



**Тихвинский
промышленно-технологический
техникум им. Е. И. Лебедева**

МЕТОДИЧЕСКАЯ РАЗРАБОТКА

Тема: «Основы программирования промышленных средств автоматизации»

Автор: Чекаенков Владимир Александрович.

Должность: преподаватель.

Аннотация к методической разработке

Цель: углубление теоретических знаний, умения работы со справочной и нормативно-правовой литературой, получение навыков программирования ПЛК (программируемого логического контроллера).

Аудитория: Обучающиеся по ППССЗ «Техническая эксплуатация и обслуживание электрического и электромеханического оборудования (по отраслям)».

Актуальность: 1. В 2019 году в техникуме приобретено новое лабораторное оборудование, которое позволяет обучать студентов навыкам работы в современной производственной среде.

2. Важная социальная задача - освобождения человека от работ, связанных с опасностями для здоровья или с тяжелым физическим трудом, а также от простых монотонных операций, не требующих высокой квалификации путем внедрения автоматизации.

Условия реализации: Лабораторный стенд «Пневмоэлектрический манипулятор».

Содержание:

1. Программируемый контроллер.
2. Состав учебного роботизированного комплекса.
3. Программируемый логический контроллер в роботизированном комплексе.
4. Основы программирования промышленных средств автоматизации.
5. Базовые элементы программирования на языке LED.
6. Основные элементы, используемые в TIAPORTALV14 и их значение.
7. Основы построения систем позиционного управления.
8. Составление программы «Пневмоэлектрического манипулятора» в соответствии с техническим заданием.

Введение

Промышленные роботы предназначены для замены человека при выполнении основных и вспомогательных технологических операций в процессе промышленного производства. Гибкие автоматизированные производства, создаваемые на базе промышленных роботов, позволяют решать задачи автоматизации на предприятиях с широкой номенклатурой продукции при мелкосерийном и штучном производстве. Копирующие манипуляторы, управляемые человеком-оператором, необходимы при выполнении различных работ в опасной для него среде. Эти устройства незаменимы при выполнении работ в космосе, под водой, в химически активных средах. Таким образом, промышленные роботы и копирующие манипуляторы являются важными составными частями современного промышленного производства.

В настоящее время робототехнические системы находят достаточно широкое применение в машиностроении, промышленности, производящей средства производства, а также в легкой и пищевой промышленности. Вместе с тем на основе данных системного анализа может быть представлен перечень других отраслей, также нуждающихся в роботизации.

Для автоматического управления промышленными роботами, как правило, используется промышленный логический контроллер (ПЛК) или встроенный контроллер.

1 Программируемый контроллер.

Программируемый контроллер— специальная разновидность электронной вычислительной машины. Чаще всего ПЛК используют для автоматизации технологических процессов. В качестве основного режима работы ПЛК выступает его длительное автономное использование, зачастую в неблагоприятных условиях окружающей среды, без серьёзного обслуживания и практически без вмешательства человека.

1. В отличие от микроконтроллера— микросхемы, предназначенной для управления электронными устройствами — ПЛК являются самостоятельным устройством, а не отдельной микросхемой.
2. В отличие от компьютеров, ориентированных на принятие решений и управление оператором, ПЛК ориентированы на работу с машинами через развитый ввод сигналов датчиков и вывод сигналов на исполнительные механизмы.

Упрощенное представление состава и принципа действия ПЛК хорошо демонстрирует рисунок 1. Из него видно, что ПЛК имеет три основные секции:

- входную; выходную; центральную.



Рис. 1.1 ПЛК

Имеется еще источник питания. Возможно подключение к ПЛК внешнего ПК для программирования и отладки.

По конструктивному исполнению ПЛК делят на моноблочные (рисунок 2) и модульные. В корпусе моноблочного ПЛК наряду с ЦП, памятью и блоком питания размещается фиксированный набор входов/выходов. В модульных ПЛК используют отдельно устанавливаемые модули входов/выходов. Согласно требованиям МЭК 61131, их тип и количество могут меняться в зависимости от поставленной задачи и обновляться с течением времени. ПЛК подобной концепции представлены на рисунке 1.2. Подобные ПЛК могут действовать в режиме «ведущего» и расширяться «ведомыми» ПЛК через интерфейс Ethernet.



Рис. 1.2. Моноблочные программируемые логические контроллеры

2 Состав учебного роботизированного комплекса.

Лабораторный стенд «Пневмоэлектрический манипулятор» (рис. 2.1) представляет собой мехатронный комплекс, реализованный на основе средств промышленной автоматизации компаний SMC, Siemens и др.

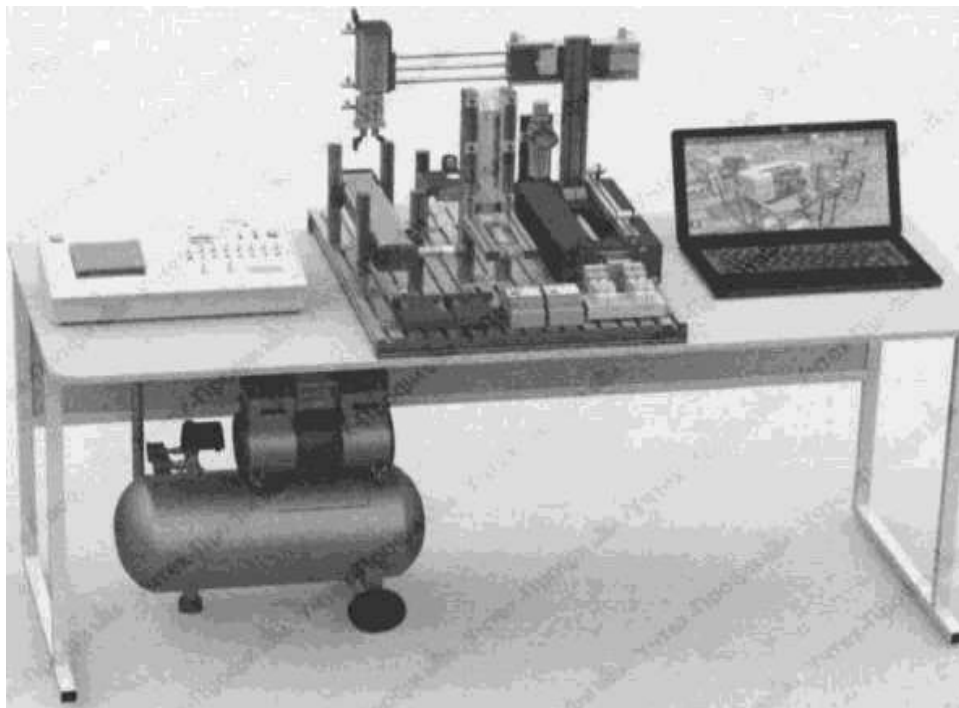


Рис. 2.1. Лабораторный стенд «Пневмоэлектрический манипулятор»:

- Моноблок «Программируемый логический контроллер»;
- Мехатронный модуль с пневмоэлектрическим манипулятором;
- Ноутбук;
- Компрессор;
- Лабораторный стол.

К основным узлам относятся:

1. узел выдачи заготовок;
2. узел трехосевого пневмосхвата;
3. узел определения цвета и типа материала;
4. узел складирования.

К вспомогательному оборудованию относятся:

5. блок подготовки воздуха;
6. блок управления и коммутации;
7. алюминиевая профильная плита;
8. кабель-каналы;
9. набор заготовок.

В качестве сортируемых заготовок используются детали цилиндрической формы, отличающиеся цветом и материалом.

