

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт машиностроения  
(наименование института полностью)

---

Кафедра «Промышленная электроника»  
(наименование)

11.03.04 Электроника и нанoeлектроника  
(код и наименование направления подготовки/ специальности)

---

Электроника и робототехника  
(направленность (профиль) / специализация)

---

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

по теме Система управления транспортным роботизированным модулем  
на базе ПЛК Siemens

---

Студент

А.И. Золотов  
(И.О. Фамилия)

---

(личная подпись)

Руководитель

к.т.н., доцент, Д.Г. Токарев  
(ученая степень, звание, И.О. Фамилия)

---

Консультант

к.ф.н , доцент, М.М. Бажутина  
(И.О. Фамилия)

---

Тольятти 2022

## Аннотация

Название бакалаврской работы: "Система управления транспортным роботизированным модулем на базе ПЛК Siemens".

Работа состоит из пояснительной записки на 83 страницах, включающей 56 рисунков, 10 таблиц, списка литературы из 30 наименований, в том числе 7 иностранных источников, 6 приложений и графической части на 10 листах формата А1.

Целью работы является разработка и проектирование учебно-лабораторного стенда на базе ПЛК Siemens, а также разработка методических рекомендаций к нему.

В первой части рассмотрены готовые решения, сформирована концепция разрабатываемого устройства и подобрана необходимая компонентная база.

Во второй части рассмотрены объекты управления, входящие в состав разрабатываемого комплекса, их основные части, принципы функционирования и некоторые характеристики.

В третьей части рассматривается компонентная база и аппаратное обеспечение, разработанного интерфейсного модуля,

В четвертой части рассмотрен промышленный протокол связи Modbus TCP/IP и его программная реализация для client и server устройств.

В пятой части рассмотрены алгоритмы работ транспортного роботизированного комплекса и их реализация.

В шестой части рассмотрены методические рекомендации по работе со стендом.

Результатом выполнения бакалаврской работы является программа, методические рекомендации и практическая реализация транспортного роботизированного комплекса, который при дополнительной настройке и доработке может быть включен в перечень лабораторных стендов, используемых при обучении.

## **Abstract**

The title of the bachelor's work is "The control system of transport robotic module based on PLC Siemens".

The work consists of an explanatory note on 83 pages, including 56 figures, 10 tables, a list of 30 references, including 7 foreign sources, 6 appendices and a graphic part on 10 sheets of A1 format.

The purpose of the work is to develop and design a training-laboratory bench on the basis of PLC Siemens, as well as to develop methodological recommendations for it.

In the first part, ready-made solutions are considered, the concept of the device under development is formed and the necessary component base is selected.

In the second part, the control objects included in the developed complex, their main parts, principles of functioning and some characteristics are considered.

In the third part, the component base and hardware of the developed interface module are considered.

In the fourth part, the industrial communication protocol Modbus TCP/IP and its software implementation for client and server devices are considered.

In the fifth part, the algorithms of the transport robotic complex and their implementation are considered.

The sixth part considers methodological recommendations for work with the stand.

The result of the bachelor's work is the program, methodological recommendations and practical implementation of the transport robotic complex, which with additional adjustment and modification can be included in the list of laboratory simulators used in the training.

## Содержание

Введение.....	6
1 Состояние вопроса .....	7
1.1 Формулирование актуальности, цель и задачи проекта .....	7
1.2 Обзор существующих технологий и решений.....	8
1.2.1 Применение промышленных манипуляторов.....	8
1.2.2 Классификация промышленных роботов-манипуляторов .....	11
1.2.3 Системы управления промышленными манипуляторами.....	12
1.2.4 Анализ данных по проекту и определение его концепции.....	14
2 Объекты управления .....	19
2.1 Конвейер .....	19
2.2 Манипулятор МП-11 .....	20
3 Интерфейсный модуль.....	27
3.1 Аппаратное обеспечение интерфейсного модуля .....	29
3.2 Компонентная база интерфейсного модуля .....	32
3.2.1 Изоляционный модуль DI .....	35
3.2.2 Релейный модуль DO.....	37
3.2.3 Питание интерфейсного модуля.....	39
3.2.4 Ethernet sheild.....	40
4 Промышленный протокол связи Modbus TCP/IP.....	43
4.1 Настройка соединения client устройства .....	44
4.2 Настройка соединения server устройства.....	49
5 Система управления.....	55
5.1 Программируемый логический контроллер Siemens S7-1200 .....	56

5.2 Система прослеживаемости .....	61
5.3 Оценка времени выполнения микроконтроллером Atmel ATmega2560 команд высокого уровня .....	62
5.4 Первый алгоритм работы.....	67
5.5 Второй алгоритм работы.....	70
5.6 Демонстрационный режим работы.....	73
5.6 Режим ручного управления .....	75
5 Методические рекомендации по работе с лабораторным стендом.....	78
Заключение .....	80
Список используемой литературы .....	81
Приложение А Программный код первого алгоритма работы.....	84
Приложение Б Программный код второго алгоритма работы.....	90
Приложение В Программный код демонстрационного режима .....	95
Приложение Г Программный код режима ручного управления .....	98
Приложение Д Программный код client устройства .....	103
Приложение Е Примеры лабораторных работ .....	106
Приложение Ж Перечень элементов к схеме электрической соединений и подключения манипулятора МП-11 .....	111
Приложение И Перечень элементов к схеме электрической соединений интерфейсного модуля .....	112
Приложение Л Перечень элементов к схеме электрической соединений разъемов интерфейсного модуля .....	113

## Введение

В современном мире роботизация является одной из самых востребованных и быстроразвивающихся частей комплексной автоматизации производств. Практически любые автоматические линии можно оснастить или дополнить промышленными роботами, что крайне положительно скажется на качестве их функционирования и производительности.

Промышленный робот представляет собой автоматическое устройство, состоящее из манипулятора и программируемого устройства управления и используется при перемещении объектов в пространстве или выполнения более сложных функциональных операций на различных производствах. Применение таких промышленных роботов в разы упрощает процесс производства и повышает его эффективность.

Среди основных способов, позволяющих сократить ручной труд, важное место отводится промышленным роботам-манипуляторам, как составному средству комплексной автоматизации. Промышленные роботы-манипуляторы заменяют работников на тех производственных участках, где необходимо выполнение тяжелого, монотонного или наоборот очень точного и быстрого выполнения работы. Также они активно применяются в условиях агрессивной окружающей среды, условиях с повышенной влажностью и температурой, сильным производственным шумом или риском взрывоопасности и утечки токсичных веществ.

Помимо этого, промышленные роботы-манипуляторы активно используются в сфере образования, в настоящее время стоит острая потребность в квалифицированных кадрах, имеющих навыки в автоматизации, программировании и работе с роботами. Работа с учебными манипуляторами и комплексами на их основе дает весомую теоретическую и практическую базу в области автоматизации технологических процессов и управления промышленными роботами для молодых специалистов, а также рабочих кадров и инженеров, проходящих повышение квалификации.

## **1 Состояние вопроса**

### **1.1 Формулирование актуальности, цель и задачи проекта**

Актуальность данной работы заключается в активном и повсеместном применении промышленных манипуляторов и роботизированных комплексов на их основе в сфере автоматизации технологических процессов, в большой востребованности специалистов, обладающих навыками в области автоматизации производственных процессов, а также разработке и программировании систем управления.

Целью работы является разработка и проектирование учебно-лабораторного стенда на базе ПЛК Siemens, включающего в свой состав промышленный робот-манипулятор, а также разработка методических рекомендаций к нему, для повышения компетенций студентов кафедры «Промышленная электроника» Тольяттинского государственного университета в области электроники и автоматизации в период обучения.

Для достижения поставленной цели были выполнены следующие задачи:

- анализ исходных данных и известных решений;
- разработка структуры комплекса лабораторного стенда;
- подбор электронной и электрической компонентной базы;
- проработка и реализация электрических связей комплекса;
- проработка и реализация электронных связей комплекса
- физическая реализация лабораторного стенда;
- разработка методических рекомендаций по работе со стендом.

## 1.2 Обзор существующих технологий и решений

### 1.2.1 Применение промышленных манипуляторов

Применение роботизированных комплексов, включающих в свой состав различные виды роботов-манипуляторов и управляемые с помощью контроллеров и специализированных систем управления заключается в решении ряда технологических задач, которые возникают в промышленности, сфере образования, науке и медицине [19].

Самое активное применение роботов-манипуляторов прослеживается на предприятиях различного производственного назначения. В перечень допустимых к выполнению производственных операций входят все задачи с непосредственным участием машин в производстве. На рисунке 1 представлен пример функционирования робота-манипулятора в машиностроительном производстве.

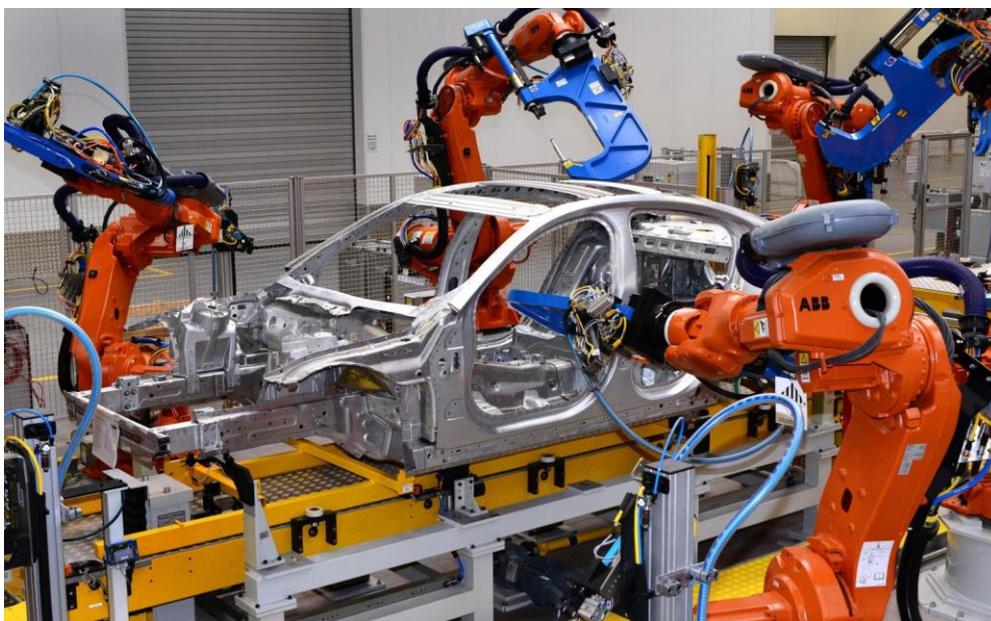


Рисунок 1 – Применение манипулятора в сборочно-кузовных работах

Современные промышленные роботы способны выполнять достаточно разнообразный перечень функциональных операций, таких как сварочные, транспортные операции, фрезеровка и обработка изделий, контроль качества и нанесение лакокрасочных покрытий [15].

Один из самых больших пластов задач, выполняемых комплексами промышленной робототехники на производствах, связан со вспомогательными операциями. В разряд вспомогательных операций входят: обслуживание станков и оборудования, выполнение транспортных функций, погрузка и сортировка. На рисунке 2 представлен манипулятор, выполняющий транспортные функции.



Рисунок 2 – Манипулятор, выполняющий транспортные функции

Возможности роботизации промышленности не ограничиваются данным списком. Гибкость систем управления в сочетании с вариативностью их использования, дают обширный спектр возможностей по подборке необходимого функционала, в соответствии с требованиями каждого отдельно взятого предприятия.

Помимо промышленности и задействования роботов-манипуляторов в сложных технологических операциях, активное применение они находят в области образования и подготовке квалифицированных инженерных и рабочих кадров. На рисунке 3 представлен учебный манипулятор от компании FISCHERTECHNIK.

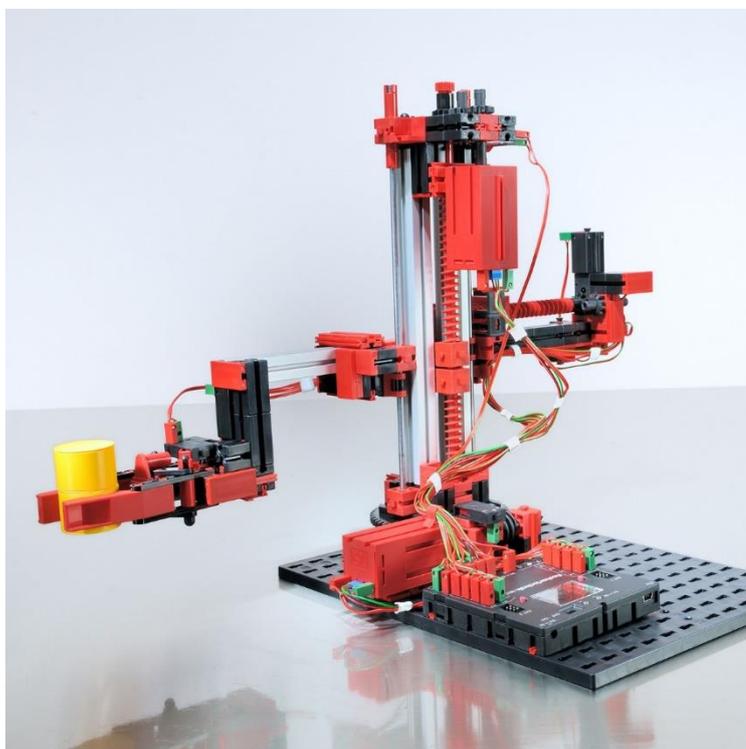


Рисунок 3 – Учебный манипулятор FISCHERTECHNIK

На данный момент существует множество производителей обучающих систем, наборов и конструкторов, позволяющих изучать основы автоматизации производственных процессов и принципы работы и управления манипуляторами и комплексами на их основе.

### **1.2.2 Классификация промышленных роботов-манипуляторов**

Классификация промышленных роботов осуществляется по достаточно обширному списку различных признаков, мы же, рассмотрим лишь те, которые непосредственно влияют на систему управления, задают конструктивные, функциональные особенности и влияют на степени свободы (подвижности) и ориентацию манипулятора в пространстве.

Конструктивное исполнение роботов-манипуляторов в первую очередь зависит от набора двигательных функций и последовательности, в которой необходимо будет использовать их в процессе работы. К примеру, для осуществления простейшей транспортной функции применительно к объекту манипулирования, без ориентации его в пространстве, достаточно будет обеспечить работа только тремя степенями свободы. В этом случае не принципиален вид используемых степеней свободы, они могут быть либо поступательными, либо вращательными. Данные степени свободы называются региональными. Региональные степени свободы определяют систему координат, в которой будет функционировать манипулятор. В зависимости от вида каждой из используемых степеней свободы и сочетании последовательностей их задействования, а также взаимной ориентации степеней подвижности складывается базовая система координат, в которой работает манипулятор. Эта база имеет свои характерные особенности и уникальную форму пространственной фигуры, которую описывает рабочий орган манипулятора в процессе движения. Руководствуясь вышеизложенным, можно сделать вывод, что вид базовой системы координат определяет конструктивное исполнение манипулятора и его систему управления [21].

