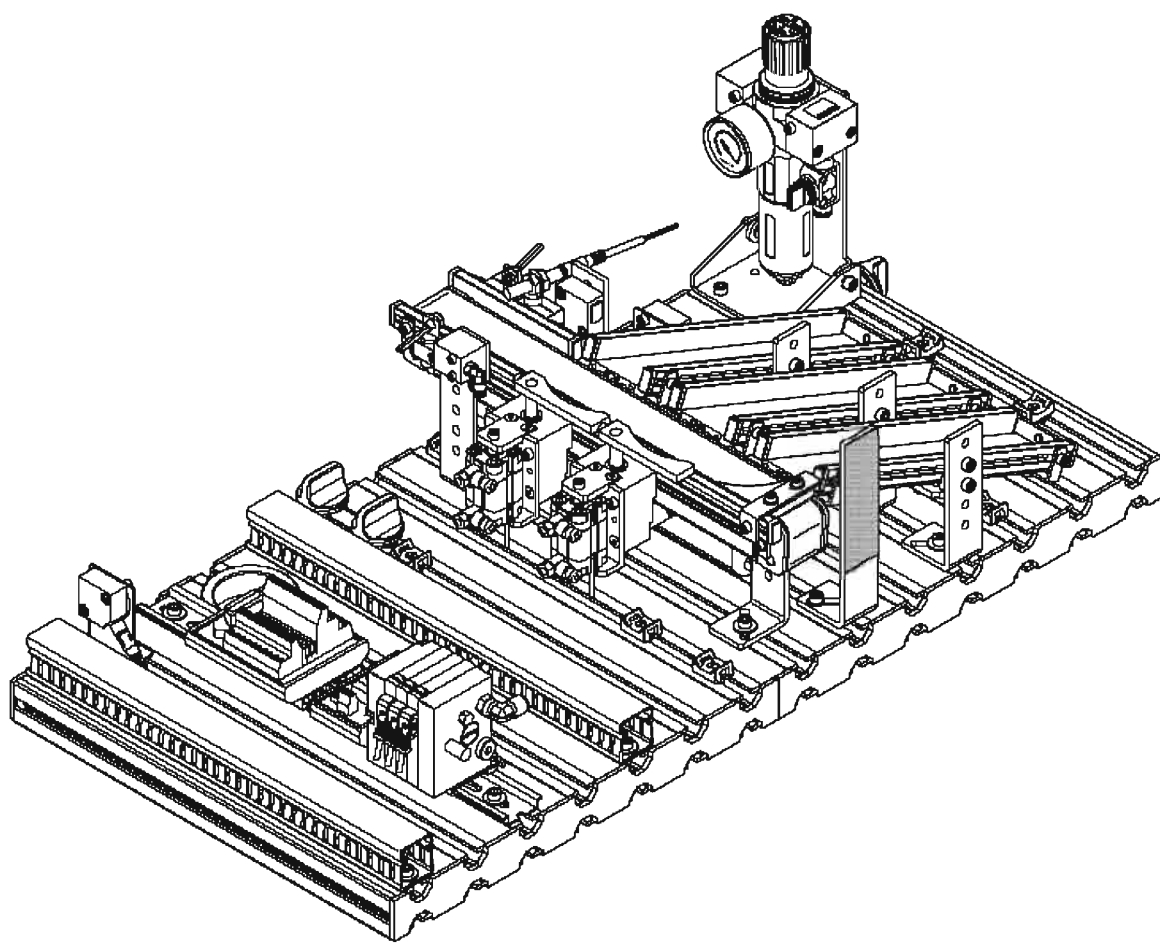


**И.В. РЕМИЗОВА А.И. НОВИКОВ**

**ПРОГРАММИРОВАНИЕ  
И НАЛАДКА ЛОГИЧЕСКОГО  
КОНТРОЛЛЕРА ФИРМЫ FESTO**

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ**



**Санкт-Петербург  
2009**

## ВВЕДЕНИЕ

Качество промышленного производства сегодня определяется не только степенью автоматизации, но и квалификацией персонала. Инвестиции как в автоматизацию, так и в подготовку специалистов в области автоматизации создают надежную основу для достижения прибыли.

Автоматизация технологических процессов – наукоемкая сфера деятельности, сложная для понимания и требующая особых подходов при обучении. Известно, что эффективное обучение предполагает максимальное использование полученных знаний на практике. Это возможно при применении обучающего оборудования, которое позволяет имитировать реальные производственные процессы в сочетании с современными техническими средствами.

Настоящий практикум предназначен для лабораторных работ по курсу «Автоматизированные системы управления работой оборудования». Лабораторные работы направлены на ознакомление и получение студентами начальных навыков программирования логических контроллеров. Используется лабораторный стенд сортировочной станции на базе контроллера FESTO.

Международный концерн FESTO является одним из крупнейших в области пневматической и электронной техники автоматизации, который, кроме того, занимается разработкой и внедрением обучающих программ и оборудования.

Дидактическая обучающая автоматизированная система FESTO спроектирована с учетом ряда тренировочных и профессиональных требований. Системы и станции Модульной Системы Производства (MPS) облегчают промышленно-ориентированные профессиональные тренировки, поскольку оборудование состоит из пригодных к обучению промышленных компонентов.

# 1. ОПИСАНИЕ ЛАБОРАТОРНОГО СТЕНДА

## 1.1. Сортировочная станция

Сортировочная станция (см. рис.1.1) предназначена для сортировки символических заготовок по материалу и цвету, по трем желобам.

Представленный стенд является макетом реальной промышленной установки, что позволяет студентам практически увидеть работу сортировочной станции. Стенд содержит датчики, исполнительные механизмы, счетчики, контроллер. В каждой последующей лабораторной работе студентам дается задание более сложное, чем предыдущее, таким образом, в лабораторных работах задействуются все больше элементов стенда. Нахождение на данном стенде перепрограммируемого контроллера дает возможность написания и загрузки программ в контроллер самими студентами, что в свою очередь положительно скажется на профессиональной подготовке будущих специалистов.

Оптический датчик обнаружения определяет, когда заготовка попала в начало конвейера.

Характеристики заготовок (красный, черный, металл) определяются сенсорами спереди ограничителя, после чего заготовки сортируются в соответствующие желоба через ветвления. Ветви двигаются посредством цилиндров короткого удара с использованием реверсирующего механизма.

Оптический датчик заполнения следит за уровнем заполнения желобов.



Рис.1.1. Сортировочная станция с консолью управления

В сортировочной станции, символические заготовки сортируются по материалу и цвету.

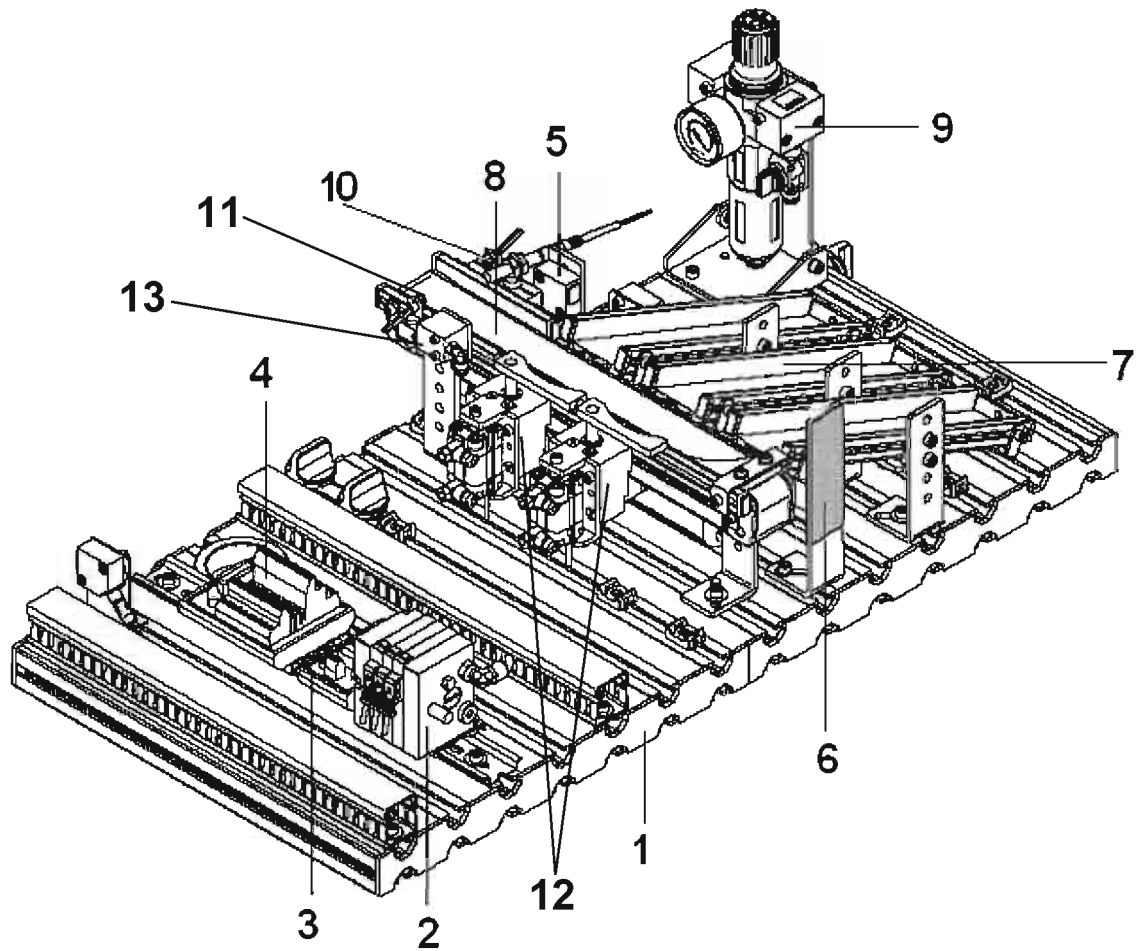


Рис.1.2. Состав сортировочной станции:

- 1 – профильная панель
- 2 – электропневматический преобразователь
- 3 – пусковой предохранитель
- 4 – терминал ввода/вывода
- 5 – датчик заполнения
- 6 – внешний отражатель
- 7 – модуль желобов
- 8 – модуль конвейера
- 9 – индикатор давления
- 10 – датчики металла и цвета
- 11 – датчик обнаружения
- 12 – пневмоцилиндры с ветвями
- 13 – стопор

## 1.2. Электропневматический преобразователь

Электропневматический преобразователь (ЭПП) (рис. 1.3) предназначен для пропорционального преобразования непрерывного

электрического сигнала постоянного тока в унифицированный пневматический сигнал. Он применяется в пневматических системах регулирования и контроля производственных процессов.

Технические характеристики ЭПП представлены в табл. 1.1.



Рис.1.3. Электропневматический преобразователь

Технические характеристики ЭПП  
(CPV-10VI 70332)

Таблица 1.1

Параметр	Значение
Управляющее напряжение	24 вольт
Рабочее давление	600 кПа
Порт подача сжатого воздуха	D=6мм
Рабочий порт подключения	D=4мм

### 1.3. Датчики

#### Оптический датчик заполнения с отражателем

Датчик заполнения оптический с отражателем (рефлектором) (рис.1.4) используется для наблюдения за уровнем заполнения желобов.

Он состоит из передатчика и приемника в одном корпусе и внешнего отражателя. Устройство излучает видимый красный свет, отражаемый внешним отражателем.

