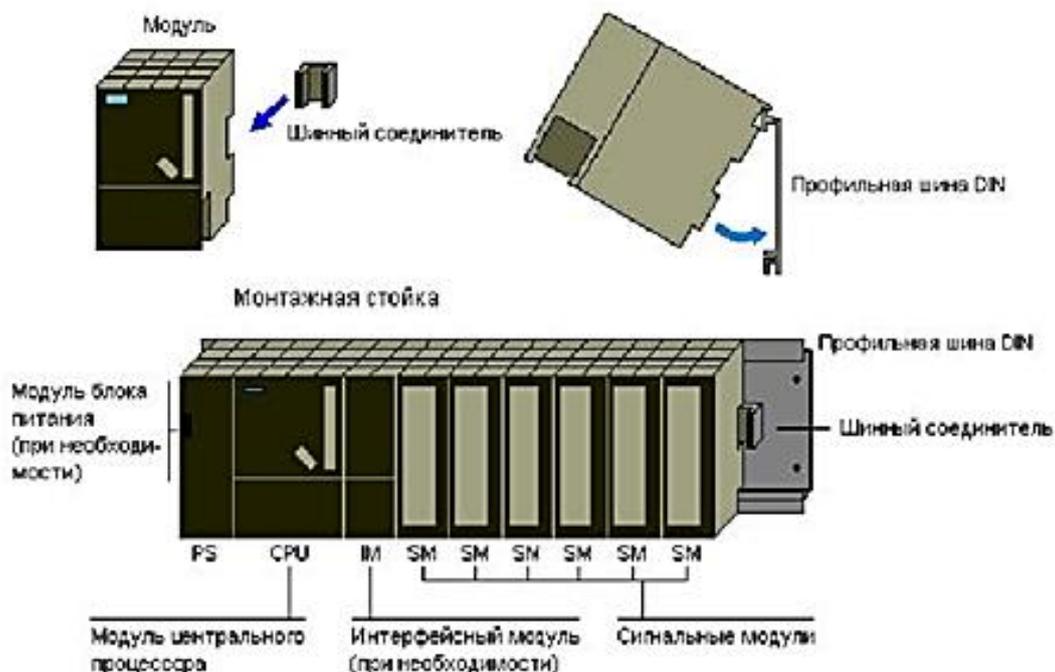
Коммуникационные процессоры (CP) для подключения к сетям PROFIBUS, Industrial Ethernet, AS-Interface или организации связи по PtP (point to point) интерфейсу.

Функциональные модули (FM), способные самостоятельно решать задачи автоматического регулирования, позиционирования, обработки сигналов.

Интерфейсные модули (IM), обеспечивающие возможность подключения к стойке с CPU стоек расширения ввода-вывода. Контроллеры SIMATIC S7-300 позволяют использовать в своем составе до 32 сигнальных и функциональных модулей, а также коммуникационных процессоров, распределенных по 4 монтажным стойкам. Все модули работают с естественным охлаждением.



## Конструкция S7-300



**Конструкция контроллера отличается высокой гибкостью и удобством обслуживания:**

- Все модули легко устанавливаются на профильную рейку SIMATIC и фиксируются в рабочем положении винтом
- Во все модули (кроме модулей блоков питания) встроены участки внутренней шины контроллера
- Соединение этих участков выполняется шинными соединителями, устанавливаемыми на тыльной стороне корпуса
- Наличие фронтальных соединителей, позволяющих производить замену модулей без демонтажа внешних соединений и упрощающих выполнение операций подключения внешних цепей модулей
- Подключение внешних цепей через фронтальные соединители с контактами под винт или контактами-защелками. Механическое кодирование фронтальных соединителей, исключающее возможность возникновения ошибок при замене модулей
- Применение модульных и гибких соединителей SIMATIC TOP Connect, существенно упрощающих монтаж шкафов управления
- Единая для всех модулей глубина установки. Все кабели располагаются в монтажных каналах модулей и закрываются защитными дверцами

- Произвольный порядок размещения модулей в монтажных стойках. Фиксированные места должны занимать только блоки питания, центральные процессоры и интерфейсные модули.

### **Функциональные особенности**

- Удобная для обслуживания конструкция и работа с естественным охлаждением.
- Свободное наращивание возможностей при модернизации системы.
- Встроенный переключатель режимов работы.
- Встроенный MPI интерфейс со скоростью передачи данных 187.5Кбит/с. Может быть использован для построения простейшей сети с подключением до 16 центральных процессоров и передачей глобальных данных.
  - Загружаемая память в виде микро карты памяти NVFlash-EEPROM (MMC) емкостью до 8Мбайт: сохранение основной программы, данных конфигурации, всего проекта, также любых файлов, хранимых на внешних носителях.
  - Необслуживаемое сохранение резервной копии данных: при перебоях в питании в микро карту памяти записываются состояния флагов, таймеров, счетчиков и содержимое блоков данных (работа без буферной батареи).
  - Парольная защита: обеспечивает защиту программы от несанкционированного доступа, попыток копирования и модификации программы.
  - Диагностический буфер: в буфере сохраняется 100 последних сообщений об отказах и прерываниях. Содержимое буфера используется для анализа причин, вызвавших остановку центрального процессора.
  - Часы реального времени: все диагностические сообщения могут снабжаться отметками даты и времени.
  - Встроенные коммуникационные функции: PG/OP функции связи, стандартные и расширенные (только сервер) S7-функции связи.
  - Дополнительный встроенный интерфейс ведущего/ ведомого устройства PROFIBUS DP (в CPU 31xC-2 DP) со скоростью обмена данными до 12 Мбит/с.
  - Возможность использования распределенных структур ввода-вывода.

### **Набор поддерживаемых функций**

Контроллеры SIMATIC S7-300 оснащены широким набором функций, позволяющих в максимальной степени упростить процесс разработки программы, ее отладки, снизить затраты на обслуживание контроллера в процессе его эксплуатации:

- Высокое быстроедействие и поддержка математики с плавающей запятой, позволяющие выполнять эффективную обработку данных.
- Удобный интерфейс для настройки параметров: для настройки параметров всех модулей используется единый набор инструментальных средств с общим интерфейсом.
- Человеко-машинный интерфейс. Функции обслуживания человеко-машинного интерфейса встроены в операционную систему контроллера. Эти функции позволяют существенно упростить программирование.

- Диагностические функции, встроенные в операционную систему контроллера. С их помощью осуществляется непрерывный контроль функционирования системы, и выявляются все возникающие отказы. Все диагностические сообщения с отметками даты и времени накапливаются в кольцевом буфере для последующего анализа.

## 2. Система ввода-вывода

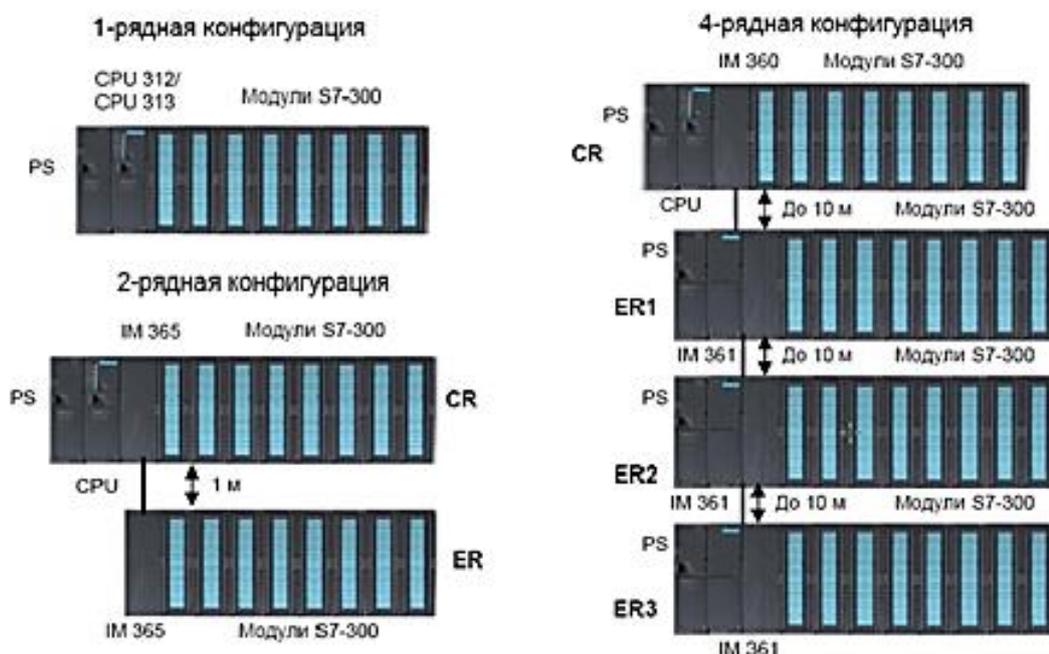
Система ввода-вывода программируемого контроллера S7-300 может включать в свой состав две части: систему **локального** и систему **распределенного** ввода-вывода.

Система локального ввода-вывода образуется модулями, устанавливаемыми непосредственно в монтажные стойки контроллера. Система распределенного ввода-вывода включает в свой состав станции распределенного ввода-вывода и приборы полевого уровня, подключаемые к контроллеру через сети PROFIBUS-DP или AS-Interface.

Системы локального ввода-вывода программируемых контроллеров S7-300 с CPU 313C/ CPU 314 или более мощными процессорами могут включать в свой состав до 32 сигнальных, функциональных и коммуникационных модулей. Все модули устанавливаются в монтажные стойки контроллера, функции которых выполняют профильные шины S7-300.

- В состав системы может входить одна базовая (CR) и до трех стоек расширения (ER). В каждой стойке может размещаться до 8 сигнальных, функциональных и коммуникационных модулей. В стойке CR устанавливается центральный процессор.

- Соединение стоек осуществляется с помощью интерфейсных модулей. Каждая стойка снабжается собственным интерфейсным модулем. Применение интерфейсных модулей IM 365 позволяет подключать к базовой стойке одну стойку расширения, удаленную на расстояние не более 1 м. Интерфейсные модули IM 360 и IM 361 позволяют подключать к базовой стойке до 3 стоек расширения. Расстояние между стойками в этом случае может достигать 10 м.



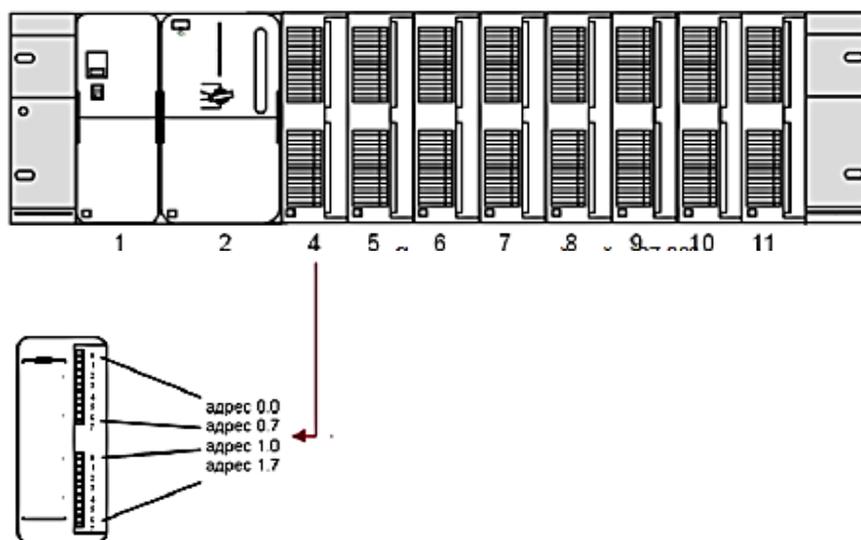
## Подготовка SIMATIC S7 к запуску программы

### Инструкция для монтажа S7-300

1. Монтируйте профильную шину.
2. Установите источник питания.
3. Установите шинный соединитель для CPU и монтируйте модуль.
4. Установите шинные соединители периферийных модулей и монтируйте модули.
5. Установите фронтштекер, описание и номера слотов.
6. Провести разводку модулей (ист. питания, CPU, периферийные модули).

### Адресация в S7-300

Нумерация слотов (установочных мест) облегчает определение адресов модулей ввода-вывода.



Слоты в монтажной стойке S7-300 имеют логические номера. Слот для CPU имеет всегда номер 2. Номер 3 резервируется за модулем IM.

Размещение модулей DI/DO начинается со слота № 4 и до слота № 11. Для каждого слота под установку DI/DO резервируется 4 байта адресов (достаточно для отображения состояний всех дискретных входов или выходов 32-канального модуля).

Базовый (начальный) адрес модуля DI/DO определяется по формуле:

$$\text{БА} = [\text{№слота} - 4] \times 4 + \text{№стойки} \times 32,$$

где №слота  $\in$  4..11, №стойки  $\in$  0..3

### Примеры адресации:

- Если в слот № 5 установлен модуль ввода-вывода дискретных сигналов (16DI/16DO), то его 8 первых входных каналов будут иметь адреса I4.0 ... I4.7, а первые 8 выходных каналов - адреса Q4.0 ... Q4.7.
- Первые 8 каналов модуля вывода дискретных сигналов, установленного в слот № 6, будут иметь адреса Q8.0 ... Q8.7.

### Пример адресации модулей DI/DO (S7-300, 2-х стойки)

Слот	3	4	5	6	7	8	9	10	11
Источник питания	IM (Приемник)	32.0	36.0	40.0	44.0	48.0	52.0	56.0	60.0
		DO 35.7	DO 39.7	DO 43.7	DO 47.7	DO 51.7	DO 55.7	DO 59.7	DO 63.7
Стойка 1									
Источник питания и CPU	IM (Передача)	0.0	4.0	8.0	12.0	16.0	20.0	24.0	28.0
		DO 3.7	DO 7.7	DO 11.7	DO 15.7	DO 19.7	DO 23.7	DO 27.7	DO 31.7
Стойка 0									

### 3. Инструкция для ввода в эксплуатацию SIMATIC S7

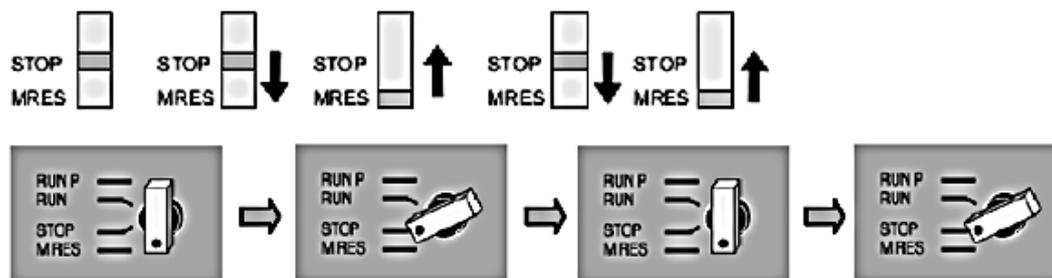
1. Включить питание.
2. Сбросить CPU.
3. Провести старт.
4. Выполнить визуальный контроль индикаторов на CPU, источнике питания и модулях.
5. Подключить программатор и запустить программу SIMATIC Manager.
6. Проверить входы/выходы с помощью утилиты HW Config.
7. Выполнить загрузку программы в CPU.

### Сброс CPU

При сбросе CPU происходит очистка программы и данных в рабочей памяти (RAM) и области системной памяти. Затем проводится тест аппаратуры и, если модуль памяти (FLASH-память или MMC) вставлен в CPU и имеет программу, то программа и конфигурационные данные из модуля памяти копируются в рабочую память.

### Выполнение сброса

Установите переключатель работы в режим "STOP". На CPU должен гореть желтый светодиод "STOP".



Переведите переключатель режимов работы в положение **MRES** и удерживайте его там не менее 3 секунд (пока желтый светодиод "STOP" не мигнет 2 раза). Отпустите переключатель (он перейдет в позицию "STOP") и сразу (максимум 2 секунды) снова поверните его в положение **MRES**. Если светодиод "STOP" мигает быстро (частота 2Гц), то CPU сброшен. Снова отпустите переключатель.

## Выполнение старта

Поверните переключатель режимов работы в RUN. Загорится зеленый светодиод "RUN", а желтый светодиод "STOP" погаснет.

При новом старте удаляются области отображения процесса (PII, PIQ) и вся системная память, кроме тех областей памяти, которые назначены как сохраняемые, а затем начинается циклическая работа CPU.

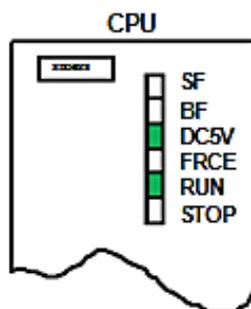
Примечание: во всех новых моделях CPU переключатель режимов не имеет положения RUN-P. В ранних моделях CPU изменение программы или любых данных без остановки CPU было возможно только в позиции RUN-P, а в позиции RUN - невозможно.

## Режим RUN

В режиме RUN операционная система CPU выполняет циклический алгоритм, который предусматривает системные процедуры передачи данных из внутренней памяти PIQ (области отображения выходов) на модули вывода и передачи сигналов с модулей ввода в системную память PII (отображение входов). Затем управление передается на первую команду программы пользователя.



## Визуальный контроль индикаторов в S7-300



**SF** - Суммарная ошибка, программная ошибка в CPU или ошибка от модуля с внутренней диагностикой.

**BF** - Ошибка сети PROFIBUS DP (для модулей с дополнительным интерфейсом DP). Светится постоянно - при коротком замыкании кабеля или неправильной конфигурации. Мигает - при отсутствии связи с каким-либо участником по PROFIBUS DP.

**DC5V** - Индикация о наличии 5V.

**FRCE** - Индикация об установленном режиме Force (по меньшей мере, один вход или выход управляется принудительно в соответствии с таблицей значений).

**RUN** - Run-режим: мигает при старте CPU, светится постоянно в Run-режиме.

**STOP** - Stop-режим: медленно мигает, если требуется сброс (например, при отсутствии модуля MMC); быстро мигает во время сброса, светится постоянно в Stop-режиме.

## Подключение программатора к CPU



Если Вы имеете USB-адаптер (6ES7972-0CB20-0XA0), то в качестве устройства программирования может использоваться любой персональный компьютер или ноутбук, который удовлетворяет требованиям к установке пакета STEP7 (WIN2000 PROF/ XP PROF, 512Mb RAM). Начиная с версии STEP 7 V5.4 SP3, поддерживается работа с Windows Vista Ultimate/Business. Версия STEP 7 V5.5 SP1 работает также с MS Windows 7 32/64-Bit.

USB-адаптер поддерживает протоколы MPI (S7-300/400), PPI (S7-200) и PROFIBUS-DP. Максимальная скорость передачи 1.5Mbps.

Для связи с S7-CPU Вы можете также использовать специальные коммуникационные модули SIMATIC NET, например, CP 5512 (PC-CARD) или CP 5611 (PCI-CARD).

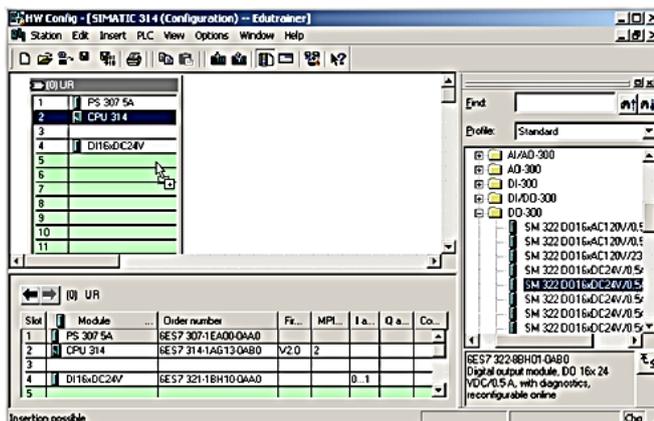
## 4. Конфигурирование и параметрирование SIMATIC S7

Если Вы знаете состав модулей станции (в том числе заказные номера, расположенные на фронтальной панели модулей), то проектирование задачи автоматизации мы рекомендуем начать с конфигурирования станции. Преимущество этого подхода состоит в том, что редактор конфигурирования аппаратуры в STEP 7 отображает возможные адреса сигналов и позволяет сразу задать для них символьные адреса.

При конфигурировании аппаратуры Вы не только можете определять адреса, но и можете также изменять параметры и свойства модулей. Например, если вы хотите работать с несколькими CPU, то вы должны согласовывать адреса MPI этих CPU.

### Конфигурирование:

1. Выделить новую станцию. В правом подокне выделить объект "Hardware".
2. Открыть объект "Hardware". Откроется окно "HW Config".
3. В окне "HW Config" открыть каталог блоков View -> Catalog.
4. Сначала выбрать носитель модулей (rack) и разместить его в пустом окне конфигуратора (кликнуть мышью) Затем выбрать остальные модули и разместить их в соответствующие слоты корзины. Как минимум, укажите только CPU.



Примечание: при конфигурировании станции S7-300 слот №3 резервируется для интерфейсного модуля (IM).

### Обновление каталога элементов

Для новых модулей, например CPU или I/O-модулей, которых нет в каталоге элементов, Вы можете выполнить команду обновления каталога элементов Options -> Install HW Updates. Обновление возможно через Internet (стандартная страница <http://www.siemens.com/automation/step7-hwconfig2>), либо через каталог файлов на жестком диске. При выборе Internet создается каталог C:\HWUpdates, который в дальнейшем можно использовать для обновления HW-библиотеки на компьютерах без подключения к Internet.