### **1.2.3 Системы управления промышленными манипуляторами**

Все оборудование роботизированных комплексов, за исключением объектов управления, предназначено для формирования и передачи управляющих сигналов. Таким образом, система управления обрабатывает пользовательскую программу и в совокупности с сигналами от датчиков и программно заданными условиями, формирует управляющие сигналы на исполнительные механизмы манипулятора.

Система управления обобщенно представляет из себя последовательность позиций, через которые проходит манипулятор в процессе выполнения необходимых работ. Все манипуляторы, движение которых осуществляется немеханическими приводами оснащены системами управления. Системы управления современных манипуляторов состоят из нескольких подсистем, каждая из которых выполняет свои функции: информационные, управляющие, защитно-предупредительные. Основное устройство управления и прочие компоненты системы при всех видах системы управления могут базироваться на одинаковых или разных принципах, а также электронных и электрических элементных базах.

Существует три основных вида систем управления: цикловое, позиционное и контурное. При реализации цикловой системы управления программно задается определенная последовательность выполнения движений и условия их начала и окончания. Крайняя точка перемещения манипулятора в каждой из степеней свободы определяется самим манипулятором и никак не учитывается в программе, возможно лишь отслеживание крайних положений при наличии подключенных концевых выключателей или электромагнитных контактов. Скорость выполнения движений также не зависит от программы и определяется лишь параметрами исполнительных механизмов [22].

При реализации позиционной системы управления исполнительные команды подаются таким образом, что перемещение руки манипулятора происходит от одной точки, задаваемой программным путем, к другой. Скорость выполнения движений, как и в цикловой системе управления, определяется лишь параметрами исполнительных механизмов.

При реализации контурной системы управления, движения манипулятора происходят по заранее заданной траектории с необходимой скоростью. Траектория, скорость и режимы движения задаются программным путем. Контурные системы управления применяются в тех технологических процессах, где критичен такой параметр, как точность выполнения работы.

К специализированному виду систем управления можно отнести адаптивное управление. В роботизированных комплексах на его основе при поступлении внешней информации, например, с датчиков, определяющих положение робота в пространстве, осуществляется автоматическое изменение управляющей программы. Также, возможна реализация ручного управления при помощи операторского пульта. Оператор, с помощью пульта ручного управления, осуществляет управление и контроль за перемещениями манипулятора при помощи набора обратных связей.

В процессе развития промышленности, на производствах требуется обеспечение все большей точности и гибкости выполнения операций, что приводит к внедрению систем управления с элементами искусственного интеллекта. Данные системы, с помощью сенсоров и систем машинного зрения позволяют анализировать внешние условия и подстраивать специфику выполнения операции с учетом изменяющейся окружающей среды. Роботы с системами управления на основе искусственного интеллекта, способны выполнять более сложные технологические операции эффективнее, с большей точностью, скоростью и следуя более рациональному алгоритму.

В условиях реального производства используются именно специализированные модули управления (рис. 4), позволяющие легко и эффективно осуществлять управление роботами и производственными установками [30].



Рисунок 4 – Модуль управления роботом KUKA

Также, в качестве основы для построения систем управления могут выступать микроконтроллеры различных архитектур, программируемые логические контроллеры и специализированные модули управления.

#### **1.2.4 Анализ данных по проекту и определение его концепции**

Для выполнения выпускной квалификационной работы, были исследованы данные о видах манипуляторов, их конструктивных особенностях и системах управления. Ознакомившись с наличием материально-технической базы кафедры «Промышленная электроника» и техническим заданием, в качестве основного объекта управления был выбран промышленный пневматический робот-манипулятор МП-11 с цикловой

системой управления, осуществляющий свои перемещение в цилиндрической системе координат. Такие промышленные манипуляторы и роботизированные комплексы на основе их являются хорошим инструментом в области обучения, повышения квалификации студентов и работников, изучающих принципы автоматизации производств и основы управления роботизированными комплексами. При этом они просты в обслуживании, эксплуатации и не требуют больших затрат, специализированных систем и коммуникаций.

В качестве системы подачи воздуха будет использоваться компрессор Jun Air 6-25. На рисунке 5 представлен используемый компрессор и его основные характеристики.

Модель 6-25



Технические характеристики	
Рабочее давление, бар	8
Производительность, л/мин	50
Объём ресивера, л	25
Потребляемая мощность, кВт	0,34
Напряжение/частота, В/Гц	230/50
Максимальная сила тока, А	2,9
Уровень шума, дБ	45
Вес*, кг	29
Размеры* (ДхШхВ), мм	380x380x550

Рисунок 5 – Компрессор Jun Air 6-25.

Система управления будет основываться на программируемом логическом контроллере Siemens S7-1200. Для более приближенной имитации реального технологического процесса в состав разрабатываемого учебно-лабораторного комплекса также будет включен конвейер, реализованный в виде ленточного транспортера и человеко-машинный интерфейс с обеспечением системы прослеживаемости.

Для обеспечения питания комплекса будет использоваться блок питания SIMATIC PM 1507, представленный на рисунке 6.



Рисунок 6 – Блок питания SIMATIC PM 1507

На рисунках 7 и 8 представлены основные характеристики выбранного блока питания.

Вход	
Вход	1-фазный переменный ток
Напряжение питания	
1 при переменном токе номинальное значение	120 V
2 при переменном токе номинальное значение	230 V
Примечание	Автоматическое переключение диапазона
Входное напряжение	
1 при переменном токе	85 ... 132 V
2 при переменном токе	170 ... 264 V
Вход с широким диапазоном возможностей	нет
Устойчивость к перенапряжению	2,3 x U <sub>e</sub> ном, 1,3 мс
Резервное питание при исчезновении напряжения сети при номинальном I <sub>a</sub> , мин.	20 ms, при U <sub>e</sub> = 93/187 В
Номинальная частота сети 1	50 Hz
Номинальная частота сети 2	60 Hz
Диапазон частоты сети	45 ... 65 Hz
Входной ток	
при номинальном значении входного напряжения 120 В	3,7 А
1,7 А	
Ограничение пускового тока (+ 25 °C), макс.	62 А
длительность ограничения тока включения при 25 °C	
максимальное	3 ms
I <sub>t</sub> , макс.	12 А·с
Встроенный предохранитель при входе	T 6,3 A/250 В (недоступно)
Защита предохранителями в сетевой подводке (IEC 898)	рекомендованный LS-переключатель: 16 А характеристика В или 10 А характеристика С

Рисунок 7 – Входные характеристики блока питания

Выход	
Выход	регулируемое постоянное напряжение без потенциала
Номинальное значение напряжения $U_a$ Nenn DC	24 V
Общий допуск, статический $\pm$	1 %
сетевое статическое регулирование, ок.	0,1 %
регулирование статической нагрузки, ок.	0,1 %
Остаточная пульсация пиков амплитуды, макс.	50 mV
Пики амплитуды, макс. (ширина полосы пропускания ок. 20 МГц)	150 mV
Функция продукта выходное напряжение можно регулировать	нет
Индикаторное табло	светодиод зеленый для 24 В О.К.; светодиод красный для ошибки; светодиод желтый для дежурного режима
Режим включения/отключения	без отклонения напряжения $U_a$ (плавное включение)
Задержка запуска максимальная	1,5 s
Повышение напряжения, тип.	10 ms
Номинальная величина тока $I_a$ ном.	8 A
Диапазон тока	0 ... 8 A
отдаваемая активная мощность типовое	192 W
кратковременный ток перегрузки	
при коротком замыкании во время разгона типовое	35 A
при коротком замыкании во время эксплуатации типовое	35 A
длительность перегрузочной способности тока перегрузки	
при коротком замыкании во время разгона	70 ms
при коротком замыкании во время эксплуатации	70 ms
Пригодность для параллельной работы для повышения мощности	нет

Рисунок 8 – Выходные характеристики блока питания

Разрабатываемый лабораторный комплекс будет осуществлять имитацию процесса сортировки деталей на производстве. Каждая деталь, будет снабжаться штрих-кодом и цветовой маркировкой для идентификации в системе прослеживаемости и в самом комплексе соответственно. В зависимости от значения, считанного с цветовой маркировки, при помощи оптического датчика цвета, установленного на конвейере, манипулятор будет перемещать деталь с ленты конвейера в соответствующую тару.

При разработке и проектировании транспортного роботизированного комплекса стоит отдавать предпочтения модульности составных частей, что обеспечит простоту сборки, обслуживания и снизит экономические затраты.

В качестве готового аналога можно рассмотреть, к примеру, учебный роботизированный комплекс от компании FISCHERTECHNIK, однако, разрабатываемый комплекс будет более приближен как по электрической, так и по электронной компонентной базе к используемым на реальных предприятиях, что даст более весомую базу теоретических и практических навыков начинающим специалистам. На рисунке 9 представлен учебный стенд от компании FISCHERTECHNIK.

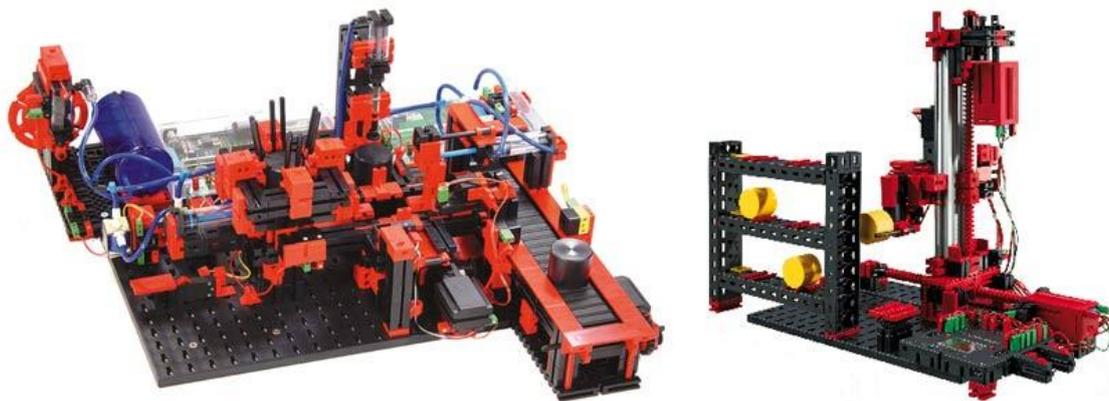


Рисунок 9 – Учебный комплекс от компании FISCHERTECHNIK.

#### Выводы по разделу

Таким образом, после постановки цели ВКР и задач по её осуществлению, было проведено исследование темы, связанной с промышленными манипуляторами и системами их управления. Был осуществлен обзор и анализ существующих решений, подбор компонентной базы, отвечающей необходимым нам требованиям. Определилась ключевая концепция разрабатываемого устройства, опираясь на которую будет происходить разработка электронных и электрических соединений комплекса и его техническая и программная реализация.

## 2 Объекты управления

### 2.1 Конвейер

Конвейер, включенный в состав транспортного роботизированного комплекса, реализован в виде ленточного транспортера и состоит из блока индуктивных датчиков, детектирующих наличие детали на ленте конвейера, а также оптического датчика цвета от компании FISCHERTECHNIK и кронштейна для его крепления. Лента приводится в движение при помощи мотор-редуктора от стеклоподъемника автомобиля. На рисунке 10 представлен используемый ленточный транспортер.

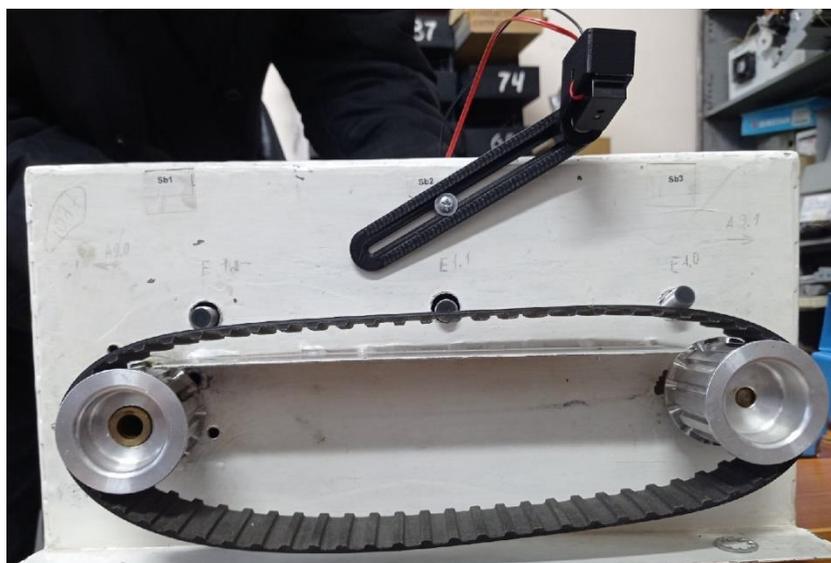
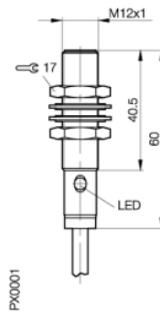


Рисунок 10 – Ленточный транспортер

В состав блока индуктивных датчиков ленточного транспортера входят 3 индуктивных датчика BALLUF BES 516-370-BO-C, закрепленных в специальных отверстиях, расположенных на лицевой части корпуса, немного выше конвейерной ленты. На рисунке 11 представлены основные технические характеристики используемых индуктивных датчиков приближения.



**Electrical data:**

Connection	cable
Rated operational voltage (Ue)	24 DC V
Load current capacity (Ie)	200 mA
Time delay before availability	10 ms
Electrical type	DC
Utilization category	DC 13
Hysteresis max (H)	15 %
No-load supply current damped	= 1 mA
Off-state current max (Ir)	10 µA
Ripple max. of Ue	= 15 %
Switching output	PNP
Switching element function	NC
Operating frequency (f)	3000 Hz
Voltage drop max. static	1.5 V
Supply voltage max. (Ub)	30 V
Supply voltage min. (Ub)	10 V

**Mechanical data:**

Number of wires	3-wire
Rated operating distance (sn)	2 mm
Diameter	M12x1 mm
Assured operating distance	0...1.6 mm
Cable code designation	L1YY-O
Mounting	flush mountable
Ambient temperature max	+85 °C
Ambient temperature min	-40 °C
Degree of pollution	3
Sensing face material	PA 12
Housing material	corrosion resistance Steel
Repeat accuracy max. (R)	= 5 %

**General data:**

Output indication	yes
Short circuit protected	yes
Degree of protection IP	IP68 per BWN Pr 20
Insulation class	2
Protected against polarity reversal	yes
Approval	CE,c_ul_us

**wiring**



Рисунок 11 – Характеристики датчика BALLUF BES 516-370-BO-C

Данные датчики обеспечивают детектирование металлических деталей на ленте конвейера.