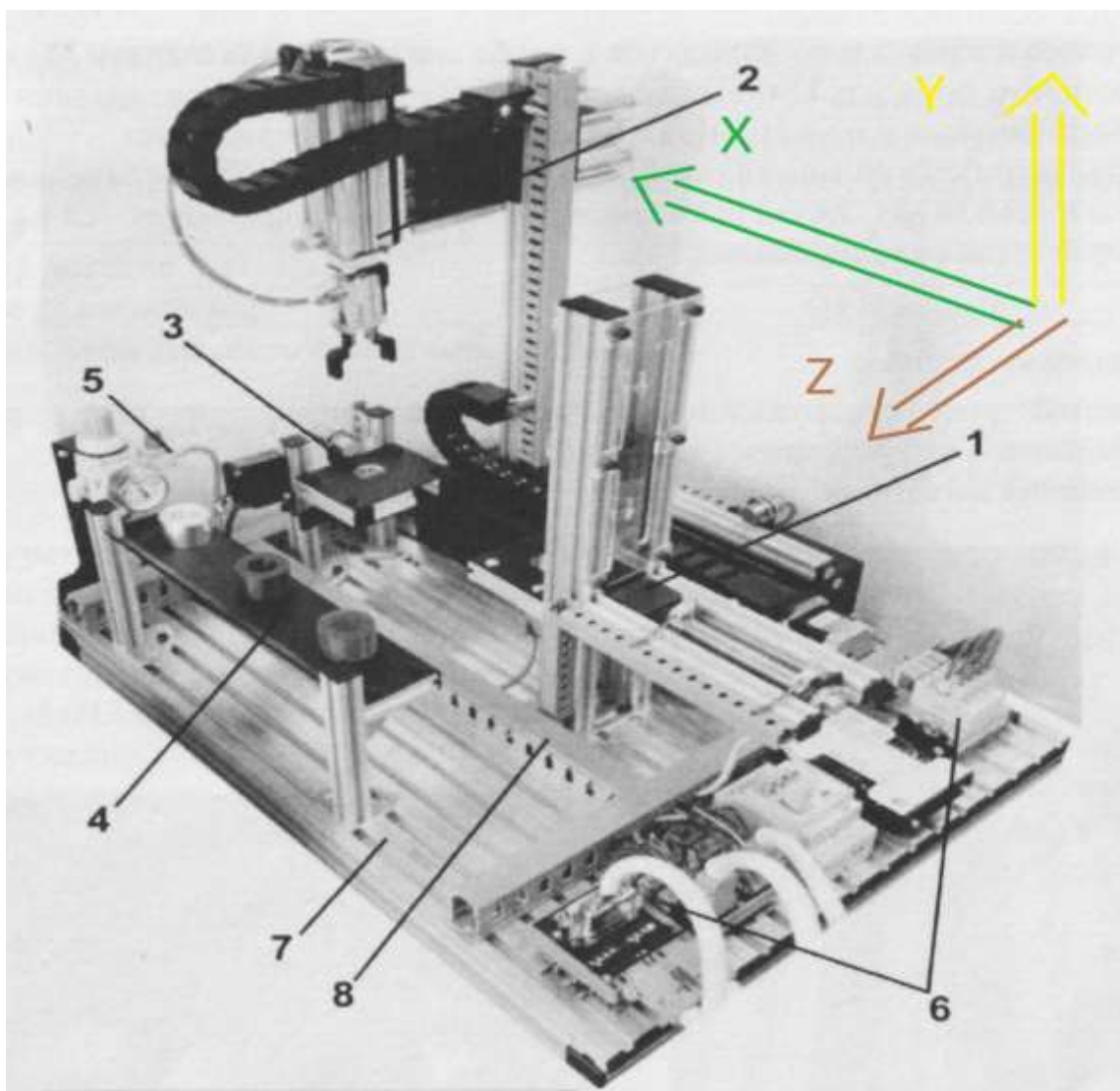


Рис. 2.2. Состав мехатронного модуля

Определим основные показатели манипулятора:

- по системе координат руки манипулятора: прямоугольная;
- по числу подвижностей манипулятора: 3 подвижные части;
- по грузоподъемности: сверхлегкие (до 10 Н);
- по подвижности основания: стационарная;
- по виду программы: перепрограммируемый;
- по типу силового привода: комбинированный (электрический и пневматический);

Почему же здесь используется комбинированный привод? Затем чтобы максимально использовать достоинства каждого привода. Двигатели горизонтального перемещения (по оси Z) и вертикального перемещения (по оси Y) - работают на пневматике. У обоих задача переместить из одной точки в другую без регулирования положения. А за перемещение по оси X отвечает шаговый двигатель, от него уже требуется переместить деталь в конкретное место на площадке (4) в зависимости от цвета и материала детали. Программа позволяет отсчитывать кол-во импульсов подаваемых на шаговый двигатель и тем самым рассчитать, куда приедет манипулятор. Контроллер шагового двигателя осуществляет подачу питания на ШД при получении сигнала идущего с ПЛК.

3 Программируемый логический контроллер в роботизированном комплексе.

Моноблок «Программируемый логический контроллер»

Конструктивно моноблок «Программируемый логический контроллер» (далее «ПЛК») представляет собой короб с металлическим кожухом и дюралюминиевой передней панелью (рис. 3).

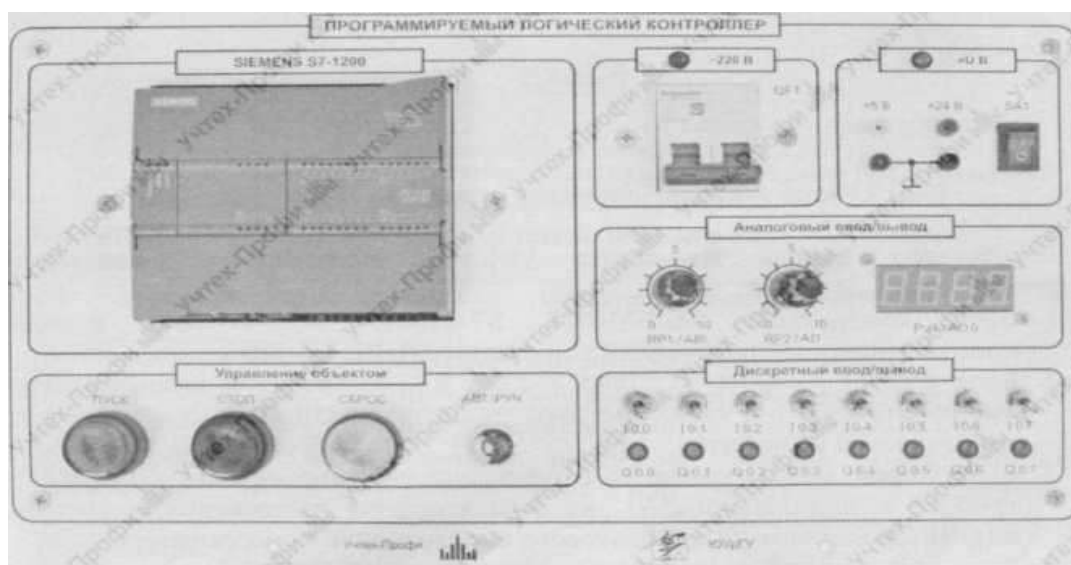


Рис. 3 Внешний вид ПЛК.

Лицевая панель моноблока условно разделена на следующие функциональные блоки:

1. Блок контроллера «SIEMENS S7-1200», включающий в себя программируемый логический контроллер (ПЛК) S7-1200.
2. Блок подачи питания на стенд « ~ 220 В», содержащий автоматический выключатель QF1, а также световой индикатор наличия высоковольтного напряжения в моноблоке.
3. Блок подачи низковольтного напряжения на стенд «=UB», содержащий тумблер с индикацией SA1, клеммы контроля напряжений и световой индикатор наличия низковольтного напряжения в моноблоке.
4. Блок «Аналоговый ввод\вывод», содержащий два потенциометра RP1, RP2 для имитации входных аналоговых сигналов ПЛК и цифровой индикатор PV1 для отображения напряжения выходного аналогового сигнала ПЛК.

5. Блок «Дискретный ввод\вывод» предназначен для управления мехатронным модулем в ручном режиме. Блок включает в себя тумблеры, имитирующие включение дискретных выходов ПЛК и светодиодные индикаторы, отображающие состояние датчиковой аппаратуры мехатронного модуля.
6. Блок «Управление объектом» предназначен для использования в качестве пульта управления мехатронным модулем. Блок включает в себя кнопки управления «Пуск», «Стоп» и «Сброс» со встроенными световыми индикаторами, а также переключатель режима работы «Авт\Руч».

В лабораторном стенде используется программируемый логический контроллер Siemens STMATTC S7-1200 1214C.

4 Основы программирования промышленных средств автоматизации.

4.1 Знакомство с интерфейсом TIA Portal V14

Программное обеспечение Totally Integrated Automation Portal (TIA Portal V14) предназначено для программирования и обслуживания логических контроллеров серии фирмы Siemens. В частности для программирования контроллеров используется программное обеспечение STEP7 Basic, встроенное в SIMATIC TIA PORTAL. Программа способна работать с программами в одном из трех видов: LAD (язык релейных схем), FBD (язык функциональных блоков) или STL (текстовый язык ассемблерного типа).