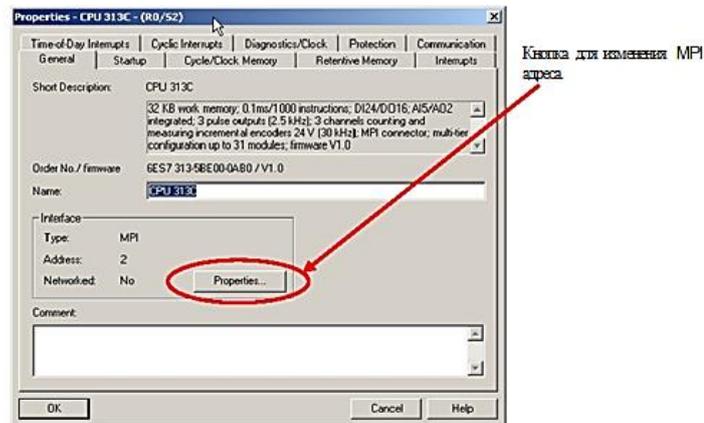
### Параметрирование станции

Вы назначаете модулям параметры, чтобы отказаться от системных настроек модулей устанавливаемых при сбросе CPU (например, адресация модулей), или еще при производстве модуля (MPI адрес).

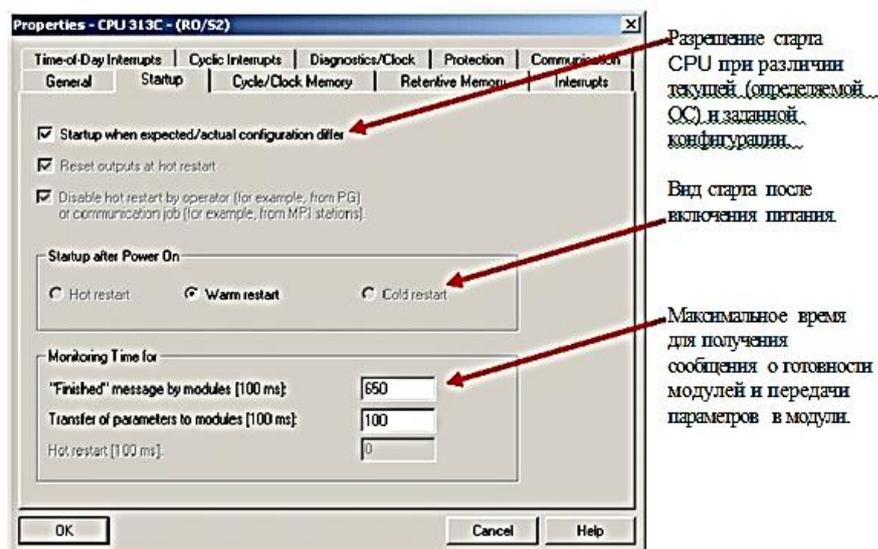
### Параметрирование CPU

Выбрать модуль CPU и открыть окно свойств (Properties).

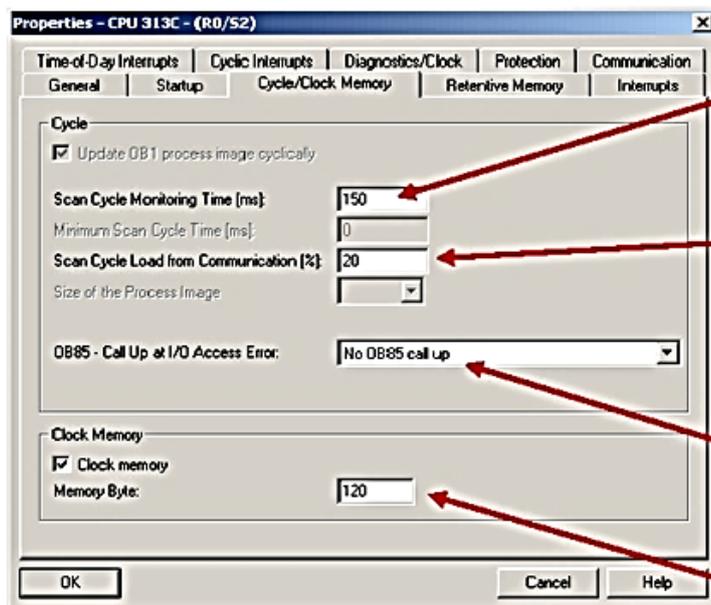
- Вкладка **General** (общие)



- Вкладка **Startup** (запуск)



- Вкладка **Cycle/Clock Memory** (Цикл/ Байт синхронизации)



Максимальное время цикла (150 ms)/Если это время превышено, CPU переходит в STOP (при отсутствии блока ошибки OB80).

Коэффициент (K) увеличения времени цикла за счет коммуникационной нагрузки (Н%)  

$$K = \frac{100}{100 - Н\%}$$

Реакция CPU на ошибки доступа при работе с областями отображения PII/PIQ.

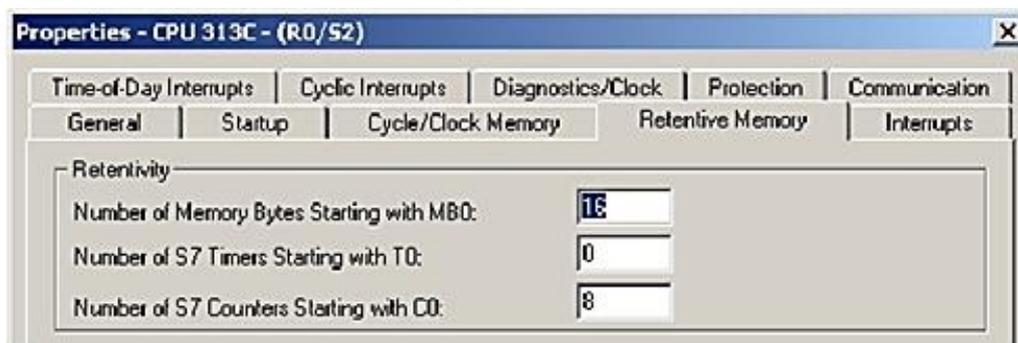
Байт синхронизации.

Байт синхронизации (Clock Memory) задается в области М (Memory Byte) через указание его адреса. При задании синхробайта, операционная система CPU изменяет 8 бит этого байта с различной частотой (имеем 8 генераторов фиксированной частоты).

Синхробит	7	6	5	4	3	2	1	0
Частота (Гц)	0.5	0.62	1	1.25	2	2.5	5	10
Период (с)	2	1.6	1	0.8	0.5	0.4	0.2	0.1

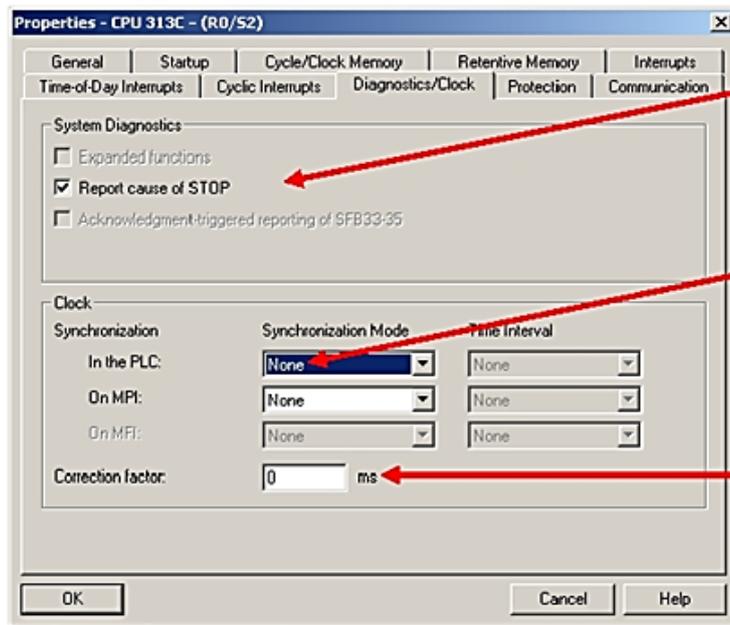
**Пример:** Мигание лампы с частотой 1 Гц => M10.5(меняем байт синхронизации на 10)

- Вкладка **Retentive Memory** (Сохраняемая память)



Вы определяете диапазон флагов, таймеров, счетчиков (начиная с 0), содержимое которых сохраняется при выключении питания и восстанавливается при старте CPU.

- Вкладка **Diagnostic/Clock** (Диагностика/Часы)

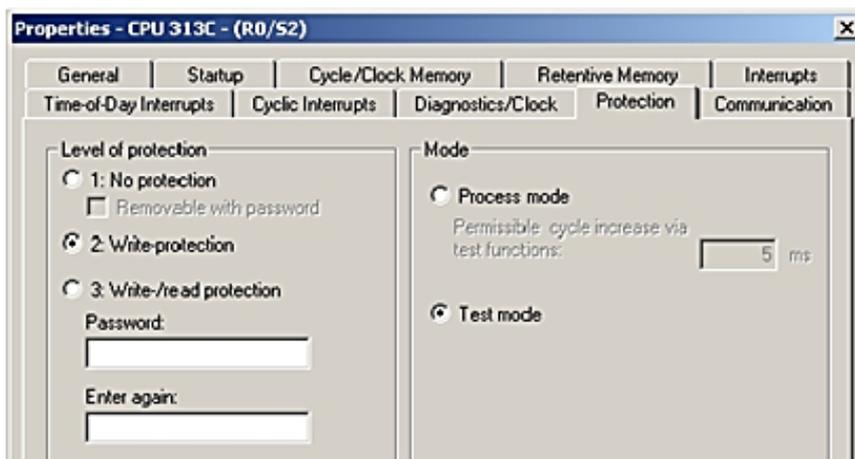


Разрешение CPU для передачи сообщения на PG/OP при переходе в STOP.

Настройки для синхронизации часов для нескольких CPU, работающих в сети MPI или между модулями внутри станции.

Показатель коррекции для учета погрешности часов. Пример: Если часы за 24 часа убегают на 3 секунды, то это может быть скорректировано показателем "-3000ms".

- Вкладка **Protection** (Защита)



Вы можете сделать доступ к CPU на запись или чтение/запись, зависящим от ввода правильного пароля. С помощью команды меню *PLC -> Access Rights* Вы можете вызвать диалоговое окно "Enter Password (Введите пароль)".

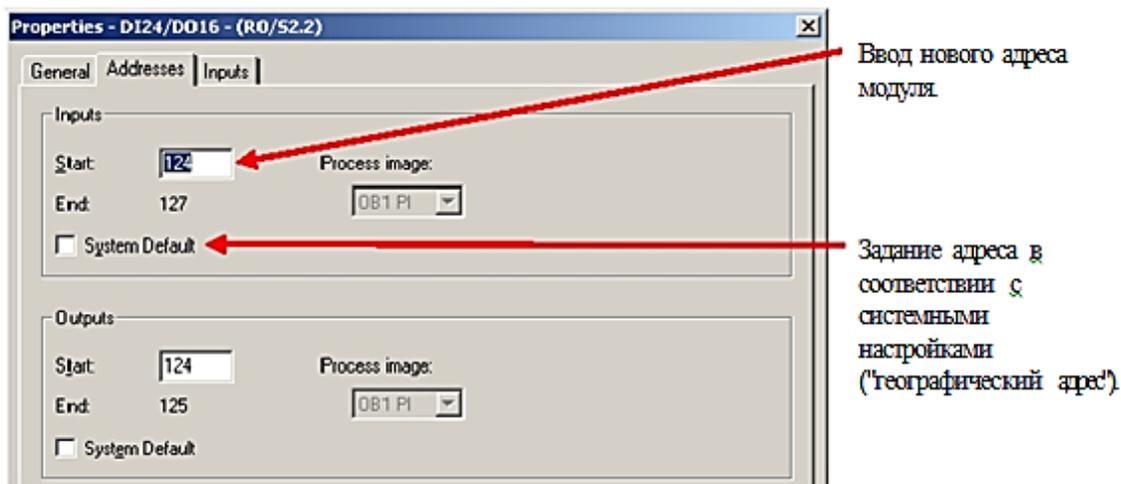
В режиме **Process mode** тестовые функции, такие как статус программы или управление/наблюдение переменных, ограничены временем, что не позволяет увеличить текущий цикл CPU на установленный предел.

В режиме **Test mode** все тестовые функции могут использоваться без ограничений, даже если они вызывают существенное увеличение времени цикла.

### Параметрирование модулей ввода/вывода

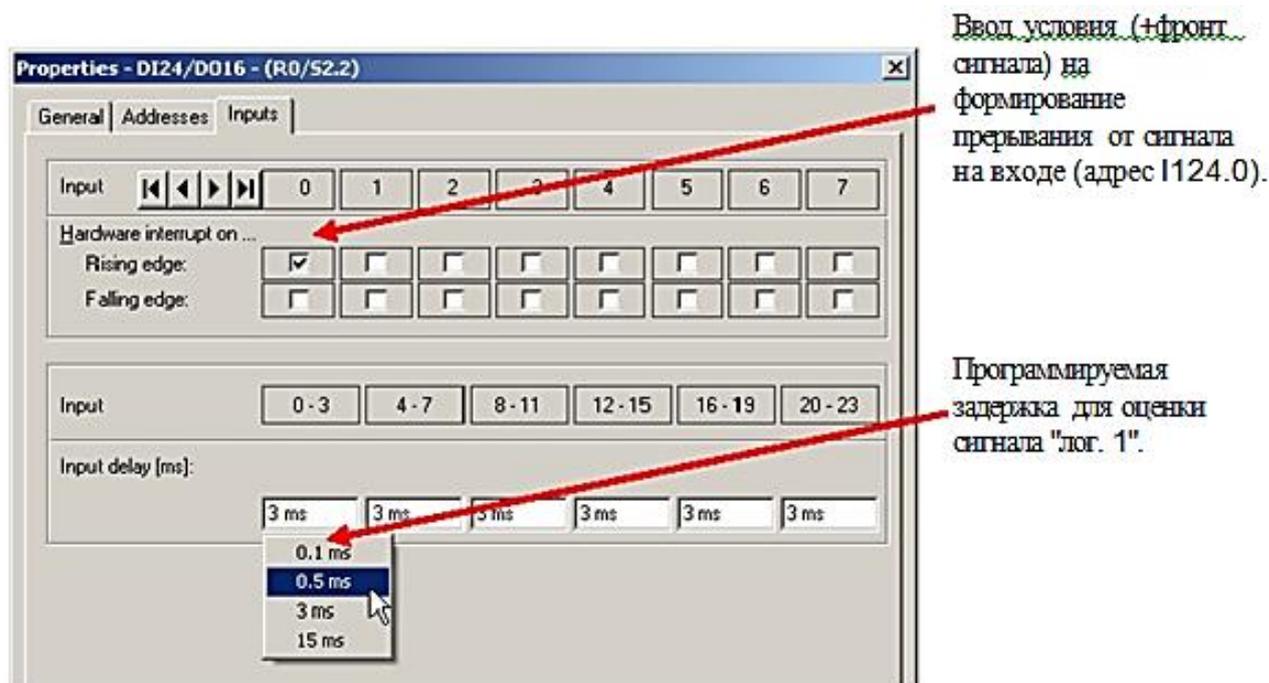
Выбрать соответствующий модуль и открыть окно свойств (Properties). В открывшемся окне Вы можете задать адрес модуля и, если позволяет модуль, задать, например, условия аппаратных / диагностических прерываний.

## Параметрирование модуля DI/DO (вкладка Addresses)



(start=0, end=1)

## Параметрирование модуля DI/DO (вкладка Inputs)



## Сохранение и загрузка заданной конфигурации

С помощью команды меню *Station->Save* Вы можете сохранить текущую конфигурацию в текущем проекте (без генерации системных блоков данных).

Когда Вы выбираете команду *Station->Save and Compile* или нажимаете на соответствующую кнопку в панели инструментов, конфигурация и назначенные параметры сохраняются также в системных блоках данных (System Data).



Вы выбираете команду меню *PLC ->Download* или нажимаете на соответствующую кнопку в панели инструментов, чтобы загрузить выбранную конфигурацию в PLC (PLC должен

быть в режиме "STOP"). 

Вы можете выбрать команду меню *Station -> Consistency Check* , чтобы проверить результат конфигурирования на ошибки.

### Чтение текущей конфигурации и параметров из CPU



Через меню *PLC -> Upload* или с помощью соответствующей кнопки Вы можете прочитать текущую конфигурацию S7-станции.

Фактическая конфигурация, считанная из контроллера, размещается в выбранном проекте как новая станция.

*Примечание.* При чтении реальной конфигурации модули могут быть определены не полностью. По этой причине Вы должны проверить конфигурацию и, если необходимо, указать точный тип существующих модулей. Для того чтобы сделать это, выберите модуль, а затем выберите команду *Options -> Specify Module*.

### Диагностика аппаратуры

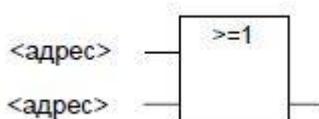


Для чтения диагностической информации об аппаратуре кликните на соответствующую кнопку. В результате откроется окно с online информацией. Вы получите тот же результат через online-функцию *PLC-> Hardware Diagnostics*.

## 5. Программирование на STEP 7

### 5.1. $\geq 1$ : Логическая операция OR (ИЛИ)

*Обозначение*



Параметры	Тип данных	Область памяти	Описание
<адрес>	BOOL	I, Q, M, T, C, D, L	Адрес определяет состояние опрашиваемого бита .

### Описание

В случае инструкции ИЛИ Вы можете опросить состояния сигналов по двум или более адресам, указанным на входах блока ИЛИ.

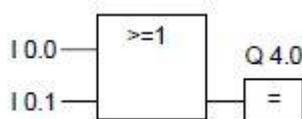
Если состояние сигнала хотя бы по одному из адресов равно 1, то условие удовлетворяется и инструкция выдает результат 1. Если состояние сигнала по всем адресам равно 0, то условие не удовлетворяется и инструкция дает результат 0.

Если инструкция ИЛИ является первой в цепи логических операций, то она сохраняет результат опроса состояния сигнала в бите RLO. Каждая инструкция ИЛИ, не являющаяся первой в цепи логических операций, комбинирует результат опроса состояния сигнала со значением, хранящимся в бите RLO. Эти значения комбинируются в соответствии с таблицей истинности для И.

### Биты слова состояния

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Записывает	-	-	-	-	-	X	X	X	1

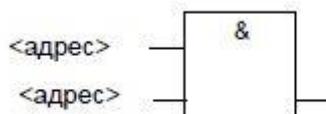
### Пример



Выход Q4.0 установлен, когда равно 1 состояние сигнала на входе I0.0 ИЛИ на входе I0.1.

## 5.2. & : Логическая инструкция И

### Обозначение



Параметры	Тип данных	Область памяти	Описание
<адрес>	BOOL	I, Q, M, T, C, D, L	Адрес определяет состояние опрашиваемого бита .

### Описание

В случае инструкции И, Вы можете опросить состояния сигналов по двум или более адресам, указанным на входах блока И.

Если состояния сигналов по всем адресам равны 1, то условие удовлетворяется и операция выдает результат 1. Если состояние сигнала хотя бы по одному адресу равно 0, то условие не удовлетворяется и инструкция выдает результат 0.

Если инструкция И является первой в цепи логических операций, то она сохраняет результат опроса состояния сигнала в бите RLO.

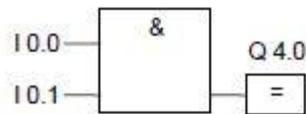
Каждая инструкция И, не являющаяся первой в цепи логических операций, комбинирует

результат опроса состояния сигнала со значением, хранящимся в бите RLO. Эти значения комбинируются в соответствии с таблицей истинности для И.

*Биты слова состояния*

	<b>BR</b>	<b>CC 1</b>	<b>CC 0</b>	<b>OV</b>	<b>OS</b>	<b>OR</b>	<b>STA</b>	<b>RLO</b>	<b>FC</b>
Записывает	-	-	-	-	-	X	X	X	1

*Пример*



Выход Q4.0 установлен, когда равны 1 состояния сигналов на входах I0.0 и I0.1.

### 5.3. Логические инструкции И перед ИЛИ и ИЛИ перед И

*Описание*

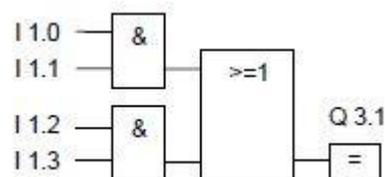
В случае инструкции И-перед-ИЛИ результат инструкции определяется в соответствии с таблицей истинности для ИЛИ.

В случае логической инструкции И-перед-ИЛИ состояние сигнала равно 1, если удовлетворяется хотя бы одна логическая операция И.

*Биты слова состояния*

	<b>BR</b>	<b>CC 1</b>	<b>CC 0</b>	<b>OV</b>	<b>OS</b>	<b>OR</b>	<b>STA</b>	<b>RLO</b>	<b>FC</b>
Записывает	-	-	-	-	-	X	X	X	1

*Пример*



Состояние сигнала на выходе Q3.1 равно 1, когда удовлетворяется хотя бы одна операция И.

Состояние сигнала на выходе Q3.1 равно 0, когда не удовлетворяется ни одна операция И.

*Описание*

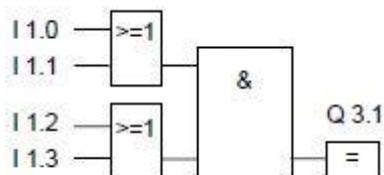
В случае инструкции ИЛИ-перед-И результат инструкции определяется в соответствии с таблицей истинности для И.

В случае логической инструкции ИЛИ-перед-И состояние сигнала равно 1, когда удовлетворяются все логические инструкции ИЛИ.