Технические характеристики датчика заполнения оптического с рефлектором представлены в табл. 1.2.



Рис.1.4. Оптический датчик заполнения с рефлектором

Технические характеристики оптического датчика  
заполнения с рефлектором  
(SOEG-RSP-Q30-PS-S-2L)

Таблица 1.2

Параметр	Значение
1	2
Рабочее расстояние до объекта	0 - 2000 мм
Цвет луча	Красный, поляризованный

1	2
Размер	30x30x15 мм
Разъем для электрического соединения	3-пиновый; М8х1
Защищен от смены полярности для всех электрических соединений	Да
Индикатор рабочего состояния	Желтый светодиод
Резервный рабочий индикатор	Зеленый светодиод
Максимальная частота переключения	1000 Гц
Питание постоянного тока (DC)	10 - 30 В
Максимальный выходной ток	200 мА
Класс защиты корпуса	IP65
Рабочий диапазон температур	-5 ... +55 °С
Вес продукта	18 г
Тип крепления	со сквозным отверстием

### Внешний отражатель (рефлектор)

Внешний отражатель (рис.1.5) служит для отражения видимого красного света, испускаемого датчиком заполнения.

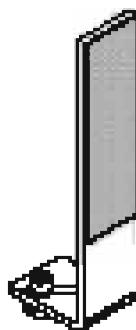


Рис.1.5. Внешний отражатель



## Датчик металла

Датчик металла (рис.1.6) используется для обнаружения металлических объектов. Рабочее расстояние до объекта зависит от материала и чистоты поверхности исследуемого объекта.

Технические характеристики датчика металла представлены в табл. 1.3.



Рис.1.6. Датчик металла

### Технические характеристики датчика металла

(SIEN-M8NB-PS-S-L)

Таблица 1.3

Параметр	Значение
1	2
Рабочее расстояние до объекта	0 - 2,5 мм
Разъем для электрического соединения	3-пиновый; M8x1
Индикатор рабочего состояния	Желтый светодиод
Максимальная частота переключения	900 Гц
Защищен от смены полярности для всех электрических соединений	Да
Питание постоянного тока (DC)	15 - 34 В

1	2
Максимальный выходной ток относительно температуры	200 мА при $\leq 85^{\circ}\text{C}$ 150 мА при $\leq 85^{\circ}\text{C}$
Класс защиты корпуса	IP67
Рабочий диапазон температур	$-25 \dots +85^{\circ}\text{C}$
Вес продукта	20 г
Тип крепления	с контргайкой
Информация о материале корпуса	Легированная сталь, нержавеющая

### Оптический датчик цвета

Датчик цвета оптический (рис.1.7) используется для распознавания цветов заготовок.



Рис. 1.7. Датчик цвета

Волоконно-оптический кабель (рис.1.8) соединяется с волоконно-оптическим устройством. Оно излучает видимый красный свет. Датчик цвета обнаруживает свет, отраженный от заготовки. Поверхности разных цветов в различной степени отражают падающий на них свет.

Технические характеристики датчика цвета представлены в табл.1.4.

Технические характеристики оптического датчика цвета  
(SOEG-L-Q30-PA-5-2C)

Таблица 1.4

Параметр	Значение
Размер	30x30x15 мм
Рабочее расстояние до объекта	0 - 120 мм
Разъем для электрического соединения	4-пиновый
Защищен от смены полярности для всех электрических соединений	Да
Индикатор рабочего состояния	Желтый светодиод
Резервный рабочий индикатор	Зеленый светодиод
Максимальная частота переключения	1000 Гц
Питание постоянного тока (DC)	10 - 30 В
Максимальный выходной ток	200 мА
Класс защиты корпуса	IP65
Рабочий диапазон температур	-5 ... +55 °С
Вес продукта	18 г
Тип крепления	со сквозным отверстием

## Волоконно-оптический кабель

Волоконно-оптический кабель используется для передачи светового луча на расстояние. Позволяя, таким образом, разместить оптического датчика цвета на расстоянии от объекта измерения.

Технические характеристики волоконно-оптического кабеля представлены в табл. 1.5.



Рис.1.8. Волоконно-оптический кабель

### Технические характеристики волоконно-оптического кабеля (SOEZ-LLK-RT-2,0-M6)

Таблица 1.5

Параметр	Значение
Цвет луча	Красный
Рабочее расстояние до объекта	0 - 120мм
Рабочий диапазон температур	-40 ... +70 °С
Материал корпуса	латунь
Покрытие	никель
Вес	0,02кг
Класс защиты корпуса	IP65

## Датчик обнаружения

Датчик обнаружения оптический (рис.1.9) используется для обнаружения заготовок в начале конвейера.



Рис.1.9. Датчик обнаружения

Волоконно-оптический кабель (рис.1.8) подсоединен к волоконно-оптическому устройству. Оно излучает видимый красный свет. Датчик обнаружения распознает свет, отраженный от заготовки.

Технические характеристики датчика обнаружения представлены в табл. 1.6.

Технические характеристики датчика положения оптического  
(SOEG-L-Q30-PA-S-2L)

Таблица 1.6

Параметр	Значение
1	2
Размер	30x30x15 мм
Рабочее расстояние до объекта	0 - 120 мм

1	2
Разъем для электрического соединения	4-пиновый
Защищен от смены полярности для всех электрических соединений	Да
Индикатор рабочего состояния	Желтый светодиод
Резервный рабочий индикатор	Зеленый светодиод
Максимальная частота переключения	1000 Гц
Питание постоянного тока (DC)	10 - 30 В
Максимальный выходной ток	200 мА
Класс защиты корпуса	IP65
Рабочий диапазон температур	-5 ... +55 °С
Вес продукта	18 г
Тип крепления	со сквозным отверстием

#### 1.4. Устройство регулирования давления

Устройство регулирования давления (рис.1.10) предназначено для подготовки сжатого воздуха к работе в установке. Оно оборудовано воздушным фильтром и аналоговым манометром.

Технические характеристики регулятора давления представлены в табл. 1.7.



Рис.1.10. Устройство регулирования давления  
(LRF-1/8-D-MINI)

Технические характеристики устройства регулирования давления

Таблица 1.7

Параметр	Значение
Рабочая среда	Сжатый воздух
Монтажное положение	$\pm 5^\circ$ (от вертикали)
Расход воздуха	750 л/мин
Максимальное верхнее давление	1600 кПа
Максимальное рабочее давление	1200 кПа

## 1.5. Контроллер FESTO FEC-FC-640-FST

Контроллер FESTO FEC-FC-640-FST (рис.1.11), обладает всеми качествами, которыми должен обладать мини-контроллер: простотой программного управления, при одновременном наличии сложной программы и обширной памяти данных.



Рис. 1.11. Внешний вид контроллера FESTO FEC-FC-640-FST

Контроллер FESTO FEC-FC-640-FST небольших компактных размеров. С помощью разъемов (рис.1.12) на лицевой стороне датчики и исполнительные устройства подключаются без промежуточных клеммников, что экономит время монтажа и пространство в шкафу управления.

Технические характеристики рассматриваемого контроллера FESTO представлены в табл. 1.8.



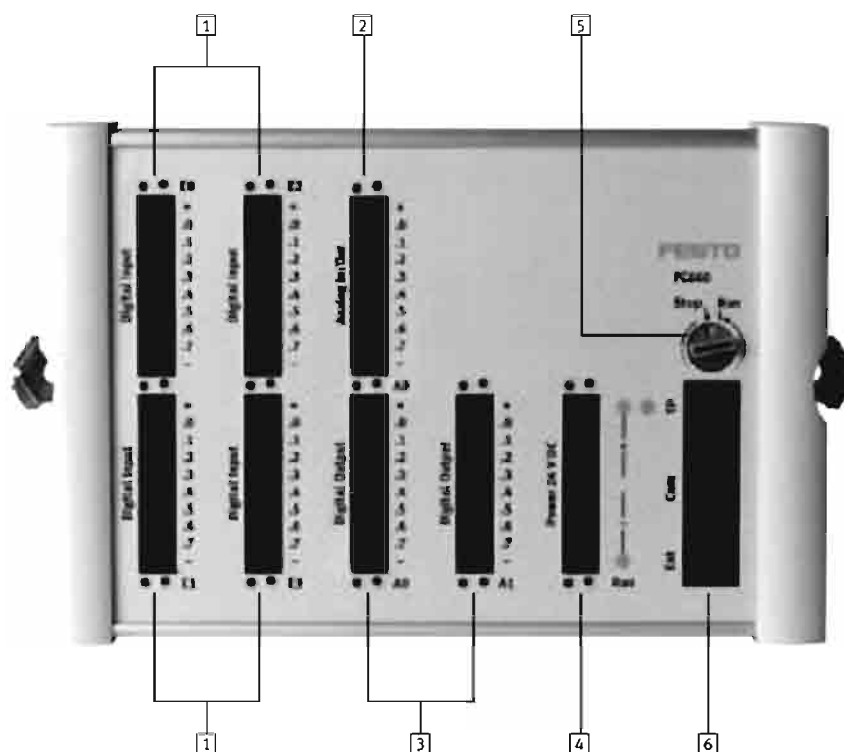


Рис.1.12. Назначение разъемов на лицевой панели контроллера

FESTO FEC-FC-640-FST:

- 1 – 32 цифровых входа, 24В постоянного тока;
- 2 – 3 аналоговых входа / 1 аналоговый выход (*в рассматриваемой модели FEC-FC-640 отсутствуют*);
- 3 – 16 цифровых выходов;
- 4 – электропитание;
- 5 – поворотный переключатель RUN/STOP (Работа/Остановка);
- 6 – два последовательных интерфейса, поддержка (подключение к сети) Ethernet.

**Аппаратная часть:**

FESTO FEC-FC-640-FST имеет зажим для крепления на DIN-рейке (рис.1.13) и отверстия в углах для болтов при использовании монтажной плиты. Все подключения спереди; отсутствует необходимость в дополнительном пространстве сверху или снизу.



Рис.1.13. Крепление контроллера FESTO FEC-FC-640-FST

### **Последовательные интерфейсы:**

FESTO FEC-FC-640-FST имеет два последовательных интерфейса COM и EXT. Они являются универсальными TTL-интерфейсами с максимальной скоростью передачи данных 115КБит/с. В зависимости от потребностей, их можно использовать как RS232с или RS485 интерфейсы. Рассматриваемая лабораторная установка соединяется с персональным компьютером по RS2232с.

### **Электропитание:**

Питание контроллера FESTO FEC-FC-640-FST осуществляется постоянным напряжением 24В. Питание для входных сигналов составляет 24В постоянного тока. Выходные сигналы – 24В постоянного тока (400мА). Предусмотрена защита от короткого замыкания и низкоомных нагрузок.

### **Сетевой интерфейс:**

FESTO FEC-FC-640-FST включает в себя интерфейс Ethernet 10BaseT с разъемом RJ45 и скоростью передачи данных 10МБит/с (рис.1.14).

Комбинированный светодиод “Link/Active (Сеть/В действии)” отображает статус соединения. Контроллер поддерживает передачу данных, программирование и поиск неисправностей через Ethernet-интерфейс.