## 2.2 Манипулятор МП-11

Промышленные роботы-манипуляторы, перемещающиеся в цилиндрической системе координат, в простейшем исполнении, имеет одну вращательную и две поступательные степени подвижности с перпендикулярно расположенными векторами направлений движений. На рисунке 12 представлен робот-манипулятор, функционирующий в цилиндрической системе координат

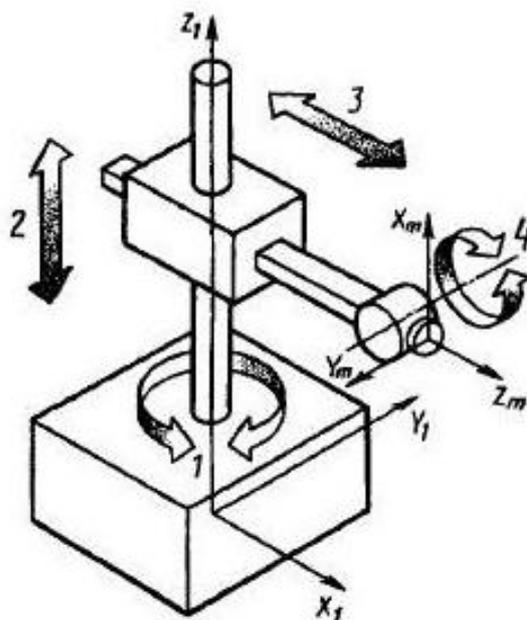


Рисунок 12 – Работа робота в цилиндрической системе координат

Фигура, определяющая область рабочей зоны и образующаяся при перемещении рабочего органа манипулятора, представляет собой неполный цилиндр. Манипуляторы, осуществляющие работу в данной системе координат широко распространены, это обусловлено простотой их конструктивного исполнения, удобством и относительной простотой разработки системы управления. Манипуляторы, функционирующие в цилиндрической системе координат, способны обслуживать большой объем рабочей зоны, а наличие двух поступательных функций перемещения, наряду с вращательной, облегчает планировку рабочих мест и компоновку оборудования при создании роботизированных комплексов [6].

Промышленный пневматический манипулятор МП-11, изображенный на рисунке 13, оснащен двумя универсальными механическими схватами клещевого типа, которые используются в операциях, где необходима транспортировка и ориентирование деталей, при автоматизации таких технологических процессов, как сборка, штамповка деталей и др.



Рисунок 13 – Манипулятор МП-11.

Манипулятор МП-11 функционирует в цилиндрической системе координат и имеет шесть степеней свободы, четыре из которых обеспечивают региональных движений (обеспечивают перемещение), две оставшиеся степени свободы являются локальными и обеспечивают ориентацию в пространстве.

Основной задачей разрабатываемого роботизированного комплекса является сортировка и перемещение объекта манипулирования в определенную позицию рабочей зоны, для чего достаточно будет задействовать только четыре региональных степени свободы, исходя из этого, оставшиеся три степени свободы для реализации проекта в рамках выпускной квалификационной работы являются избыточными. Их использование позволяет расширить функциональные возможности робота.

Таким образом, при работе комплекса мы будем использовать следующие региональные степени свободы:

- подъем рук манипулятора;
- поворот рук манипулятора;
- поступательное движение верхней и нижней руки.

Избыточными же, применительно к данному проекту, стоит считать локальные степени свободы:

- ротация схвата;
- сдвиг схвата.

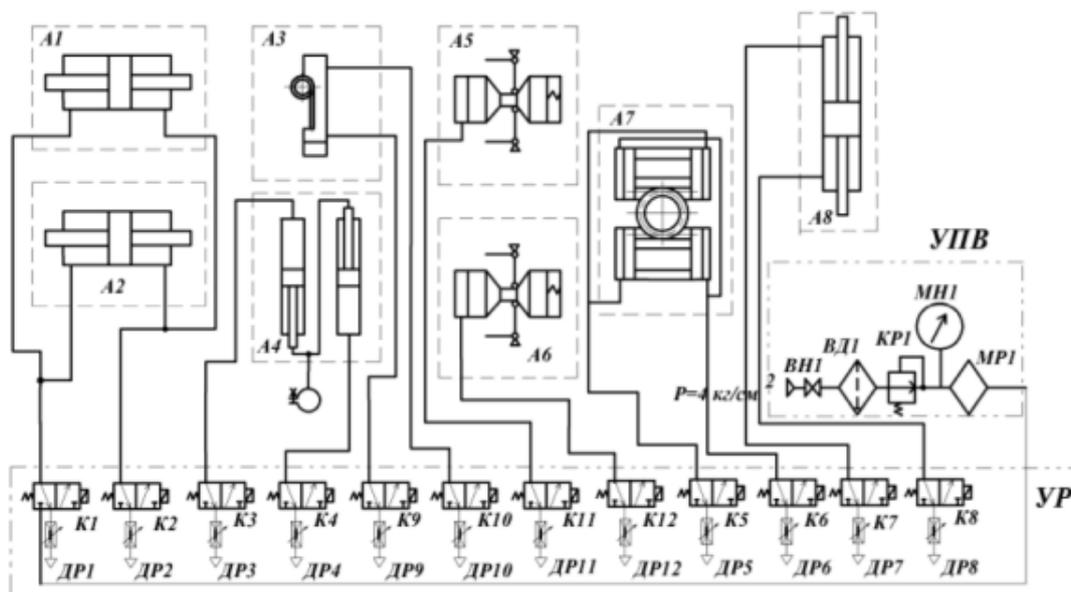
Основные характеристики промышленного пневматического робота МП-11 приведены в таблице 1 [11].

Таблица 1 – Основные характеристики манипулятора МП-11

Параметр	Значение
Номинальная грузоподъемность/полезная грузоподъемность, кг	2 / 1
Максимальная величина горизонтального перемещения, мм	200
Максимальный угол поворота в горизонтальной плоскости, °	120
Максимальная величина вертикального перемещения (подъема рук), мм	65
Максимальная величина сдвига схвата, мм	25
Максимальный угол поворота схвата вокруг продольной оси (ротация), °	180
Максимальный угол между руками, °	100
Минимальный угол между руками, °	20
Габаритные размеры, мм	990x990x445
Напряжение управления электропневмораспределителями, В	24
Род тока	Постоянный
Сила тока, не более, А	0,5
Масса, кг	70
Рабочее давление пневмопитания, МПа (кг/см <sup>2</sup> )	0,4 – 0,5 (4 – 5)

Состав манипулятора МП-11 и функционирование его основных частей подробно рассмотрены в работе «Изучение конструкций манипуляторов промышленных роботов РБ-241 и МП-11.01» А.Г. Кравцова [11].

На рисунке 14 представлена схема пневматическая принципиальная робота-манипулятора МП-11 [11].



А1, А2 – модули поступательные (руки); А3 – модуль вращательный (ротация схвата); А4 – модуль поступательный (сдвиг схвата); А5, А6 – схват; А7 – механизм поворота; А8 – механизм подъема; УР – узел распределения; ДР1, ДР2,...ДР12 – дроссели; К1, К2,...К12 – пневмораспределители; УПВ – узел подготовки воздуха; ВН1 – вентиляционный запорный, ВД1 – фильтр – влагоотделитель, КР1 – пневмоклапан, МН1 – манометр, МР1 – маслораспылитель.

Рисунок 14 – Схема пневматическая принципиальная промышленного манипулятора МП-11

Узел распределения и подачи воздуха манипулятора состоит из двенадцати электропневмоклапанов, обозначены на рисунке 16 символами «К1», «К2», ..., «К12», установленных на специальной распределительной плите. На них установлены дроссели для регулировки, обозначенные на рисунке 16 символами «ДР1», «ДР2», ..., «ДР12». Все элементы подключены согласно схеме пневматической принципиальной, приведенной на рисунке 16. Электропневмоклапаны обеспечивают функционирование какой-либо степени подвижности при подключении шлангов, соединяющих полость соответствующего пневмоцилиндра с нагнетающей магистралью компрессора через каналы подачи, выполненные в распределительной плите. Дроссели позволяют регулировать скорость срабатывания по каждой из степеней подвижности. Рабочее давление промышленного робота манипулятора МП-11 составляет порядка 2,5 атмосфер [11].

К основным характерным особенностям приводов пневматического типа относится конструктивная простота и сравнительно низкая себестоимость, быстрота выполнения движений, простота обслуживания и высокая надежность в процессе эксплуатации. Пневматические приводы в основном используются в манипуляторах, потребляемые мощности которых лежат в пределах 60 - 800 Вт.

К основным недостаткам приводов данного типа относятся:

- сложность реализации привода следящего типа;
- отсутствие точного позиционирования манипулятора;
- относительно большие размеры исполнительных механизмов;
- необходимость использования демпферов;
- низкий КПД, порядка 15 - 20%;

Выводы по разделу

В данном разделе были рассмотрены составные части объектов управления, входящих в состав транспортного роботизированного комплекса и их основные особенности, проработаны электрические соединения транспортного роботизированного комплекса. В соответствии с этим, для объектов управления были разработаны схемы электрические соединений и подключения и обеспечены все необходимые электрические соединения, а также, была определена первичная концепция электронных связей, которые будут реализованы в системе управления.

### **3 Интерфейсный модуль**

Как было определено ранее, в качестве основы системы управления транспортного роботизированного комплекса выступает программируемый логический контроллер Siemens S7-1200. Соединение осуществляется по сети Ethernet. В качестве объектов управления выступают конвейер и промышленный манипулятор МП-11.

Создание автоматизированных комплексов и сложных систем управления большим количеством объектов на базе программируемых логических контроллеров неизбежно влечет за собой необходимость применения разнообразных коммутационных, интерфейсных модулей и расширителей. Применимо к данной задаче, использование готовых решений в виде модулей расширения дискретных входов и выходов является экономически невыгодным. Помимо того, уход с российского рынка крупных производителей электроники и средств автоматизации делает крайне актуальным вопрос поиска аналогов и замен уже существующим решениям, для поддержания функционирования ранее введенных в эксплуатацию производственных комплексов и запуска новых. Разрабатываемый объект должен обладать всеми функциональными возможностями модулей расширения дискретных входов и выходов и при этом, себестоимость разрабатываемого модуля должна быть ниже аналогов, а также иметь возможность подключения к сети Ethernet для связи с устройством управления транспортным роботизированным комплексом.

В качестве замены модулей расширения дискретных входов и выходов, активно используемых в проектах автоматизации, в том числе и транспортных роботизированных комплексах, был разработан и спроектирован интерфейсный модуль. Стандартный вид, конвертированный из 3D модели сборки интерфейсного модуля, представлен на рисунке 15.

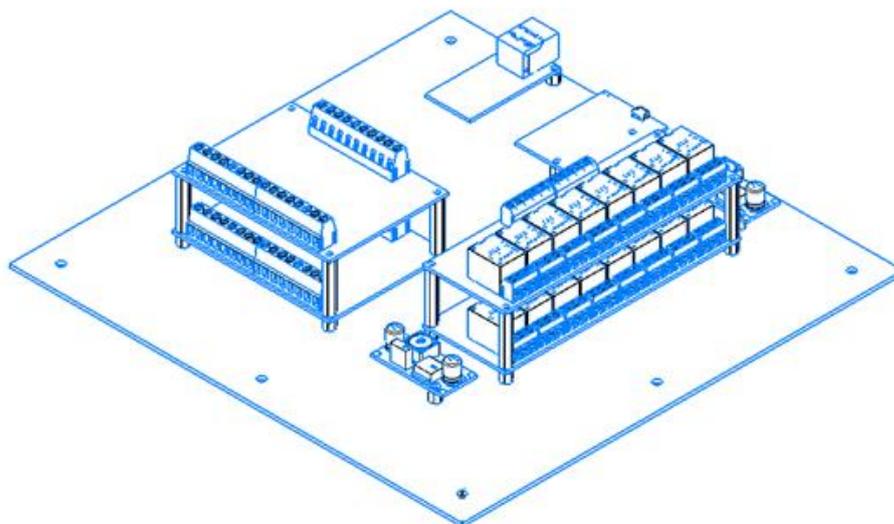


Рисунок 15 – Интерфейсный модуль

На рисунке 16 представлен чертеж интерфейсного модуля в сборе.

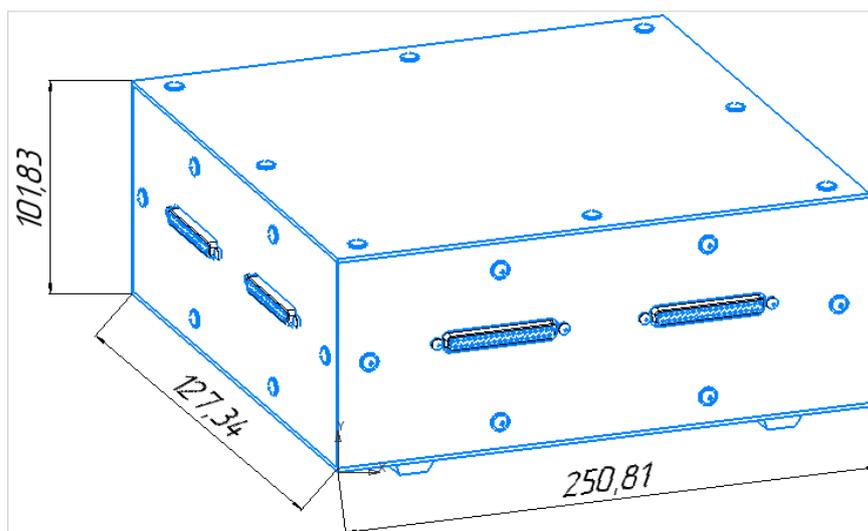


Рисунок 16 – Интерфейсный модуль в сборе

Данный модуль, в некоторых случаях, может являться хорошей заменой стандартным решениям, имеющимся на рынке.

### **3.1 Аппаратное обеспечение интерфейсного модуля**

Превалирующие критерии при построении системы управления (СУ) для устройства могут быть сведены к следующим:

- достаточность технических характеристик, для реализации управления устройством;
- низкая цена компонентов;
- доступность и гибкость средств программирования СУ.