*Биты слова состояния*

	<b>BR</b>	<b>CC 1</b>	<b>CC 0</b>	<b>OV</b>	<b>OS</b>	<b>OR</b>	<b>STA</b>	<b>RLO</b>	<b>FC</b>
Записывает	-	-	-	-	-	X	X	X	1

*Пример*

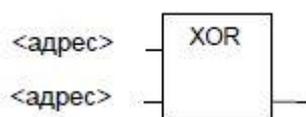


Состояние сигнала на выходе Q3.1 равно 1, когда удовлетворяются обе логические инструкции ИЛИ.

Состояние сигнала на выходе Q3.1 равно 0, когда по крайней мере одна из логических операций ИЛИ не удовлетворяется.

**5.4. XOR : Логическая инструкция исключающее ИЛИ**

*Обозначение*



Параметры	Тип данных	Область памяти	Описание
<адрес>	BOOL	I, Q, M, T, C, D, L	Адрес определяет состояние опрашиваемого бита .

*Описание*

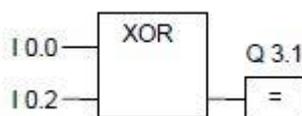
В случае инструкции исключающее ИЛИ результат инструкции определяется в соответствии с таблицей истинности для исключающего ИЛИ

В случае инструкции исключающее ИЛИ состояние сигнала равно 1, когда состояние сигнала одного из двух указанных адресов равно 1.

*Биты слова состояния*

	<b>BR</b>	<b>CC 1</b>	<b>CC 0</b>	<b>OV</b>	<b>OS</b>	<b>OR</b>	<b>STA</b>	<b>RLO</b>	<b>FC</b>
Записывает	-	-	-	-	-	X	X	X	1

*Пример*



Состояние сигнала на выходе Q3.1 равно 1 когда равно 1 состояние сигнала ИЛИ на входе I0.0 ИЛИ на входе I0.2 (но не на обоих одновременно).

## 5.5. XOR : Логическая инструкция исключающее ИЛИ

Обозначение



Описание

Инструкция Инверсия двоичного входа инвертирует RLO.

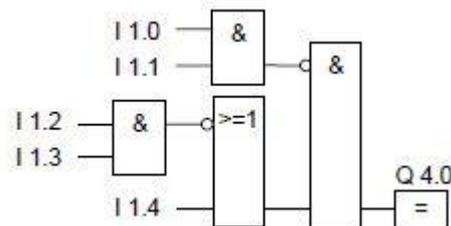
При отрицании результата логической инструкции Вы должны помнить следующие правила:

- Если инвертируется результат логической инструкции на первом входе блока И или ИЛИ, то логического сопряжения с RLO не производится.
- Если инвертируется результат логической инструкции не на первом входе блока ИЛИ, то вся логическая операция перед этим входом включается в логическую операцию ИЛИ.
- Если инвертируется результат логической инструкции не на первом входе блока И, то вся логическая операция перед этим входом включается в логическую операцию И.

Биты слова состояния

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Записывает	-	-	-	-	-	-	1	X	-

Пример



Выход Q4.0 равен 1 когда:

- состояния сигналов на I1.0 И I1.1 не равны одновременно 1
- И состояния сигналов на I1.2 И I1.3 не равны одновременно 1
- ИЛИ состояние сигнала на I1.4 не равно 1.

## 5.6. = : Инструкция присвоения

Обозначение



Параметры	Тип данных	Область памяти	Описание
<адрес>	BOOL	I, Q, M, D, L	Адрес указывает бит, которому присваивается значение результата цепи логических операций.

### Описание

Инструкция Присвоение выдает результат логической инструкции. Этот блок в конце логической инструкции имеет состояние 1 или 0 в соответствии со следующими критериями:

- Сигнал на выходе равен 1, когда удовлетворяются условия логической инструкции перед выходным блоком
- Сигнал на выходе равен 0, когда условия логической инструкции перед выходным блоком не удовлетворяются.

Эта логическая операция FBD присваивает состояние сигнала выходу, указанному в качестве адреса. Если условия логической инструкции FBD удовлетворяются, то значение сигнала 1 присваивается адресу, указанному в инструкции. В противном случае состояние сигнала равно 0. На инструкцию Присвоение оказывает влияние главное управляющее реле (MCR).

За более подробной информацией о функциях MCR обратитесь к соответствующему разделу.

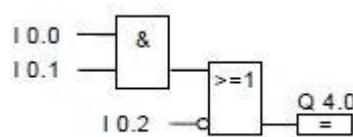
Блок Присвоение можно поместить только справа в конце цепочки логических операций. Однако, Вы можете использовать несколько таких блоков.

Вы можете создать инверсное присвоение с помощью инструкции Инверсия двоичного входа.

### Биты слова состояния

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Записывает	-	-	-	-	-	0	X	-	0

### Пример



Состояние сигнала на выходе Q4.0 равно 1, когда: равны 1 состояния сигналов на входах I0.0 и I0.1, ИЛИ I0.2 равен 0.

## 5.7. R: Сброс бита

### Обозначение



Параметры	Тип данных	Область памяти	Описание
<адрес>	BOOL TIMER COUNTER	I, Q, M, T, C, D, L	Адрес указывает бит который будет сброшен

### Описание

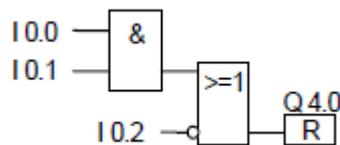
Инструкция Сбросить бит выполняется только тогда, когда RLO равен 1. Если RLO равен 0, эта инструкция сбрасывает указанный адрес в 0. Если RLO равен 1, эта инструкция сбрасывает указанный адрес в 0. Если RLO равен 0, то инструкция не влияет на указанный адрес, который остается неизменным.

На инструкцию Сбросить бит оказывает влияние Главное управляющее реле (MCR). За более подробной информацией о функциях MCR обратитесь к соответствующему разделу.

### Биты слова состояния

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Записывает	-	-	-	-	-	0	X	-	0

### Пример



Состояние сигнала на выходе Q4.0 сбрасывается в 0 только тогда, когда:

- Равны 1 состояния сигналов на входе I0.0 И I0.1
- ИЛИ состояние сигнала на входе I0.2 равно 0

Если RLO этих веток равен 0, то состояние сигнала на выходе Q4.0 не меняется.

## 5.8. S : Установка бита

### Обозначение



Параметры	Тип данных	Область памяти	Описание
<адрес>	BOOL	I, Q, M, D, L	Адрес указывает, какой бит будет установлен.

### Описание

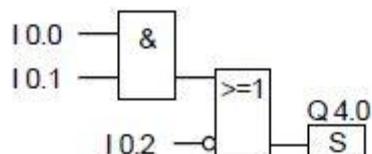
Инструкция Установить бит выполняется только тогда, когда RLO равен 1. Если RLO равен 0, эта инструкция устанавливает указанный адрес в 1. Если RLO равен 1, эта инструкция устанавливает указанный адрес в 1. Если RLO равен 0, то инструкция не влияет на указанный адрес, который остается неизменным.

На инструкцию Установить бит оказывает влияние Главное управляющее реле (MCR). За более подробной информацией о функциях MCR обратитесь к соответствующему разделу.

### Биты слова состояния

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Записывает	-	-	-	-	-	0	X	-	0

### Пример



Состояние сигнала на выходе Q4.0 устанавливается в 1 только тогда, когда:

- равны 1 состояния сигналов на входах I0.0 И I0.1
- ИЛИ равно 0 состояние сигнала на входе I0.2

Если RLO этих веток равен 0, то состояние сигнала на выходе Q4.0 не меняется.

## 5.9. RS : RS- Триггер

Обозначение



Параметры	Тип данных	Область памяти	Описание
<адрес>	BOOL	I, Q, M, D, L	Адрес указывает, какой бит будет установлен.
S	BOOL	I, Q, M, D, L, T, C	Инструкция установки
R	BOOL	I, Q, M, D, L, T, C	Инструкция сброса
Q	BOOL	I, Q, M, D, L	Состояние сигнала <адрес>

### Описание

Инструкция Сбросить / установить триггер выполняет установку (S) или сброс (R) только тогда, когда RLO = 1. RLO, равный 0, не оказывает влияния на эти инструкции, адрес, указанный в команде остается неизменным.

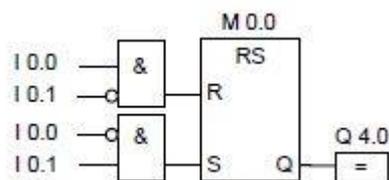
RS -триггер сбрасывается, когда состояние сигнала на входе R равно 1, а состояние сигнала на входе S равно 0. Если вход R равен 0, а вход S равен 1, то триггер установлен. Если RLO на обоих входах равно 1, то триггер установлен.

На инструкцию Сбросить / установить триггер оказывает влияние Главное управляющее реле (MCR). За более подробной информацией о функционировании MCR обращайтесь к соответствующему разделу.

### Биты слова состояния

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Записывает	-	-	-	-	-	X	X	X	1

### Пример



Если I0.0 = 1, а I0.1 = 0, то меркер M0.0 сброшен и выход Q4.0 равен 0.

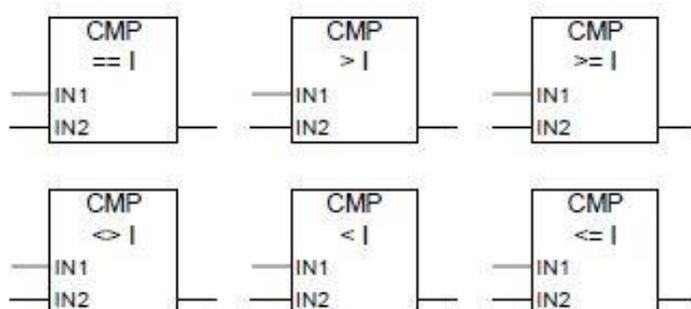
Если I0.0 = 0 а I0.1 = 1, то меркер M0.0 установлен и выход Q4.0 = 1.

Если оба сигнала равны 0, то изменения отсутствуют. Если оба сигнала равны 1, то благодаря порядку следования команд приоритетом обладает инструкция установки.

Если M 0.0 установлен, то и Q4.0 равен 1.

### 5.10. RS : RS- Триггер

#### Обозначение



Параметры	Тип данных	Область памяти	Описание
IN1	INT	I, Q, M, D, L или константа	Первое сравниваемое значение
IN2	INT	I, Q, M, D, L или константа	Второе сравниваемое значение
Выход блока	BOOL	I, Q, M, D, L	Результат сравнения

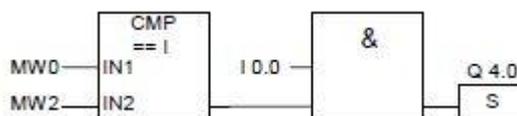
#### Описание

Инструкция Сравнить целые числа сравнивает значения двух 16-битных чисел с фиксированной точкой. Эта инструкция сравнивает входы IN1 и IN2 в соответствии с типом сравнения, выбираемым из окна списка.

#### Биты слова состояния

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Записывает	X	X	X	0	-	0	X	X	1

*Пример*



Q 4.0 устанавливается если: MW0 равно MW2 и состояние сигнала на входе I0.0 равно 1.

### 5.11. S\_CUD: Назначение параметров и прямой/обратный счет

*Обозначение*



Параметры English	Параметры German	Тип данных	Область памяти	Описание
no.	Nr.	COUNT ER	C	Номер счетчика . Диапазон номеров зависит от CPU.
CU	ZV	BOOL	I, Q, M, D, L	ZV вход : прямой счет
CD	ZR	BOOL	I, Q, M, D, L	ZR вход: обратный счет
S	S	BOOL	I, Q, M, D, L, T, C	Вход предустановки счетчика
PV	ZW	WORD	I, Q, M, D, L или константа	Значение счетчика от 0 до 999 или Значение счетчика, введенное как C#<значение> в формате BCD
R	R	BOOL	I, Q, M, D, L, T, C	Вход сброса
CV	DUAL	WORD	I, Q, M, D, L	Текущее значение счетчика (целый формат)
CV_BCD	DEZ	WORD	I, Q, M, D, L	Текущее значение счетчика (формат BCD)
Q	Q	BOOL	I, Q, M, D, L	Состояние счетчика

*Описание*

Нарастающий фронт (изменение сигнала с 0 на 1) на входе S устанавливает счетчик прямого/обратного счета на значение, указанное на входе предварительного задания PV. Счетчик увеличивается на 1, если состояние сигнала на входе CU изменяется с 0 на 1 (нарастающий фронт) и значение счетчика меньше 999. Счетчик уменьшается на 1, если состояние сигнала на входе CD изменяется с 0 на 1 (нарастающий фронт) и значение счетчика больше 0. Если имеет место нарастающий фронт на обоих счетных входах, то выполняются обе инструкции и счетчик

сохраняет прежнее значение. Счетчик сбрасывается, если нарастающий фронт появляется на входе R.

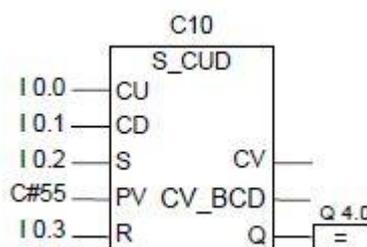
Сброс счетчика устанавливает его значение в 0.

Опрос на 1 состояния сигнала на выходе Q дает 1, если значение счетчика больше 0; опрос дает результат 0, если значение счетчика равно 0.

*Биты слова состояния*

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Записывает	-	-	-	-	-	X	X	X	1

*Пример*

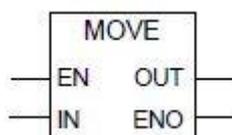


Изменение состояния сигнала с 0 на 1 на входе I0.2 устанавливает C10 со значением 55. Если состояние сигнала на входе I0.0 меняется с 0 на 1, то значение счетчика C10 увеличивается на 1, кроме случая, когда значение счетчика C10 уже равно 999. Если вход I0.1 меняется с 0 на 1, то счетчик C10 уменьшается на 1 кроме случая, когда значение счетчика C10 уже равно 0. Если I0.3 меняется с 0 на 1, то значение счетчика C10 устанавливается в 0. Выход Q4.0 равен 1, когда C10 не равен 0.

**Замечание:** Для предотвращения ошибок не используйте один номер счетчика в нескольких местах программы.

## 5.12. MOVE : Передача значения

*Обозначение*



Параметры	Тип данных	Область памяти	Описание
EN	BOOL	I, Q, M, D, L, T, C	Деблокировка входа
IN	Все элементарные типы данных 8, 16 или 32-битовые	I, Q, M, D, L или константа	Исходная область
OUT	Все элементарные типы данных 8, 16 или 32-битовые	I, Q, M, D, L или константа	Целевая область
ENO	BOOL	I, Q, M, D, L	Деблокировка выхода

### Описание

С помощью команды Передать значение, Вы можете присваивать конкретные значения переменным.

Значение, указанное на входе IN, копируется в адрес, указанный на выходе OUT. ENO имеет то же состояние сигнала, что и EN.

С помощью блока MOVE команда Передать значение может копировать все типы данных длиной 8, 16 или 32 бита. Типы данных, определенные пользователем, такие как массивы или структуры, должны копироваться с помощью системной функции SFC 20 "BLKMOV". На команду Передать значение оказывает влияние Главное управляющее реле (MCR). За дополнительной информацией о функциях MCR обращайтесь к разделу Вкл./Выкл. Master Control Relay.

### Биты слова состояния

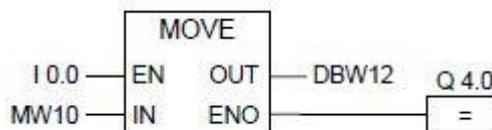
	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Записывает	1	-	-	-	-	0	1	1	1

### Замечание

При передаче значений между переменными различных типов данных, старшие байты отсекаются или заполняются нулями:

<b>Пример: Двойное слово</b>	<b>1111 1111</b>	<b>0000 1111</b>	<b>1111 0000</b>	<b>0101 0101</b>
<b>Передать</b>	<b>Результат</b>			
В двойное слово:	1111 1111	0000 1111	1111 0000	0101 0101
В байт:				0101 0101
В слово:			1111 0000	0101 0101
<b>Пример: Байт :</b>				<b>1111 0000</b>
<b>Передать</b>	<b>Результат</b>			
В байт:				1111 0000
В слово:			0000 0000	1111 0000
В двойное слово:	0000 0000	0000 0000	0000 0000	1111 0000

### Пример



Инструкция выполняется, если вход I0.0 = 1. Содержимое MW10 копируется в слово данных 12 открытого блока данных DB.

Если инструкция выполняется, выход Q4.0 устанавливается в 1.

### 5.13. S\_PULSE: Задание параметров и запуск таймера «Импульс»

Обозначение



Параметры English	Параметры German	Тип данных	Область памяти	Описание
no.	Nr.	TIMER	T	Номер таймера. Диапазон номеров зависит от CPU.
S	S	BOOL	I, Q, M, D, L, T, C	Вход запуска
TV	TW	S5TIME	I, Q, M, D, L или консанта	Установка времени(от 0-9990)
R	R	BOOL	I, Q, M, D, L, T, C	Вход сброса
BI	DUAL	WORD	I, Q, M, D, L	Остаток времени (значение в целом формате)
BCD	DEZ	WORD	I, Q, M, D, L	Остаток времени (значение в формате BCD )
Q	Q	BOOL	I, Q, M, D, L	Состояние таймера

Описание

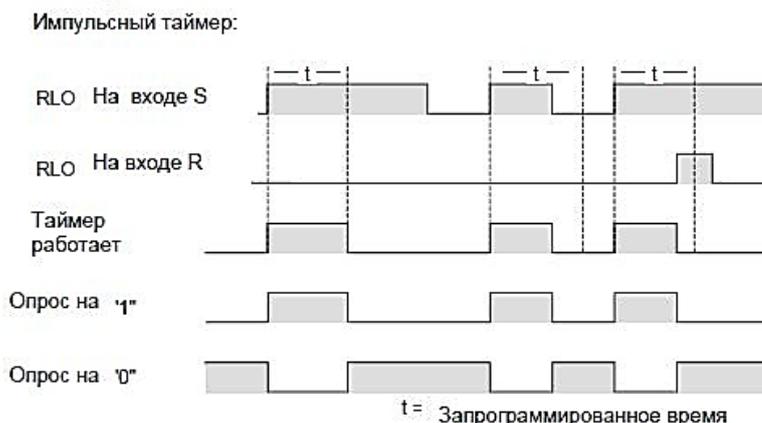
Инструкция Задание параметров и запуск таймера «Импульс» запускает заданный таймер, если имеется нарастающий фронт (изменение состояния сигнала с 0 на 1) на входе запуска (S). Для запуска таймера всегда необходимо изменение сигнала. Таймер продолжает работать в течение времени, заданного на входе TV, пока состояние сигнала на входе S остается равным 1. Пока таймер работает, опрос состояния сигнала на 1 на выходе Q дает 1. Если на входе S сигнал меняется с 1 на 0 до истечения заданного времени, таймер останавливается. Тогда опрос состояния сигнала на 1 на выходе Q дает 0. Если во время работы таймера происходит изменение с 0 на 1 сигнала на входе сброса (R), то таймер сбрасывается. Это изменение сбрасывает в ноль время и базу времени. Единица на входе R таймера не оказывает никакого влияния, если таймер не работает.

Текущее значение времени может быть опрошено на выходах BI и BCD. Значение времени на BI представлено в двоичном формате, а на BCD - в двоично-десятичном формате.

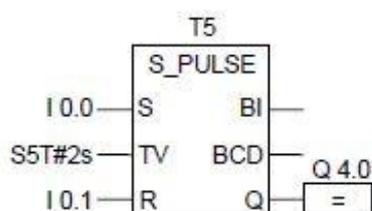
Биты слова состояния

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Записывает	-	-	-	-	-	X	X	X	1

## Временные диаграммы



## Пример



Если состояние сигнала на входе I0.0 меняется 0 на 1 (нарастающий фронт RLO), таймер T5 запускается. Таймер продолжает работать с указанным временем (2s) пока вход I0.0 равен 1. Если состояние сигнала на входе I0.0 меняется с 1 на 0 до истечения заданного времени, таймер останавливается. Если состояние сигнала на входе I0.1 меняется с 0 на 1, когда таймер работает, то таймер сбрасывается. Состояние сигнала Q4.0 равно 1 пока таймер работает.

### Задание:

#### **Разработать и отладить релейную систему управления бойлером.**

С помощью входного и выходного клапанов необходимо циклически наполнять и опустошать резервуар, непрерывно регулируя температуру воды между двумя пограничными величинами.

Требования по управлению уровнем воды в баке:

1) для наполнения бака водой при уровне воды ниже минимального (I0.0) необходимо открывать оба вентиля (Q0.0, Q0.1).

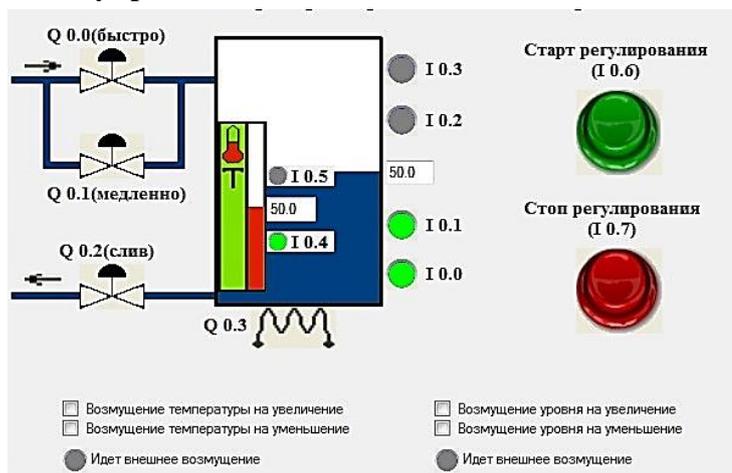
2) Если уровень воды выше датчика верхнего уровня (I0.2), то дальнейшее заполнение водой до максимального уровня должно происходить медленнее, поэтому должен быть включен только клапан медленного заполнения (Q0.1).