Рис.1.14. Подключение контроллера FESTO к Ethernet

Технические характеристики контроллера FESTO FEC-FC-640-FST

Таблица 1.8

Параметр	Значение
1	2
Рабочее напряжение постоянного тока (DC)	24 V (+25% / -15%)
Потребление энергии	< 5Вт
Рабочий диапазон температур	0 ... 55 °С
Относительная влажность воздуха	0 ... 95%
Класс защиты корпуса	IP20
Вес продукта	500 г
Поворотный переключатель STOP/RUN (СТОП/РАБОТА)	0 = Стоп (STOP) 1 ... F = РАБОТА (RUN)
Подключение входов выходов	Система с пружиной натяжения
Отображение состояния	LED

Продолжение табл.1.8

1	2
Цифровые входы, количество	32
Цифровые входы, задержка входного сигнала	5 ms
Цифровые входы, входное напряжение/ток	24 В постоянного тока (DC) / 5 мА
Цифровые входы, номинальное значение для True (истинный)	$\geq 15$ В постоянного тока (DC)
Цифровые входы, номинальное значение для False (ложный)	$\leq 5$ В постоянного тока (DC)
Цифровые входы, электрическая изоляция через оптосоединение	Да
Цифровые выходы, количество	16
Цифровые выходы, выходное напряжение/ток	24 В постоянного тока (DC) / 400 мА
Цифровые выходы, суммарный ток	3,2 А
Цифровые входы, электрическая изоляция через оптосоединение	Да
Цифровые выходы, защита от короткого замыкания	Да
Цифровые выходы, защита от перегрузки	Да
Цифровые выходы, стойкость к нагрузкам с низким сопротивлением	Да, до 5 Вт
Последовательный интерфейс, количество	2

1	2
Последовательный интерфейс, подключение	штекерная розетка RJ12
Последовательный интерфейс, характеристики COM / EXT	Последовательный, асинхронный, уровень TTL, без электрической изоляции
Ethernet, количество	1
Ethernet, подключение	штекерная розетка RJ45
Ethernet, скорость передачи данных	10 Мбит/с
Ethernet, поддерживаемые протоколы	TCP/IP, EasyIP, http(только FST)
Программирование, количество программ и задач на проект	64(0...63)
Программирование, количество регистров (слов)	256(0...255) адресуемые как слова
Программирование, количество таймеров и счетчиков	256(0...255)
Программирование, поддержка для C/C++ и FST	Да

## 2. ЯЗЫК ОПИСАНИЯ ЗАДАЧ FST

Контроллер FESTO FEC-FC-640-FST программируется с помощью FST 4.02.21 DEMO фирмы FESTO (рис.2.1). FST является уникальным языком программирования, имеющим богатые традиции и очень легким в применении, позволяющим осуществлять программирование "как думаешь": ЕСЛИ...ТО...ИНАЧЕ. FST также поддерживает пошаговые операции для последовательного

программирования. FST можно использовать для программирования через Ethernet; также доступен веб-сервер.

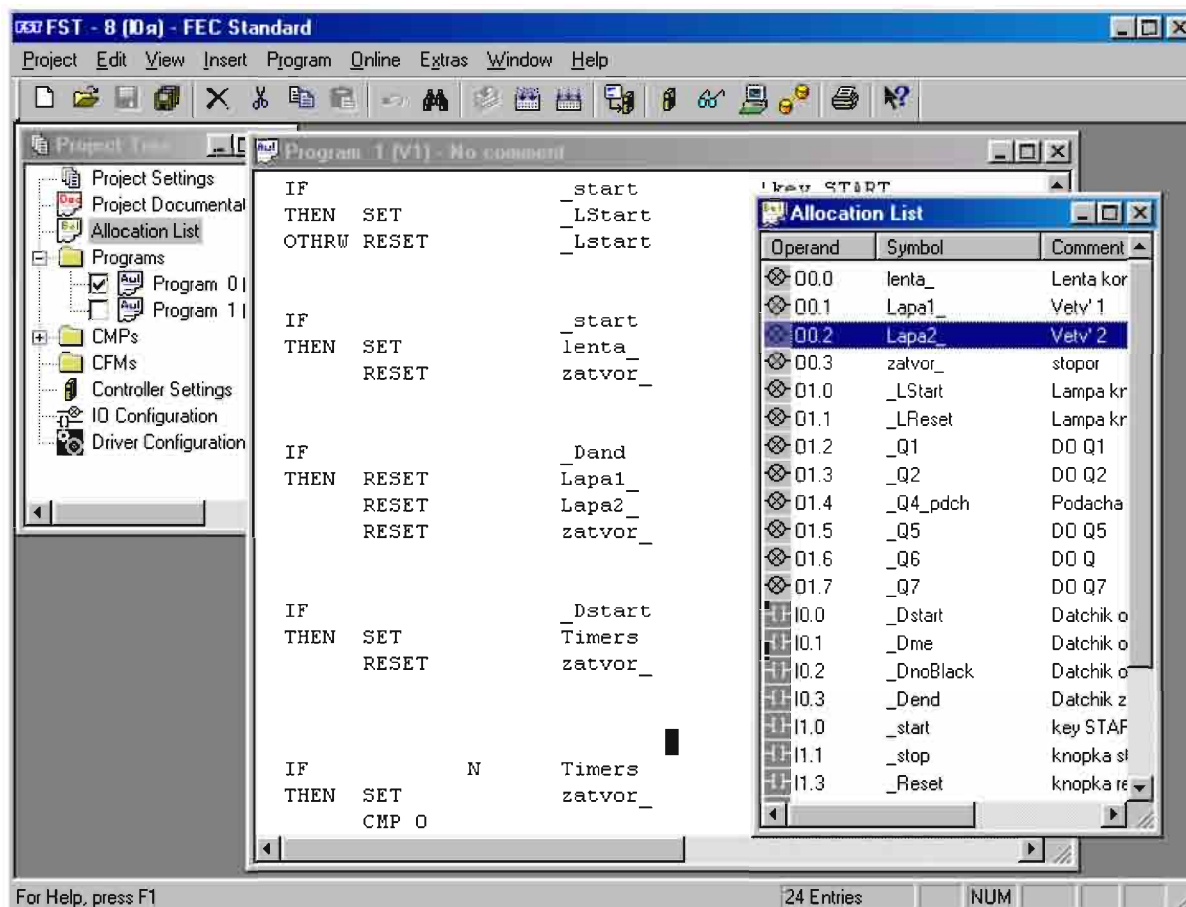


Рис.2.1. Внешний вид программы FST 4.02.21

## 2.1. Характеристики программы

Программирование в данном пакете может осуществляться на языке FST с помощью команд (табл. 2.1).

Программирование осуществляется путем ввода команд в окне программы. Команды также могут быть выбраны из окна быстрого ввода команд (рис. 2.2).

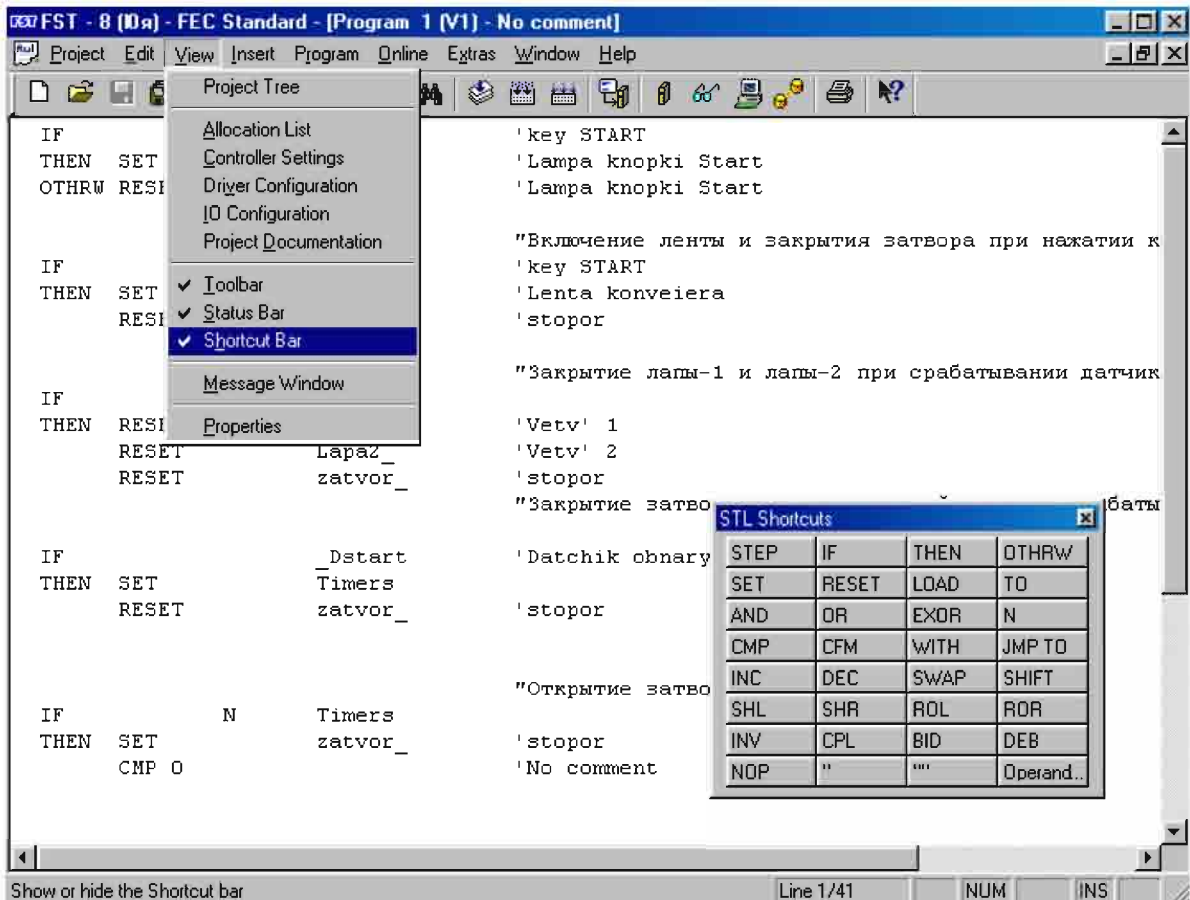


Рис.2.2. Программа FST 4.02.21 с открытым окном быстрого ввода команд

## 2.2. Список команд

Список доступных для использования команд и их описание приведены в табл. 2.1. Пример использования команд приведен в разделе 3.3.

### Описание команд

**STEP** Оператор STEP (*шаг*) весьма важен в последовательности команд, так как STEP определяет структуру программы или (при наличии ветвления),

последовательность выполнения.

Оператор STEP должен сопровождаться меткой шага, содержащей максимум 9 символов или номеров. Метка шага необходима при наличии ветвлений в программе и переходом на этот шаг.

**Пример:**

```

STEP      Setup
...
...
THEN     JMP TO Setup
    
```

где Setup – метка шага.

Во время процесса трансляции шаги внутренне перенумеровываются, и их номер начинается с 1 и т.д. до максимального номера. В программе FST шаг не выполняется, пока не будет выполнено условие THEN или OTHRW в последнем предложении перед этим шагом.

**IF** IF (*если*) всегда представляет условную часть. Операнды будут опрошены и связаны с логическими операторами в предложении.

**Пример:**

```

IF              I1.0
                AND N   I1.1
...
    
```

где I1.0 и I1.1 – входные переменные

**THEN** THEN (*тогда*) представляет исполнительную часть. Эта часть выполняется, если условие есть *истина*. Оператор может содержать команды для модификации выходов или флагов и т.д., выполнение арифметических



операций, активизации таймеров или счетчиков, вызова других программ или программных модулей.