Главное окно программы TIA Portal по умолчанию содержит (рис. 4.1):

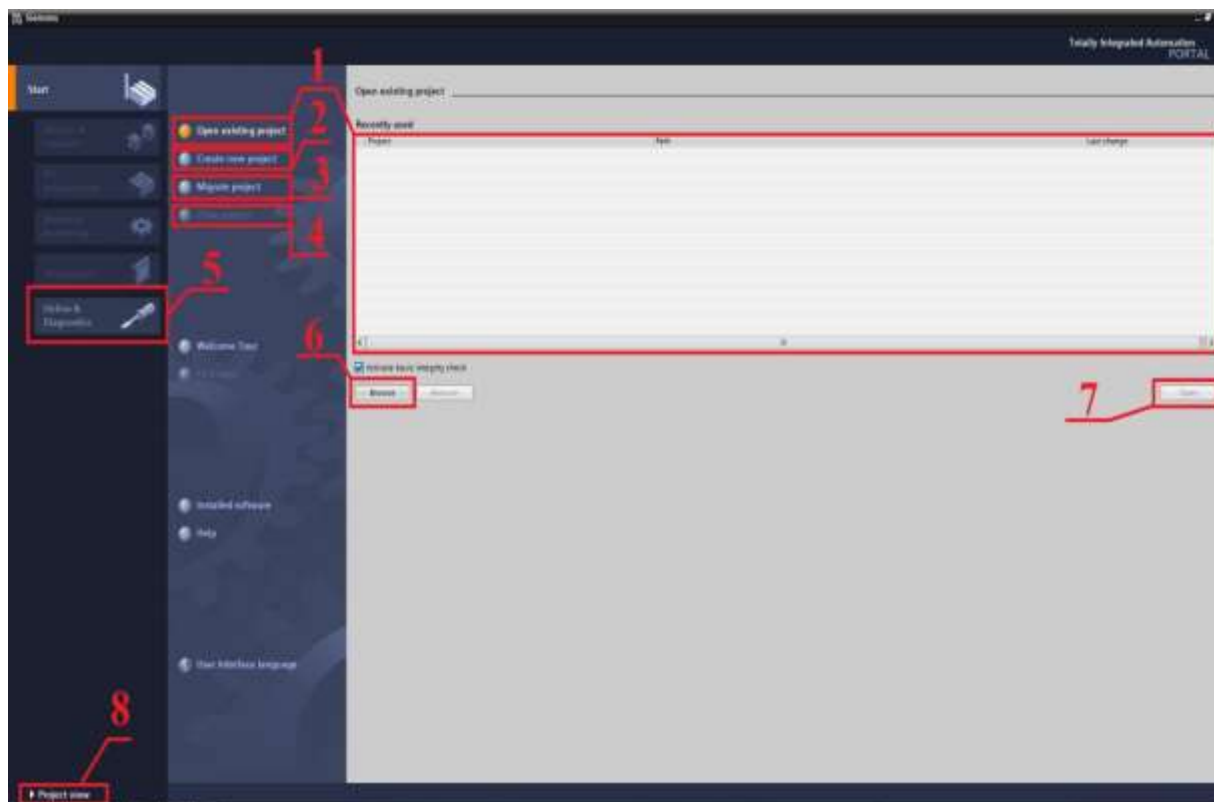
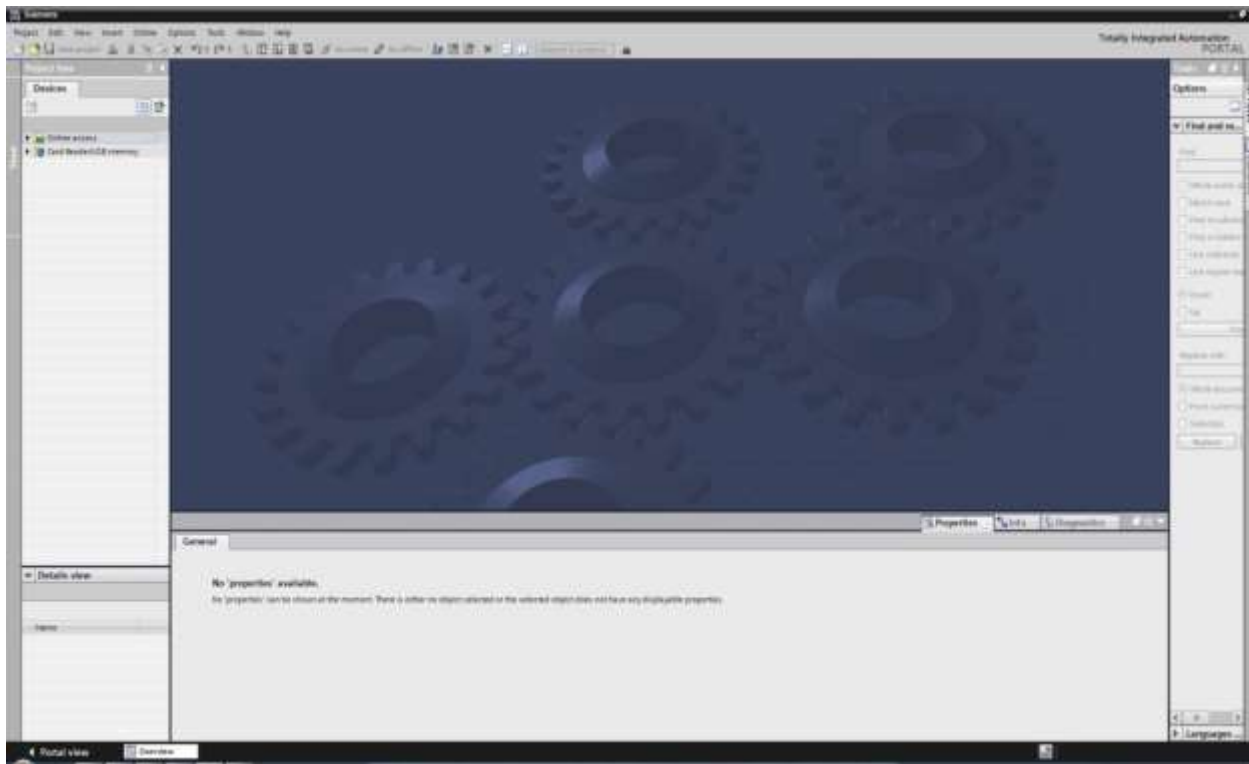


Рис.4.1. Главное окно TIA Portal / 1. — поле для выбора проекта; 2 — создание нового проекта; 3 — иммигрировать проект; 4 — Закрытие проекта; 5 — окно подключение и диагностики «ПЛК»; 6 — Поиск гото-

вого проекта на ПК; 7 — открытие выбранного проекта (в п.1.); 8 — показать проект(древо проекта).

Для начала работы нужно нажать на кнопку Projectview (8 на рис.4.2)



Откроется окно проекта:

Для открытия нового проекта нужно нажать на кнопку «Project» (рис. 4.3.) (находится в верхнем левом углу экрана)