3) Если уровень воды выше датчика максимального уровня (I0.3), то выключаются оба входных вентиля (Q0.0, Q0.1) и включается клапан слива (Q0.2).

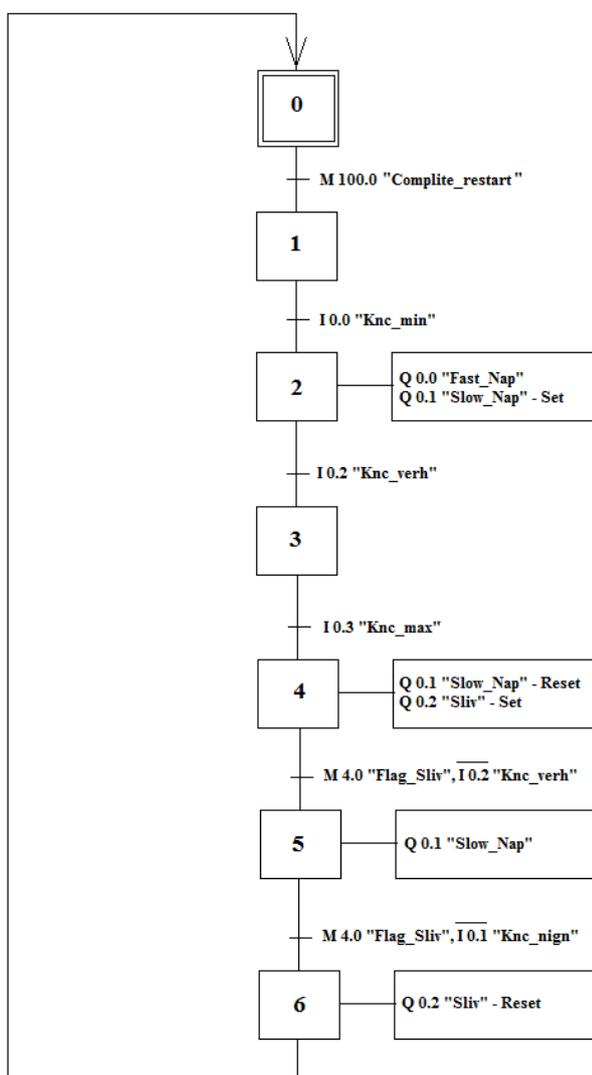
4) Если клапан слива (Q0.2) включён, то выключается он только тогда, когда уровень воды станет ниже нижнего уровня (I0.1). Так же совместно с клапаном слива отключается и клапан медленного заполнения (Q0.1).

5) Если включен клапан слива и уровень воды ниже верхнего уровня (I0.2), то для замедления процесса слива включается входной клапан медленного заполнения (Q0.1).

### Визуализация объекта управления



### Graph Set управления уровнем воды в баке

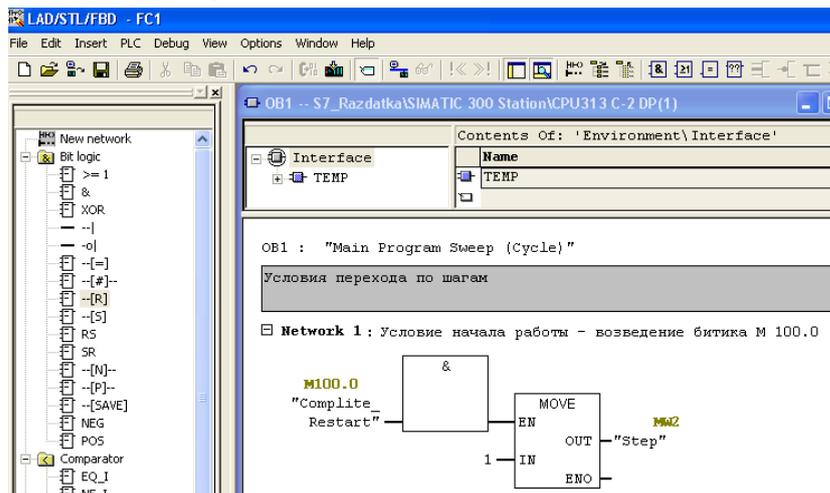


**Решение:** Разработка и отладка релейной системы управления бойлером. Программа управления уровнем воды в баке

1. Заполнение символьной таблицы

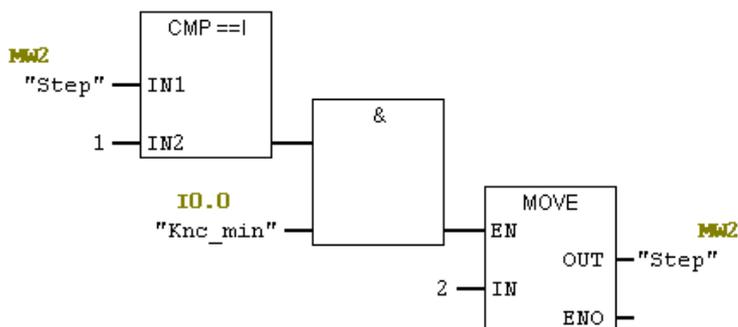
Status	Symbol	Address	Data type	Comment
1	Fast_nap	Q 0.0	BOOL	
2	Flag_sliv	M 4.0	BOOL	
3	Knc_max	I 0.3	BOOL	
4	Knc_min	I 0.0	BOOL	
5	Knc_nign	I 0.1	BOOL	
6	Knc_verh	I 0.2	BOOL	
7	Sliv	Q 0.2	BOOL	
8	Slow_nap	Q 0.1	BOOL	
9	Step	MW 2	INT	
10	Complite_Restart	M 100.0	BOOL	
11				

2. Для открытия двух окон одновременно – нажать клавиши Shift + F3
3. В организационном блоке OB 1 прописываются условия переходов с одного шага на другой. Все действия и управляющие сигналы прописываются в функции FC1.
4. На первый шаг мы переходим при условии наличия бита М 100.0 «Complite\_Restart», возводимого в организационном блоке OB 100.



5. Переход на 2 шаг происходит при наличии сигнала на датчике минимального уровня воды в баке, при этом открываются оба вентиля: быстрого и медленного набора.

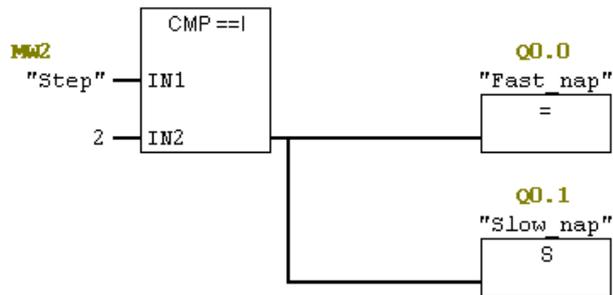
Network 2 : Условие перехода на 2 шаг - минимальный уровень воды



FC1 : Title:

События на каждом шаге

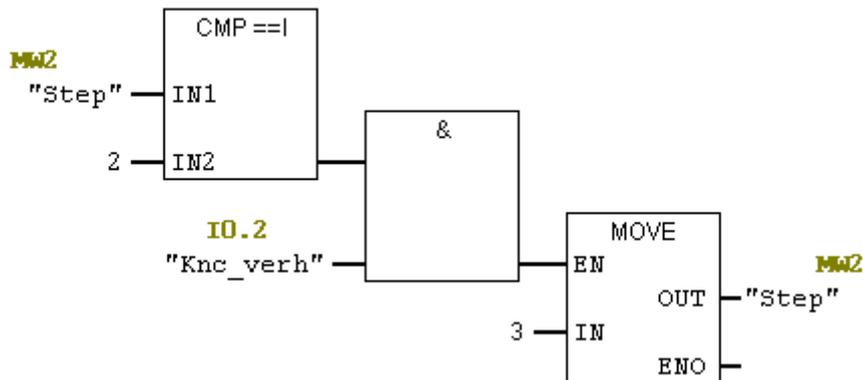
□ **Network 1** : При минимальном уровне включаются оба насоса



6. Переход на 3 шаг происходит при наличии сигнала на датчике верхнего уровня воды в баке, при этом закрывается вентиль быстрого набора.

ОВ 1.

□ **Network 3** : Срабатывает датчик верхнего уровня

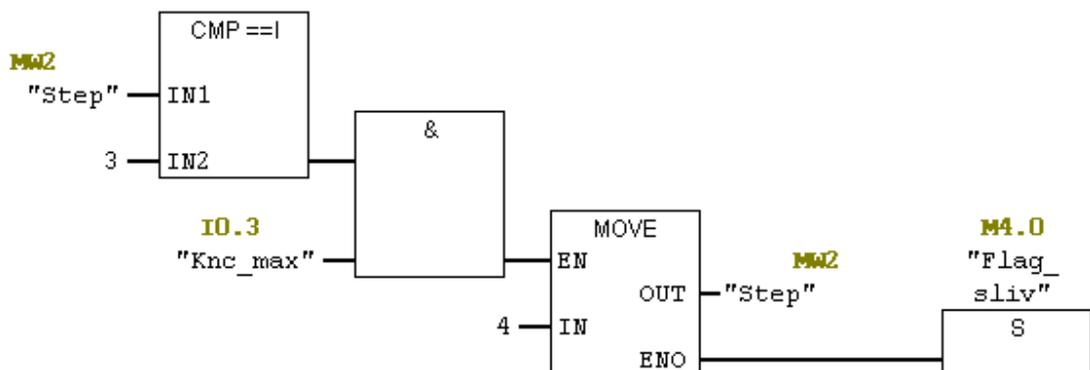


7. При переходе на 3 шаг сигнал с выхода Q 0.0 снимется автоматически, так как не был зафиксирован Set, поэтому прописывать события 3-го шага в блоке FC1 не надо.

8. Переход на 4 шаг происходит при наличии сигнала на датчике максимального уровня воды в баке, при этом закрывается вентиль медленного набора и открывается сливной вентиль. Открытие сливного вентиля фиксируется установкой флага M 4.0 «Flag\_Sliv», который используется на последующих шагах для проверки открытия сливного вентиля.

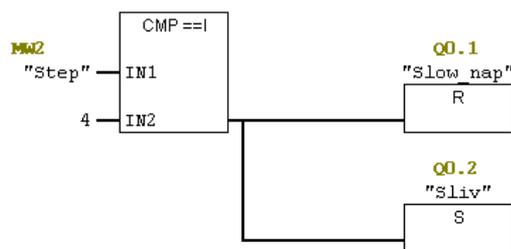
ОВ1

□ **Network 4** : Достигается максимальный уровень воды. Вводится флаг - "Слив"



FC1

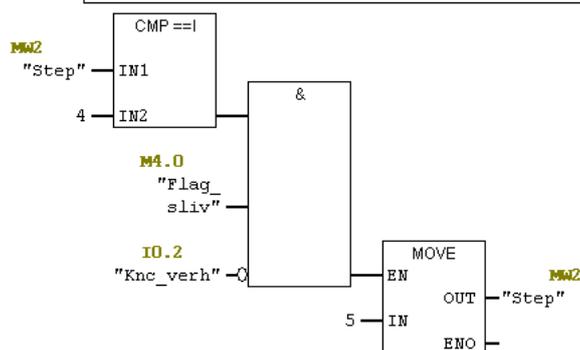
Network 2: Выкл. насос медленного набора, включается слив



9. Переход на 5 шаг происходит при условии, что открыт сливной вентиль (наличие флага M 4.0 «Flag\_Sliv») и уровень воды опустился ниже датчика верхнего уровня (сигнал датчика пропал - инверсия). При этом открывается вентиль медленного наполнения.

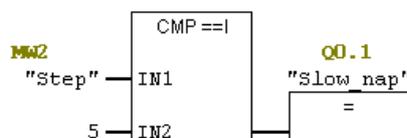
OB1

Network 5: При наличии флага "Слив", вода опускается ниже верхнего уровня



FC1

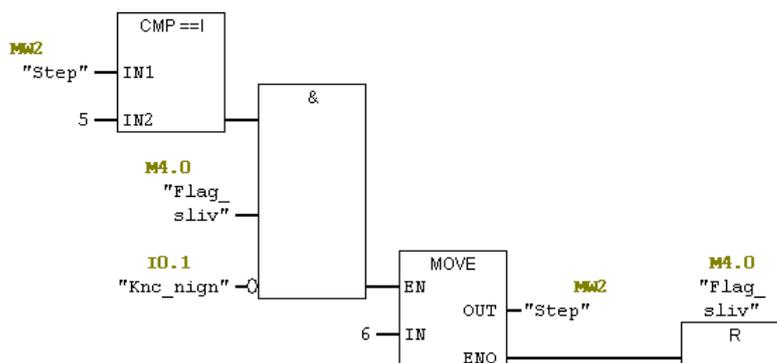
Network 3: Включается насос медленного набора



10. Переход на 6 шаг происходит при условии, что открыт сливной вентиль (наличие флага M 4.0 «Flag\_Sliv») и уровень воды опустился ниже датчика нижнего уровня (сигнал датчика пропал - инверсия). При этом закрываются оба вентиля. Сбрасывается флаг M 4.0 «Flag\_Sliv».

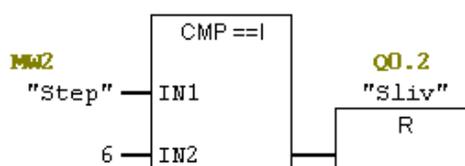
OB1

Network 6: При наличии флага "Слив", вода опускается ниже нижнего уровня



## FC1

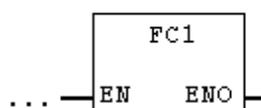
□ Network 4: Выключается слив



При переходе на 6 шаг сигнал с выхода Q 0.1 снимется автоматически, так как не был зафиксирован Set, поэтому прописывать события 6-го шага в блоке FC1 не надо.

11. Для установки связи между организационным блоком OB1 и функцией FC1, помещаем ее в последний Network организационного блока.

□ Network 7: Связь с функцией FC1



### Порядок выполнения практической работы:

1. Изучить краткие теоретические сведения.
2. Разработать и отладить релейную систему управления бойлером.
3. Подготовить отчет.

### Отчет по работе должен содержать:

1. Название работы.
2. Цель работы.
3. Результаты выполнения практической работы.

### Вопросы для самопроверки:

1. Для чего служит Модуль центрального процессора (CPU)?
2. Что обеспечивают Модули блоков питания (PS)?
3. Для чего предназначены Сигнальные модули (SM)?
4. Для чего служат Коммуникационные процессоры (CP)?
5. Какие задачи решают Функциональные модули (FM)?
6. Что обеспечивают Интерфейсные модули (IM)?

### Средства обучения:

1. **Оборудование и материалы:** Раздаточный материал, Рабочая тетрадь
2. **Вербальные средства обучения:**

Программирование в функциональном плане (FBD) для S7-300 и S7-400: Справочное руководство, с.208

3. **Технические средства обучения (ТСО):** -

## Практическая работа № 4

**Тема:** Программирование станции перемещения материалов

### Цель работы:

- уметь читать и составлять принципиальные схемы электрических, гидравлических и пневматических приводов несложного технологического оборудования;
- уметь составлять управляющие программы для программируемых логических контроллеров в SCADA – системе;
- уметь распознавать, классифицировать и использовать датчики, реле и выключатели в системах управления;
- уметь правильно эксплуатировать мехатронное оборудование.

### Задание: Станция «Buffer»



Рисунок 4.1 - Станция «Buffer»

### Алгоритм функционирования станции

#### Режим работы

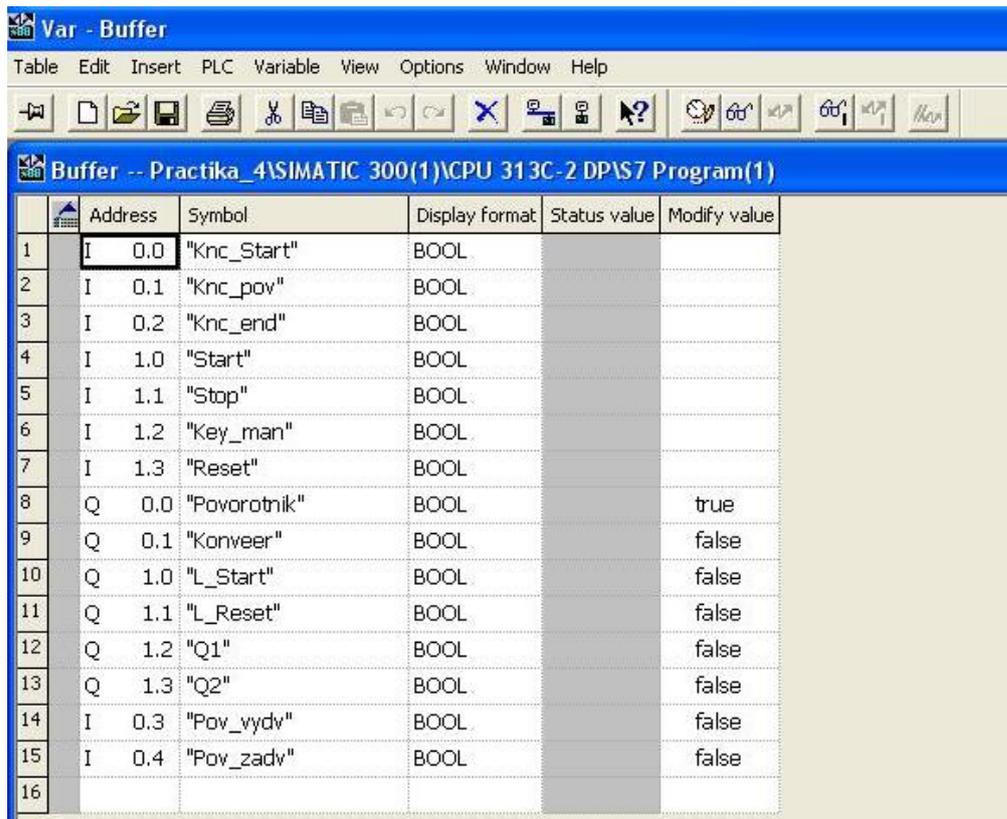
- Лампа Reset мигает с частотой 0,5 Гц, попеременно мигают лампы Q1 и Q2 с частотой 2 Гц;
- Переключить ключ в позицию MAN, перестают мигать лампы Q1 и Q2;
- Нажать кнопку Reset, тогда приводы системы перемещаются в исходное положение (исходное положение - поворотник выдвинут!);
- При условии, что приводы системы переместились в исходное положение, лампа Reset гаснет, лампа Start начинает мигать с частотой 1 Гц.

#### Основной алгоритм работы

- Нажать кнопку Start, при условии, что в начале конвейерной ленты есть деталь, конвейер приводится в движение, при этом загорается лампа Q1;
- При достижении деталью поворотника, загорается лампа Q2, деталь удерживается поворотником;
- Спустя 3 секунды поворотник задвигается, пропуская деталь дальше;
- При условии, что деталь сошла с ленты останавливается конвейер, гаснут лампы Q1 и Q2, лампа Start начинает мигать с частотой 1 Гц.

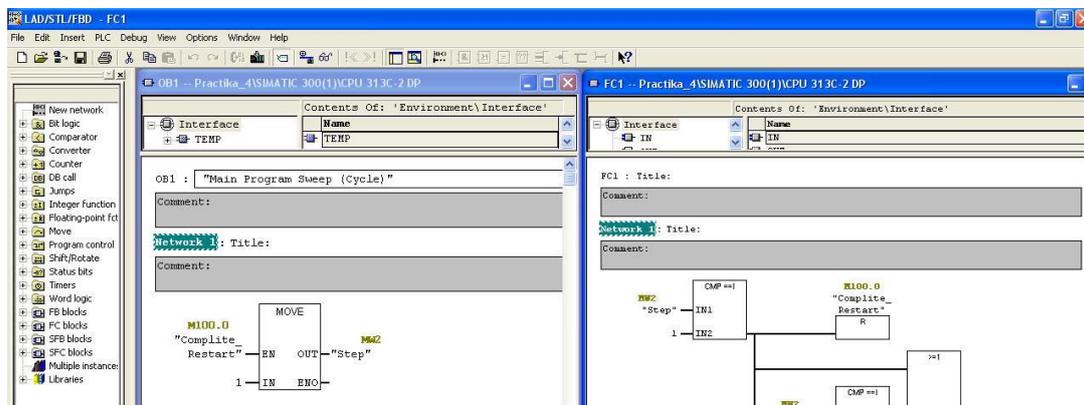
## Программная реализация алгоритма функционирования станции

1. Создание таблицы состояний Variable Table, позволяющей отслеживать текущие состояния сигналов датчиков и управляющих сигналов, а также дистанционно вносить изменения в эти состояния.