**Пример:**

```

THEN   LOAD   V100
        TO     TP7
...

```

Значение 100 загружается в переменную TP7

**OTHRW** OTHRW (*другой путь*) представляет вторую, альтернативную исполнительную часть. Эта часть используется, если результат в условной части шага не *истина* и условие THEN не может быть выполнено.

**Пример:**

```

...
THEN   SET     O1.0
OTHRW  RESET   O1.0

```

где O1.0 – выходная переменная

**AND** AND (*и*) – логический оператор. Позволяет проводить логическое суммирование большое количество входных условий. Условия выполняются, когда результат логического суммирования есть *истина*.

**Пример:**

```

IF                I1.0
                AND    I1.1
THEN   SET     O1.0
OTHRW  SET     O1.7

```

где I1.0, I1.1 – входные переменные,  
O1.0, O1.7 – выходные переменные.

**OR** OR (*или*) – логический оператор. Позволяет логически сравнить несколько входных условий. Условие

выполняется, когда минимум одно условие является *истиной*.

**Пример:**

IF		I1.0
	<b>OR</b>	I1.1
	<b>OR</b>	I1.7
THEN	SET	O1.0
OTHRW	SET	O1.7

где I1.0, I1.1, I1.7 – входные переменные,  
O1.0, O1.7 – выходные переменные.

**EXOR** EXOR (*исключающее или*) – логический оператор. Позволяет логически сравнить два (и только два) входных условия. Условие выполняется, когда только одно исходное условие есть *истина*.

**Пример:**

IF		I1.0
	<b>EXOR</b>	I1.1
THEN	SET	O1.0
OTHRW	SET	O1.7

где I1.0, I1.1 – входные переменные,  
O1.0, O1.7 – выходные переменные.

**NOP** NOP означает *No OPeration*. Вы можете ввести этот оператор, когда вводите исполнительную часть без входного условия.

**Пример:**

IF		<b>NOP</b>
THEN	SET	O1.0

где O1.0 – выходная переменная

**NOT** NOT (*не*) – отрицание, инвертирующее входное условие.

В показанном примере: если выход O1.0 не активен, то программа переходит к шагу *Setup*.

**Пример:**

```
IF          N          O1.0
THEN       JMP TO     Setup
```

где O1.0 - выходная переменная

**SET** SET активизирует однобитовый операнд. SET используется, например, при установки выхода в логическую единицу.

**Пример:**

```
...       SET          O1.0
```

где O1.0 - выходная переменная

**RESET** RESET есть противоположность SET и деактивирует однобитовый операнд. RESET используется, например при установки выхода в логический ноль.

**Пример:**

```
...       RESET        O1.0
```

где O1.0 - выходная переменная

**LOAD** LOAD –загружает в аккумулятор число. Обычно после этого оператора следует оператор TO, показывающий цель операции.

**Пример:**

```
THEN      LOAD        V500
          TO          TP31
```

Значение 500 загружается в переменную TP31

**TO** Оператор TO предписывает значение слову операнда. TO определяет цель операции.

**Пример:**

```
THEN      LOAD        V100
```

**TO** R6

Значение 100 загружается в регистр R6.

**JMP TO** Оператор JMP TO вызывает ветвление и переход программы на определенный шаг.

**Пример:**

```
STEP    Mark
IF      I1.0
THEN    SET    O1.0
        JMP TO Start
...
...
STEP    Start
...
```

Программа переходит на шаг с меткой Start, и продолжает использование с этого шага.

**SWAP** SWAP - меняет местами старший и младший байт.

**Пример:**

```
THEN    LOAD    V$55AA
        TO      OW0
        SWAP
        TO      OW1
```

Первые две строчки загружают значение \$55AA в OW0. После чего SWAP переставляет местами старший и младший байт (шестнадцатеричные числа \$55 и \$AA). Таким образом, оставшиеся две строчки загружают значение \$AA55 в OW1.

**SHL** SHL – сдвиг влево. Этот оператор сдвигает содержимое аккумулятора влево на один бит. Самый правый бит заполняется нулем. Результат равносильно умножению на

2. При вызове оператора SHL три раза подряд, это эквивалентно умножению на 2x2x2, т.е. на 8.

**Пример:**

```

THEN      LOAD      V16
          SHL
          TO        R7
    
```

Значение R7 теперь равно 32, что равносильно умножению числа 16 на 2

**SHR** SHR – сдвиг вправо. Этот оператор сдвигает содержимое аккумулятора вправо на один бит. Самый левый бит заполняется нулем. Результат равносильен делению на 2. Повторение оператора SHL приводит к делению на 2 каждый раз.

**Пример:**

```

THEN      LOAD      V16
          SHR
          TO        R7
    
```

Значение R7 теперь равно 8, что равносильно делению числа 16 на 2

**ROL** Результат действия – такой же, как у SHL за тем исключением, что самый старший, крайний левый бит выводится из аккумулятора и вводится на место крайнего правого, самого младшего бита.

**ROR** Как у SHR, биты в аккумуляторе сдвигаются на этот раз вправо, крайний правый бит выводится из аккумулятора и вводится опять в качестве самого старшего бита.

**VID** Оператор VID конвертирует двоичное содержимое аккумулятора в двоично-десятичное. Это можно использовать, например, для индикации двоично-

десятичного кода на светодиодном индикаторе панели управления.

**Пример:**

```

...
THEN      LOAD      IW0
           BID
           TO        OW7
    
```

**DEB** Оператор DEB конвертирует двоично-десятичное содержимое аккумулятора в двоичное. Эта операция нужна при соединении BCD – кодеров со входами контроллера и Вы хотите записать код из кодеров в счетчик.

**Пример:**

```

...
THEN      LOAD      IW7
           DEB
           TO        CW7
    
```

**INC** Счет с увеличением. Увеличивает значение переменной на единицу.

**Пример:**

```

...
THEN      INC       CW0
    
```

**DEC** Счет с уменьшением. Уменьшает значение переменной на единицу.

**Пример:**

```

...
THEN      DEC       CW0
    
```

**WITH** Передача параметров в вызываемый модуль. Используется для установки начального времени

таймера:

```
THEN      SET      T0  "старт таймера 0
          WITH     1s  "на 1 секунду
```

### 2.3. Таймеры и счетчики

Язык FST 4 позволяет использовать 256 таймеров (T0-T255) и 256 счетчиков (C0-C255).

Проиллюстрируем работу таймера на следующем примере:

Пусть необходимо изменять состояние дискретного выхода O0.0 на противоположное с интервалом в одну секунду («мигающий» выход). Программа, реализующая данную функцию, состоит из двух частей, отличающихся между собой лишь действиями, выполняемыми по истечении таймера (включить или выключить выход O0.0). После окончания второй части программы необходимо вновь перейти к началу программы и повторить действия.

Пример:

"" Запускаем таймер T0 на 1 секунду

**STEP setTmr0**

```
THEN  SET      T0
      WITH     1s
```

"" и ждем окончания работы таймера

**STEP WaitSet**

```
IF    N        T0
```

"" Выполняем необходимые действия

```
THEN  SET      O0.0  " Включить O0.0
```

"" Запускаем таймер T0 на 1 секунду

#### **STEP resetTmr1**

```
THEN SET T0  
      WITH 1s
```

"" и ждем окончания работы таймера

#### **STEP WaitReset**

```
IF N T0
```

"" Выполняем необходимые действия

```
THEN RESET O0.0 " Выключить O0.0
```

```
JMP TO setTmr0 " Повторяем снова
```

Особенностью приведенной программы является возможность отдельно задавать время для первой и второй частей. То есть время, на которое выключается выход O0.0 может отличаться от времени включения.

Специально для реализации счетных функций в языке FST предусмотрены 256 счетчиков (C0-C255)

Для реализации функций счета могут быть использованы и другие переменные – переменные «счетчики» (например FW0).

Для увеличения и уменьшения значения счетчика на единицу используются команды INC и DEC (см. табл. 2.1.).

**Для определения четности двоичного числа необходимо посмотреть на его младший бит** (если 0 – число четное, если 1 – число нечетное).

Например, для переменной счетчика FW0 четность определяется битом F0.0 .



## 3. ПРИМЕР ПРОГРАММИРОВАНИЯ ЛАБОРАТОРНОЙ УСТАНОВКИ

### 3.1. Функции системы управления

Функциональная схема автоматизации соответствующая описанным ниже функциям управления представлена на рис. 3.1.

Условие старта

- Заготовка в начале конвейера

Начальная позиция

- Стопор выдвинут
- Ветвь 1 отведена
- Ветвь 2 отведена
- Мотор конвейера выключен

Последовательность работы

1. Заготовка обнаружена
2. Мотор конвейера запущен
3. Определение цвета / материала

Обнаружена черная заготовка, поместить в желоб в конце конвейера

4. Стопор отводится
5. Заготовка выпускается
6. Шаг ожидания

Обнаружена металлическая заготовка, поместить в желоб в центре конверта

7. Ветвь 2 выдвигается
8. Стопор отводится
9. Заготовка выпускается

10. Шаг ожидания

Обнаружена красная заготовка, поместить в желобе в начале конвейера

- 11. Ветвь 1 выдвигается
- 12. Стопор отводится
- 13. Заготовка выпускается
- 14. Шаг ожидания
- 15. Двигатель конвейера выключается
- 16. Стопор выдвигается
- 17. Ветвь 1 отводится
- 18. Ветвь 2 отводится

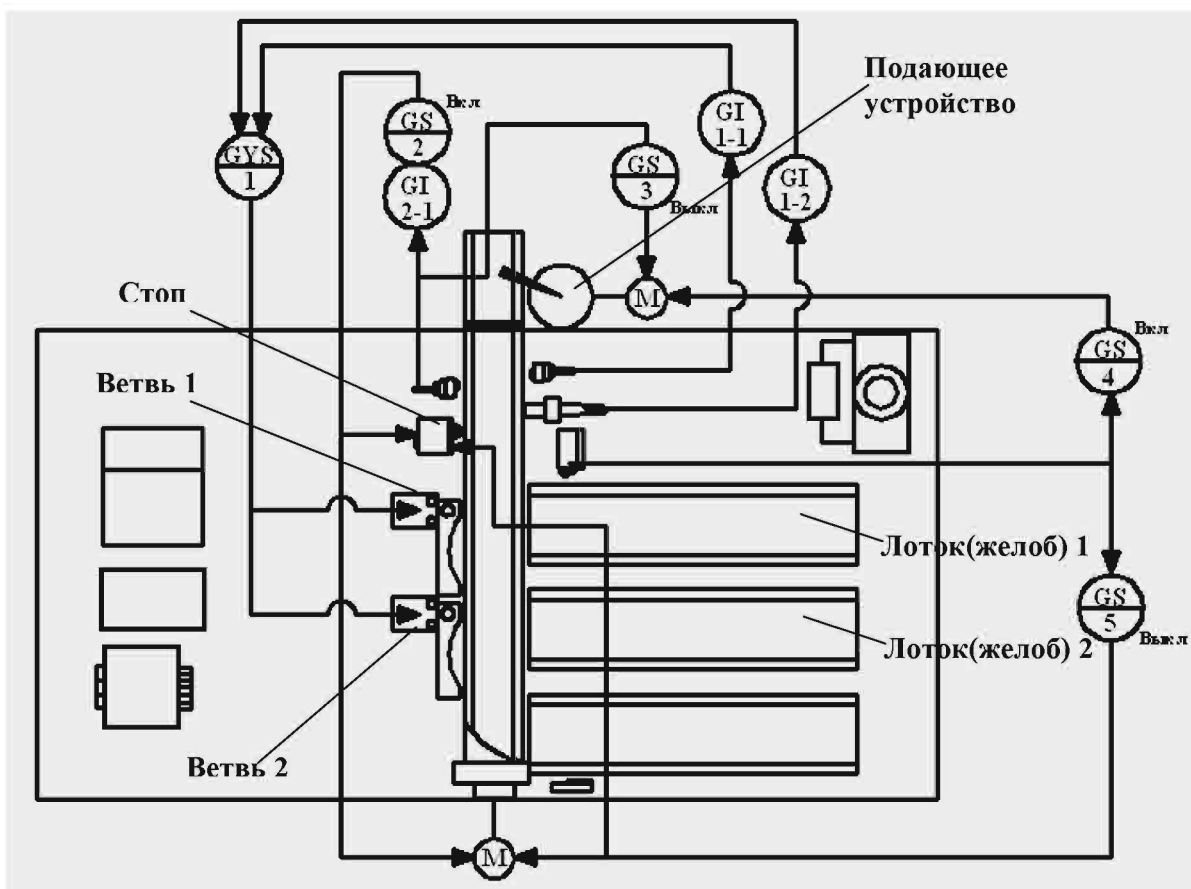


Рис 3.1. Функциональная схема автоматизации объекта

### 3.2. Алгоритм управления

Система управления должна выполнять следующие функции:

#### I – ЗАПУСК:

- При нажатии кнопки Start или срабатывании датчика обнаружения (в начале конвейера) запускается таймер и конвейер.