Исходя из обозначенных критериев, нельзя обойти вниманием аппаратную платформу Arduino, сыскавшую мировую известность, как самый распространённый инструмент для проектирования электронных устройств, взаимодействующих с физической средой, используемый в огромном числе реальных технических приложений как любителями электронной техники для реализации различных устройств автоматизации, так и инженерами для быстрого прототипирования разработок [1,2,5].

Arduino предоставляет пользователю возможность приема сигналов от различных цифровых и аналоговых датчиков и внешних устройств, а также управления различными исполнительными механизмами и дополнительными устройствами.

На данный момент на рынке микроконтроллеров и готовых платформ существует множество представителей со схожим функционалом, что и у Arduino: Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard и т.п.

Однако, следует отметить ряд принципиальных преимуществ перед другими устройствами подобного функционала:

- низкая стоимость. Стоимость в рублях для самых популярных моделей плат составляет порядка 4000 Р ;
- кроссплатформенность.
- большое разнообразие плат расширения и всевозможных датчиков;
- простая среда для разработки программного кода, основанная на среде программирования Processing, с возможностью дополнения библиотеками C++, а также перехода на язык AVR-C.

В настоящее время рынок аппаратных платформ Arduino представлен версиями плат, отличающихся как по размеру, так и по техническим характеристикам микроконтроллеров, являющихся основой платформ [7-10].

Главными критериями выбора аппаратной платформы для реализации на ней системы управления разработанной модели устройства является достаточная производительность и объем памяти микроконтроллера, а также наличие необходимого количества входов/выходов для реализации функции управления реальным устройством.

В ходе анализа существующих на данный момент на рынке вариантов аппаратных платформ Arduino, учитывая критерий стоимости, для указанных целей была выбрана конфигурация, соответствующая Arduino Mega [23]. Описание выводов данной платформы представлено на рисунке 17.

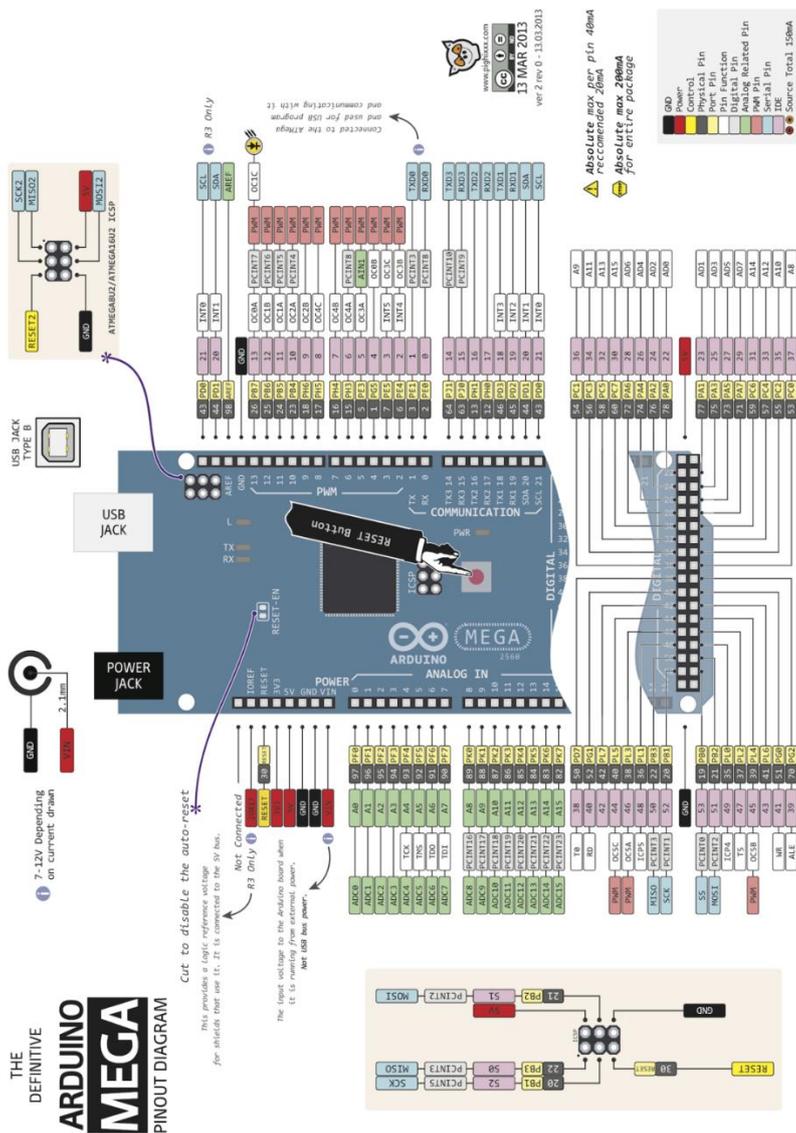


Рисунок 17 – Описание выводов аппаратной платформы Arduino Mega

На сегодняшний день рынок, в частности электроники наполнен разнообразными имитациями оригинальных продуктов. Некоторая часть этих имитаций даже имеет преимущество перед оригиналами при рассмотрении в плоскости цена – качество – функционал. Примером в отношении аппаратных платформ Arduino может служить имитация компании RobotDyn. На рисунке 18 представлена плата Arduino MEGA 2560 PRO от компании RobotDyn.

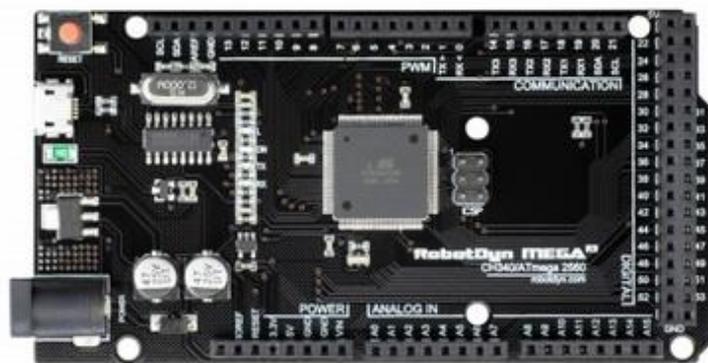


Рисунок 18 – Внешний вид Arduino MEGA 2560 PRO

Некоторая часть этих имитаций даже имеет преимущество перед оригиналами при рассмотрении в плоскости цена – качество – функционал.

### 3.2 Компонентная база интерфейсного модуля

Для сборки интерфейсного модуля были подобраны все необходимые элементы, позволяющие обеспечить полный функционал модуля расширения дискретных входов и выходов для программируемых логических контроллеров. Состав разработанного интерфейсного модуля представлен на рисунке 19.

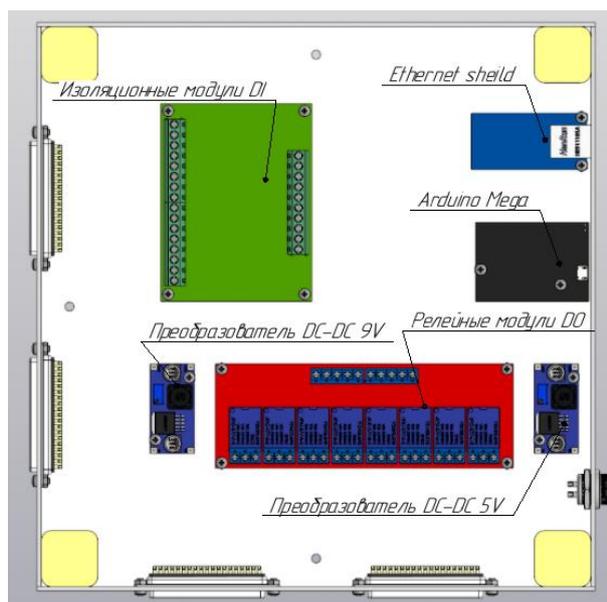


Рисунок 19 – Состав интерфейсного модуля

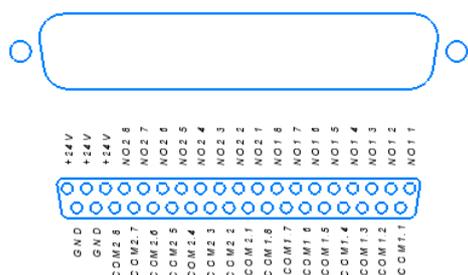
Разработанный интерфейсный модуль поддерживает 16 каналов дискретных входов и 16 каналов дискретных выходов и является кроссплатформенным, все это при более низкой себестоимости производства по сравнению с существующими на рынке решениями, что наглядно представлено в таблице 2.

Таблица 2 - Сравнение интерфейсного модуля с готовыми решениями

Наименование	DI	DO	Цена
Интерфейсный модуль	16 каналов	16 каналов	6000 руб.
ОВЕН ПРМ-220.1	8 каналов	8 каналов	10000 руб.
Siemens SM1223	16 каналов	16 каналов	62405 руб.
МОХА E1212	8 каналов	8 каналов	20825 руб.

Для подключения объектов управления к интерфейсному модулю предусмотрены разъемы DB-37F, также присутствуют разъем для подключения питания, устройства управления транспортным роботизированным комплексом и программирования интерфейсного модуля. На рисунках 20 – 22 представлено расположение разъемов на боковых панелях интерфейсного модуля и отображено подключение их выводов.

### Дискретные выходы DO



### Дискретные выходы NC

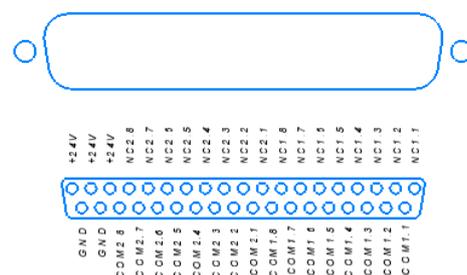


Рисунок 20 – Разъемы для подключения манипулятора



### 3.2.1 Изоляционный модуль DI

Следует понимать, что аппаратная платформа Arduino, отвечающая за контроль и управление интерфейсным модулем, может принимать и обрабатывать лишь сигналы в диапазоне от 0 до 5 В, а также не воспринимает входные отрицательные (относительно общей точки) сигналы, приходящие от внешних устройств. Однако, в случае вышеприведенного устройства, возникает необходимость обработки реального напряжения от составных частей комплекса, параметры которого выходят за рамки ограничений аппаратной платформы и составляют 24 В. Таким образом, для обеспечения корректного функционирования системы необходимо обеспечить преобразование сигналов до момента обработки их микроконтроллером с помощью стороннего устройства преобразования сигнала. Для решения данной задачи, в состав интерфейсного модуля были включены два изоляционных 8-и канальных модуля с оптопарами. На рисунке 24 изображен модуль гальванической развязки DST-1R8P.

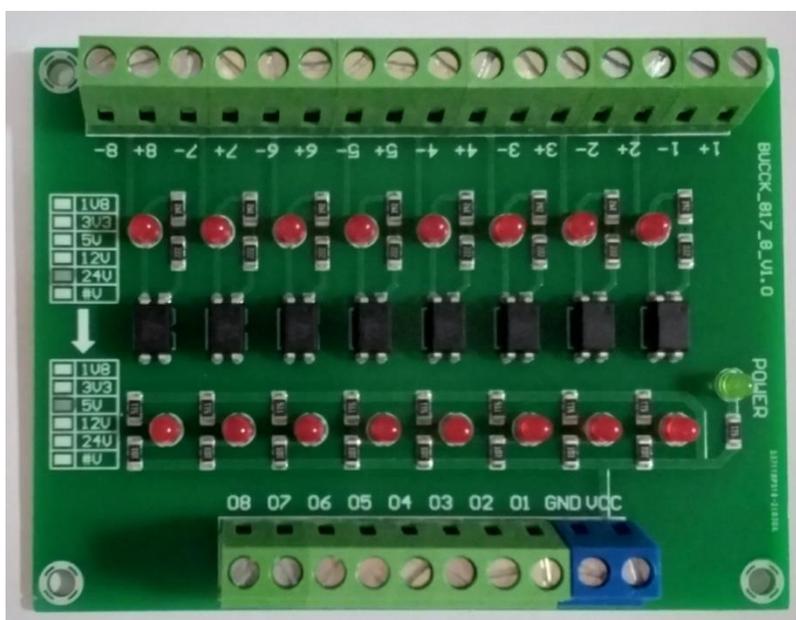


Рисунок 24 – Модуль DST-1R8P

К контактам +1...+8 и -1...-8 подключаются датчики, источники сигналов и их питание, к контактам O1...O8 подключаются цифровые входа Arduino, а к контактам GND и VCC – питание модуля. На рисунке 25 представлена схема электрическая принципиальная оптического изолятора, реализованного в модуле DST-1R8P.

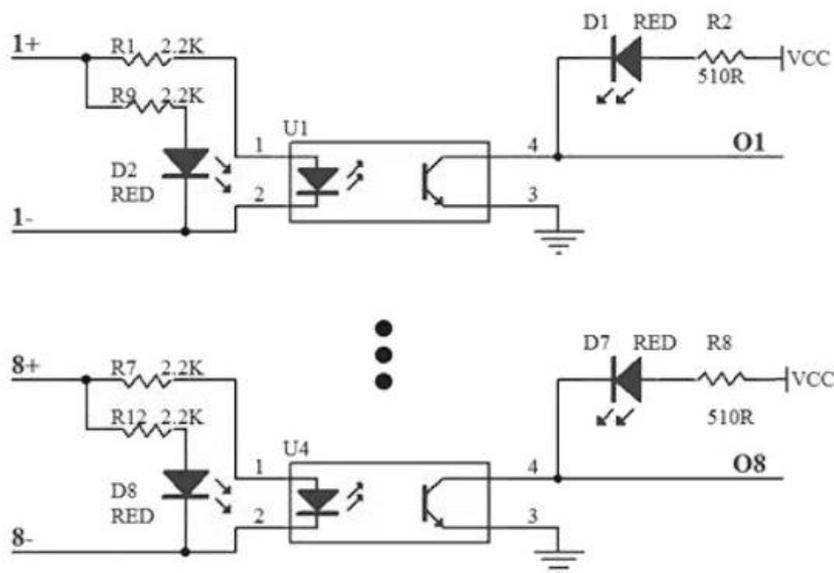


Рисунок 25 – Схема оптического изолятора

В таблице 3 приведены основные характеристики данного изоляционного модуля.

Таблица 3 – Основные характеристики модуля DST-1R8P

Вид гальванической развязки	Фотоэлектрический
Напряжение питания	5 В
Частота передачи сигнала	До 20 КГц
Допустимое входное напряжение	От 5 до 24 В
Допустимое выходное напряжение	От 3 до 7 В
Разъемы для подключения	8 шт.
Габаритные размеры	95 x 72 x 18 мм

Данные модули, в совокупности с платой Arduino Mega, образуют расширитель дискретных входов.