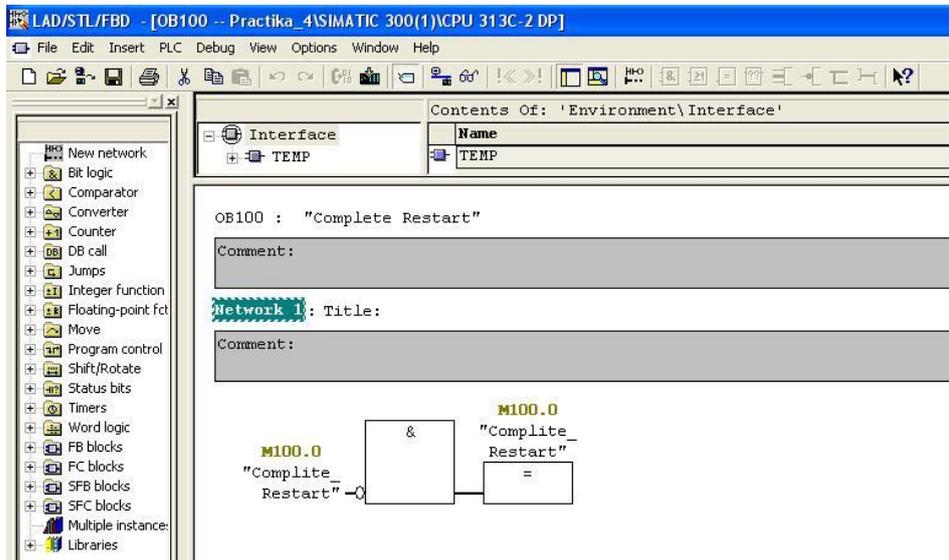


	Address	Symbol	Display format	Status value	Modify value
1	I 0.0	"Knc_Start"	BOOL		
2	I 0.1	"Knc_pov"	BOOL		
3	I 0.2	"Knc_end"	BOOL		
4	I 1.0	"Start"	BOOL		
5	I 1.1	"Stop"	BOOL		
6	I 1.2	"Key_man"	BOOL		
7	I 1.3	"Reset"	BOOL		
8	Q 0.0	"Povorotnik"	BOOL		true
9	Q 0.1	"Konveer"	BOOL		false
10	Q 1.0	"L_Start"	BOOL		false
11	Q 1.1	"L_Reset"	BOOL		false
12	Q 1.2	"Q1"	BOOL		false
13	Q 1.3	"Q2"	BOOL		false
14	I 0.3	"Pov_vydv"	BOOL		false
15	I 0.4	"Pov_zadv"	BOOL		false
16					

2. В организационном блоке OB 1 прописываются условия переходов с одного шага на другой. Все действия и управляющие сигналы прописываются в функции FC1.

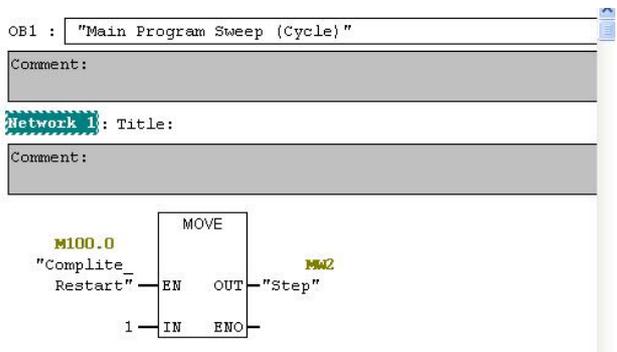


3. Для организации стартового состояния по прерыванию создается организационный блок OB 100, в котором взводится бит маркерной памяти M 100.0 «Complite\_Restart».

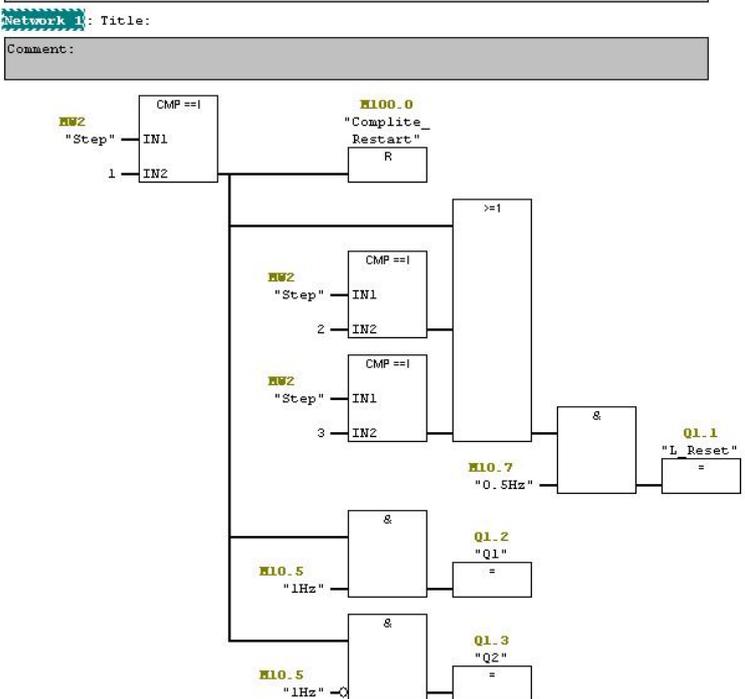


4. На первый шаг мы переходим при условии наличия бита М 100.0 «Complite\_Restart». При этом мигает с частотой 0,5 Гц Лампа Reset, попеременно мигают лампы Q1 и Q2 с частотой 2 Гц. Лампа Reset продолжает мигать так же на 2 и на 3 шагах.

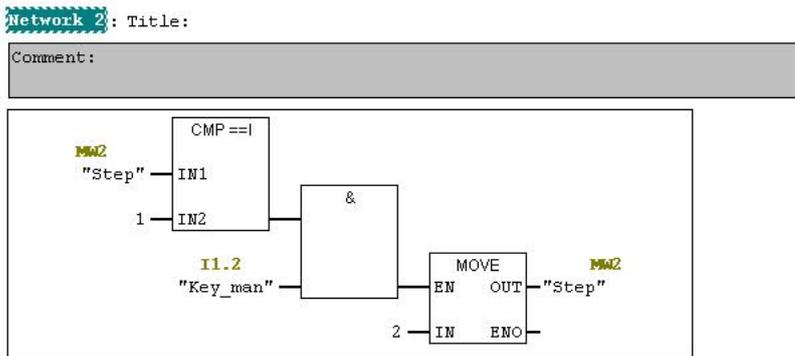
#### OB1



#### FC1



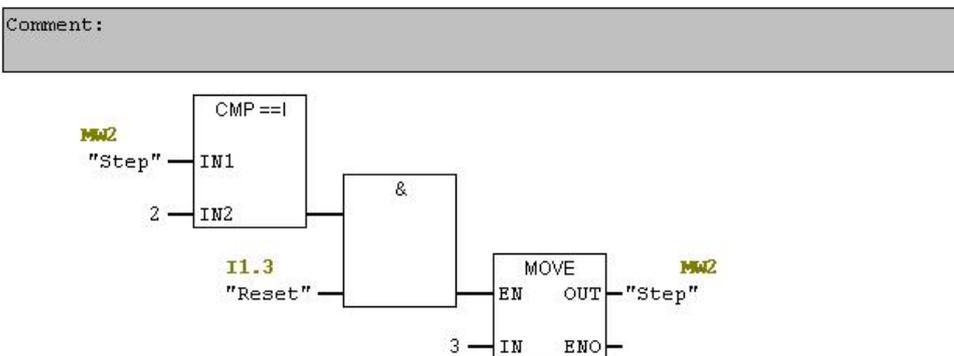
5. Переход на 2 шаг происходит при повороте ключа в положение MAN.  
OB 1.



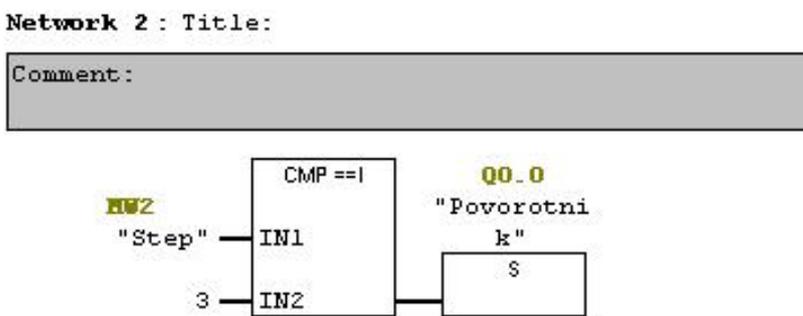
При переходе на 2 шаг сигналы с ламп Q1 и Q2 снимется автоматически, так как не был зафиксирован Set, поэтому прописывать события 2-го шага в блоке FC1 не надо.

6. Переход на 3 шаг происходит при нажатии кнопки Reset. При этом поворотник перемещаются в исходное положение (исходное положение - поворотник выдвинут!).

OB1



FC1

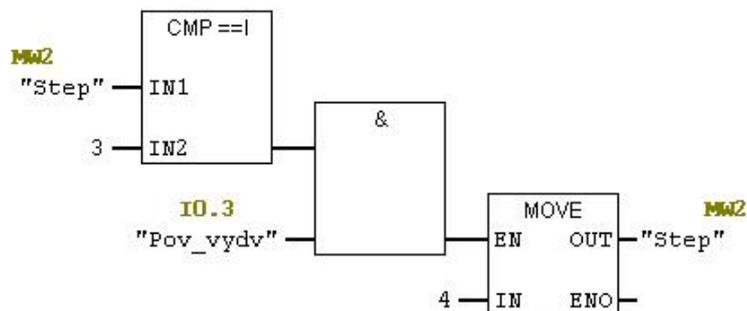


7. Условием перехода на 4 шаг является наличие сигнала с концевика «Povorotnik\_vydv», что свидетельствует о перемещении поворотника в исходное положение. При этом лампа Reset гаснет, лампа Start начинает мигать с частотой 1 Гц.

OB1

Network 4 : Title:

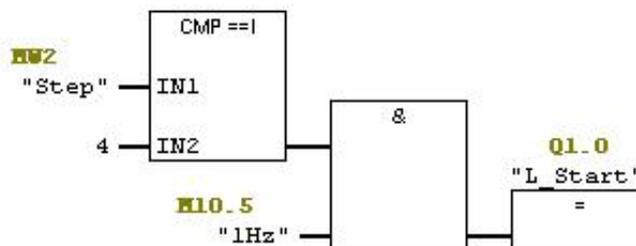
Comment:



FC1

Network 3 : Title:

Comment:

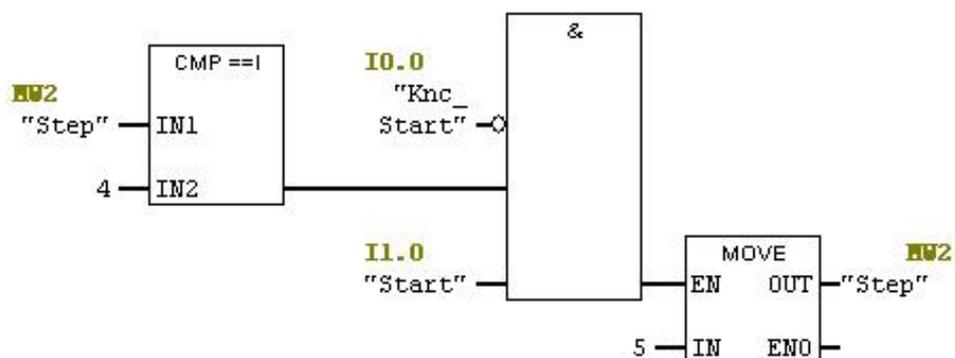


8. Переход на 5 шаг происходит по нажатию кнопку Start, при отсутствии сигнала с датчика наличия детали в начале конвейерной ленты «Knc\_Start» (отсутствие сигнала свидетельствует о наличии детали – инверсный сигнал). При этом конвейер приводится в движение и загорается лампа Q1.

OB1

Network 5 : Title:

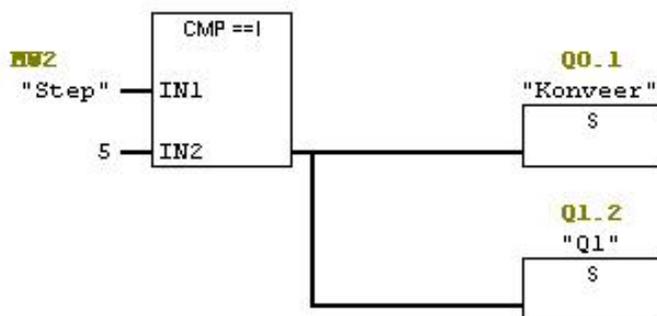
Comment:



FC1

Network 4 : Title:

Comment:

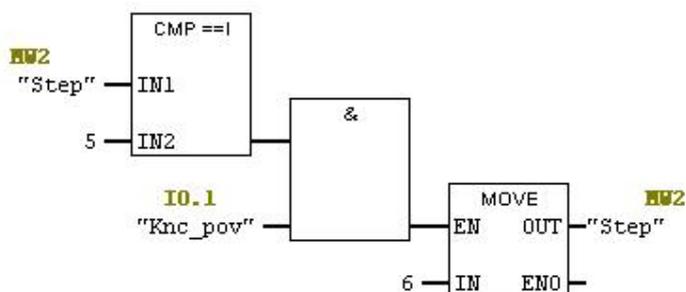


9. Переход на 6 шаг происходит при наличии сигнала с датчика «Кнс\_Ров», свидетельствующий о достижении деталью поворотникалампа. При этом загорается лампа Q2 и запускается таймер с задержкой включения, выставленный на 2 секунды, спустя это время снимается сигнал с привода поворотника и он задвигается, пропуская деталь дальше.

OB1

Network 6 : Title:

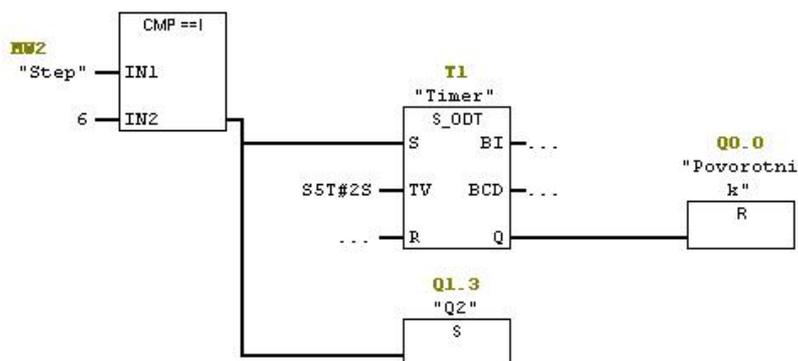
Comment:



FC1

Network 5 : Title:

Comment:

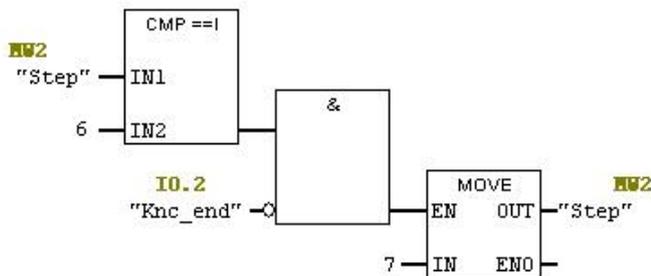


10. Условием перехода на 7 шаг является отсутствие сигнала с датчика наличия детали в конце конвейерной ленты «Knc\_End» (отсутствие сигнала свидетельствует о наличии детали – инверсный сигнал). При этом запускается таймер с задержкой включения, выставленный на 3 секунды – за которые деталь успеет переместиться с конвейерной ленты на следующую станцию, спустя это время снимается сигнал с двигателя конвейера и он останавливается, гаснут лампы Q1 и Q2.

OB1

Network 7 : Title:

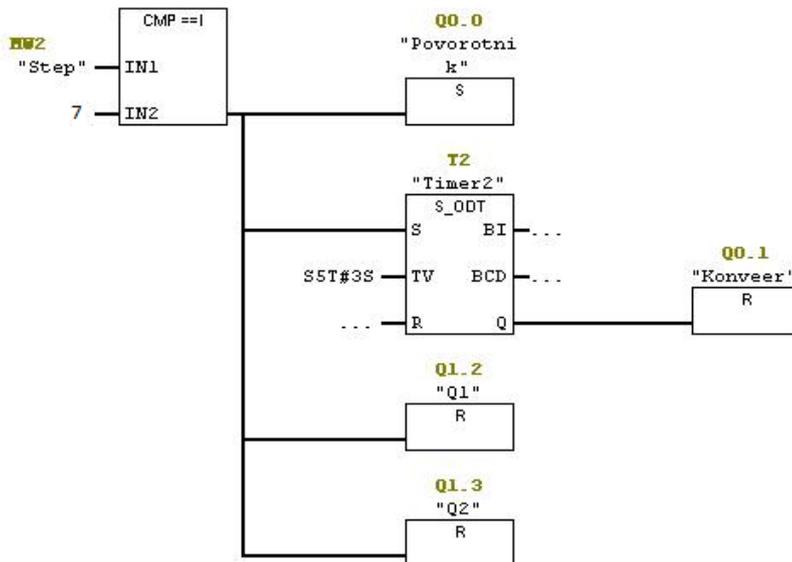
Comment:



FC1

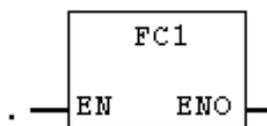
Network 6 : Title:

Comment:



11. Для установки связи между организационным блоком OB1 и функцией FC1, помещаем ее в последний Network (8) организационного блока.

: Связь с функцией FC1



**Порядок выполнения практической работы:**

1. Изучить краткие теоретические сведения.
2. Разработать программу управления станцией перемещения материалов.
3. Подготовить отчет.

**Отчет по работе должен содержать:**

1. Название работы.
2. Цель работы.
3. Результаты выполнения практической работы.

**Вопросы для самопроверки:**

1. Опишите логическую операцию OR (ИЛИ).
2. Опишите логическую операцию И.
3. Опишите инструкция присвоения.
4. Для чего служит команда MOVE?

**Средства обучения:**

1. **Оборудование и материалы:** Раздаточный материал, Рабочая тетрадь
2. **Вербальные средства обучения:**

Программирование в функциональном плане (FBD) для S7-300 и S7-400: Справочное руководство, с.208

3. **Технические средства обучения (ТСО):** -

## Практическая работа № 5

**Тема:** Построение системы контроля и управления на участках термической обработки, хранения и дозирования в SCADA TRACE MODE

### **Цель работы:**

- уметь читать и составлять принципиальные схемы электрических, гидравлических и пневматических приводов несложного технологического оборудования;
- уметь составлять управляющие программы для программируемых логических контроллеров в SCADA – системе;
- уметь распознавать, классифицировать и использовать датчики, реле и выключатели в системах управления;
- уметь правильно эксплуатировать мехатронное оборудование.

### **Задание**

Технологический процесс (ТП) ведется на трех участках: термической обработки, хранения и дозирования.

***Необходимо спроектировать мнемосхему системы контроля и управления ТП.***

**Участок термической обработки.** Технологическая задача - поддержание постоянной температуры в аппарате - технологической установке осуществляющей длительный нагрев исходного сырья (функция регулирования). Регулирование температуры производится путем изменения расхода поступающего в аппарат теплоносителя. Примем, что для измеряемой датчиком температуры коды соответствуют диапазону (0 - 100) С. АРМ контролирует подключенные к PC-based контроллеру технологические параметры (функция мониторинга) и задает настройки регулятора (функция управления).

**Участок хранения** обслуживается контроллером с традиционной архитектурой - PLC (используются четыре аналоговых входных - 4AI) . АРМ выполняет только функцию мониторинга. PLC контроллер SIEMENS S7-200 содержит в своем составе центральный процессор CPU222 (8DI, 6DO) и модуль EM231 (4AI). С помощью программного пакета STEP7 разработана программа управления поддержанием уровня в хранилище и организована связь с АРМ с использованием имеющегося «на борту» процессорного блока свободно конфигурируемого коммуникационного интерфейса PPI по стандартному протоколу обмена Modbus RTU. В двухбайтовых ячейках с адресами 0x1, 0x3, 0x5 и 0x7 содержатся данные, характеризующие такие параметры, как уровень заполнения, температура в хранилище, давление и влажность воздуха. АЦП в EM231 12-ти разрядный, представление в кодах (0 - 4095) и соответственно контролируемые величины имеют диапазоны (0 - 5) м, (0 - 100) С", (0 - 5) атм. и (0 - 100) % соответственно. Настройки коммуникационного последовательного порта PPI CPU222 - 19200,п,8,1.

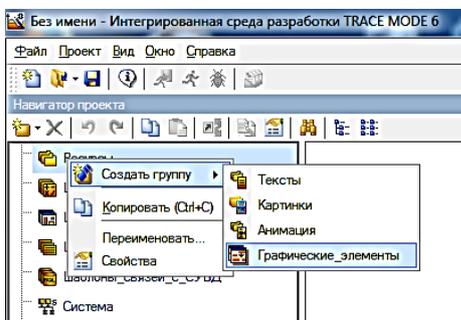
**Участок дозирования.** По заданию, передаваемому с АРМ, производится отпуск готового продукта, контроль отпуска ведется по показаниям расходомера, соответствующий им расход (0 - 100 л/мин), в качестве управляемого оборудования выступает дозирующий насос (включение - подача от контроллера логической «1» на модуль гальванической развязки, имеющий в качестве нагрузки пускатель насоса, отключение - логический «0»).

## Программная реализация мнемосхемы системы контроля и управления ТП

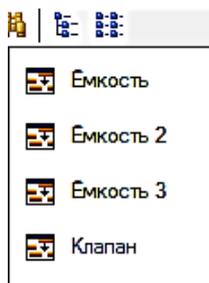
### Создание экранов АРМ

1. Откроем интегрированную систему разработки и с помощью щелчка ЛК по иконке создадим новый проект. В качестве стиля разработки выберем Стандартный.