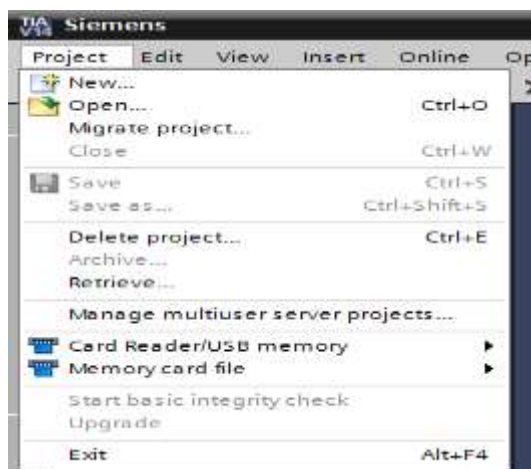
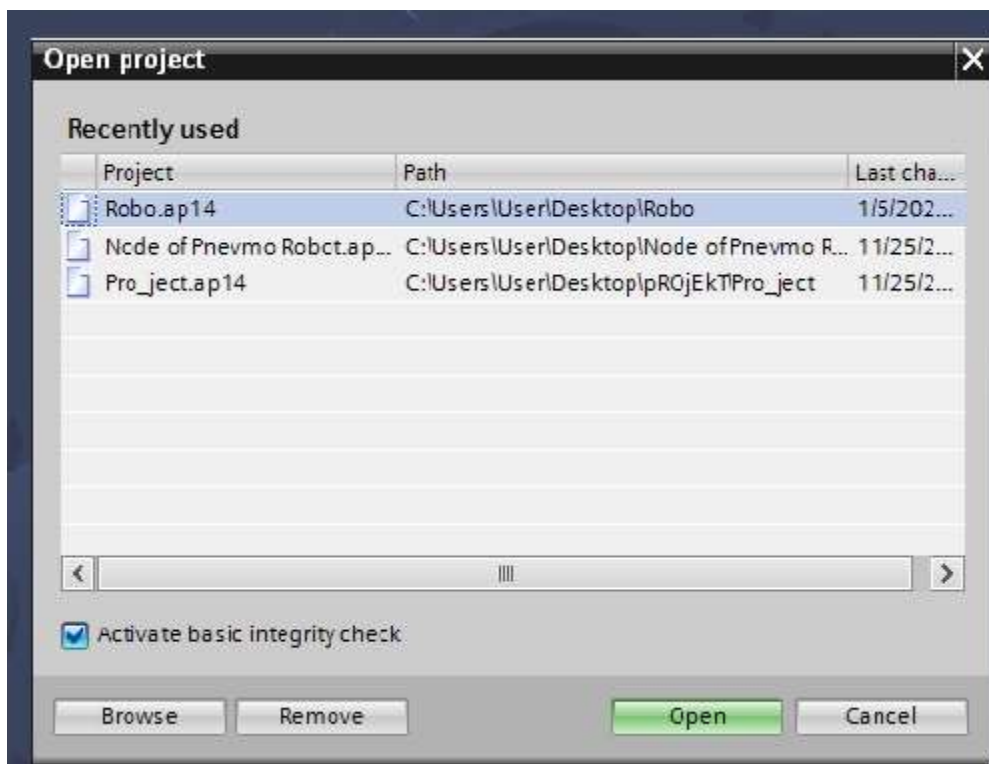


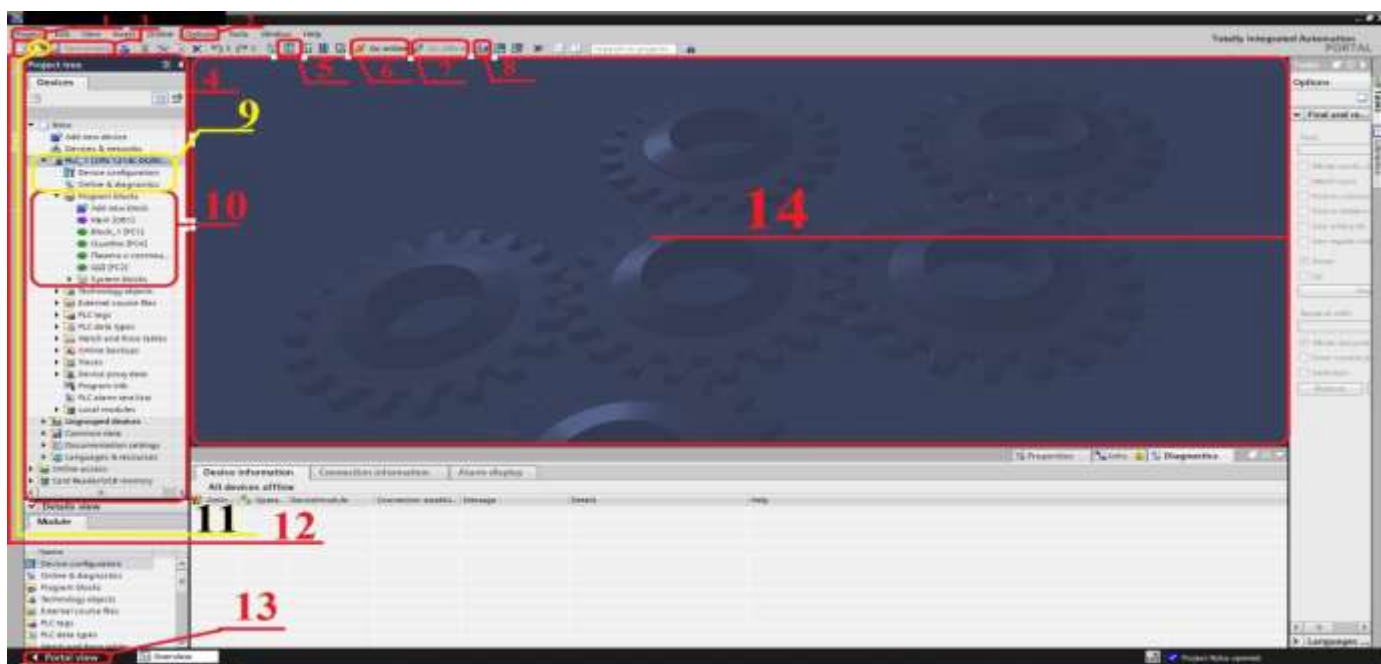
Рис. 4.3. Вкладка программы

Далее нажимаем New... (для создания нового проекта) или Open... (для открытия уже существующего «созданного» проекта). В нашем случае нажимаем Open...



Открывается окно в котором надо выбрать из существующих программ нужную нам и нажать кнопку Open выделенную зеленым цветом.

Откроется полноценное окно проекта с созданной программой (рис. 4.4)



которое выглядит следующим образом:

1 – Открытие, сохранение, удаление проекта, выход из программы (Рис. 8); 2 – Настройки; 3 – Сохранение проекта; 4 – Дерево проекта; 5 – Загрузка программы из компьютера в ПЛК; 6 – подключение в онлайн режим с ПЛК (позволяет следить за программой в реальном времени); 7 – оффлайн режим; 8 – поиск доступных устройств (ПЛК); 9 – окно ПЛК; 10 – Программные блоки; 11 – иконка открытия проектов; 12 – иконка создания нового проекта; 13 – возврат на главную; 14 – рабочее окно.

Чтобы подключиться к ПЛК следует нажать на 8 «accessibledevices». Откроется окно (рис 2.10) в котором нужно нажать «startsearch», после нажатия программа будет искать подключенный ПЛК и отображать его в этом же окне (Выдает ошибку если ПЛК был плохо подключен и оранжевый индикатор Rx/Tx горит, не мигая). После этого нужно выбрать этот ПЛК, нажав на «show», тем самым отобразим его в древе проекта на главном окне проекта.

5 Базовые элементы программирования на языке LAD.

LAD (LadderDiagram) – графический язык, основанный на принципах релейно-контактных схем (элементами релейно-контактной логики являются: контакты, обмотки реле, вертикальные и горизонтальные перемычки и др.) с возможностью использования большого количества различных функциональных блоков. Достоинствами языка LAD являются: представление программы в виде электрического потока (близко специалистам по электротехнике), наличие простых правил, использование только булевых выражений. На рис. 5.1 приведён пример программы на языке LAD (слева) и ее эквивалент в виде электрической цепи с реле и выключателями (справа).

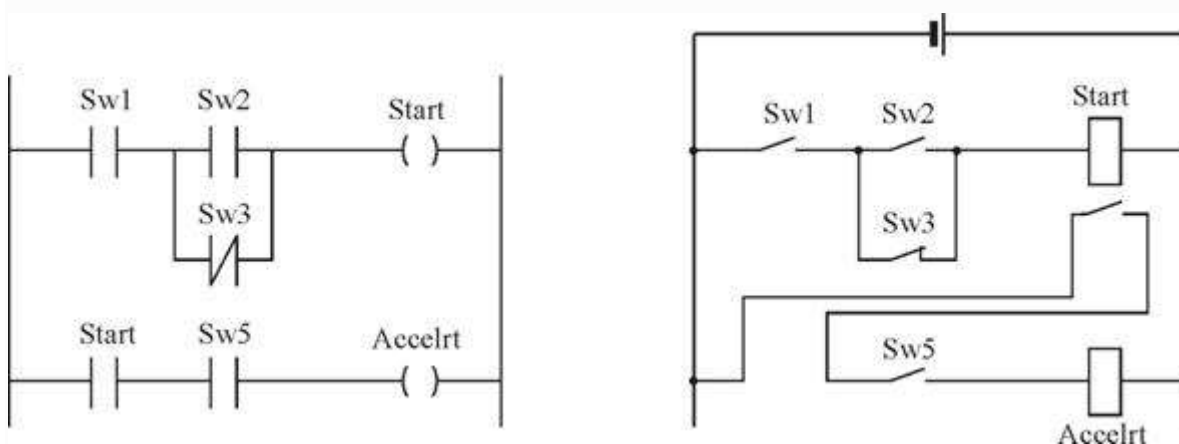


Рис. 5.1 Программа на языке LAD (слева) и ее эквивалент в виде электрической (справа).