### 3.2.2 Релейный модуль DO

Как и в случае с изоляционным модулем дискретных входов, возникает проблема различия рабочих напряжений между аппаратной платформой, контролирующей интерфейсный модуль, и внешними цепями. Для коммутации катушек электропневмоклапанов манипулятора ПМ-11 и двигателя конвейера необходимо подавать постоянное напряжение 24 В. Для решения данной задачи, в состав устройства были включены два изоляционных 8-и канальных релейных модуля, расположенных друг над другом и в сочетании с платой Arduino Mega выступающие в качестве расширителя дискретных выходов. На рисунке 26 представлен используемый релейный модуль.



Рисунок 26 – Релейный модуль

Данный модуль сконструирован на базе реле JQC-3FF-S-Z. Управление каждым из каналов релейного модуля происходит независимо от других. Реле модуля можно использовать как нормально замкнутые, нормально разомкнутые и переключающее. При реализации проекта по выпускной квалификационной работе, будут использоваться нормально разомкнутые контакты (NO), то есть, вся подключенная нагрузка будет коммутироваться только при поступлении соответствующего управляющего сигнала. Имеется возможность настройки уровня сигнала, обеспечивающего срабатывание реле (HIGH-LOW level trigger).

На рисунке 27 представлена схема электрическая принципиальная нескольких каналов релейного модуля.

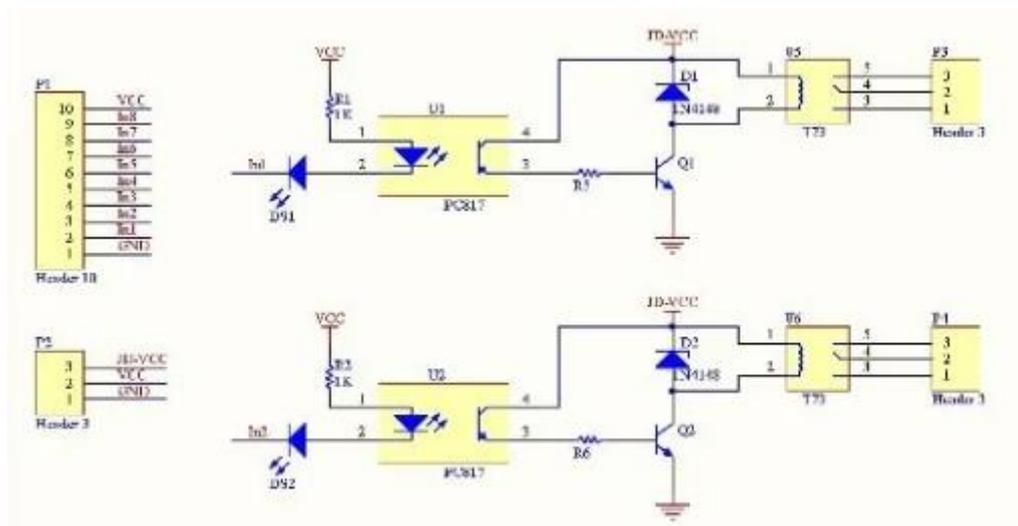


Рисунок 27 - Схема электрическая принципиальная релейного модуля

В таблице 4 приведены основные характеристики используемого релейного модуля.

Таблица 4 – Основные характеристики релейного модуля

Напряжение питания	5 В
Тип реле	Электромеханическое
Номинальный ток нагрузки	10 А
Коммутируемая нагрузка	250 В AC/ 30 В DC
Габаритные размеры	142 x 50 мм

Данные модули, в совокупности с платой Arduino Mega, образуют расширитель дискретных выходов.

### 3.2.3 Питание интерфейсного модуля

На базе интерфейсного модуля установлены 2 преобразователя напряжения DC-DC типа, на 5В для питания платы Arduino и составных частей модуля, а также на 9В для питания оптического датчика цвета, установленного на конвейере. На рисунке 28 представлен регулируемый преобразователь напряжения на базе микросхемы LM2596.



Рисунок 28 – Преобразователь напряжения

В таблице 5 представлены основные характеристики приведенного выше преобразователя.

Таблица 5 – Основные характеристики преобразователя напряжения

Тип микросхемы	LM2596
Входное напряжение	От 4 до 35 В
Выходное напряжение	От 1,23 до 30 В
Максимальный выходной ток	3 А
КПД преобразования	92%
Частота переключения	150 кГц
Пульсация на выходе	Менее 30 мВ
Диапазон рабочей температуры	От -45 до + 85 °С
Габаритные размеры	43 x 21 x 12 мм

На рисунке 29 представлена схема электрическая принципиальная используемого понижающего DC-DC преобразователя.

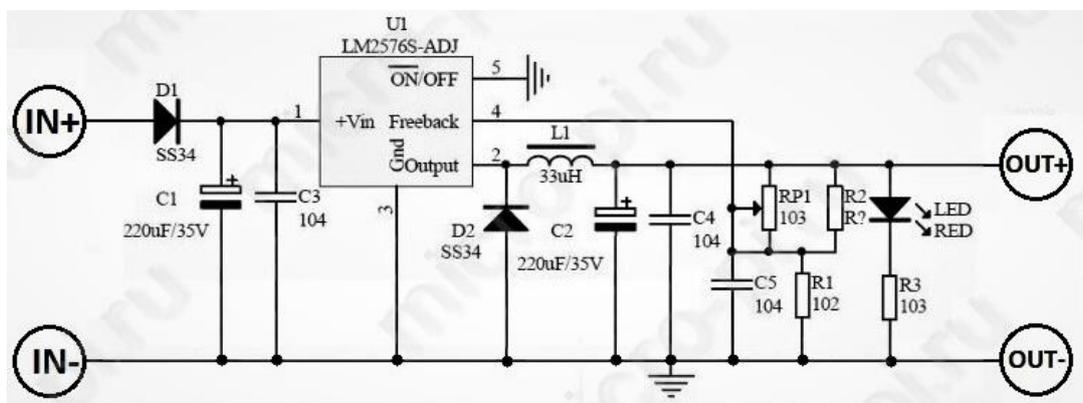


Рисунок 29 – Схема электрическая принципиальная преобразователя

Преобразователи напряжения обеспечивают необходимым питанием оптический датчик цвета и устройство управления интерфейсного модуля

### 3.2.4 Ethernet shield

Программируемый логический контроллер Siemens S7-1200, являющийся устройством управления всего транспортного роботизированного комплекса, должен связываться с устройством управления интерфейсным модулем по сети Ethernet. Однако, Arduino и аналоги этой платформы не имеют встроенных PROFINET разъемов для подключения кабелей данного типа. Именно с этой целью в состав интерфейсного модуля было добавлено такое устройство связи как Ethernet shield, обеспечивающий целостность и корректность работы системы управления. Он преобразует сигналы, передаваемые ПЛК в понятную для AVR микроконтроллеров TTL логику. На рисунке 30 представлен сетевой модуль Ethernet shield.



Рисунок 30 - Ethernet shield

В таблице 6 приведены основные характеристики используемого сетевого модуля.

Таблица 6 – Основные характеристики сетевого модуля

Тип микросхемы	W5500
Напряжение питания	3,3 В / 5 В
Протоколы TCP/IP	TCP, UDP, ICMP, IP4, ARP, IGMP, PPPoE
Интерфейс	SPI, режим SPI 0, 3
Внутренняя память для буферов RX/TX	32 Кбайт
Габаритные размеры	55x28 мм

Данный сетевой модуль подключается по SPI интерфейсу. В таблице 7 приведен пример подключения выводов Ethernet shield к выводам платы Arduino MEGA 2560 PRO, используемый в рамках данного проекта.

Таблица 7 – Подключение Ethernet shield к Arduino

Выводы Ethernet shield	Выводы платы Arduino MEGA 2560 PRO
+ 5 V	+ 5 V
GND	GND
SCS	53
SCLK	52
MOSI	51
MISO	50

#### Выводы по главе

В данной главе был рассмотрен разработанный интерфейсный модуль, выступающий в качестве замены готовых решений в области модулей расширения дискретных входов и выходов, его аппаратное обеспечение, компонентная база, а на основе нее была разработана схема электрическая соединений и подключения интерфейсного модуля.

## 4 Промышленный протокол связи Modbus TCP/IP

Как было рассмотрено выше, физическая связь между устройством управления транспортно-роботизированного комплекса и интерфейсным модулем происходит по сети Ethernet, программно же, связь реализуется при помощи промышленного протокола связи Modbus TCP/IP. Архитектура данного протокола основана на принципе ведущий-ведомый, в данном случае client-server соответственно. Структура передаваемых данных кодируются в двоичный формат, на основе которого формируется TCP-пакет, передающийся по сети. Данный протокол не используется имеет встроенный механизм контроля целостности передачи данных. Запрос на инициализацию работы и обмена данными может передавать только client устройство, server устройство может лишь отвечать на формируемые запросы [17]. На рисунке 31 представлена основная составляющая структуры пакета Modbus.

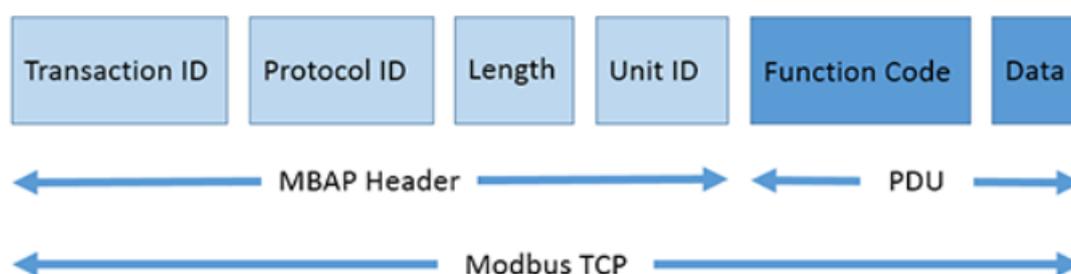


Рисунок 31 – Структура пакета Modbus TCP/IP

Идентификатор транзакции (Transaction Identifier): 2 служебных байта, к устанавливаемые client устройством, они необходимы для корректной идентификации отправленных запросов. Данный элемент пакета дублируются server устройством в ответном сообщении. Использование идентификатора запроса необходимо, поскольку, при наличии множества server устройств, порядок ответов от них могут не совпадать с порядком запросов [28].

Идентификатор протокола (Protocol Identifier): два служебных байта, устанавливаемые client устройством, данный элемент пакета в всегда будет равняться 00 00, что обозначает связь по протоколу Modbus [29].

Длина (Length): два служебных байта, устанавливаемые client устройством, и идентифицирующие количество байтов в пакете, следующим от Unit ID и до окончания информационного сообщения.

Идентификатор адреса устройства (Unit Identifier): один байт, устанавливаемый client устройством и повторяемый server устройством для окончательной идентификации объекта назначения передачи данных.

Пакет данных (Data): пользовательская информация, которую необходимо передать или считать.

#### **4.1 Настройка соединения client устройства**

Настройка соединения client устройства ПЛК Siemens S7-1200 и всех управляющих команд проводилось в среде разработки TIA Portal V15.1. Для написания программных кодов протокола связи и системы управления будет использоваться язык программирования Ladder (язык лестничной логики) [26]. К основным командам, отвечающим за режим работы промышленного протокола связи Modbus TCP/IP относятся:

- MB\_MODE,
- MB\_DATA\_ADDR,
- MB\_DATA\_LEN.

Функция MB\_MODE содержит информацию о режиме работы протокола Modbus, в MB\_DATA\_ADDR прописывается начальный адрес памяти для поиска информационного сообщения, а MB\_DATA\_LEN отображает длину информации, которую необходимо обработать client устройству [25]. В таблице 8 представлены комбинации, которые могут быть использованы при настройке работы протокола в среде TIA Portal.

Таблица 8 – Режимы работы протокола связи Modbus TCP/IP

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Номер функции	Назначение
0	1 – 9999	1 – 2000	01	Чтение выходных битов
0	10001 – 19999	1 – 2000	02	Чтение входных битов
0	40001 – 49999 400001 – 465535	1 – 125	03	Чтение регистров
0	30001 – 39999	1 – 125	04	Чтение сообщений типа word
1	1 – 9999	1	05	Запись бита
1	40001 – 49999 400001 – 465535	1	06	Запись регистра
1	1 – 9999	2 – 1968	15	Запись двух битов
1	40001 – 49999 400001 – 465535	2 – 123	16	Запись двух регистров
2	1 – 9999	1 – 1968	15	Запись бита
2	40001 – 49999 400001 – 465535	1 – 123	16	Запись регистра
11	Не указываются		11	Чтение статусов и сообщений
80	-	1	08	Проверка статуса server
81	-	1	08	Сброс счетчика событий
101	0 – 65535	1 – 2000	01	Чтение выходных битов
102	0 – 65535	1 – 125	02	Чтение входных битов
103	0 – 65535	1 – 125	03	Чтение регистров
104	0 – 65535	1	04	Чтение сообщений типа word
105		1	05	Запись бита
106	0 – 65535	1	06	Запись регистра
115	0 – 65535	1 – 1968	15	Чтение входных битов
116	0 – 65535	1 – 123	16	Запись регистров

В рамках реализации данного проекта, для осуществления чтения данных с интерфейсного модуля был использован режим работы под номером 103 (holding register read), позволяющий считывать информационные сообщения в количестве до 125 и размером 2 байта каждое, располагающиеся в области памяти с адресацией от 0 до 65535. Для реализации функции записи управляющих команд в интерфейсный модуль был использован режим работы под номером 106 (holding register write), позволяющий считывать одно информационное сообщение размером 2 байта, располагающееся в области памяти с адресацией от 0 до 65535.

На рисунках 32 и 33 представлены программные блоки, реализующие функции чтения и записи информации соответственно.