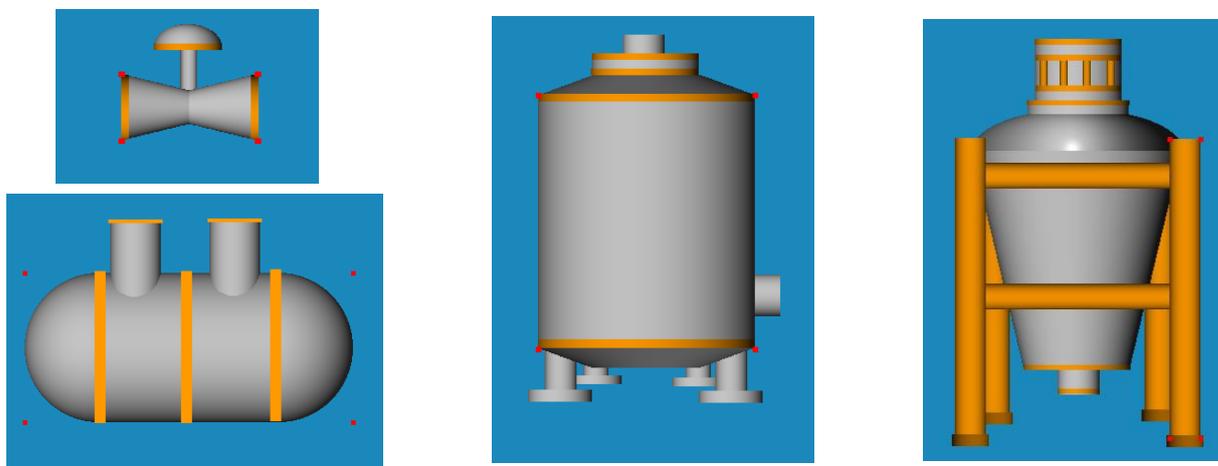
Перейдем в слой Ресурсы создадим группу **Графические элементы** необходимый для дальнейшей разработки набор графических объектов - изображения клапанов, емкостей, двигателей и т.д.



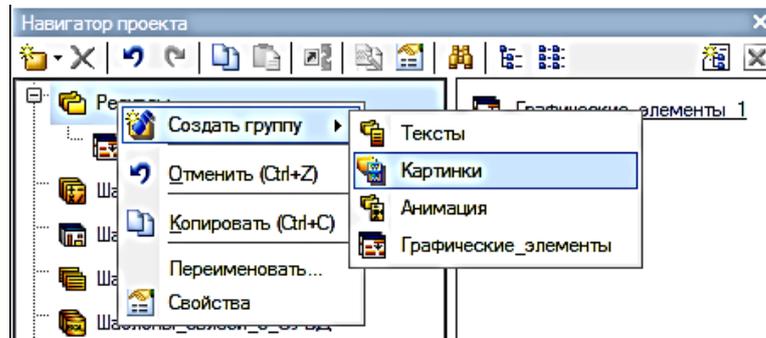
Создадим 4 компонента, переименуем соответственно:



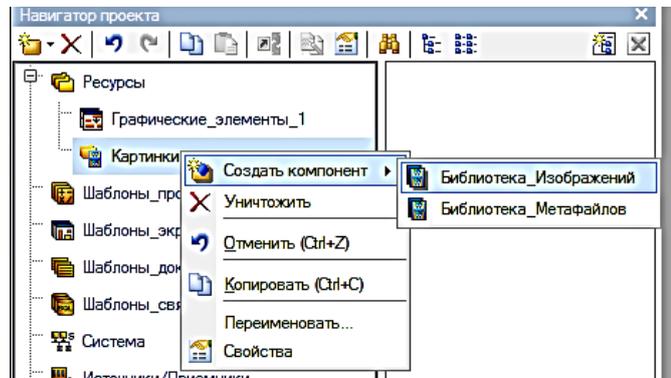
Двойным нажатием левой кнопки мыши открываем графические элементы и создаём объекты:



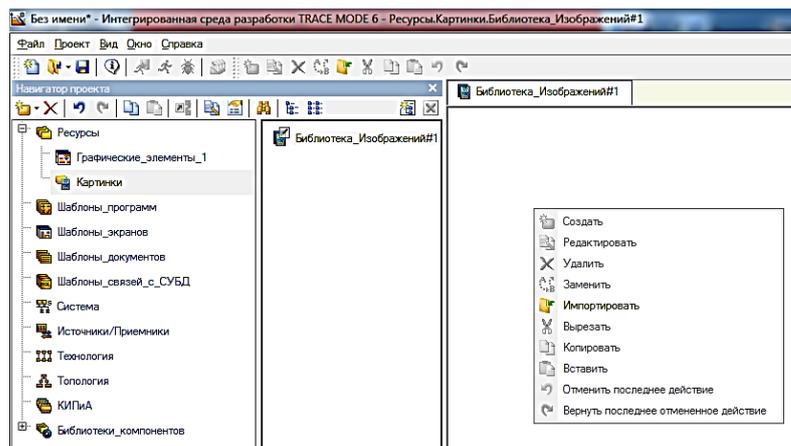
Здесь же в слое Ресурсы создадим группу **Картинки** для помещения в нее текстур, которые будут применены в оформлении создаваемых графических экранов.



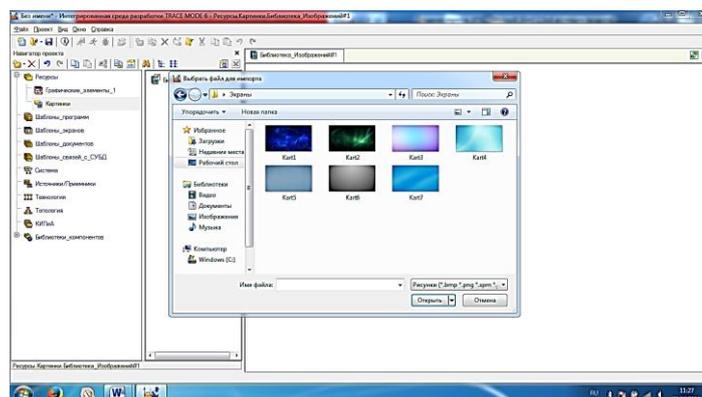
Создадим в группе **Картинки** **новый компонент**  
**Библиотека\_Изображений#1.**



Откроем двойным щелчком ЛК вновь созданную библиотеку для редактирования. Для ее наполнения воспользуемся правой кнопкой мыши.



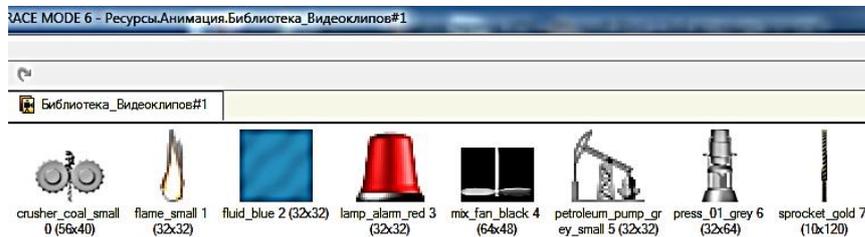
В открывшемся диалоге выбора файлов для импорта выберем все файлы и нажмем экранную кнопку **Открыть**.



Содержимое библиотеки **Библиотека\_Изображений#1** станет следующим:

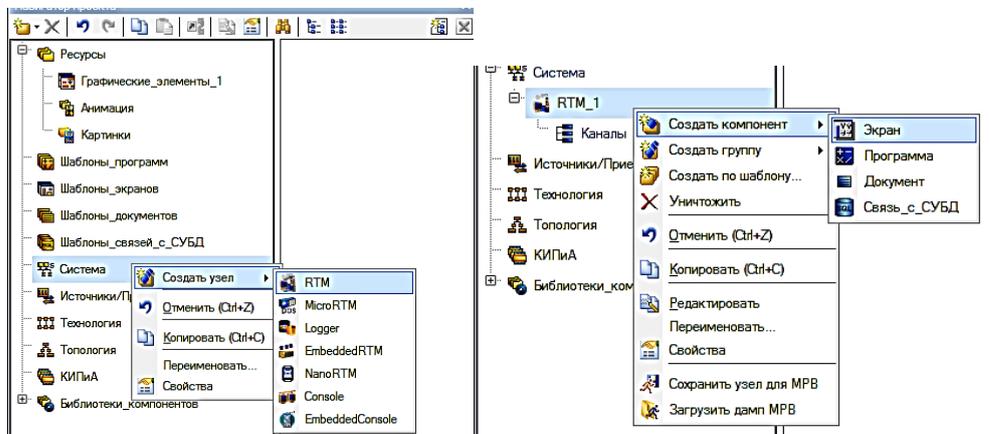


Подобным описанному выше способу создадим в слое **Ресурсы** группу **Анимация**, в ней - библиотеку **Библиотека\_Видеоклипов#1**. Наполним ее содержимым ... \Lib\Animation.



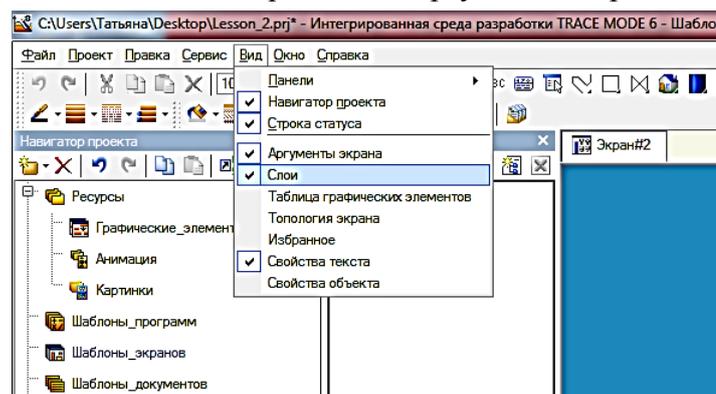
После проведения подготовительных мероприятий сохраним выполненную работу, нажав ЛК и указав имя **Lesson\_2.prj**.

1. В слое **Система** создадим узел **RTM**, а в нём создадим компонент **Экран**:

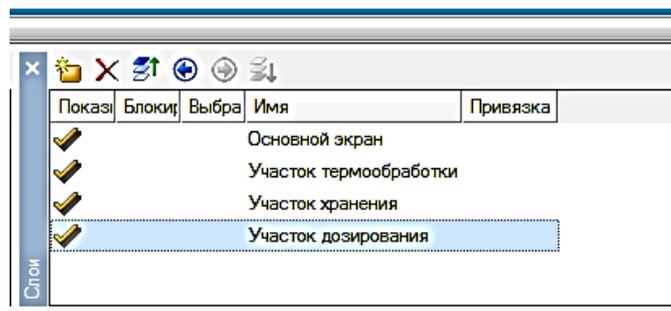


Откроем компонент **Экран#1** на редактирование.

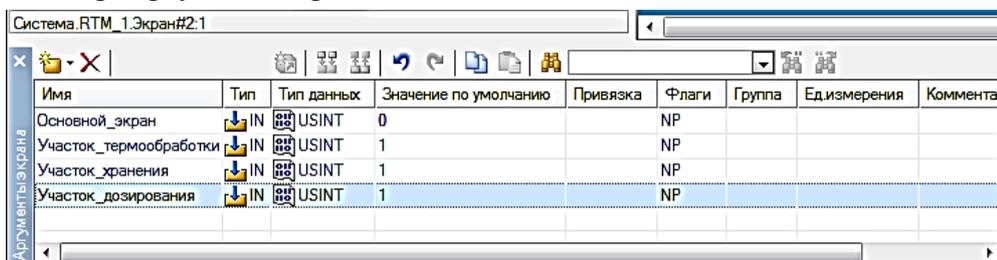
Через пункт **Вид** основного меню откроем окно аргументов экрана и графических слоев:



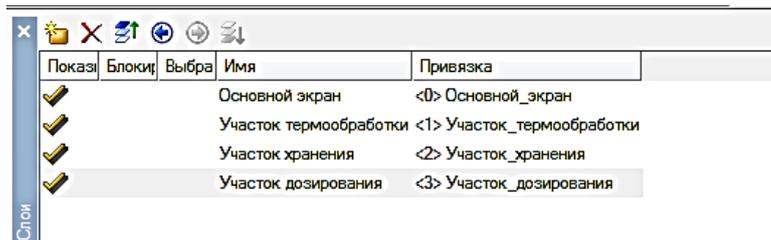
Создадим четыре слоя экрана:



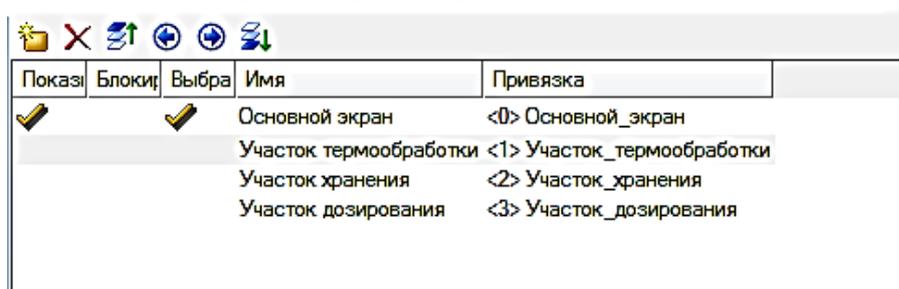
Создадим четыре аргумента экрана:



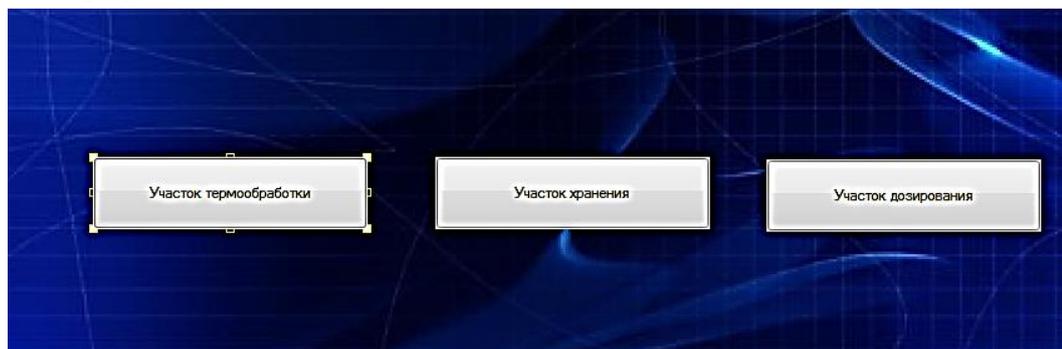
С помощью механизма **drag-and-drop** привяжем аргументы экрана к слоям экрана:



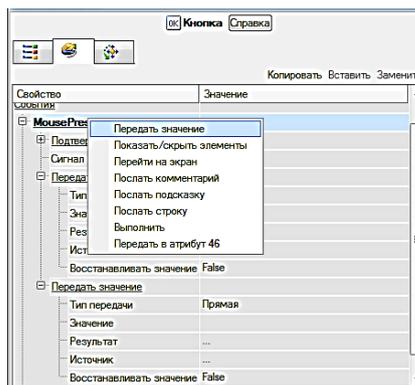
Оставим Слой «Основной экран» на редактирование:



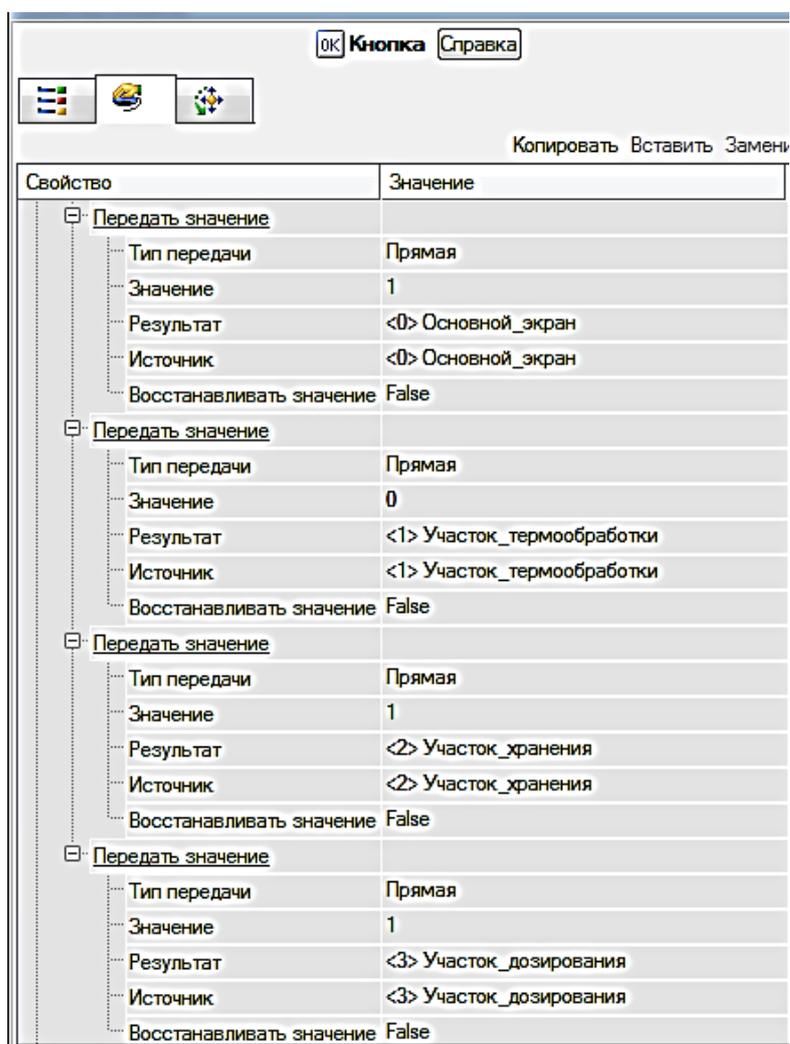
Создаём 3 графических объекта **Кнопка**:



Откроем бланк События для кнопки «Участок термообработки» и ПК раскроем меню **По нажатию** (mouse Pressed); выберем из списка команду Передать значение для 4 слоёв экрана.



Сделаем привязки для кнопки «Участок термообработки»:



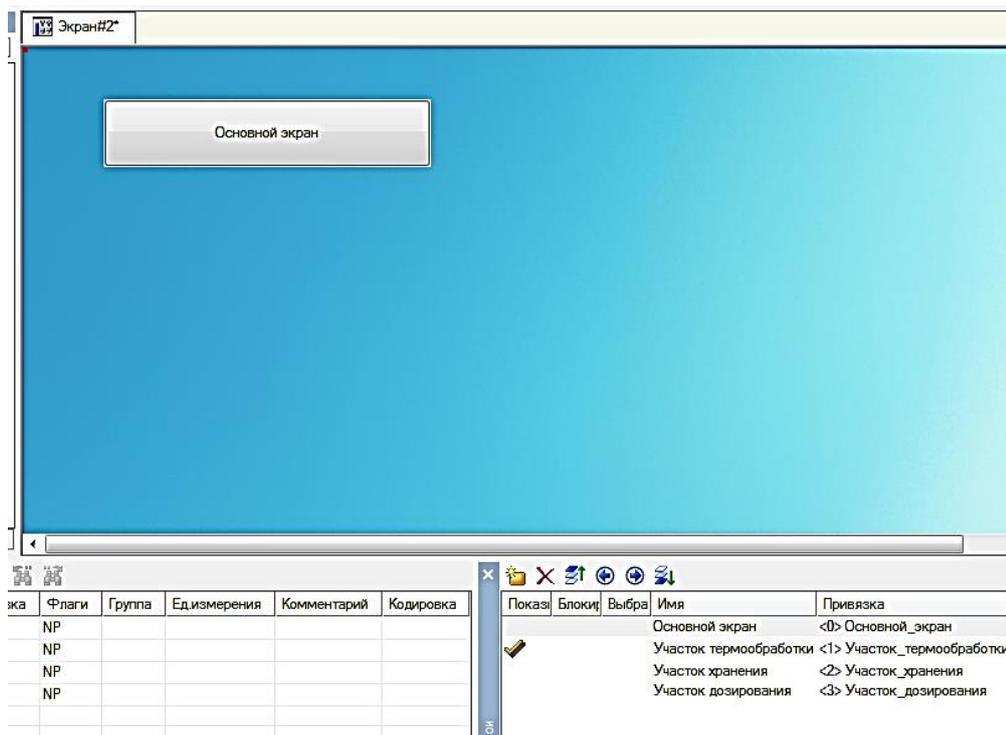
Подобным описанному выше способу редактируем кнопку «Участок хранения»:

Свойство	Значение
Передать значение	
Тип передачи	Прямая
Значение	1
Результат	<0> Основной_экран
Источник	<0> Основной_экран
Восстанавливать значение	False
Передать значение	
Тип передачи	Прямая
Значение	1
Результат	<1> Участок_термообработки
Источник	<1> Участок_термообработки
Восстанавливать значение	False
Передать значение	
Тип передачи	Прямая
Значение	0
Результат	<2> Участок_хранения
Источник	<2> Участок_хранения
Восстанавливать значение	False
Передать значение	
Тип передачи	Прямая
Значение	1
Результат	<3> Участок_дозирования
Источник	<3> Участок_дозирования
Восстанавливать значение	False

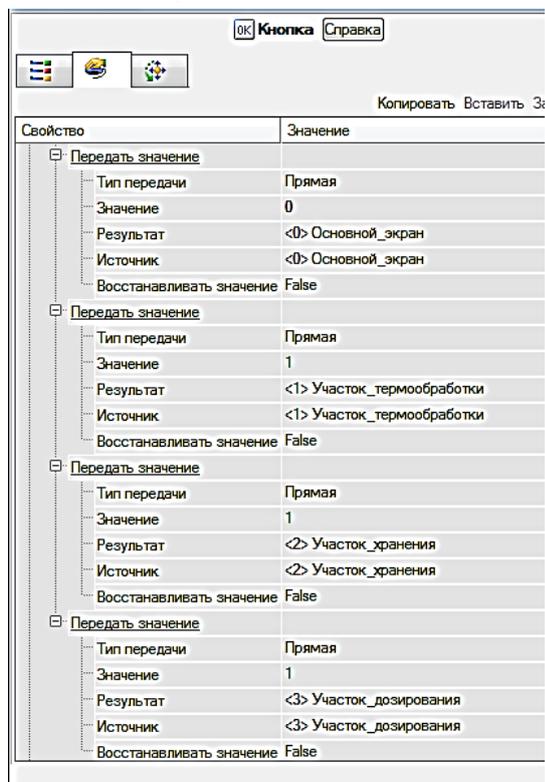
и кнопку «Участок дозирования»:

Свойство	Значение
Передать значение	
Тип передачи	Прямая
Значение	1
Результат	<0> Основной_экран
Источник	<0> Основной_экран
Восстанавливать значение	False
Передать значение	
Тип передачи	Прямая
Значение	1
Результат	<1> Участок_термообработки
Источник	<1> Участок_термообработки
Восстанавливать значение	False
Передать значение	
Тип передачи	Прямая
Значение	1
Результат	<2> Участок_хранения
Источник	<2> Участок_хранения
Восстанавливать значение	False
Передать значение	
Тип передачи	Прямая
Значение	0
Результат	<3> Участок_дозирования
Источник	<3> Участок_дозирования
Восстанавливать значение	False

Откроем на редактирование слой «Участок термообработки», и создадим в нём ГЭ Кнопка:

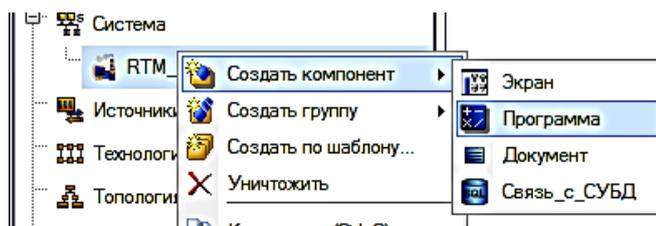


Отредактируем кнопку «Основной экран»:

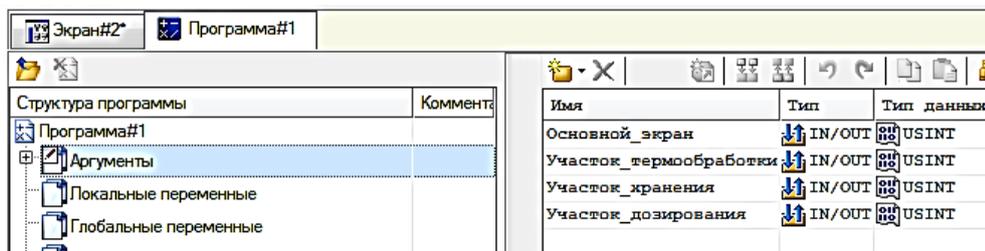


Далее скопируем ГЭ кнопку «Основной экран» и вставим в слои «Участок хранения» и «Участок дозирования».