#### II - ОПРЕДЕЛЕНИЕ ТИПА ЗАГОТОВКИ:

- Если датчик индуктивной близости определил деталь как металлическую, срабатывает ветвь 1 и включается зеленый индикатор в конце лотка 1.

- Если оптический датчик цвета определил деталь, как цветную, срабатывает ветвь 2 и включается желтый индикатор в конце лотка 2.

- Если деталь определена, как черная включается красный индикатор в конце лотка 3.

- При истечении время таймера затвор задвигается.

#### III - ЗАВЕРШЕНИЕ ЦИКЛА:

- При попадании в лоток или нажатии кнопки Stop закрываются ветви 1,2, выдвигается затвор, останавливается конвейер и гаснет индикатор Start.

- При работе в автоматическом режиме подающее устройство включается.

Перечисленные кнопки управления (Start, Stop, Rreset), а также ключ переключения режимов (Key) и индикаторы состояния работы установки расположены на передней панели. Дублирующая кнопка Start (№2) расположена на подающем устройстве.

Алгоритм, выполняющий данные функции, представлен на рис. 3.2.

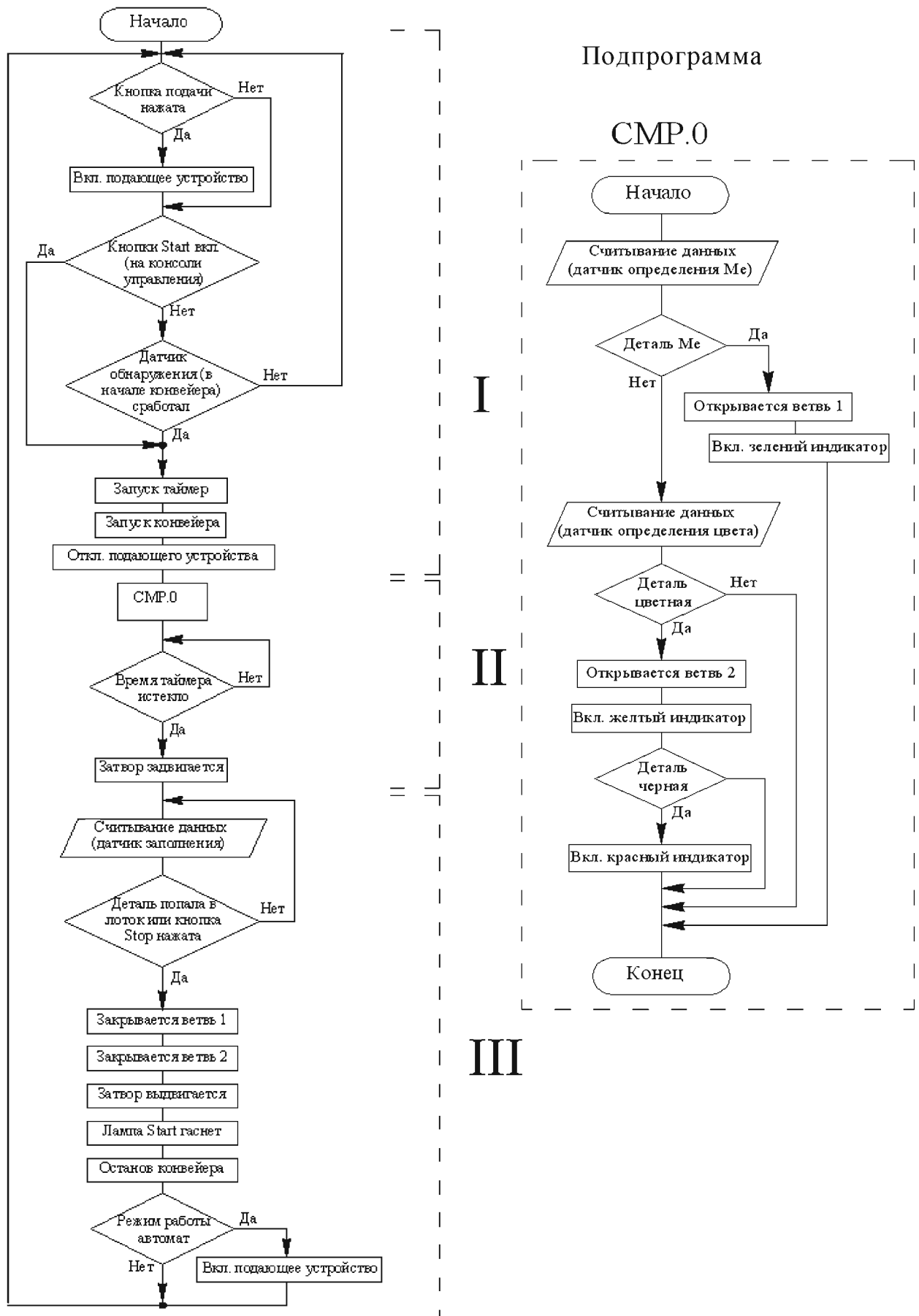


Рис 3.2. Алгоритм управления

### 3.3. Пример программы

Список переменных и внешний вид окна переменных представлены в табл. 3.1 и на рис. 3.3.

Таблица переменных

Таблица 3.1

Операнд	Обозначение	Комментарий
O0.0	lenta_	Запуск ленты
O0.1	Lapa1_	Первая ветвь
O0.2	Lapa2_	Вторая ветвь
O0.3	zatvor_	Стопор заготовок
O0.4	green_L	Зеленый индикатор
O0.5	yellow_L	Желтый индикатор
O0.6	red_L	Красный индикатор
O1.0	LStart_	Индикатор кнопки Start
O1.1	LReset_	Индикатор кнопки Reset
O1.2	_Q1	Выход на передней панели Q1
O1.3	_Q2	Выход на передней панели Q2
O1.4	_Q4_pdch	Подача заготовок (выход на передней панели Q4)
O1.5	_Q5	Выход на передней панели Q5
O1.6	_Q6	Выход на передней панели Q6
O1.7	_Q7	Выход на передней панели Q7
I0.0	_Dstart	Начальный датчик обнаружения заготовок
I0.1	_Dme	Датчик металла
I0.2	_DnoBlack	Датчик цвета
I0.3	_Dend	Конечный датчик заполнения
I1.0	_start	Кнопка Start
I1.1	_stop	Кнопка Stop
I1.2	KEY	Ключ переключения режимов
I1.3	_Reser	Кнопка Reset
I1.4	_I4start2	Дополнительная кнопка запуска на подающем устройстве (выход на передней панели I4)
T0	Timer	Таймер

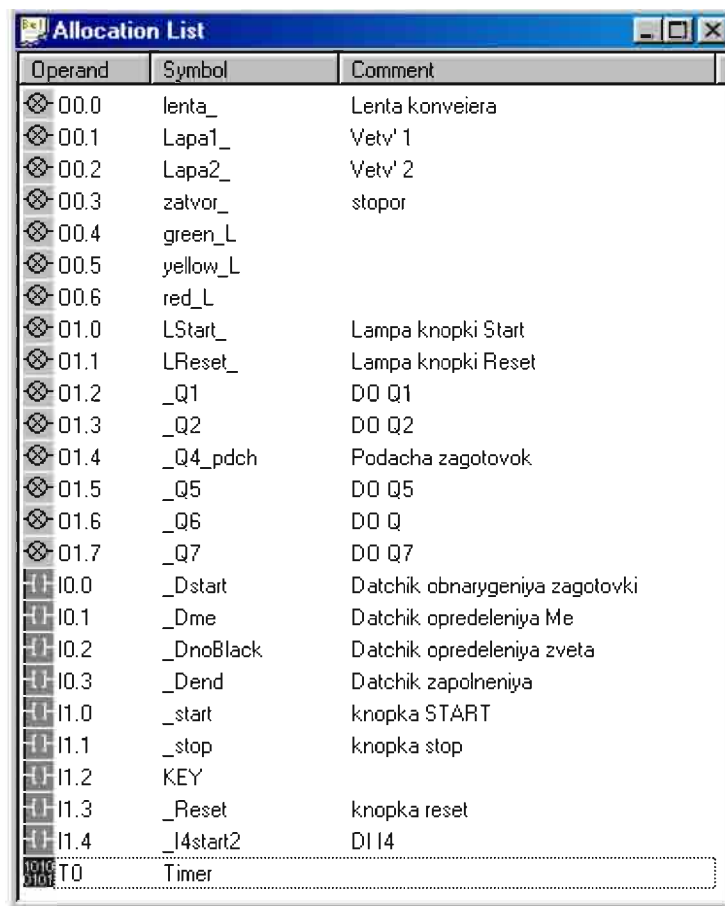


Рис.3.3. Внешний вид окна переменных

Код программы:

**STEP 1**

```

IF          _I4start2      "Кнопка START №2 (DI I4)
THEN SET    _Q4_pdch      "Подача заготовок

IF          N  _stop       "Кнопка STOP
THEN RESET  lenta_        "Лента конвейера
           RESET  LStart_  "Индикатор кнопки START
           RESET  _Q4_pdch "Подача заготовок

IF          _Dstart       "Датчик обнаружения заготовок
OR          _start       "Кнопка START

```

```

THEN RESET  _Q4_pdch  "Подача заготовок
      SET    Lenta_    "Лента конвейера
      SET    LStart_   "Индикатор кнопки START
      SET    Timer     "Включение таймера

```

### STEP 2

```

IF      N      _stop    "Кнопка STOP
THEN RESET Lenta_     "Лента конвейера
      RESET LStart_   "Индикатор кнопки START

IF      N      Timer    "Время таймера истекло
THEN CMP 0            "Подпрограмма распознаения заготовки
      SET    Zatvor_  "Стопор