Схемы, реализованные на данном языке, называются многоступенчатыми. Они представляют собой набор горизонтальных цепей, напоминающих ступеньки лестницы, соединяющих вертикальные шины питания. Объекты языка программирования LAD обеспечивают средства для структурирования программного модуля в некоторое количество контактов, катушек. Эти объекты взаимосвязаны через фактические параметры или связи. Порядок обработки индивидуальных объектов в LAD-секции определяется потоком данных внутри секции. Ступени, подключенные к левой шине питания, обрабатываются сверху вниз (соединение к левой шине питания). Ступени внутри секции, которые не зависят друг от друга, обрабатываются в порядке размещения.

Основные конструкции языка.

Слева и справа схема на языке LD ограничена вертикальными линиями – шинами питания. Между ними расположены цепи, образованные контактами и катушками реле, по аналогии с обычными электронными цепями. Слева любая цепь начинается набором контактов, которые посылают слева направо состояние «ON» или «OFF», соответствующие логическим значениям TRUE или FALSE. Каждому контакту соответствует логическая переменная (типа BOOL). Если переменная имеет значение TRUE, то состояние передается через контакт. Иначе – правое соединение получает значение выключено (“OFF”).

Контакты могут быть соединены параллельно, тогда соединение передает состояние «логическое ИЛИ». Если контакты соединены последовательно, то соединение передает «логическое И».

Контакт может быть инвертируемым. Такой контакт обозначается с помощью символа $|/|$ и передает состояние “ON”, если значение переменной FALSE.

Язык LD позволяет:

- выполнять последовательное соединение контактов;
- выполнять параллельное соединение контактов;
- применять нормально разомкнутые или замкнутые контакты;
- использовать переключаемые контакты;
- записывать комментарии;
- включать Set/Reset-выходы (Установка/Сброс);
- переходы;
- включать в диаграмму функциональные блоки;
- управлять работой блоков по входам EN.

Контактом является LD-элемент, который передает состояние горизонтальной связи левой стороны горизонтальной связи на правой стороне. Это состояние – результат булевой AND-операции состояния горизонтальной связи с левой стороны с состоянием ассоциированной переменной или прямого адреса. Контакт не изменяет значения связанной переменной или прямого адреса.

Для нормальных контактов (см. рис. 5.2) состояние левой связи передается в правую связь, если состояние связанного логического фактического параметра TRUE. Иначе, состояние правой связи FALSE.

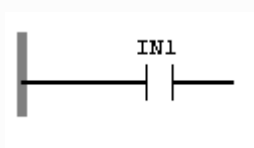


Рис. 5.2 Нормальный контакт

Для инверсных контактов (см. рис. 5.3) состояние левой связи передается в правую связь, если состояние связанного логического фактического параметра FALSE. Иначе, состояние правой связи TRUE.

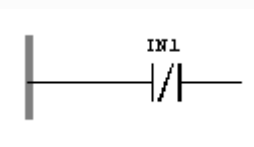


Рис. 5.3 Инверсный контакт

В контактах для обнаружения нарастания фронта (см. рис 5.4) правая связь устанавливается в состояние TRUE, если переход связанного фактического параметра происходит из FALSE в TRUE, и в то же время состояние левой связи TRUE. Иначе, состояние правой связи FALSE.

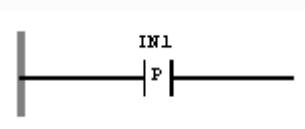


Рис. 5.4 Контакт для обнаружения нарастания фронта

В контактах для обнаружения спада фронта (см. рис. 5.5) правая связь устанавливается в состояние TRUE, если переход связанного фактического параметра происходит из True в False, и состояние левой связи True в то же время. Иначе, состояние правой связи FALSE.

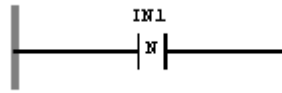


Рис. 5.5 Контакт для обнаружения спада фронта

Катушка является LD-элементом, который передаёт состояние горизонтальной связи на левой стороне неизменяемым горизонтальной связи на правой стороне. В этом процессе состояние связанной переменной или прямого адреса будет сохранено.

В нормальных катушках (см. рис. 5.6) состояние левой связи передается в связанный логический фактический параметр и в правую связь.

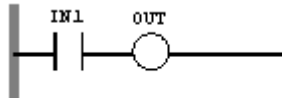


Рис. 5.6 Нормальная катушка

В инвертирующей катушке (см. рис. 5.7) состояние левой связи копируется в правую связь. Инвертированное состояние левой связи копируется в связанный логический фактический параметр. Если связь находится в состоянии FALSE, тогда правая связь тоже будет находиться в состоянии FALSE, и связанный логический фактический параметр будет находиться в состоянии TRUE.

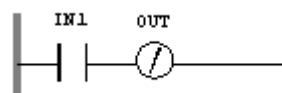


Рис. 5.7 Инвертирующая катушка

В катушке установки (см. рис. 5.8) состояние левой связи копируется в правую связь. Связанный логический фактический параметр устанавливается в состояние TRUE, если левая связь имеет состояние TRUE, иначе он не изменяется. Связанный логический фактический параметр может сбрасываться только катушкой сброса.

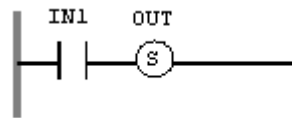


Рис. 5.8 Катушка установки

В катушке сброса (см. рис. 5.9) состояние левой связи копируется в правую связь. Связанный логический фактический параметр устанавливается в состояние FALSE, если левая связь имеет состояние TRUE, иначе он не изменяется. Связанный логический фактический параметр может устанавливаться только катушкой установки.

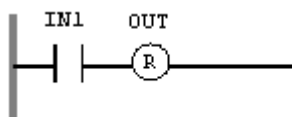


Рис. 5.9 Катушка сброса

В катушке обнаружения нарастания фронта (см. рис. 5.10) состояние левой связи копируется в правую связь. Связанный фактический параметр типа данных BOOL будет установлен в состояние TRUE для цикла программы, если произошел переход левой связи из FALSE в TRUE.

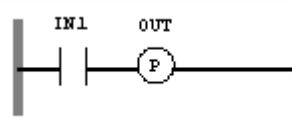


Рис. 5.10 Катушка обнаружения нарастания фронта

В катушке обнаружения спада фронта (см. рис. 5.11) состояние левой связи копируется в правую связь. Связанный фактический параметр типа данных BOOL будет установлен в состояние TRUE для цикла программы, если произошел переход левой связи из TRUE в FALSE.

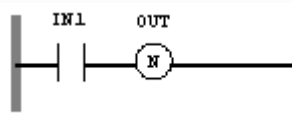


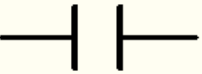




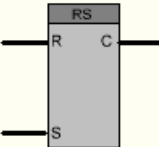
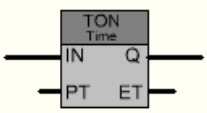
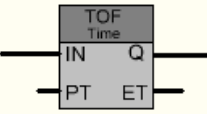
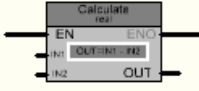
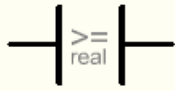
Рис. 5.11 Катушка обнаружения спада фронта

Слово «катушка» имеет обобщенный образ исполнительного устройства, поэтому в русскоязычной документации обычно говорят о выходе цепочки, хотя можно встретить и частные значения термина, например катушка реле.

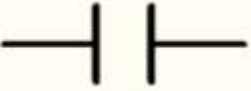

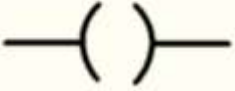
Левая шина питания соответствует единичному сигналу. Ступени, подключённые к левой шине питания, обрабатываются сверху вниз (соединение к левой шине питания).

6 Основные элементы, используемые в TIA PORTAL V14 и их значение.