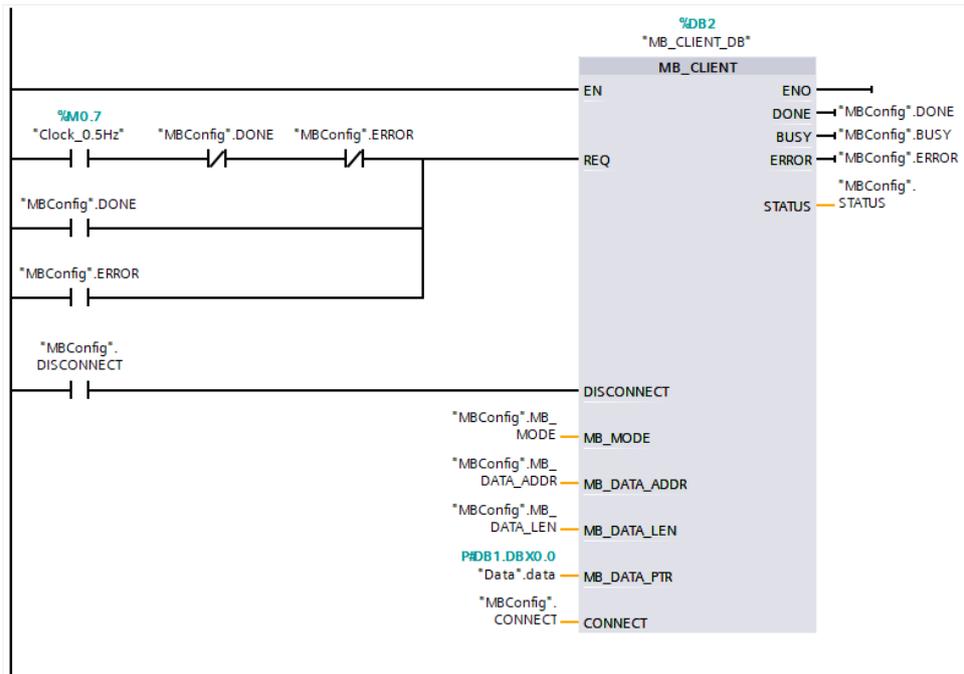


Рисунок 32 – Чтение данных через протокол Modbus TCP

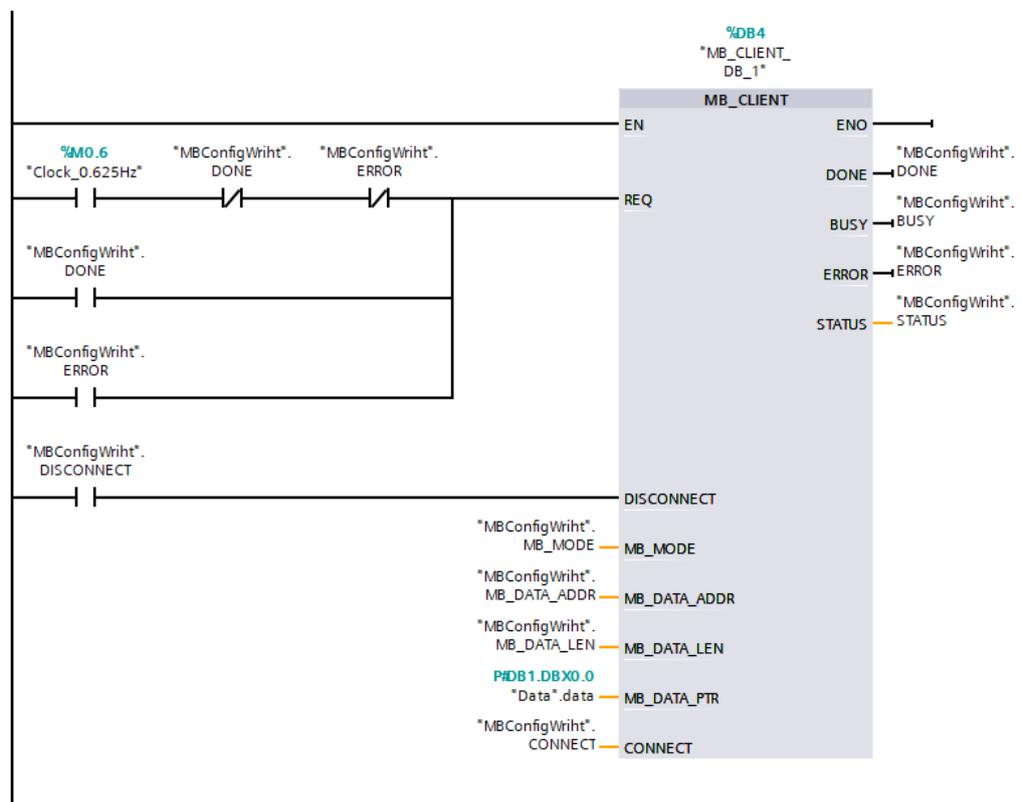


Рисунок 33 – Запись данных через протокол Modbus TCP

В среде разработки TIA Portal один регистр составляет 2 байта и является минимальным пакетом данных, которым могут обмениваться устройства. К контактам MB\_DATA\_PTR, изображенным на рисунках выше, подключается специализированный блок данных с отключенной оптимизацией, который и будет содержать информацию для обмена между устройствами. Первые 3 бита в нем являются служебными и отвечают за выбор режима работы транспортного роботизированного комплекса, а все остальные содержат информацию о срабатывании датчиков и включении электропневмоклапанов. На рисунке 34 представлен блок данных, используемый протоколом Modbus для хранения и обмена информацией.

1	Static			
2	data	Array[0..15] of Bool	0.0	
3	data[0]	Bool	0.0	false
4	data[1]	Bool	0.1	false
5	data[2]	Bool	0.2	false
6	data[3]	Bool	0.3	false
7	data[4]	Bool	0.4	false
8	data[5]	Bool	0.5	false
9	data[6]	Bool	0.6	false
10	data[7]	Bool	0.7	false
11	data[8]	Bool	1.0	false
12	data[9]	Bool	1.1	false
13	data[10]	Bool	1.2	false
14	data[11]	Bool	1.3	false
15	data[12]	Bool	1.4	false
16	data[13]	Bool	1.5	false
17	data[14]	Bool	1.6	false
18	data[15]	Bool	1.7	false

Рисунок 34 – Блок обмена данными

Помимо этого, для каждого из программных блоков необходимо создать по блоку данных, хранящих в себе настройки соединения и режимов работы. Необходимо указать ID client устройства, IP адрес server устройства для подключения, а также все необходимую информацию о режиме работы, типе протокола связи и портах подключения.

На рисунках 35 и 36 представлены настройки программных блоков, отвечающих за чтение и запись данных соответственно.

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	REQ	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	DISCONNECT	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	MB_MODE	USInt	103		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	ERROR	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	MB_DATA_ADDR	UDInt	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	MB_DATA_LEN	UInt	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	CONNECT	TCON_IP_v4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Interfaceld	HW_ANY	64		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		HWidentifier of IE-interface submodule
10	ID	CONN_OUC	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		connection reference / identifier
11	ConnectionType	Byte	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		type of connection: 11=TCP/IP, 19=UDP (17=TCP)
12	ActiveEstablished	Bool	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		active/passive connection establishment
13	RemoteAddress	IP_V4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	remote IP address (IPv4)
14	ADDR	Array[1..4] of Byte			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
15	ADDR[1]	Byte	192		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
16	ADDR[2]	Byte	168		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
17	ADDR[3]	Byte	31		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
18	ADDR[4]	Byte	240		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
19	RemotePort	UInt	502		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote UDP/TCP port number
20	LocalPort	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		local UDP/TCP port number
21	DONE	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	BUSY	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	STATUS	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Рисунок 35 – Настройка блока чтения информации

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	REQ	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	DISCONNECT	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	MB_MODE	USInt	106		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	ERROR	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	MB_DATA_ADDR	UDInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	MB_DATA_LEN	UInt	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	CONNECT	TCON_IP_v4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	Interfaceld	HW_ANY	64		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		HWidentifier of IE-interface submodule
10	ID	CONN_OUC	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		connection reference / identifier
11	ConnectionType	Byte	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		type of connection: 11=TCP/IP, 19=UDP (17=TCP)
12	ActiveEstablished	Bool	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		active/passive connection establishment
13	RemoteAddress	IP_V4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	remote IP address (IPv4)
14	ADDR	Array[1..4] of Byte			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
15	ADDR[1]	Byte	192		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
16	ADDR[2]	Byte	168		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
17	ADDR[3]	Byte	31		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
18	ADDR[4]	Byte	240		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
19	RemotePort	UInt	502		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote UDP/TCP port number
20	LocalPort	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		local UDP/TCP port number
21	DONE	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
22	BUSY	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
23	STATUS	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Рисунок 36 – Настройка блока записи информации

Все вышеперечисленное необходимо для корректного подключения и обмена данными между client и server устройствами.

## 4.2 Настройка соединения server устройства

Настройка соединения server устройства Arduino MEGA 2560 PRO и всех управляющих команд проводилось в среде разработки Arduino IDE [24]. Список режимов работы Modbus для Arduino и принципы их функционирования схожи с представленными в пункте 5.1. Ориентируясь на выбор, сделанный для client устройства, при реализации возможности чтения и записи данных server устройством, будут использоваться команды `holdingRegisterRead()` и `holdingRegisterWrite()` соответственно. Ниже представлен программный код, реализующий соединение с ПЛК по протоколу связи Modbus TCP/IP:

```
// Подключение необходимых библиотек
#include <SPI.h>
#include <Ethernet.h>
#include <ArduinoRS485.h>
#include <ArduinoModbus.h>

byte mac [] = {
    0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED // MAC адрес устройства
};

IPAddress ip (192, 168, 31, 240); // IP адрес устройства
EthernetServer ethServer (502); // TCP порт
ModbusTCPServer modbusTCPServer;

void setup() {
    // физическое подключение пина SCS Ethernet sheild к плате Arduino
    Ethernet.init(53);
    // Инициализация последовательного порта со скоростью 9600
    Serial.begin(9600);
```

```

// Ожидания подключения последовательного порта
while (!Serial)
{
;
}
Serial.println ("Ethernet Modbus TCP Example");
// Запуск соединения по сети Ethernet
Ethernet.begin(mac, ip);
// Проверка подключения Ethernet shield
if (Ethernet.hardwareStatus() == EthernetNoHardware) {
Serial.println ("Ethernet shield was not found.");
while (true) {
delay(1);
}
}
// Проверка подключение Ethernet кабеля
if (Ethernet.linkStatus() == LinkOFF) {
Serial.println("Ethernet cable is not connected.");
}
// Запуск сервера
ethServer.begin();
if (!modbusTCPServer.begin()) {
Serial.println("Failed to start Modbus TCP Server!");
while (1);
}
modbusTCPServer.configureHoldingRegisters(0x00, 2); // Конфигурация
регистров
}

```

```

void loop() {
    // Проверка подключаемых клиентов
    EthernetClient client = ethServer.available();
    // Соединение с client устройством установлено
    if (client) {
        Serial.println("new client");
        modbusTCPServer.accept(client);
        // Запуск рабочего цикла
        while (client.connected())
        {
            modbusTCPServer.poll();
        }
        Serial.println("client disconnected"); // Соединение потеряно
    }
}

```

При подключении server устройство проверяет наличие кабеля и платы связи, а также корректность подключения. При удачном подключении client устройства запускается цикл, в котором выполняется пользовательская программа, до тех пор, пока client устройство не будет отключено. На рисунках 37-39 представлены блок-схемы работы программного кода server устройства при установке соединения.

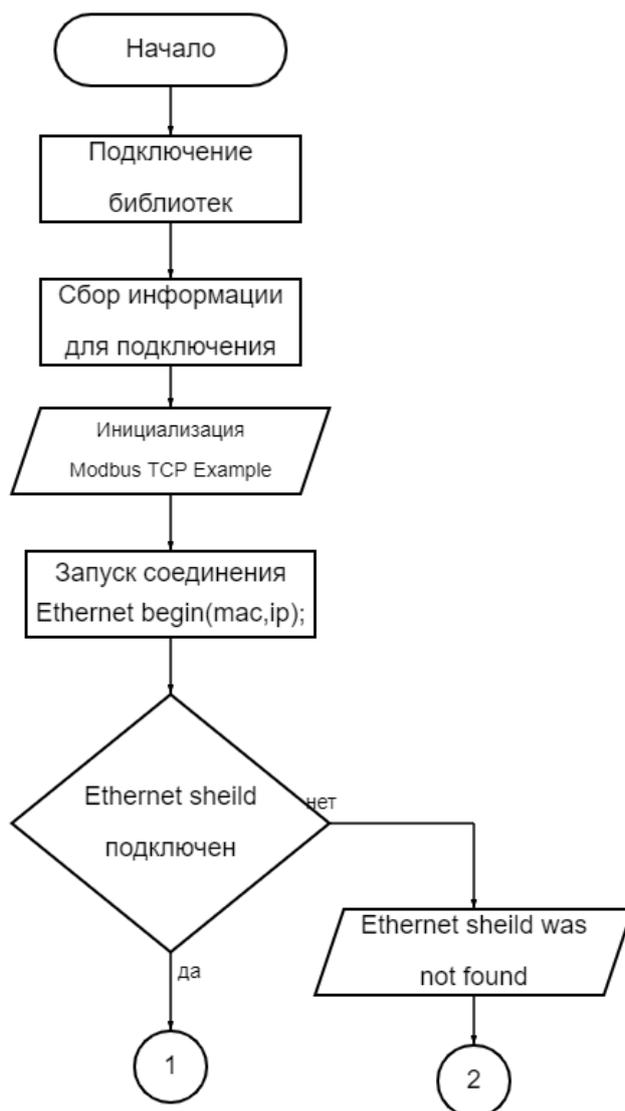


Рисунок 37 – Работа Server устройства по протоколу Modbus

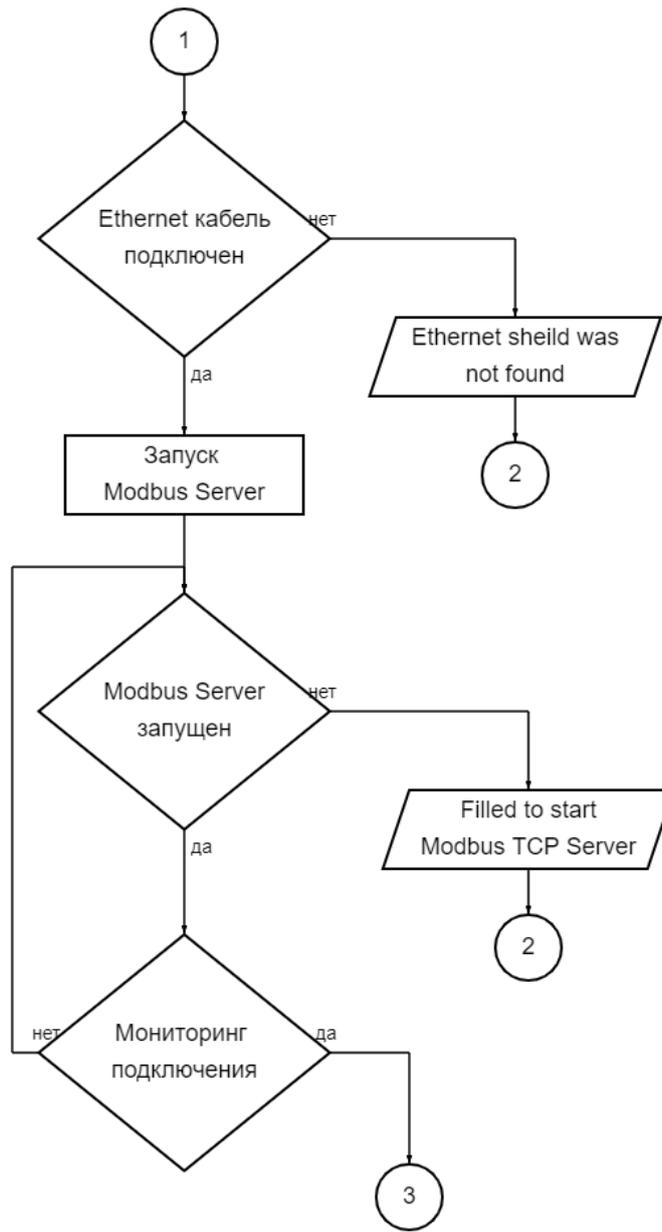


Рисунок 38 – Работа Server устройства по протоколу Modbus  
(продолжение)