```

### STEP 3

```

IF      KEY     "Ключ
      AND  ( N   _stop    "Кнопка STOP
      OR   _Dend  )      "Датчик заполнения
THEN SET  _Q4_pdch  "Подача заготовок

IF      N      _stop    "Кнопка STOP
      OR   _Dend  "Датчик заполнения
THEN RESET lenta_     "Лента конвейера
      RESET LStart_   "Индикатор кнопки START
      RESET zatvor_   "Стопор
      RESET Lapa1_    "Ветвь 1
      RESET Lapa2_    "Ветвь 2

      JMP TO 1        "Переход к шагу 1

```

Код подпрограммы CMP 0:

```
IF          _DnoBlack "Датчик определения цвета
  AND      N  _Dme     "Датчик определения металла
THEN SET    Lapa2_    "Ветвь 2
  SET      yellow_L   "Желтый индикатор лотка 2
  RESET    red_L      "Красный индикатор лотка 3
  RESET    green_L    "Зеленый индикатор лотка 1

IF          _Dme      "Датчик определения металла
  AND      _DnoBlack "Датчик определения цвета

THEN SET    Lapa1_    "Ветвь 1
  SET      green_L    "Зеленый индикатор лотка 1
  RESET    yellow_L   "Желтый индикатор лотка 2
  RESET    red_L      "Красный индикатор лотка 3

IF          N  _Dme    "Датчик определения металла
  AND      N  _DnoBlack "Датчик определения цвета
THEN SET    red_L      "Красный индикатор лотка 3
  RESET    yellow_L    "Желтый индикатор лотка 2
  RESET    green_L     "Зеленый индикатор лотка 1
```



## 4. ЛАБОРАТОРНЫЕ РАБОТЫ

### Лабораторная работа № 1

**Цель работы:** 1 — первичное ознакомление с установкой FESTO;  
2 — получение навыков написания программы для управления контроллером FESTO FEC-FC-640-FST фирмы FESTO;  
3 — разработка и отладка программы управления работой сортировочной конвейерной лентой.

Секция конвейера направлена на сортировку, где происходит переключение на разные ветви сортировочной станции в зависимости от заготовки.

В сортировочной станции символические заготовки сортируются по материалу и цвету. Заготовки сортируются по трем желобам. Датчик определяет, когда заготовка попала в начало конвейера.

Характеристики заготовок (красный, черный, металл) определяются датчиками перед стопором, после чего заготовки сортируются в соответствующие желоба через ветви. Ветви двигаются посредством цилиндров короткого удара с использованием реверсирующего механизма.

Рефлективный оптический датчик положения следит за уровнем заполнения желобов.

### Порядок выполнения работы

#### *Подготовка установки к работе*

Перед началом работы подготовить установку:

- включить источник питания,
- включить компрессор и компьютер.

## Создание проекта

1. На рабочем столе двойным щелчком по ярлыку «FST4 English Demo Version» запустить программное обеспечение «Festo FST версия 4.02» для программирования контроллера Festo/Beck FEC FC640.

Создать новый проект: в меню выбрать **Project**→**New...** (см. рис. 4.1)

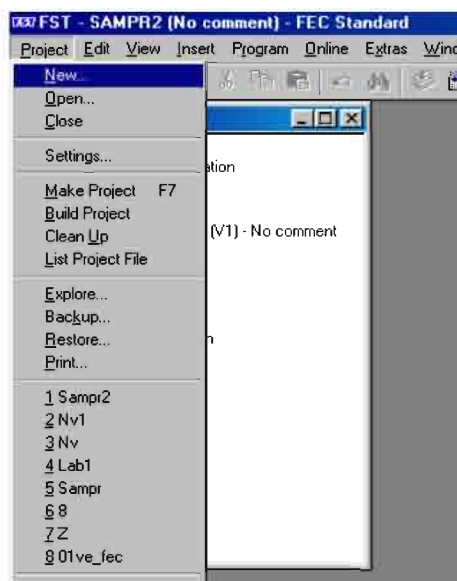


Рис.4.1. Выбор меню создания проекта

в появившемся окне (рис. 4.2) в поле **Name:** латинскими буквами ввести имя своего проекта; нажать кнопку **OK**.

Необходимо дать название проекту с учетом номера группы, подгруппы и лабораторной работы. Например, **540\_1\_L1** , где 540 – номер группы, 1 – номер подгруппы, L1 – номер лабораторной работы. Номер Вашей подгруппы спросить у преподавателя!

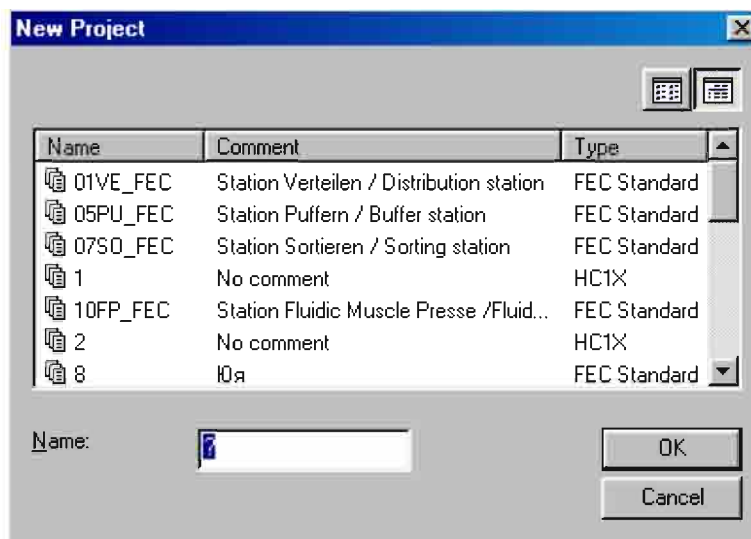


Рис.4.2. Окно создания нового проекта

В окне **Project Setting** (рис. 4.3) в поле **Controller** выбрать тип контроллера FEC Standart или FEC Compact; нажать **OK**.

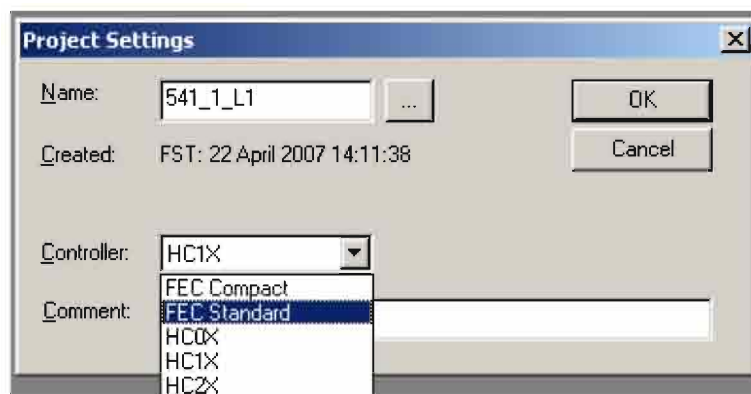


Рис.4.3. Окно настройки проекта

Через пункт меню **View**→ **IO Configuration** открывается окно выбора контроллера (или **View**→ **Project Tree** , в окне **Project Tree** двойным щелчком мыши тот же пункт).

Для определения типа и модели контроллера в окне **IO Configuration** (рис.4.4) необходимо: сделать двойной щелчок левой кнопкой мыши или правой клавишей мыши вызвать контекстное меню и выбрать **Insert IO Module**;

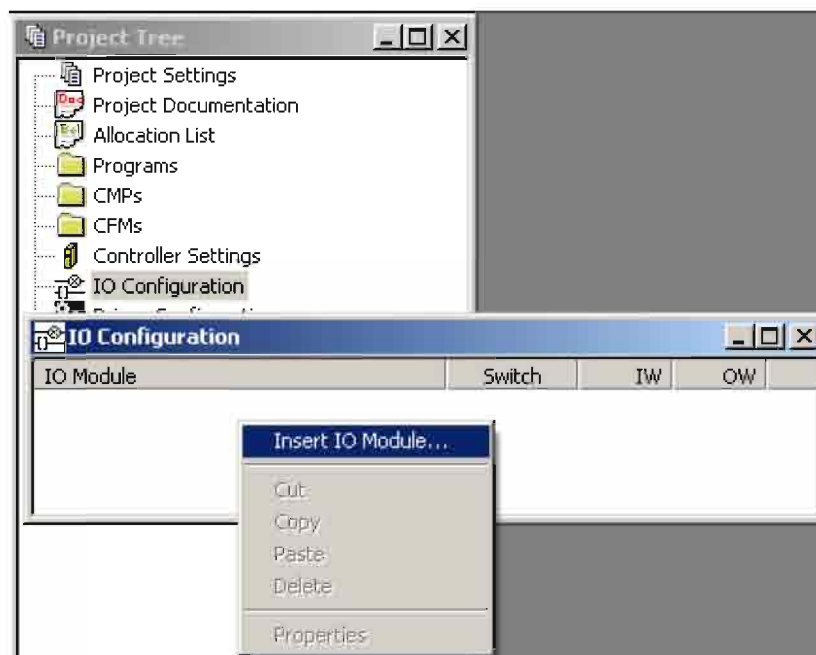


Рис.4.4. Окно конфигурации системы ввода-вывода

В появившемся окне (рис. 4.5) выбрать контроллер FC640.

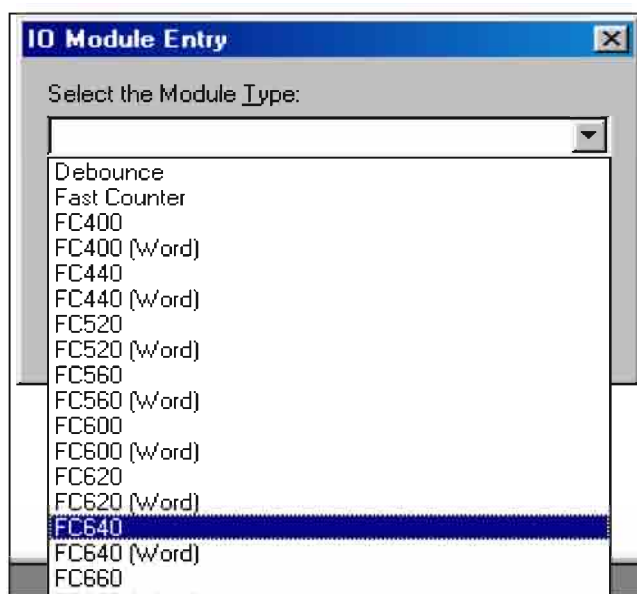


Рис.4.5. Выбор контроллера

2. Создание входных и выходных переменных, необходимых для написания программы.

Через пункт меню **View**→ **Allocation List** (или в окне **Project Tree** двойной щелчок мыши на **Allocation List**) открывается список переменных (рис. 4.6).

Для добавления новых входных и выходных сигналов в Allocation List необходимо: сделать двойной щелчок левой кнопкой мыши или правой клавишей мыши вызвать контекстное меню и выбрать **Insert Operand** (рис. 4.6);

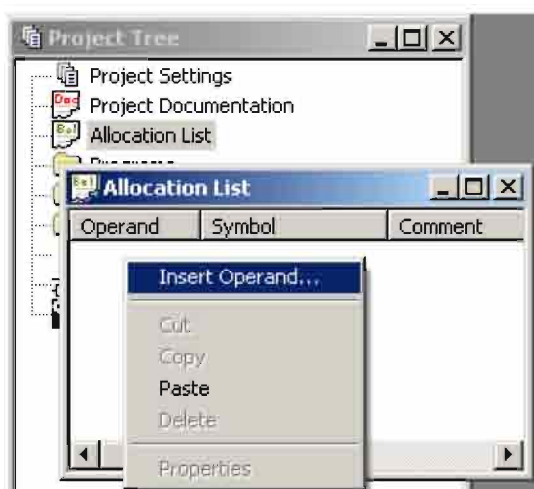


Рис.4.6. Окно списка переменных

В поле **Absolute Operand** (рис. 4.7) вводятся адреса входных/выходных сигналов (табл. 4.1 и табл. 4.2) в памяти контроллера (адреса ячеек памяти, соответствующих определенным устройствам).