Основные элементы, которые используются для программирования учебного роботизированного комплекса, приведены в таблице ниже:

	<i>Нормально разомкнутый контакт</i>
	<i>Нормально разомкнутый контакт</i>
	<i>Coil (Катушка) при получении сигнала размыкает/закрывает одноименные контакты</i>
	<i>S - присваивает указанному адресу логическую "1", пока не сработает R по такому же адресу</i>
	<i>R - Присваивает логический "0", если на указанном адресе в данный момент "1", если нет - не оказывает влияния</i>
	<i>Тот же RS триггер, объединяющий 2 предыдущих элемента в один блок. При получении сигнала на S выдает из C логическую "1", до сброса</i>
	<i>Таймер задержки включения на определенное время. При получении сигнала на IN начинает отсчет времени, затем пропускает сигнал дальше.</i>
	<i>Таймер задержки отключения на определенное время.</i>
	<i>Математический элемент, позволяющий сложить/вычесть/разделить/умножить 2 сигнала между собой.</i>
 195000.0	<i>Математический элемент позволяющий сравнить определенный сигнал с константой (или другим сигналом).</i>

Так же в TIA PORTAL V14 кроме наименования элемента используются обозначение бирюзового цвета, например %M0.5. Подробнее рассмотрим эти обозначения в таблице ниже:

<p style="text-align: center;"> %M0.5 "ЭЦ1"  </p>	<p><i>%M - Меркер т.е. это внутренняя(програмная) переменная, которая никак не связана с физическими выходами ПЛК. Она может выполнять переключения внутри программы, но не "снаружи". При получении всех необходимых условий как в релейной схеме на него подастся питание и он замкнет/разомкнет свои одноименные контакты. Цифра 0.5 означает байт и его бит занятый в карте памяти ПЛК, всего в байте 8 бит(0-7)</i></p>
<p style="text-align: center;"> %I1.0 "Пуск"  </p>	<p><i>%I - Физический вход ПЛК, который принимает сигналы датчиков/концевиков, если датчик сработал, этот контакт разомкнется т.е. сработает.</i></p>
<p style="text-align: center;"> %Q0.6 "Подача_YA4"  </p>	<p><i>%Q - это физический выход ПЛК, который при срабатывании в программе подаст управляющий сигнал на реальную эл.схему (например, подаст управляющий сигнал на контроллер шагового двигателя).</i></p>

7 Основы построения систем позиционного управления.

Позиционные системы управления являются основой многих крупных технологических линий. К ним предъявляются требования точной и быстрой отработки заданного перемещения без перерегулирования. Перерегулирование может быть недопустимо по технологии процесса или нежелательно из-за увеличения времени управления и лишних ударов в механизмах при наличии зазоров в передачах.

Позиционные системы управления являются, как правило, нелинейными системами, так как работают в режиме ограничения координат движения. Так, при средних перемещениях ограничивается момент двигателя, а при больших - его скорость. Ограничение координат движения определяет в основном время управления. Зазор между валом двигателя и выходным валом механизма, на котором установлен датчик позиции, может привести к неустойчивости системы или увеличению времени управления. Нелинейным может быть и регулятор позиции, обеспечивающий оптимальное управление при наличии ограничений координат движения.

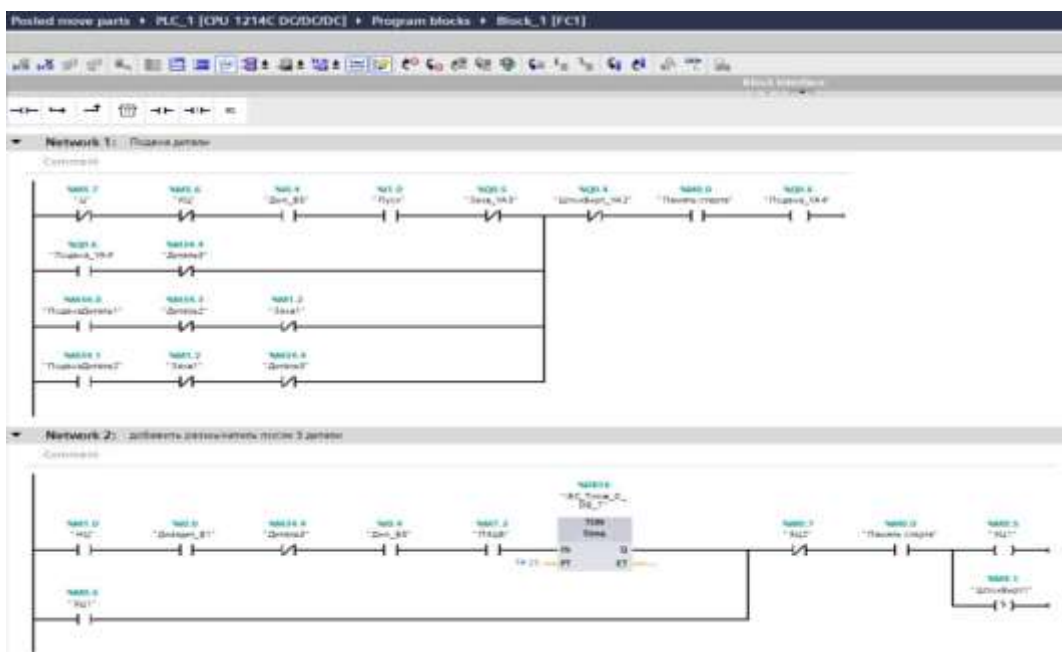
Для ряда механизмов характерно использование понижающих редукторов с большим передаточным числом. В этом случае после остановки привода нагрузка со стороны выходного вала редуктора не может вызвать перемещение со стороны входного вала. При этом статический момент, связанный с ошибкой позиционирования, определяет ток при стоящем двигателе. Это явление особенно неприятно для двигателя постоянного тока, через ряд коллекторных пластин которого постоянно протекает ток. В этом случае в регуляторе вводится зона нечувствительности и после вхождения ошибки в эту зону регуляторы блокируются. В других случаях требуемая позиция должна постоянно поддерживаться постоянным моментом.

8 Составление программы «Пневмоэлектрического манипулятора» в соответствии с техническим заданием.

По техническому заданию требуется, чтобы манипулятор равномерно разложил детали по очереди на узел складирования, не зависимо от их цвета и материала. Для простоты мы не стали убирать действие, когда он подносит деталь к датчикам.

Если будем создавать программу сначала, то следует нажать «add new block» в «program blocks» (10 на рис 4.) назовем блок «Block_1» он будет давать разрешение на работу ШД выдавать информацию, куда перемещать манипулятор, когда брать деталь, куда положить и т.п.

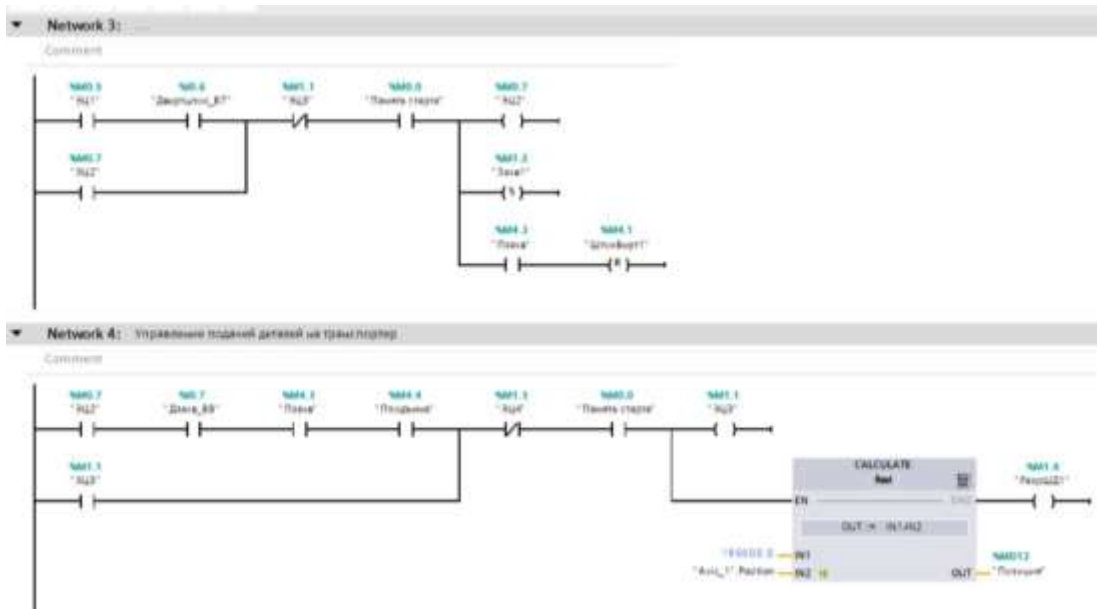
Составим уравнение для подачи детали, и поставим условия для срабатывания 1 цикла(ЭЦ1).