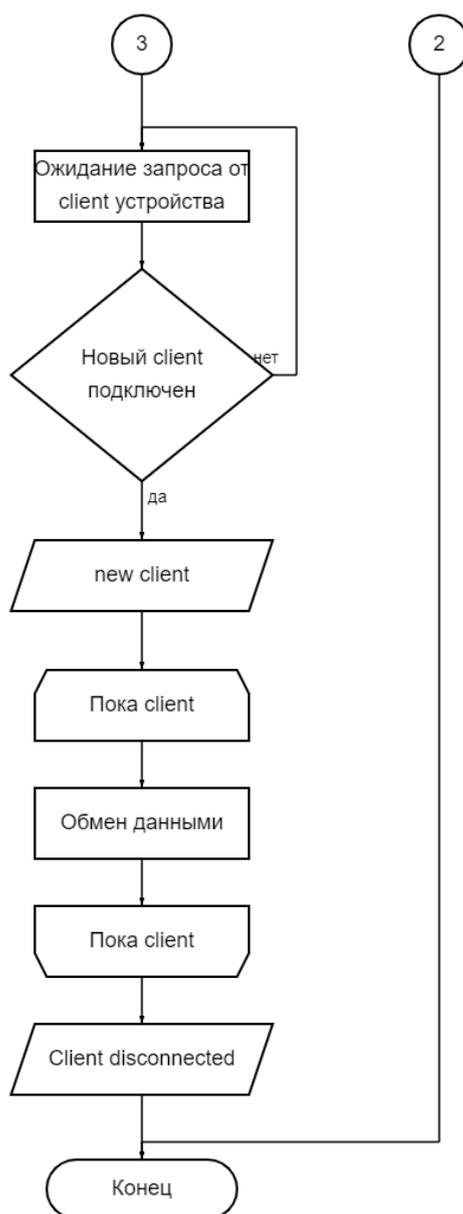


Рисунок 39 – Работа Server устройства по протоколу Modbus (продолжение)

### Выводы по главе

В данной главе были рассмотрены принципы работы промышленного протокола связи Modbus TCP/IP, а также способы его программной реализации. Были выбраны и настроены необходимые режимы работы, удовлетворяющие необходимым нам требованиям. Представлены примеры реализации программного кода для устройств client и server, обеспечивающего обмен данными и написанный в средах разработки TIA Portal и Arduino IDE соответственно.

## 5 Система управления

В рамках выполнения выпускной квалификационной работы было реализовано 4 режима работы транспортного роботизированного комплекса и разработаны алгоритмы управления промышленным манипулятором. В этот перечень входят 2 алгоритма, обеспечивающих выполнения транспортных функций и сортировки манипулируемых объектов, демонстрационный режим работы и режим ручного управления. Схваты манипулятора МП-11 не оснащены датчиками, поэтому выполнение и корректность действий будет отслеживаться по сигналу индуктивного датчика и временным задержкам, реализованным в среде разработки TIA Portal. В каждом из представленных режимов работы предусматривается симуляция системы прослеживаемости на производстве. Ручной режим управления движениями манипулятора осуществляется при помощи панели оператора Weintek eMT3070B или кнопок, интегрированных в интерфейсный модуль. Программирование панели оператора осуществляется в среде разработки EasyBuilder Pro. Соединение панели с ПЛК Siemens осуществляется по сети Ethernet. На рисунке 40 представлена используемая панель оператора.



Рисунок 40 – Панель оператора Weintek eMT3070B

## 5.1 Программируемый логический контроллер Siemens S7-1200

Программируемые логические контроллеры (ПЛК) являются микропроцессорными устройствами и предназначены для сбора, обработки, преобразования, хранения информации, выполнения пользовательской программы и формирования управляющих команд. ПЛК имеет конечное количество входов и выходов, которые подключаются к датчикам, различного вида ключам, исполнительным механизмам объекта управления, и предназначается для работы в режимах реального времени. Применительно к промышленному манипулятору МП-11 объектом управления, непосредственно взаимодействующим с системой управления на основе ПЛК, является блок электропневмоклапанов (электропневмораспределителей), а по отношению к конвейеру – электродвигатель.

На рисунке 41 представлен типичный цикл работы программируемого логического контроллера.

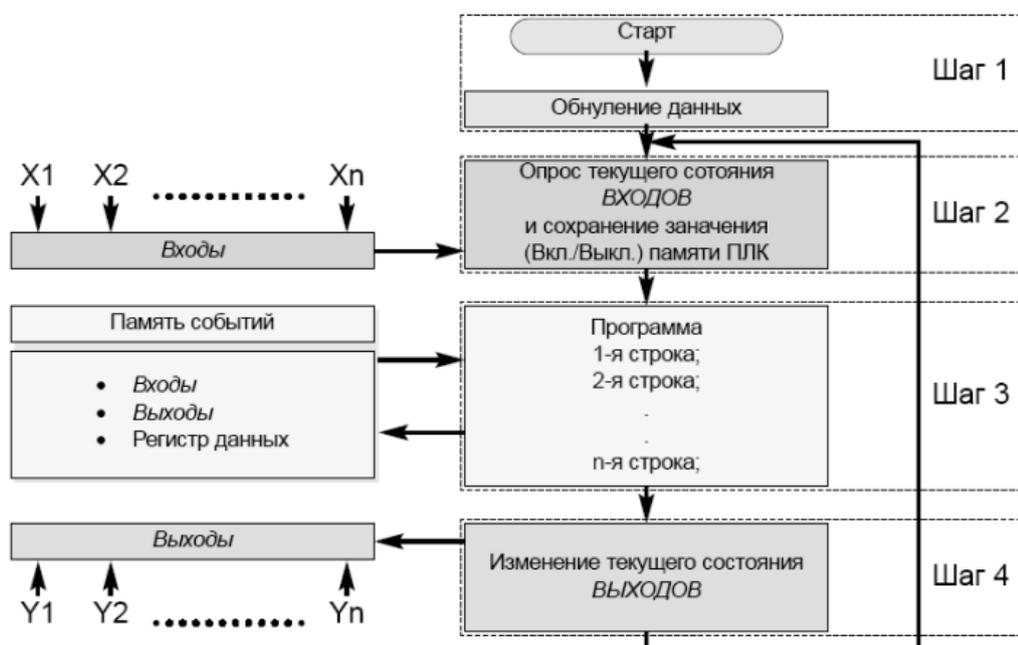


Рисунок 41 – Цикл работы ПЛК

К входам ПЛК подключаются, например, кнопки световые барьеры безопасности или разнообразные датчики, благодаря которым система ПЛК может отслеживать текущее состояние управляемых объектов и систем, выходы подключаются к объектам управления. В процессе выполнения рабочего цикла, ПЛК непрерывно опрашивает текущее состояние входов  $X_1, X_2 \dots X_n$  и в соответствии с требованиями производственного процесса и прошитой пользовательской программы изменяет состояние выходов  $Y_1, Y_2, \dots Y_n$ . Этот цикл опроса входов и изменения выходов можно разделить на четыре основных шага.

Контроллеры семейства SIMATIC S7-1200 являются, пожалуй, одними из наиболее востребованных из всей линейки, так как обладают универсальностью, должного уровня функционалом и приемлемой ценой [25].

Основой всей системы программируемых логических контроллеров является процессорный модуль, отвечающий за обработку, хранение данных и выполнение пользовательской программы. Представленное семейство ПЛК может оснащаться пятью разными моделями центральных процессоров, которые отличаются производительностью, набором входных и выходных каналов и объемом рабочей памяти. Помимо того, линейка S7-1200 имеет несколько различных модификаций, все конфигурации, доступные при программировании в среде разработки TIA Portal представлены на рисунке 42.

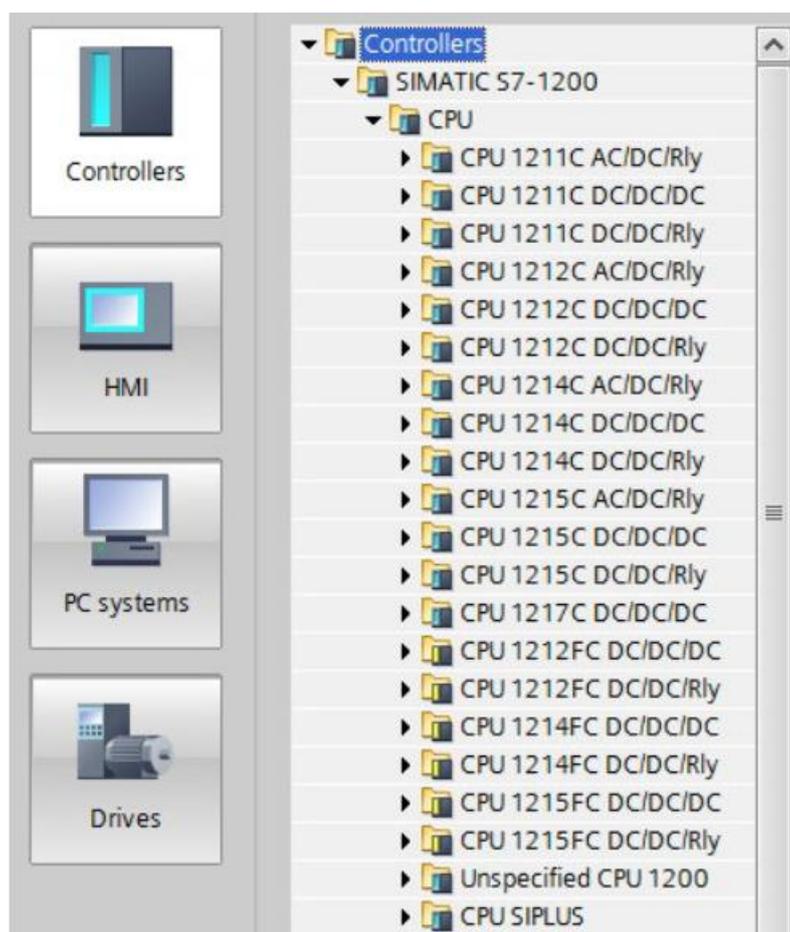


Рисунок 42 – Конфигурирование оборудования в TIAPortal

Конфигурации отличаются по типу центрального процессора, напряжению питания, а также типу дискретных входов и выходов. К примеру, контроллер CPU-1215C DC/DC/DC, который используется при выполнении бакалаврской работы, имеет напряжение питания равное 24 В, дискретные входы и выходы 24 В, на основе транзисторных ключей [18].

На рисунке 43 представлен ПЛК Siemens S7-1200 CPU-1215C с таблицей технических параметров.

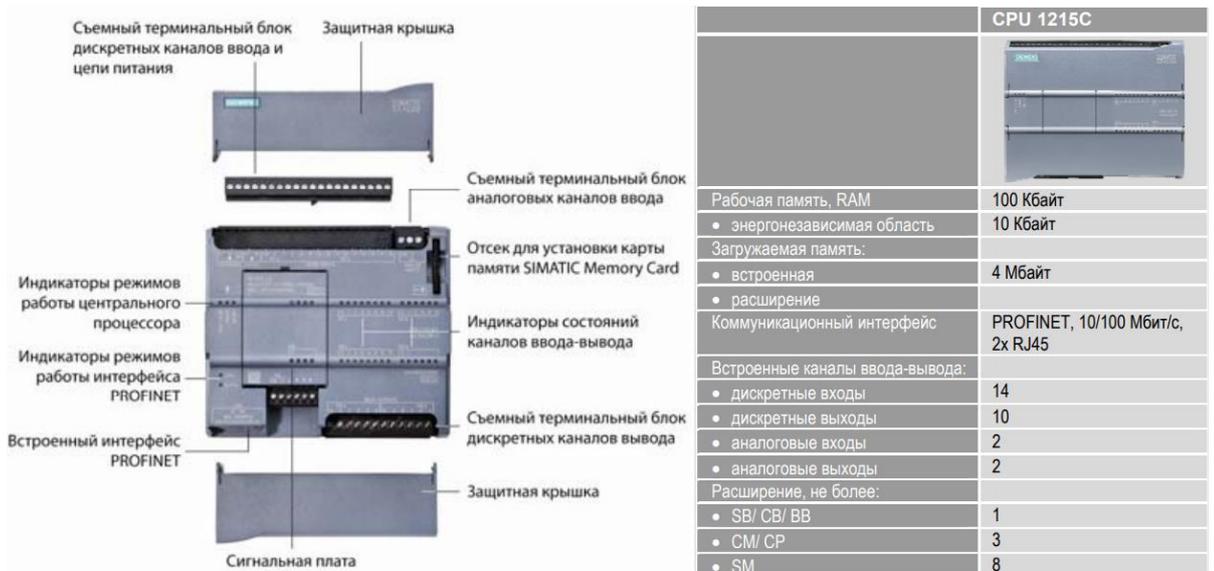


Рисунок 43 - ПЛК Siemens S7-1200 CPU-1215C.

На рисунке 44 представлена схема подключения внешних цепей к ПЛК.

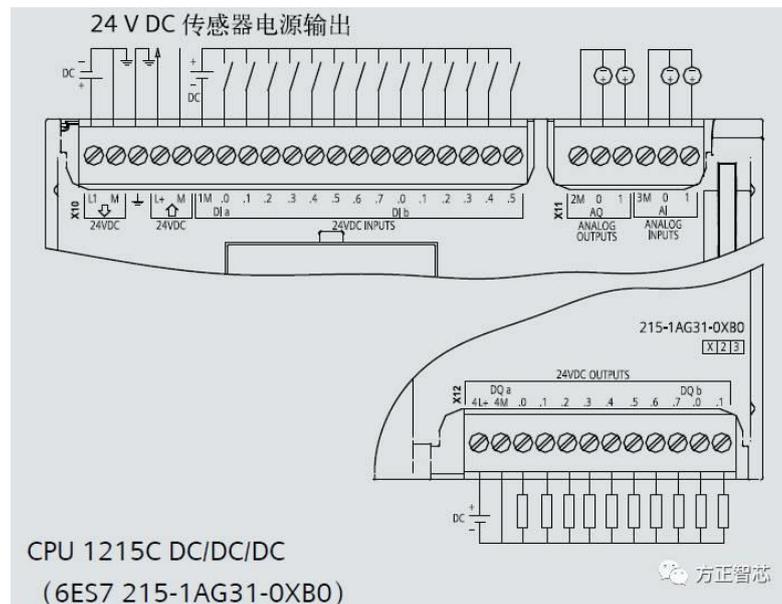


Рисунок 44 – Схема подключение внешних цепей.