В поле **Symbolic Operand** вводятся логические имена (имена, присваиваемые устройствам).

В поле **Comment** приводятся необходимые комментарии (описывается работа соответствующих устройств).

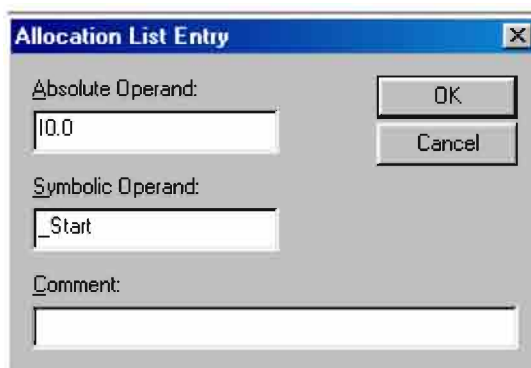


Рис.4.7. Добавление переменной

### Входные переменные

Таблица 4.1

Источники входных сигналов	Логические имена	Адреса входных сигналов в памяти
Оптический датчик положения обнаруживает, есть ли заготовка в начале конвейера	_Start	I0.0
Светоотражающий датчик, используемый для наблюдения за уровнем заполнения желобов	_End	I0.3

### Выходные переменные

Таблица 4.2

Приемники выходных сигналов	Логические имена	Адреса выходных сигналов в памяти
Сортировочная конвейерная лента служит для перемещения заготовок	Lenta_	O0.0
Пневматический стопор, останавливает заготовку с целью определения ее характеристик	Zatvor_	O0.3

### 3. Создание и отладка кода программы в контроллере

Через пункт меню **Inset** → **New Program...** или через окно **Project Tree** → правая клавиша мыши на пункте меню **Programs** → выбрать **Insert program...**

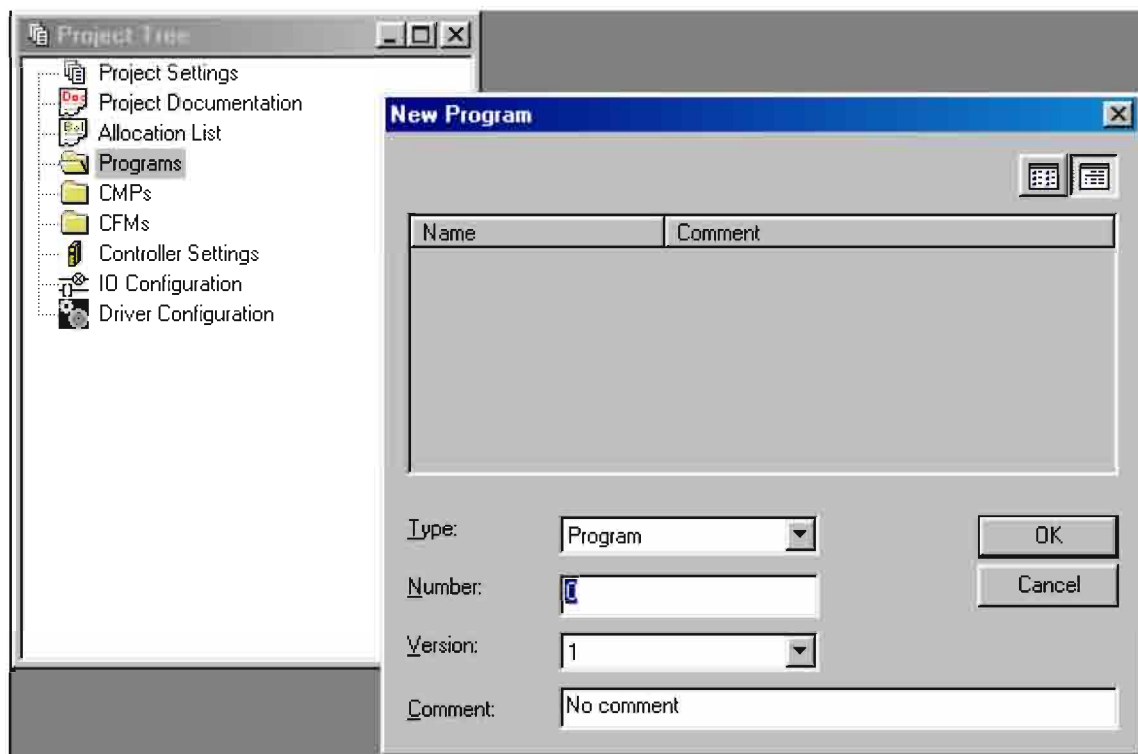


Рис.4.8. Создание новой программы

В окне **New Program** (рис. 4.8) нажать **OK** (установки остаются по умолчанию). После чего открывается окно написания кода программы.

Написать код программы согласно пункту 4. Откомпилировать его **Program** → **Compile**. В открывшемся окне прочесть о наличии ошибок, если они есть.

4. Описание условий работы сортировочной конвейерной ленты (объекта управления)

Подпункт выбирается по указанию преподавателя.

**А.** Оптический датчик положения обнаруживает, есть ли заготовка в начале конвейера. В результате чего начинается программный цикл и включается привод сортировочного конвейера. При прохождении заготовкой светоотражающего датчика, следящего за уровнем заполнения желобов, конвейер останавливается.

**Б.** Оптический датчик положения обнаруживает, есть ли заготовка в начале конвейера и при наличии заготовки срабатывает привод сортировочного конвейера. Заготовки, поступающие с модуля конвейера, хранятся в модуле желобов. Светоотражающий датчик фиксирует поступающие в желоб заготовки. При заполнении заготовками желоба конвейер останавливается.

При необходимости использования таймера (табл. 4.3), он описывается, так же как и входные/выходные сигналы в пункте меню **Allocation List**.

Таймеры

Таблица 4.3

Таймеры	Логическое имя	Адреса в памяти
Импульсный таймер	Timer	T0

7. Загрузка программы в контроллер.

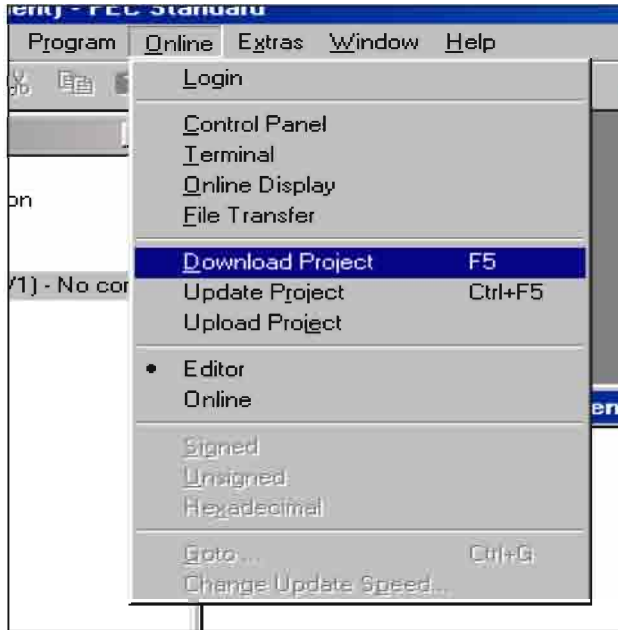


Рис.4.9. Выбор пункта меню «Загрузка программы в контроллер»



В меню выбирается пункт **Online→Download Project**; таким образом, происходит компиляция программы в контроллер; на изображенные на рис. 4.10 системные сообщения необходимо ответить «Да».

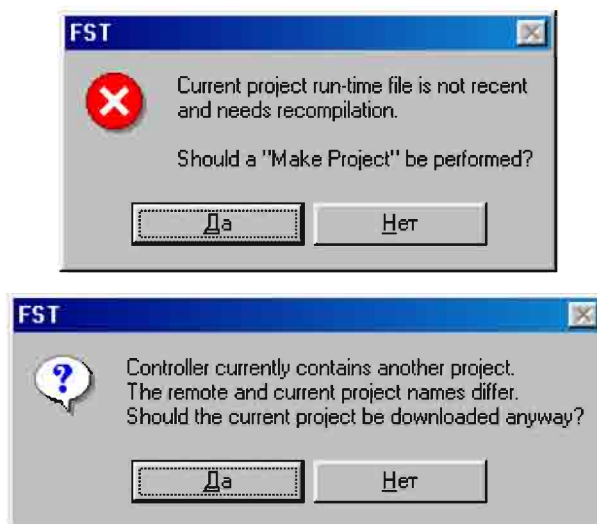


Рис.4.10. Окна вопросов загрузчика программы

После чего оборудование готово к работе.

#### 8. Проверка работоспособности программы.

Выбирается пункт меню **Online→Control panel**. Для запуска, в появившемся окне (рис. 4.11), нажать кнопку Run, по завершению нажать кнопку Stop.



Рис.4.11. Окно управления работой программы

## Оформление отчета

В отчет о работе включается:

1. Титульный лист.
2. Задание на программирование с описанием входных и выходных переменных, а также внутренние переменные и таймеры, если таковые имеются.
3. Блок-схема работы алгоритма программы.
4. Код программы на языке FST с комментариями.

## Лабораторная работа № 2

- Цель работы:**
- 1 — разработка и отладка программы для сортировки заготовок в зависимости от типа;
  - 2 — использование таймера при программировании контроллера.

### Порядок выполнения работы

1. Запустить программное обеспечение «Festo FST версия 4.02». Создать новый проект (см. Лабораторная работа №1).

Необходимо дать название проекту с учетом номера группы, подгруппы и лабораторной работы. Например, **540\_1\_L2** , где 540 – номер группы, 1 – номер подгруппы, L2 – номер лабораторной работы.  
Номер Вашей подгруппы спросить у преподавателя!

Определить тип и модель контроллера в окне **IO Configuration** (рис.4.4).

2. Создать входные и выходные переменные, необходимые для написания программы (см. табл. 4.4, табл. 4.5 и табл. 4.6).

#### Входные переменные

Таблица 4.4

Источники входных сигналов	Логические имена	Адреса входных сигналов в памяти
Оптический датчик положения обнаруживает, есть ли заготовка в начале конвейера	_Start	I0.0
Светоотражающий датчик, используемый для наблюдения за уровнем заполнения желобов	_End	I0.3
Датчик индуктивной близости обнаруживает металлические объекты.	_Metall	I0.1

## Выходные переменные

Таблица 4.5

Приемники выходных сигналов	Логические имена	Адреса выходных сигналов в памяти
Сортировочная конвейерная лента служит для перемещения заготовок	Lenta_	00.0
Пневматический стопор, останавливает заготовку с целью определения ее характеристик	Zatvor_	00.3
Ветвление для сортировки заготовок (переключается при помощи пневмоцилиндра)	Vetv1_	00.2

## Таймеры

Таблица 4.6

Таймеры	Логическое имя	Адреса в памяти
Импульсный таймер	Timer	T0

Время таймера задается одним из двух способов:

- Статически - через пункт меню **Online** → **Online Display** на вкладке **Timer** (рис. 4.12).