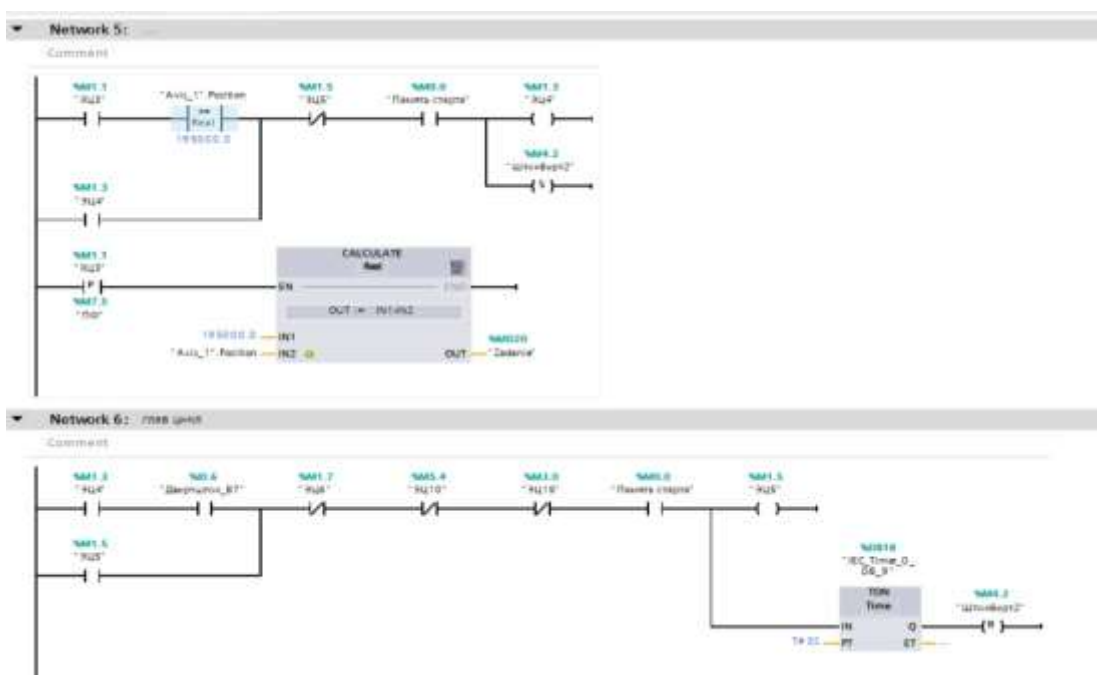
Деталь будет подаваться, только если манипулятор стоит около трубы с деталями, и нажата кнопка пуск, либо когда манипулятор возвращается, после того как переместил первую или вторую деталь и вернулся в начало. Когда последняя деталь(3) будет поставлена – следующая деталь подаваться не будет.

Условия для срабатывания НЦ – положение манипулятора в начале пути(у трубы), и срабатывание датчика наличия детали.

Нормально замкнутый контакт «Деталь3» не дает манипулятору запускать программу заново, размыкается, если 3 по счету деталь поставлена на платформу.



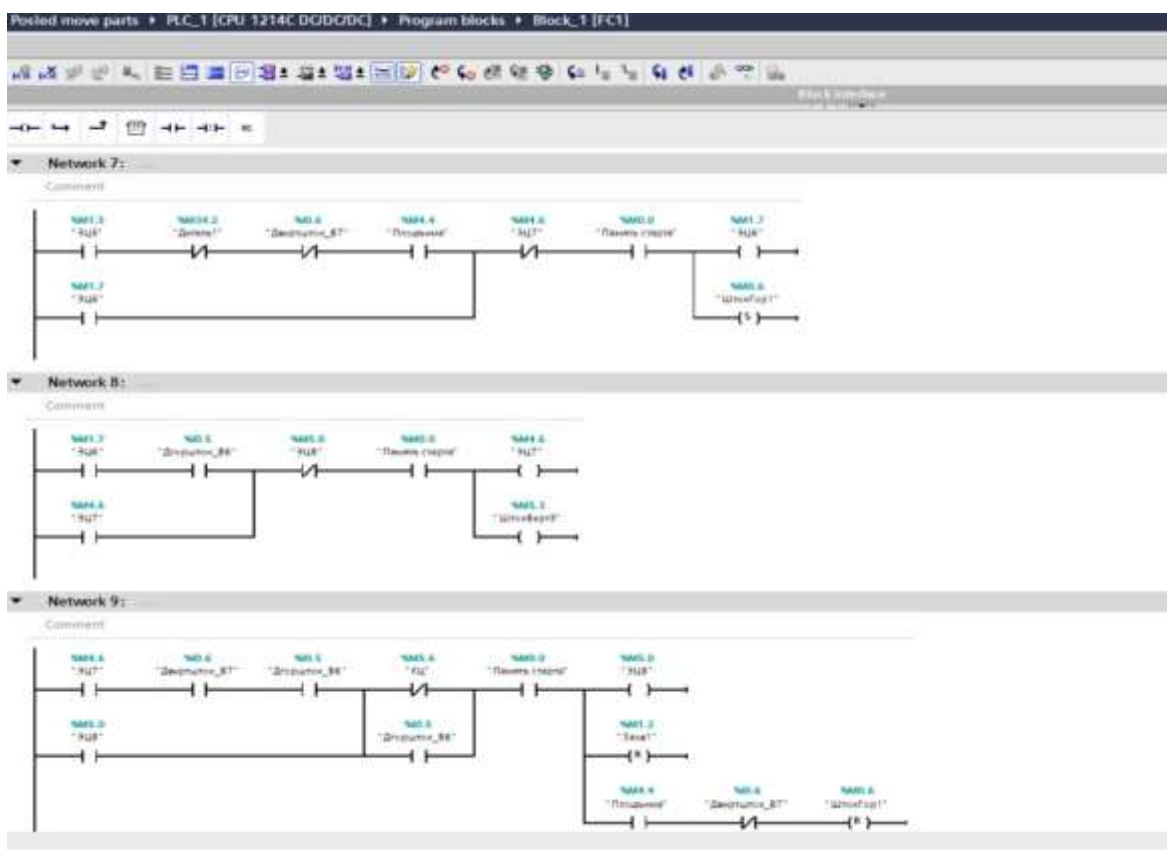
С 1 – 5 циклы манипулятор выполняет действия: опуститься, захватить деталь, поднять деталь, затем с помощью calculate высчитывается конечная позиция, ШД запускается, манипулятор движется к платформе с датчиками, пока не сработает элемент сравнения Axis_1.position конечное положение манипулятора должно быть ≥ 195000.0



Затем манипулятор остановится и опустит деталь.

Пояснение: в рамках технического задания, не убрано действие для опознавания деталей в зависимости от цвета/ материала. Оставили как есть, но значения цветового и индукционного датчика не будут влиять на программу. По техническому заданию требуется, чтобы 3 детали не зависимо от цвета и материала, были равномерно установлены по порядку на узле складирования.