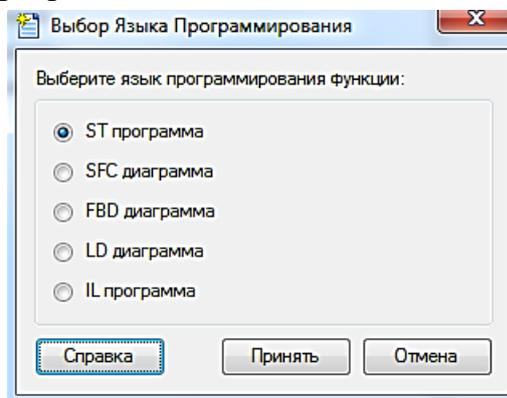
Приступим к созданию программы для смены слоёв экрана. В узле RTM создадим компонент Программа:



Создадим аргументы для данной программы:



Для разработки программы воспользуемся языком **Техно ST**, для чего в окне выбора языка программирования укажем **ST программа**.



Текст программы будет выглядеть следующим образом:

```

PROGRAM
VAR_INOUT Основной_экран : USINT; END_VAR
VAR_INOUT Участок_термообработки : USINT; END_VAR
VAR_INOUT Участок_хранения : USINT; END_VAR
VAR_INOUT Участок_дозирования : USINT; END_VAR

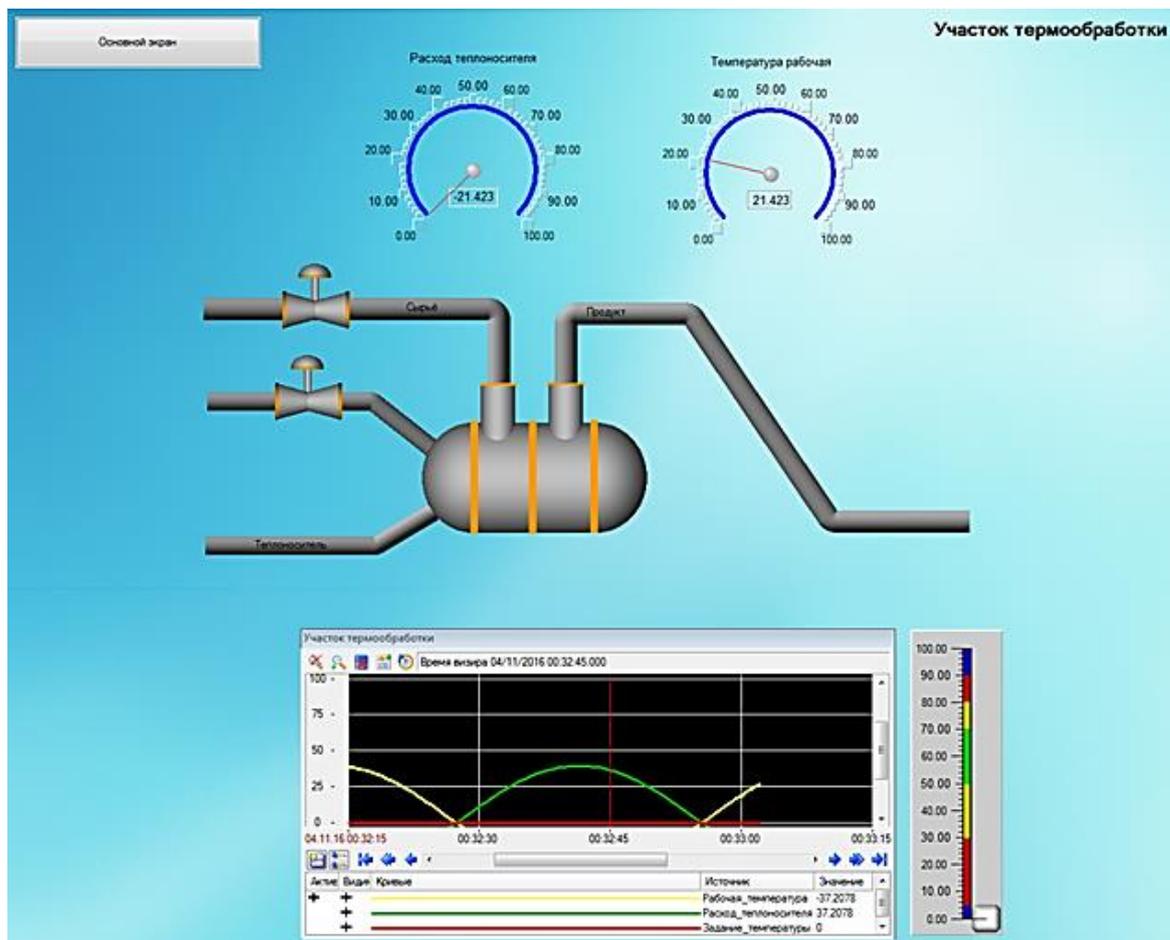
if Основной_экран==0 then
    Участок_термообработки=1;Участок_хранения=1; Участок_дозирования=1;
else if Участок_термообработки==0 then
    Основной_экран=1;Участок_хранения=1;Участок_дозирования=1;
else if Участок_хранения==0 then
    Основной_экран=1;Участок_термообработки=1;Участок_дозирования=1;
else if Участок_дозирования==0 then
    Основной_экран=1;Участок_термообработки=1;Участок_хранения=1;
end_if;end_if;end_if;end_if;

END_PROGRAM
    
```

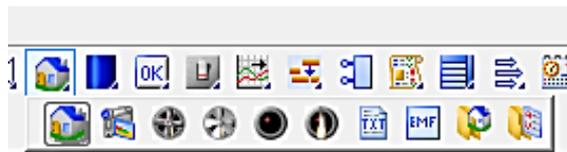
2. Создадим недостающие аргументы экрана для привязки их к динамическим объектам:

Имя	Тип	Тип данных
Основной_экран	↓ IN	USINT
Участок_термообработки	↓ IN	USINT
Участок_хранения	↓ IN	USINT
Участок_дозирования	↓ IN	USINT
Расход_теплоносителя	↕ IN/OUT	REAL
Температура_рабочая	↕ IN/OUT	REAL
Задание_температуры	↕ IN/OUT	REAL
Уровень	↕ IN/OUT	REAL
Температура	↕ IN/OUT	REAL
Давление	↕ IN/OUT	REAL
Влажность	↕ IN/OUT	REAL
Насос	↕ IN/OUT	BOOL
Расход_продукта	↕ IN/OUT	REAL
Задание_объёма	↕ IN/OUT	REAL
Объём_продукта_текущий	↕ IN/OUT	REAL
Старт_стоп	↕ IN/OUT	REAL

Приступим к созданию слоя «Участок термообработки»:



Для перехода к непосредственному созданию и редактированию содержимого экрана дважды нажмем на нем ЛК мыши. Зададим в качестве фона экрана любую текстуру. Для этого дважды нажмем на нем ЛК мыши на верхней панели экрана на **Ресурсы**.



Выбираем **Растовое изображение**, и выберем любой фон из имеющихся в библиотеке текстур.

С помощью графических объектов (ГО), сохраненных в ресурсных библиотеках и вызываемых с помощью иконки  панели инструментов, а также графических элементов (ГЭ) объемных труб и текста, создадим статическую часть экрана.

Значения расхода теплоносителя и рабочей температуры будем отображать с помощью ГЭ Показывающий прибор . Разместив их на экране, двойным щелчком ЛК откроем свойства левого ГЭ и зададим ему свойства:

Стрелочный прибор Справка

Копировать Вставить Заменить

Свойство	Значение
Отображаемая величина (<4> Расход теплоносителя)	0
Привязка	<4> Расход_теплоносителя
Положение	По центру
Угол разворота	135
<b>Заголовок</b>	
<b>Полоса</b>	Тле
Ширина	5
Верхний предел шкалы (<4> Расход теплоносителя)	100
HL (<4> Расход теплоносителя)	90
HA (<4> Расход теплоносителя)	80
HW (<4> Расход теплоносителя)	70
LW (<4> Расход теплоносителя)	50
LA (<4> Расход теплоносителя)	30
LL (<4> Расход теплоносителя)	5
Нижний предел шкалы (<4> Расход теплоносителя)	0
Цвет (HW, LW)	
Цвет (HA, HW), (LW, LA)	
Цвет (HL, HA), (LA, LL)	
Цвет >HL, <LL	
<b>Шкала</b>	Тле
<b>Указатель</b>	
<b>Единицы</b>	
Основная привязка	<4> Расход_теплоносителя

Аналогичным образом поступим с правым ГЭ:

Стрелочный прибор Справка

Копировать Вставить Заменить

Свойство	Значение
Отображаемая величина (<5> Температура рабочая)	0
Положение	По центру
Угол разворота	135
<u>Заголовок</u>	
Полоса	True
Ширина	5
<input type="checkbox"/> Верхний предел шкалы (<5> Температура рабочая)	100
<input type="checkbox"/> HL (<5> Температура рабочая)	90
<input type="checkbox"/> HA (<5> Температура рабочая)	80
<input type="checkbox"/> HW (<5> Температура рабочая)	70
<input type="checkbox"/> LW (<5> Температура рабочая)	50
<input type="checkbox"/> LA (<5> Температура рабочая)	30
<input type="checkbox"/> LL (<5> Температура рабочая)	5
<input type="checkbox"/> Нижний предел шкалы (<5> Температура рабочая)	0
Цвет (HW, LW)	
Цвет (HA, HW), (LW, LA)	
Цвет (HL, HA), (LA, LL)	
Цвет >HL, <LL	
Шкала	True
Указатель	
Единицы	
Индикатор	True
Основная привязка	<5> Температура рабочая

В нижней части экрана разместим ГЭ Тренд для вывода значений аргументов Температура рабочая, Расход теплоносителя и Задание температуры. Основные свойства ГЭ оставим заданными по умолчанию, добавив заголовок Участок термообработки.

Тренд Справка

Копировать Вставить Заменить

Свойство	Значение
Код доступа	0x0
Использовать архив	True
Ориентация	Горизонтально
Цвет фона	
Шрифт	MS Shell Dlg,8
Масштабируемый	False
Заголовок	Участок термообработки
Сетка	
Легенда	
Цвет визира	
Ось времени	
Ось значений	
Буфер	500
Масштаб дискрет (%)	100
Цвета статусов	
* Видимость	True
* Подсказка	
* Слой	Участок термообработки
* Геометрия	Скрыть

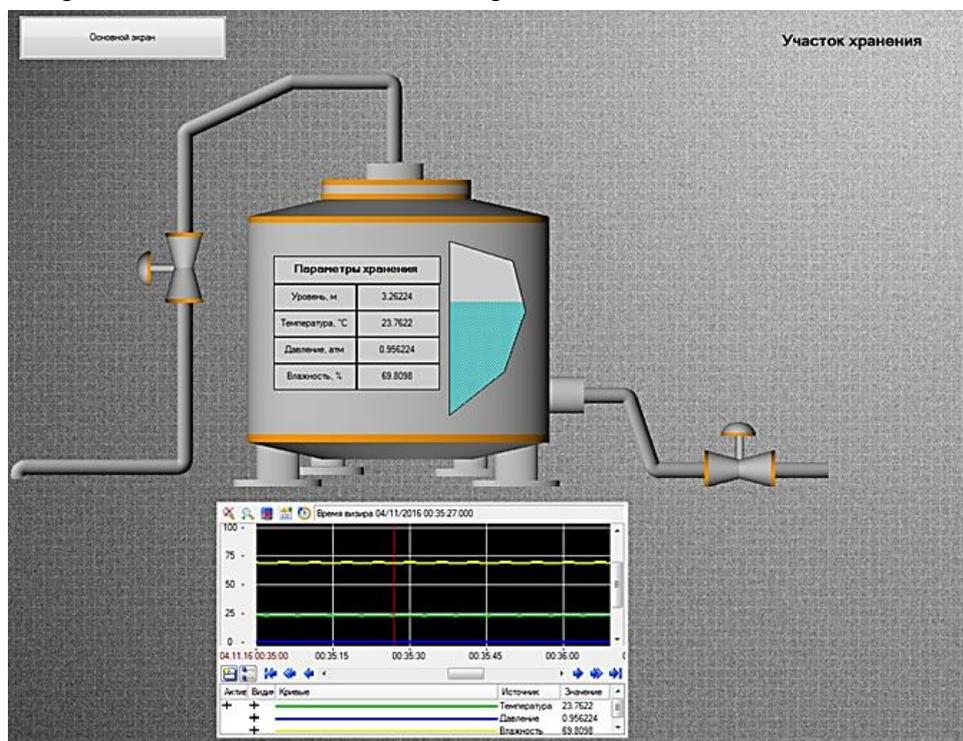
Определим для отображения на тренде три кривые, связав их с соответствующими аргументами экрана, и зададим для них цвет и толщину линий, интервалы выводимых значений.

Тренд Справка	
Копировать Вставить Заменить	
Свойство	Значение
<b>Кривые</b>	
[-] <b>Кривая1 (&lt;5&gt; Температура_рабочая)</b>	
Имя	
Привязка	<5> Температура_рабочая
Скрыть при старте	False
Интерпретировать как	Значение
Цвет	Yellow
Стиль линии	Solid
Толщина линии	3
Тип меток	Нет меток
Формат	%g
Стиль при I<0 и W=0	.....
Стиль при I=0 и W=1	-----
Стиль при I<0 и W=1	.....
Макс. значение	100
Мин. значение	0
Интерполяция	По периоду реального времени
[-] <b>Кривая2 (&lt;4&gt; Расход_теплоносителя)</b>	
Имя	
Привязка	<4> Расход_теплоносителя
Скрыть при старте	False
Интерпретировать как	Значение
Цвет	Green
Стиль линии	Solid
Толщина линии	3
Тип меток	Нет меток
Формат	%g
Стиль при I<0 и W=0	.....
Стиль при I=0 и W=1	-----
Стиль при I<0 и W=1	.....
Макс. значение	100
Мин. значение	0
Интерполяция	По периоду реального времени
[-] <b>Кривая3 (&lt;6&gt; Задание_температуры)</b>	
Имя	
Привязка	<6> Задание_температуры
Скрыть при старте	False
Интерпретировать как	Значение
Цвет	Red
Стиль линии	Solid
Толщина линии	3
Тип меток	Нет меток
Формат	%g
Стиль при I<0 и W=0	.....
Стиль при I=0 и W=1	-----

Разместим справа от ГЭ Тренд ГЭ Ползунок с помощью которого будем задавать величину задания. Свойства ГЭ Ползунок назначим следующим образом:

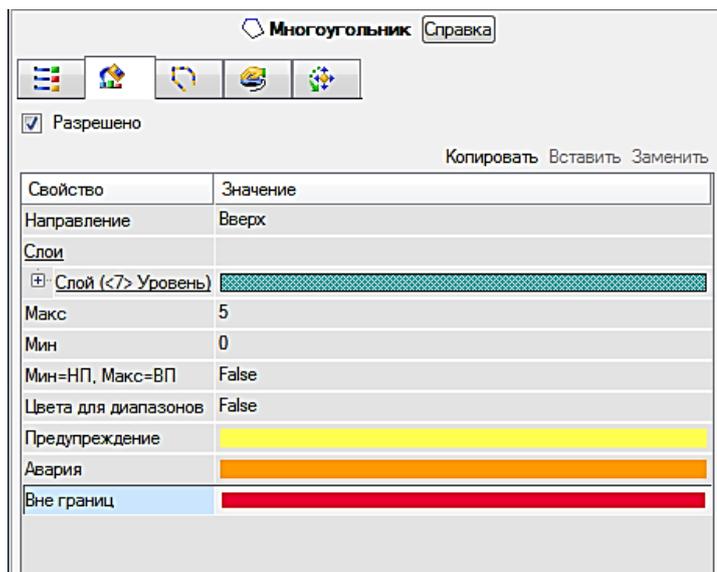
Свойство	Значение
Код доступа	0
Отображаемая величина (<B> Задание температуры)	50
Заданная величина (<B> Задание температуры)	
Положение ползунка	Справа
3D-эффекты	Где
Ползунок	Где
Тип	Квадрат
Цвет	
Размер	31
Длина ручки (%)	50
Полоса	Где
Ширина	10
Верхний предел шкалы (<B> Задание температуры)	100
HL (<B> Задание температуры)	90
HA (<B> Задание температуры)	80
HW (<B> Задание температуры)	70
LW (<B> Задание температуры)	50
LA (<B> Задание температуры)	30
LL (<B> Задание температуры)	5
Нижний предел шкалы (<B> Задание температуры)	0
Цвет (HW, LW)	
Цвет (HA, HW), (LW, LA)	
Цвет (HL, HA), (LA, LL)	
Цвет >HL, <LL	
Шкала	Где
Фон	
Рамка	
* Видимость	Где
* Подсказка	
* Слой	Участок термообработки
* Геометрия	Скрыть
Основная привязка	<B> Задание_температуры

В правом верхнем углу экрана разместим надпись: **Участок термообработки**. Подобным образом создаём слой «Участок хранения»:

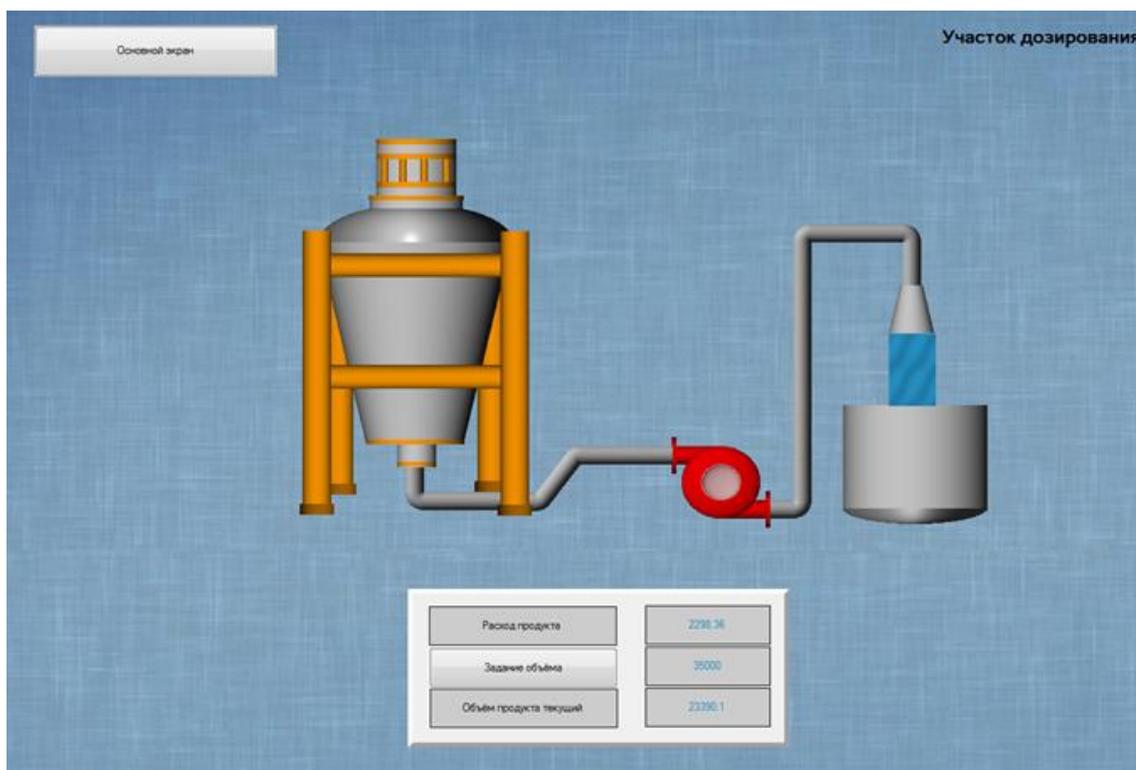


Уровень продукта в емкости будем отображать с помощью гистограммы произвольной формы, которую создадим с помощью ГЭ Многоугольник.