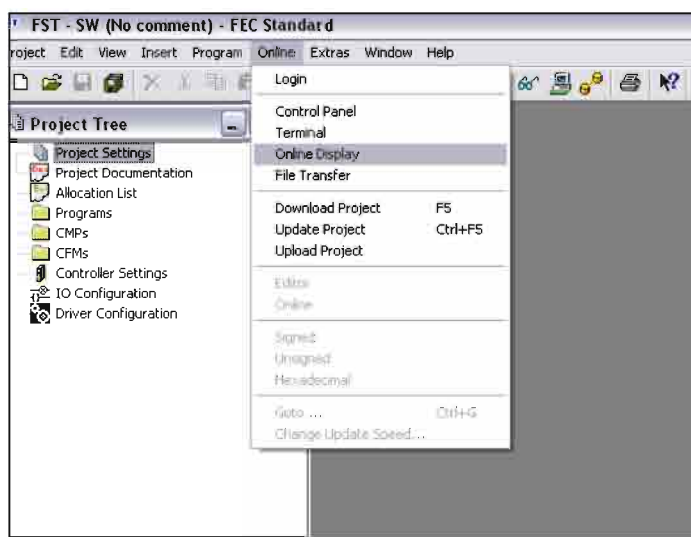


Рис.4.12. Выбор пункта меню наблюдения за переменными

- Динамически (т.е. во время выполнения программы) - с помощью команды **WITH** (см. табл. 2.1.).

По указанию преподавателя задать время отсчета таймера, например, 4 секунды.

### 3. Описание условий работы сортировочной конвейерной ленты (объекта управления)

Подпункт выбирается по указанию преподавателя.

**А.** Оптический датчик положения обнаруживает, есть ли заготовка в начале конвейера. В результате чего начинается программный цикл и включается привод сортировочного конвейера.

Заготовка останавливается с помощью пневматического стопора. Металлические заготовки определяются индуктивным сенсором близости.

В зависимости от типа заготовки, активизируются требуемая ветвь привода. Как только заготовка отпущена стопором, она доставляется в соответствующий желоб.

При прохождении заготовкой рефлексивного ретро-датчика, следящего за уровнем заполнения желобов, конвейер останавливается.

**Б.** Оптический датчик положения обнаруживает, есть ли заготовка в начале конвейера. В результате чего начинается программный цикл и включается привод сортировочного конвейера.

Заготовка останавливается с помощью пневматического стопора. Оптический датчик цвета (табл. 4.7) определяет цвет заготовки (красный или черный).

В зависимости от цвета заготовки, соответствующая ветвь активизируются. Как только заготовка отпущена стопором, она доставляется в желоб.

При прохождении заготовкой рефлексивного датчика, следящего за уровнем заполнения желобов, конвейер останавливается.

Дополнительные входные переменные

Таблица 4.7

Источники входного сигнала	Логическое имя	Адрес входного сигнала в памяти
Отражательный световой датчик распознавания используется для распознавания цветов заготовок.	_Color	10.2

## Лабораторная работа № 3

- Цель работы:**
- 1 — разработка и отладка программы для сортировки заготовок в зависимости от материала (металл-неметалл);
  - 2 — подсчет общего количества заготовок.

### Порядок выполнения работы

1. Запустить программное обеспечение «Festo FST версия 4.02». Создать новый проект (см. Лабораторная работа №1).

Необходимо дать название проекту с учетом номера группы, подгруппы и лабораторной работы. Например, **540\_1\_L3** , где 540 – номер группы, 1 – номер подгруппы, L3 – номер лабораторной работы.  
Номер Вашей подгруппы спросить у преподавателя!

Определить тип и модель контроллера в окне **IO Configuration** (рис.4.4).

2. Создать входные и выходные переменные, необходимые для написания программы (см. табл. 4.8, табл. 4.9 и табл. 4.10).

### Входные переменные

Таблица 4.8

Источники входных сигналов	Логические имена	Адреса входных сигналов в памяти
Оптический датчик положения обнаруживает, есть ли заготовка в начале конвейера	_Start	I0.0
Светоотражающий датчик, используемый для наблюдения за уровнем заполнения желобов	_End	I0.3
Датчик индуктивной близости обнаруживает металлические объекты.	_Metall	I0.1

## Выходные переменные

Таблица 4.9

Приемники выходных сигналов	Логические имена	Адреса выходных сигналов в памяти
Сортировочная конвейерная лента служит для перемещения заготовок	Lenta_	00.0
Пневматический стопор, останавливает заготовку с целью определения ее характеристик	Zatvor_	00.3
Ветвление для сортировки заготовок (переключается при помощи пневмоцилиндра)	Vetv1_	00.2

## Дополнительные переменные и функций

Таблица 4.10

Наименование	Имена	Адреса ячеек памяти
Счетчик	Schet	FW0
Импульсный таймер (устройство обратного счета)	Timer	T0
Флаг прохождения заготовки	Flag	F1.0

По указанию преподавателя задать время отсчета таймера, например, 4 секунды.

3. Описание условий работы сортировочной конвейерной ленты (объекта управления)

Датчик обнаружения распознает появление заготовки в начале конвейера. В результате чего начинается программный цикл и включается привод сортировочного конвейера. При включении привода конвейера



необходимо, чтобы пневматический стопор открывался не сразу (для этого используется устройство обратного отсчета (таймер), установленное на время срабатывания стопора). По окончании отсчета таймера, стопор отключается, деталь попадает на конвейерную ленту.

По срабатыванию таймера и датчика металла выдвигается ветвь привода.

Непосредственно сортировка заготовок осуществляется с помощью ветвей конвейера, направляющих деталь в соответствующий желоб станции.

Подсчет заготовок происходит при помощи оптического датчика заполнения с отражателем, так как этот датчик следит за попаданием заготовки в желоб.

Работа данного датчика заключается в следующем: из датчика исходят световые импульсы (лучи), которые попадают на светоотражающий элемент и улавливаются обратно датчиком. За одну секунду происходит порядка 100 таких импульсов. Попадая в желоб, деталь на некоторое время прерывает луч.

При каждом прохождении детали значение счетчика должно увеличиваться на единицу. Для увеличения значения счетчика используется команда INC (см. р. 2.2. Список команд).

Допустим, значение счетчика увеличивается всякий раз, когда луч прерван. Тогда при прохождении каждой детали значение счетчика возрастет на количество импульсов, которые были прерваны. Но таким образом не удастся подсчитать число самих заготовок, так как деталь проходит эту черту не мгновенно и имеет размер.

Поэтому необходимо обеспечить срабатывание счетчика один раз для каждой детали. Это может быть реализовано с использованием логической переменной - флага прохождения заготовки (Flag).

Для этого в начальный момент пересечения луча требуется установить флаг прохождения заготовки, который исключит повторное увеличение счетчика одной и той же деталью.

В программе необходимо учесть, что если работают таймер и стопор, то флаг прохождения заготовки, сохранившийся от предыдущего прохождения через светоотражающий датчик, должен сброситься.

При прохождении заготовкой светоотражающего датчика, следящего за уровнем заполнения желобов, конвейер останавливается.

Подсчитанное значение заготовок можно посмотреть: **Online →Online Display→Flag Words**

## Лабораторная работа № 4

- Цель работы:**
- 1 — разработка и отладка программы сортировки заготовок на четные и нечетные;
  - 2 — подсчет общего количества заготовок.

### Порядок выполнения работы

1. Запустить программное обеспечение «Festo FST версия 4.02». Создать новый проект (см. Лабораторная работа №1).

Необходимо дать название проекту с учетом номера группы, подгруппы и лабораторной работы. Например, **540\_1\_L4** , где 540 – номер группы, 1 – номер подгруппы, L4 – номер лабораторной работы.  
Номер Вашей подгруппы спросить у преподавателя!

Определить тип и модель контроллера в окне **IO Configuration** (рис.4.4).

2. Создать входные и выходные переменные, необходимые для написания программы (см. табл. 4.11, табл. 4.12 и табл. 4.13).

### Входные переменные

Таблица 4.11

Источники входных сигналов	Логические имена	Адреса входных сигналов в памяти
Оптический датчик положения обнаруживает, есть ли заготовка в начале конвейера	_Start	I0.0
Светоотражающий датчик, используемый для наблюдения за уровнем заполнения желобов	_End	I0.3

## Выходные переменные

Таблица 4.12

Приемники выходных сигналов	Логические имена	Адреса выходных сигналов в памяти
Сортировочная конвейерная лента служит для перемещения заготовок	Lenta_	O0.0
Пневматический стопор, останавливает заготовку с целью определения ее характеристик	Zatvor_	O0.3
Ветвление для сортировки заготовок, переключается при помощи пневмоцилиндра	Vetv1_	O0.2

## Дополнительные переменные и функций

Таблица 4.13

Наименование	Имена	Адреса ячеек памяти
Счетчик	Schet	FW0
Импульсный таймер (устройство обратного счета)	Timer	T0
Флаг прохождения заготовки	EndFlag	F1.0

По указанию преподавателя задать время отсчета таймера, например, 4 секунды.

3. Описание условий работы сортировочной конвейерной ленты (объекта управления)

Запуск и останов конвейера осуществляется по аналогии с предыдущей работой. Переключение ветвей перед открытием затвора осуществляется в зависимости от четности числа в счетчике. Также необходимо предусмотреть, что срабатывание счетчика происходит на фронт детали, т.е. на первой ее точке.

## Библиографический список

Электронная документация на сортировочную станцию фирмы Festo (CD-ROM)

Анашкин А.С., Кадыров Э.Д., Харазов В.Г. Техническое и программное обеспечение распределенных систем управления / под общей ред. Хазарова В.Г. - М.: Изд-во «Р-2», 2004.

Бесекерский В.А., Попов Е.П. Теория систем автоматического управления. - Изд. 4-е, перераб. и доп.-СПб.: Профессия, 2003.-752с.

Кондрашкова Г.А. Технологические измерения и приборы в целлюлозно-бумажной промышленности: учебник для вузов.-М.: Лесная промышленность, 1981.-376с.

Гусев В.Г., Гусев Ю.М. Электроника: учеб. пособие для вузов. – М.:Высшая школа, 1982.-495с.

## Оглавление

ВВЕДЕНИЕ .....	3
1. ОПИСАНИЕ ЛАБОРАТОРНОГО СТЕНДА .....	4
1.1. Сортировочная станция .....	4
1.2. Электропневматический преобразователь.....	6
1.3. Датчики .....	8
1.4. Устройство регулирования давления.....	15
1.5. Контроллер FESTO FEC-FC-640-FST.....	17
2. ЯЗЫК ОПИСАНИЯ ЗАДАЧ FST .....	22
2.1. Характеристики программы.....	23
2.2. Список команд .....	24
2.3. Таймеры и счетчики.....	32
3. ПРИМЕР ПРОГРАММИРОВАНИЯ ЛАБОРАТОРНОЙ УСТАНОВКИ .....	34
3.1. Функции системы управления .....	34
3.2. Алгоритм управления .....	36
3.3. Пример программы .....	38
4. ЛАБОРАТОРНЫЕ РАБОТЫ .....	42
Лабораторная работа № 1 .....	42
Лабораторная работа № 2 .....	52
Лабораторная работа № 3 .....	56
Лабораторная работа № 4 .....	60
Библиографический список .....	62