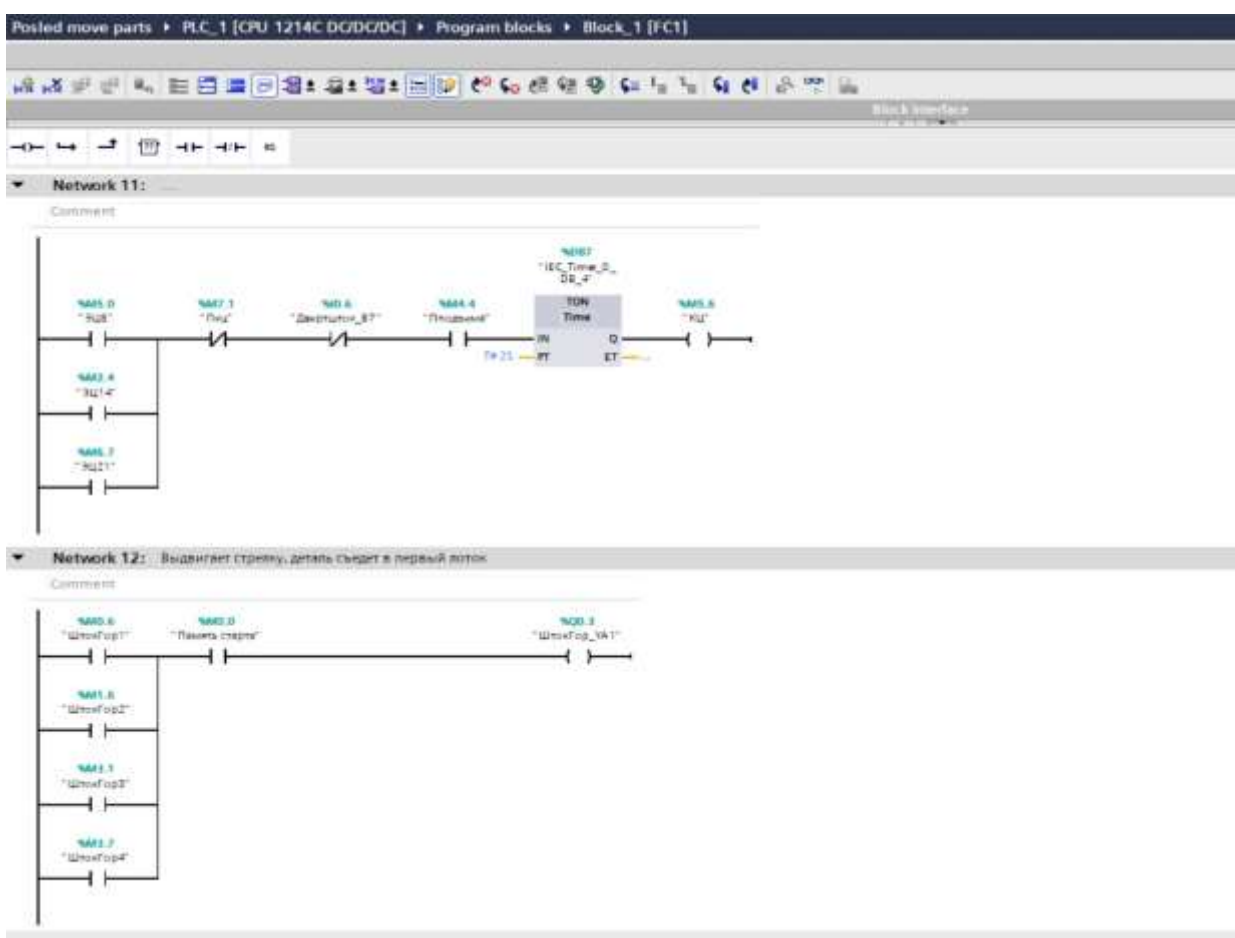
Далее манипулятор будет работать по ЭЦ6-ЭЦ8. Где он поднимет деталь с узла определения цвета и материала сразу переместит на узел складирования с помощью горизонтального штока. У него есть 2 штока горизонтальный - перемещает на узел складирования и вертикальный – поднимает и опускает деталь. Так же есть пневмозахват, он удерживает деталь.



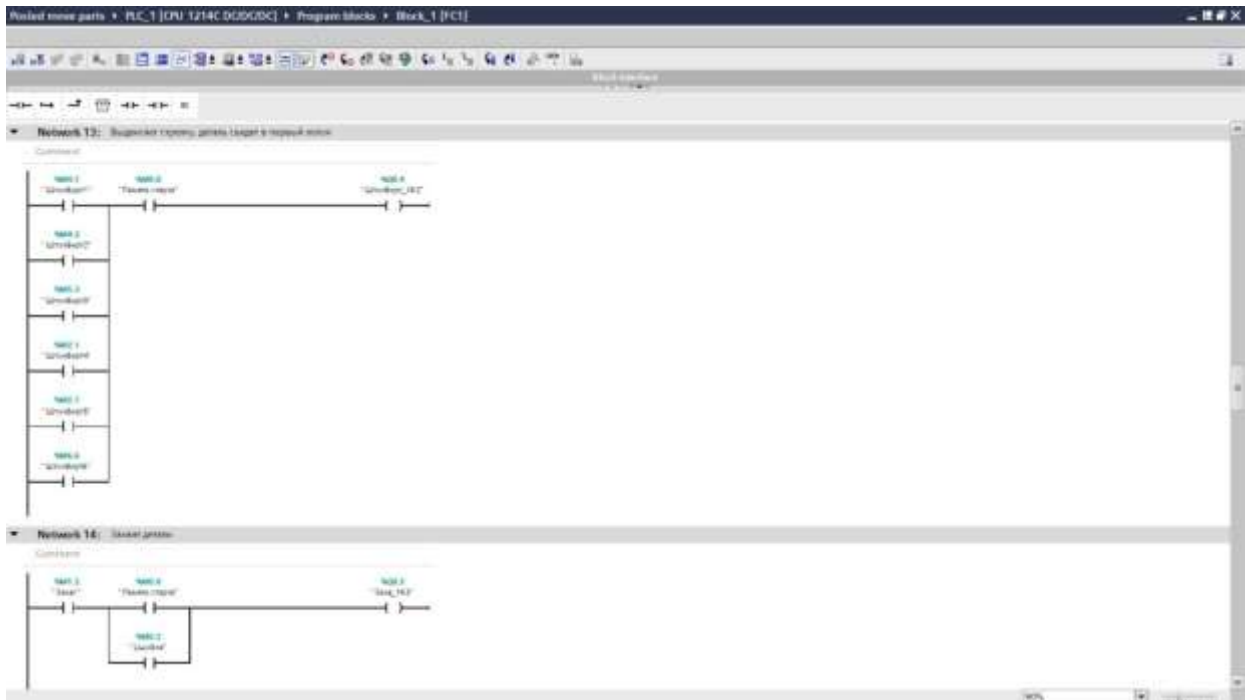
Затем, когда манипулятор отнес деталь, опустил и вернулся в прежнее состояние, срабатывает КЦ (конец цикла) условия для его работы: срабатывание ЭЦ8(1 деталь доставлена), ЭЦ14(2 деталь доставлена), ЭЦ21(3 деталь доставлена). КЦ краткосрочно сработает и разомкнет нужные контакты, сбросится позиция в блоке ШД и манипулятор поедет к узлу выдачи деталей (начальное положение). Там выполняются условия для подачи детали, датчик обнаружит деталь, и цикл ЭЦ1-5 повторится

точь-в-точь. Но из-за того, что RСтриггер запомнил, что циклы 6-8 выполнялись, деталь перенесет в другое место, по циклам ЭЦ10-14. Алгоритм тот же, но манипулятор отнесет деталь в середину узла складирования.

После этого манипулятор вернется и возьмет 3 деталь, RСтриггер запомнил, что деталь 2 была доставлена, повторять циклы 6-8 и 10-14 – нельзя. Он по-прежнему выполнит циклы 1-5 и перенесет деталь по циклам 15-21. Последний RСтриггер запомнил срабатывание ЭЦ21, манипулятор вернется в исходное положение и закончит работу. Кнопка «сброс» - сбросит RS триггеры и остальные состояния программы.



Здесь мы видим, какие условия срабатывания горизонтального, вертикального штока и захвата детали.



Кнопка «сброс» позволяет полностью обнулить все состояния в программе.

RСтриггеры - запоминают куда детали уже ставили. Есть переменные, которые выдают логическую «1» только когда манипулятор в начальном положении (Подача Деталь1-3) – это сделано для того, чтобы деталь подавалась только когда манипулятор готов забрать ее. Остальные переменные всегда выдают «1» (Деталь1-3) при срабатывании RСтриггера – они не позволяют ставить детали на одни и те же места.

Заключение

Методическая разработка предназначена для студентов 3-4 курса, изучающих тему «Электрооборудование промышленных роботов» в МДК.01.04. «Электрическое и электромеханическое электрооборудование». При изучении разработки следует обратить внимание, что в ней при программировании реализован один из возможных вариантов реализации работы пневматического робота. Обучаемым необходимо предложить разработать собственные варианты программ работы робота с последующим опробованием на лабораторной установке.