В свойствах для данного ГЭ определим динамическую заливку, привязав ее к соответствующему аргументу экрана, задав цвета фона и заполнения и указав границы:



Далее создаём слой «Участок дозирования»:



Видеоклип, изображающий поток продукта привязан к аргументу экрана следующим образом:

Свойство	Значение
<b>Видеоклип (&lt;11&gt; Насос)</b>	
Привязка	<11> Насос
Прозрачный фон	False
Пауза	0
Показывать при остановке	True
Непрерывное воспроизведение	True
* Видимость	True
* Подсказка	
* Прозрачность	0
* Слой	Участок дозирования
* Выделение в MPB	False
* Геометрия	Скрыть

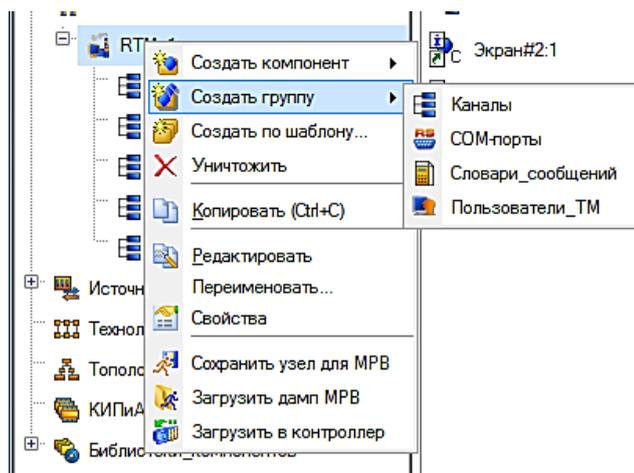
Пояснения требует и ГЭ Кнопка «Задание объема». По нажатию на нем ЛК будет осуществляться посылка значений в два аргумента экрана:

Свойство	Значение
<b>KeyPress</b>	
Подтверждение	False
Сигнал	False
Передать значение	
Тип передачи	Ввести и передать
Значение	0
Результат	<13> Задание_объёма
Источник	...
Восстанавливать значение	False
Передать значение	
Тип передачи	Прямая
Значение	0
Результат	<15> Старт_стоп
Источник	...
Восстанавливать значение	False

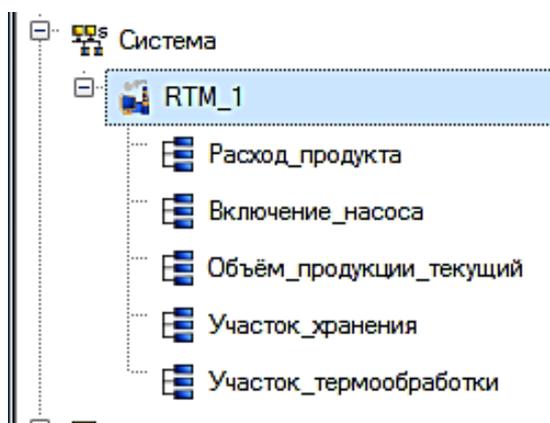
Первое - собственно для задания величины отпускаемого объема продукта, второе - для сброса накопленной в контроллере величины объема по предыдущему циклу розлива.

## Написание программ

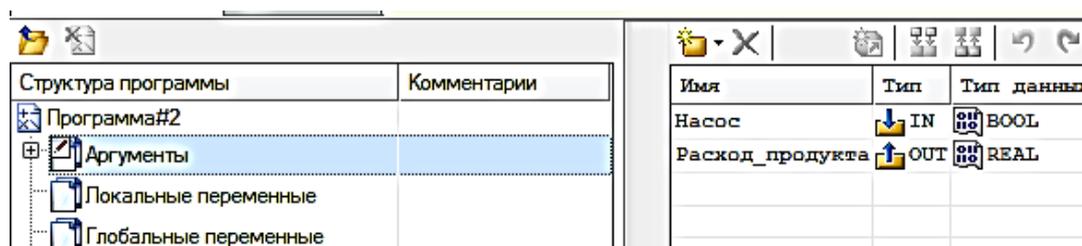
В узле RTM создаём 5 групп «Каналы»:



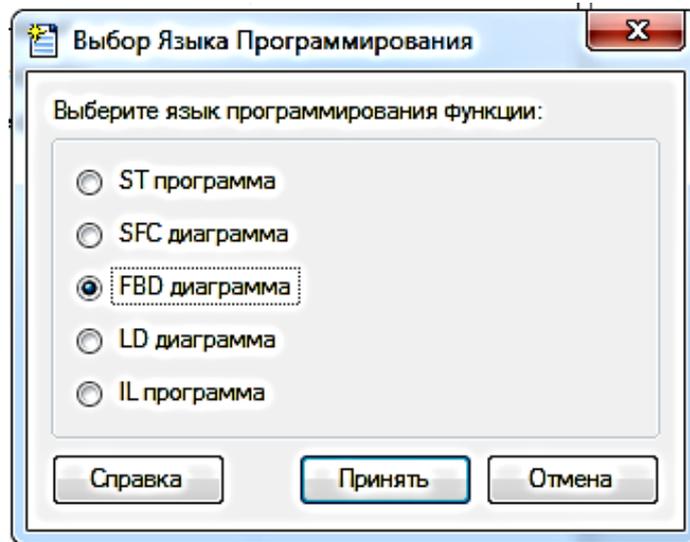
и переименовываем их соответственно:



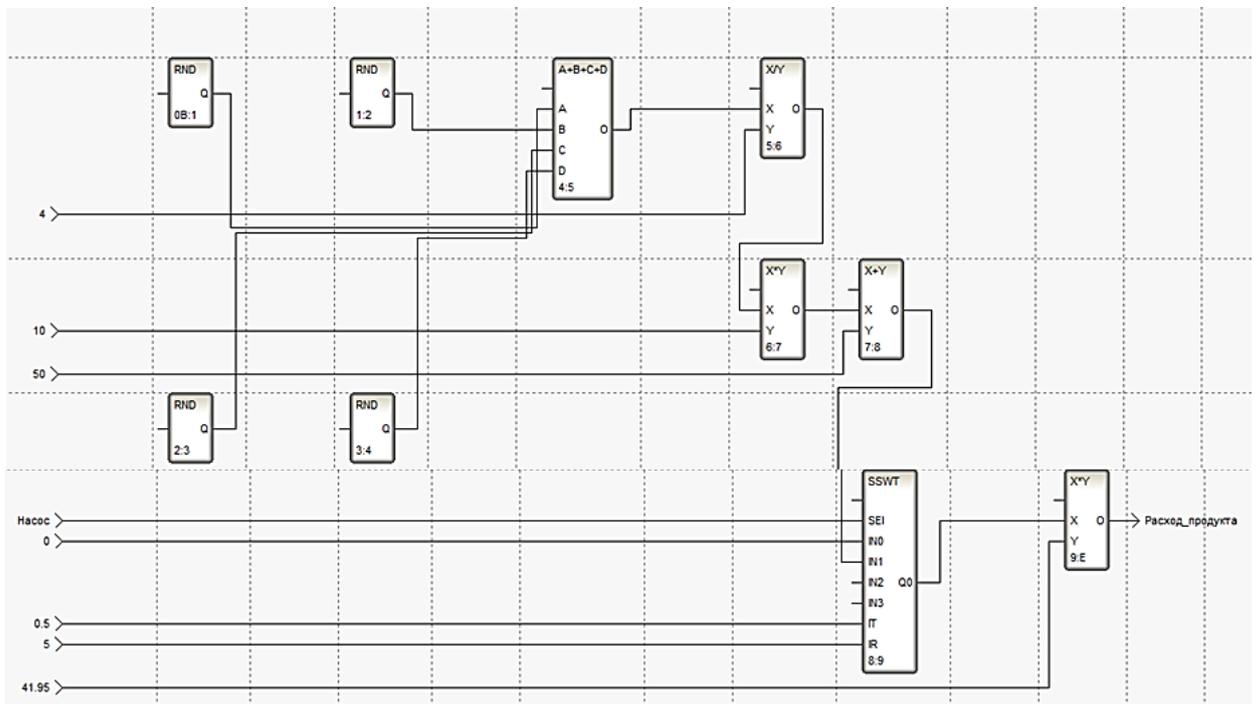
В канале «Расход продукта» создаём компонент «Программа». Двойным щелчком ЛК на компоненте Программа#2 откроем окно редактора программ и, выделив ЛК пункт Аргументы, перейдем в табличный редактор аргументов.



После определения входных и выходных аргументов приступим непосредственно к разработке программы. Для этого выделим ЛК имя созданной программы и в появившемся диалоге выбора языка программирования укажем FBD диаграмму.



В открывшемся окне редактора программ выберем ЛК иконку  для доступа к библиотекам функциональных блоков, далее выбирая ЛК необходимые блоки, перетаскиваем их в рабочее поле редактора, группируем, определяем внутренние связи между входами и выходами блоков, назначаем привязки к аргументам. Готовая программа выглядит следующим образом:

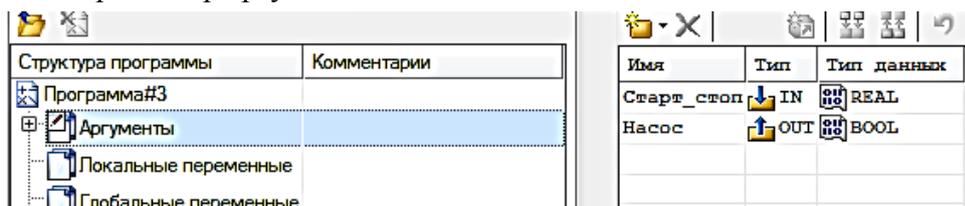


Скомпилируем программу.

Привяжем аргументы программы:

Система.RTM_1.Расход_продукта.Программа#2:3					
Информация		Аргументы		Атрибуты	
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаг
Насос	IN	BOOL		Насос:Реальное значение (Система.RTM_1.Расход_продукта)	
Расход_продукта	OUT	REAL		Расход_продукта:Входное значение (Система.RTM_1.Расход_продукта)	

В канале «Включение насоса» создаём компонент «Программа». Двойным щелчком ЛК на компоненте Программа#3 откроем окно редактора программ и, выделив ЛК пункт Аргументы, перейдем в табличный редактор аргументов.



После определения входных и выходных аргументов приступим к разработке программы. Для этого выделим ЛК имя созданной программы и в появившемся диалоге выбора языка программирования укажем ST программу. Текст программы будет выглядеть следующим образом:

```

PROGRAM
  VAR_INPUT Старт_стоп : REAL; END_VAR
  VAR_OUTPUT Насос : BOOL; END_VAR

  if Старт_стоп==0 then Насос=0;
  else Насос=1;
  end_if;

END_PROGRAM
  
```

Откомпилируем программу для включения в проект нажатием F7. Привяжем аргументы программы:

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Фла
Старт_стоп	IN	REAL		Старт_стоп:Реальное значение (Система.RTM_1.Включение_насоса)	
Насос	OUT	BOOL		Насос:Входное значение (Система.RTM_1.Включение_насоса)	

В канале «Объём продукции текущий» создаём компонент «Программа». Двойным щелчком ЛК на компоненте Программа#4 откроем окно редактора программ и, выделив ЛК пункт Аргументы, перейдем в табличный редактор аргументов.

Имя	Тип	Тип данные
Задание_объёма	IN	REAL
Расход_продукта	IN	REAL
Объём_продукции_текущий	OUT	REAL
Старт_стоп	OUT	REAL

После определения входных и выходных аргументов приступим к разработке программы. Для этого выделим ЛК имя созданной программы и в появившемся диалоге выбора языка программирования укажем ST программу. Текст программы будет выглядеть следующим образом:

```

PROGRAM
  VAR_INPUT Задание_объёма : REAL; END_VAR
  VAR_INPUT Расход_продукта : REAL; END_VAR
  VAR_OUTPUT Объём_продукции_текущий : REAL := 0; END_VAR
  VAR_OUTPUT Старт_стоп : REAL; END_VAR

  if Расход_продукта==0 then Объём_продукции_текущий=0;Старт_стоп = 0;
  else Объём_продукции_текущий = Объём_продукции_текущий + Расход_продукта;
  end_if;
  if Объём_продукции_текущий < (Задание_объёма-7000) then Старт_стоп = 1;
  else Старт_стоп = 0;
  end_if;

END_PROGRAM

```

Откомпилируем программу для включения в проект нажатием F7. Привяжем аргументы программы:

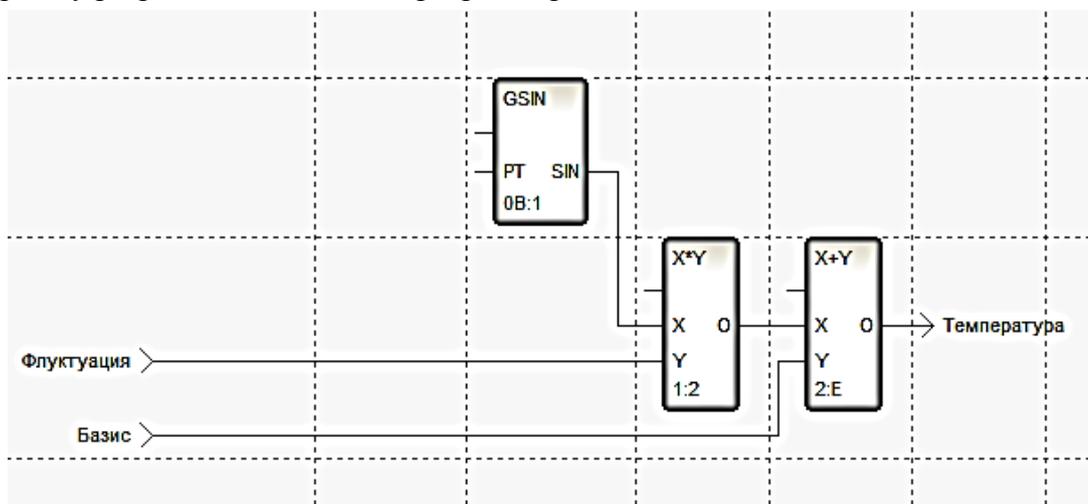
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флг
Задание_объёма	IN	REAL		Задание_объёма:Реальное значение (Система.RTM_1.Объём_продукции_текущий)	
Расход_продукта	IN	REAL		Расход_продукта:Реальное значение (Система.RTM_1.Объём_продукции_текущий)	
Объём_продукции_текущий	OUT	REAL		Объём_продукции_текущий:Входное значение (Система.RTM_1.Объём_продукции_текущий)	
Старт_стоп	OUT	REAL		Старт_стоп:Входное значение (Система.RTM_1.Объём_продукции_текущий)	

В канале «Участок хранения» создаём 4 компонента «Программа». Переименовываем их в «Температура», «Давление», «Влажность» и «Уровень».

Двойным щелчком ЛК на компоненте Программа#5 откроем окно редактора программ и, выделив ЛК пункт Аргументы, перейдем в табличный редактор аргументов. Объявим аргументы:

Имя	Тип	Тип данных	Значение по умолчанию
Температура	IN/OUT	REAL	
флуктуация	IN	REAL	0.25
Базис	IN	REAL	24

Программу разработаем на языке программирования Техно FBD.



Изменяющийся по синусоидальному закону «случайный» компонент в диапазоне (-1,1) будем масштабировать, задавая аргумент Флуктуация, и добавлять к аргументу Базис, определяющему установленное значение.

Откомпилируем программу нажатием функциональной клавиши F7.

Привяжем аргументы программы к атрибуту канала и зададим константы в столбце **Значения по умолчанию**:

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Температура	IN/OUT	REAL		Температура:Входное значение (Система.RTM_1.Участок_хранения)
Флуктуация	IN	REAL	0.25	
Базис	IN	REAL	24	

Подобным образом поступим в отношении программ «Давление», «Влажность» и «Уровень», указав для них, соответственно аргументы Флуктуация и Базис как (0.025,0.98), (0.2,70) и (0.25, 3.5).

В канале «Участок термообработки» создаём компонент «Программа». Двойным щелчком ЛК на компоненте Программа#9 откроем окно редактора программ и, выделив ЛК пункт Аргументы, перейдем в табличный редактор аргументов.

Имя	Тип	Тип данных
Входной_сигнал	IN	REAL
Задание_температуры	IN	REAL
Расход_теплоносителя	OUT	REAL
Рабочая_температура	OUT	REAL

После определения входных и выходных аргументов приступим к разработке программы. Укажем ST программу. Текст программы будет выглядеть следующим образом:

```

PROGRAM
  VAR_INPUT Входной_сигнал : REAL; END_VAR
  VAR_INPUT Задание_температуры : REAL; END_VAR
  VAR_OUTPUT Расход_теплоносителя : REAL; END_VAR
  VAR_OUTPUT Рабочая_температура : REAL; END_VAR

  Рабочая_температура=(Входной_сигнал-50)*0.8+Задание_температуры;
  Расход_теплоносителя=Рабочая_температура*(-1)+Задание_температуры*2;

END_PROGRAM

```

Откомпилируем программу. Привяжем аргументы программы:

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Входной_сигнал	IN	REAL		Входной_сигнал:Реальное значение (Система.RTM_1.Участок_термообработки)
Задание_температуры	IN	REAL		Задание_температуры:Реальное значение (Система.RTM_1.Участок_термообработки)
Расход_теплоносителя	OUT	REAL		Расход_теплоносителя:Входное значение (Система.RTM_1.Участок_термообработки)
Рабочая_температура	OUT	REAL		Рабочая_температура:Входное значение (Система.RTM_1.Участок_термообработки)

После разработки программ, привяжем аргументы программ между собой соответственно и к аргументам экрана.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Основной_экран	IN	USINT		
Участок_термообработки	IN	USINT		
Участок_хранения	IN	USINT		
Участок_дозирования	IN	USINT		
Расход_теплоносителя	IN/OUT	REAL		Расход_теплоносителя:Входное значение (Система.RTM_1.Участок_термообработки)
Температура_рабочая	IN/OUT	REAL		Рабочая_температура:Входное значение (Система.RTM_1.Участок_термообработки)
Задание_температуры	IN/OUT	REAL		Задание_температуры:Входное значение (Система.RTM_1.Участок_термообработки)
Уровень	IN/OUT	REAL		Уровень:Входное значение (Система.RTM_1.Участок_хранения)
Температура	IN/OUT	REAL		Температура:Входное значение (Система.RTM_1.Участок_хранения)
Давление	IN/OUT	REAL		Давление:Входное значение (Система.RTM_1.Участок_хранения)
Влажность	IN/OUT	REAL		Влажность:Входное значение (Система.RTM_1.Участок_хранения)
Насос	IN/OUT	BOOL		Насос:Входное значение (Система.RTM_1.Расход_продукта)
Расход_продукта	IN/OUT	REAL		Расход_продукта:Входное значение (Система.RTM_1.Расход_продукта)
Задание_объёма	IN/OUT	REAL		Задание_объёма:Входное значение (Система.RTM_1.Объём_продукции_текущий)
Объём_продукта_текущий	IN/OUT	REAL		Объём_продукции_текущий:Входное значение (Система.RTM_1.Объём_продукции_текущий)
Старт_стоп	IN/OUT	REAL		Старт_стоп:Входное значение (Система.RTM_1.Включение_насоса)

### Запуск проекта:

- сохраним проект;
- на инструментальной панели выберем ЛК иконку «Сохранить для MPB» и подготовим тем самым проект для запуска в реальном времени;
- с помощью иконки «Запустить профайлер» на инструментальной панели запустим режим исполнения.

### Порядок выполнения практической работы:

1. Провести анализ задания практической работы.
2. Спроектировать мнемосхему системы контроля и управления ТП в среде TRACE MODE 6.

### Отчет по работе должен содержать:

1. Программу мнемосхемы системы контроля и управления ТП разработанную в среде TRACE MODE 6 в формате \*.prj

### Вопросы для самопроверки:

1. Опишите структуру и состав системы TRACE MODE 6.
2. Перечислите графические элементы системы TRACE MODE 6.
3. Какие модифицированные языки поддерживает система TRACE MODE 6?

### Средства обучения:

1. **Оборудование и материалы:** Раздаточный материал
2. **Вербальные средства обучения:**

Видеоуроки: начало работы в SCADA TRACE MODE 6. Режим доступа: [http://www.adastra.ru/support/get\\_support/first\\_step/](http://www.adastra.ru/support/get_support/first_step/)

3. **Технические средства обучения (ТСО):** ПК, SCADA TRACE MODE 6.