Комплектация ПЛК Siemens S7-1200, рассмотренная выше, поддерживает высокий уровень сигнала (логическая единица), составляющий не менее 20 В, и низкий уровень сигнала (логический ноль), составляющий не более 0,1 В. Программируемый логический контроллер способен одновременно коммутировать до 10 выходов с номинальной нагрузкой 0,5 А или 5 Вт для ламповых нагрузок. В использованной комплектации отсутствует встроенная защита от перегрузок, при необходимости она может обеспечиваться внешними цепями, но для реализации данного проекта включение дополнительных цепей не требуется. При работе с панелями оператора и контроле необходимых в рамках выполнения данного проекта объектов ПЛК использует абсолютную адресацию.

Для подключения всего комплекса и осуществления управления им недостаточно стандартного набора каналов дискретных входов и выходов ПЛК Siemens S7-1200. При подключении промышленного манипулятора МП-11 задействуется 12 дискретных выходов, по 1 на каждую катушку электропневмоклапанов и 10 дискретных входов для подключения датчиков манипулятора. Для подключения конвейера требуется 1 дискретный выход, 3 дискретных и 1 аналоговый вход. Из этого можно сделать вывод, что стандартного набора дискретных каналов не хватит для реализации системы управления разрабатываемым комплексом, без подключения специализированных модулей расширения дискретных входов и выходов.

## 5.2 Система прослеживаемости

Прослеживаемость (Traceability) является одним из главных инструментов мониторинга качества на предприятии. Его суть заключается в том, что каждая партия сырья, деталей, полуфабриката и прочих товаров на производстве маркируется штрих-кодом или RFID меткой, и все производственные операции и их перемещение фиксируются в удаленных базах данных или облачных хранилищах. Имея исчерпывающую информацию о материальных потоках внутри производства, можно легко определить, на каком участке был допущен брак и при обнаружении несоответствий и отклонений от требуемых параметров своевременно остановить всю партию сырья или изделий. Система Traceability, реализованная в рамках выполнения данного проекта состоит из сканера штрих-кодов фирмы DATALOGIC и панели оператора. Подключение сканера к панели осуществляется через USB разъем. На рисунке 45 представлен используемый сканер.



Рисунок 45 – Сканер штрих-кодов quickscan DATALOGIK

Для настройки работы сканера и корректного получения данных с него, необходимо было настроить триггер срабатывания и выборку данных в среде разработки EasyBuilder Pro. На рисунке 46 представлено окно с настройками выборки данных.

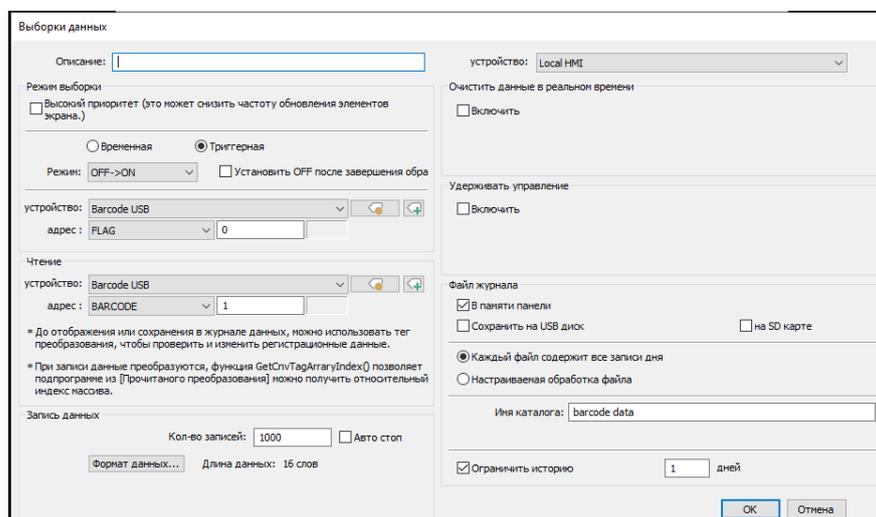


Рисунок 46 – Настройка выборки данных

Система отвечает требованиям Digital Ready, иными словами, транспортный роботизированный комплекс готов к отправке данных о транспортируемых деталях на более высокий уровень для последующего мониторинга.

### 5.3 Оценка времени выполнения микроконтроллером Atmel ATmega2560 команд высокого уровня

Соединение ПЛК Siemens и платы А протоколу связи Modbus TCP/IP, влечет за собой увеличение временных затрат на выполнение рабочего цикла программы, при дальнейших возможных доработках транспортного комплекса, например, включения в его состав световой завесы безопасности, это может существенно отразиться на корректности его работы. Была осуществлена оценка времени выполнения команд высокого уровня в среде разработки Arduino IDE, при помощи программного кода, представленного ниже:

```

    unsigned int timerValue; // Инициализация переменной, хранящей
значение таймера
    void setup() {
        Serial.begin(9600); // Инициализация последовательного порта со
скоростью 9600
        // установки таймера 1 для максимальной частоты тактирования
(16МГц)
        TCCR1A = 0;
        TCCR1B = 0;
    }
    void loop() {
        noInterrupts(); // Запрет прерываний
        TCNT1H = 0; // Сброс внутреннего таймера 1
        TCNT1L = 0;

        TCCR1B = 1; // Запуск работы внутреннего таймера 1
        // [исследуемая часть программного кода]
        TCCR1B = 0; // Остановка таймера 1 после завершения выполнения
исследуемой части программного кода
        // Чтение значения внутреннего таймера 1
        timerValue = (unsigned int) TCNT1L | ((unsigned int) TCNT1H << 8);
        interrupts(); // Включение прерываний
        // Вывод значения внутреннего таймера 1 в микросекундах на
компьютер через последовательный порт
        Serial.print( (float)(timerValue - 2) * 0.0625);
        Serial.println(" mks");
        delay(500);
    }

```

В качестве исследуемой части программного кода, поочередно использовались основные команды высокого уровня, реализованные в среде разработки Arduino IDE. Полученные результаты сведены в таблицу 8.

Таблица 9 – Время выполнения команд высокого уровня платформой Arduino MEGA 2560 PRO

Команда высокого уровня	Время выполнения (мкс)
Запись байта в память/чтение из памяти	0,125
Единичный цикл for	0,25
Сложение, вычитание, умножение данных типа int	0,438
Сравнение двух переменных типа int (с применением if)	0,188
Деление данных типа int	14,68
Сложение, вычитание, умножение данных типа long	1,38
Деление данных типа long	41,87
int => float	4,5
Сложение, вычитание, умножение данных типа float	12,5
Деление данных типа float	34,37
sin(float)	124
digitalWrite	8
digitalRead	7
pinMode	5
analogRead	111

Продолжение таблицы 9

Команда высокого уровня	Время выполнения (мкс)
Бесконечный цикл loop( )	0,5
Бесконечный цикл while(1)	0,125
Холостой цикл CPU (команда asm(«NOP»))	0,0625
Прямое обращение к регистрам данных портов (PORTB=0b10000000 (pin13 => «1»), PORTB=0b00000000 (pin13 => «0»))	0,0625
Использование регистров направления передачи данных порта: DDRB=0b10000000 (pin13 => OUTPUT)	0,0625
shiftOut	210

Самой время затратной из исследованных команд является функция analogRead (pin), предназначенная для чтения аналоговых портов микроконтроллера. Для снижения времени выполнения данной функции было выполнено изменение значения предделителя частоты АЦП микроконтроллера в одном из регистров управления АЦП – регистре ADCSRA (ADC Control and Status Register A, регистр управления и состояния), структура которого представлена на рисунке 47 [3,4,12,13,14].

Регистры	Биты							
	7	6	5	4	3	2	1	0
<b>ADCSRA</b>	ADEN	ADSC	ADATE(ADFR2)	ADIF	ADIE	ADPS[2:0]		
<b>ADMUX</b>	REFS[1:0]		ADLAR	×	MUX[3:0]			
<b>ADCSRB</b>	ADTS[2:0]			×	×	×	×	×
<b>ADCL</b>	Результат преобразования. Значение имеют только 10 бит. Старшие или младшие - зависит от значения бита ADLAR.							
<b>ADCH</b>								

Рисунок 47 – Регистры управления АЦП

Доступные варианты значений предделителя, настраиваемого с помощью трёх младших битов (LSB) ADCSRA сведены в таблицу 10.

Таблица 10 – Доступные варианты значений предделителя АЦП и соответствующие им значения битов ADPS (ADC Prescaler Select).

ADPS[2:0]	Коэффициент деления тактовой частоты CPU
000	2
001	2
010	4
011	8
100	16
101	32
110	64
111	128

По биты ADPS установлены так, что их комбинация соответствует коэффициенту деления равному 128. В программном коде для микроконтроллера отвечающего за работу интерфейсного модуля (server устройство) будет использоваться комбинация значений битов ADPS, соответствующая коэффициенту деления – 16. Таким образом, время выполнения функции analogRead (pin) сократится в 8 раз.

Стоит отметить, что уменьшение значения предделителя частоты работы АЦП микроконтроллера негативно сказывается на точность обработки данных [16,20,21,27]. В соответствии с этим, требуется настройка и проверка корректности данных, приходящих с оптического датчика цвета.

После проведенных исследований, было принято решение использовать прямое обращение к портам микроконтроллера и настройку регистра ADCSRA при разработке системы управления, что позволит сократить время выполнения рабочего цикла пользовательской программы при снижении точности функции analogRead (pin), не превышающим допустимые нормы.

## 5.4 Первый алгоритм работы

Оптический датчик цвета, установленный на конвейере, передает данные о считанном с цветовой маркировки значении в систему управления, которая, при получении данной информации идентифицирует деталь и в соответствии с этим выбирает режим работы. Ниже представлен порядок действий, выполняемых комплексов в процессе отработки первого алгоритма:

- занесение данных о детали в систему прослеживаемости;
- идентификация детали комплексом по цветовой маркировке и сигналу с индуктивного датчика, детектирующего наличие детали на ленте конвейера;
- запуск конвейера и перемещение детали в область рабочей зоны, определяемой по срабатыванию индуктивного датчика;
- выполнение рабочего цикла манипулятора.

Каждый шаг работы пары клапан-датчик в манипуляторе происходит следующим образом. При подаче управляющего сигнала, соответствующий электропневмоклапан открывается и воздух поступает в полость пневмоцилиндра до тех пор, пока не сработает концевой выключатель или геркон. После чего происходит переход к следующему шагу программы.

Программный код, реализующий выполнение первого алгоритма для server устройства представлен в приложении А. Client устройство осуществляет контроль, обработку и передачу информации при помощи блока данных, используемого для реализации связи по протоколу Modbus. На рисунках 48 и 49 представлена упрощенная блок схема алгоритма выполнения действий промышленным роботом манипулятором в ходе выполнения первого алгоритма работы.

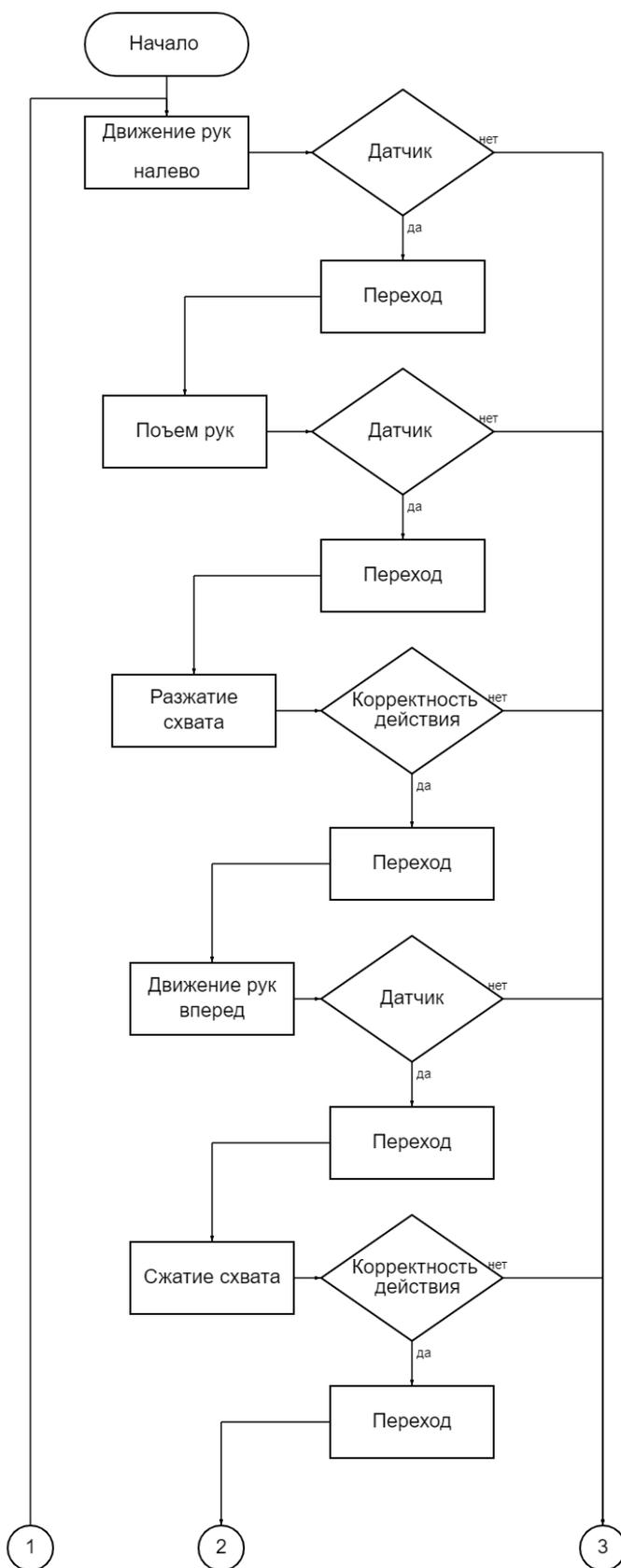


Рисунок 48 – Блок схема действий, выполняемых манипулятором

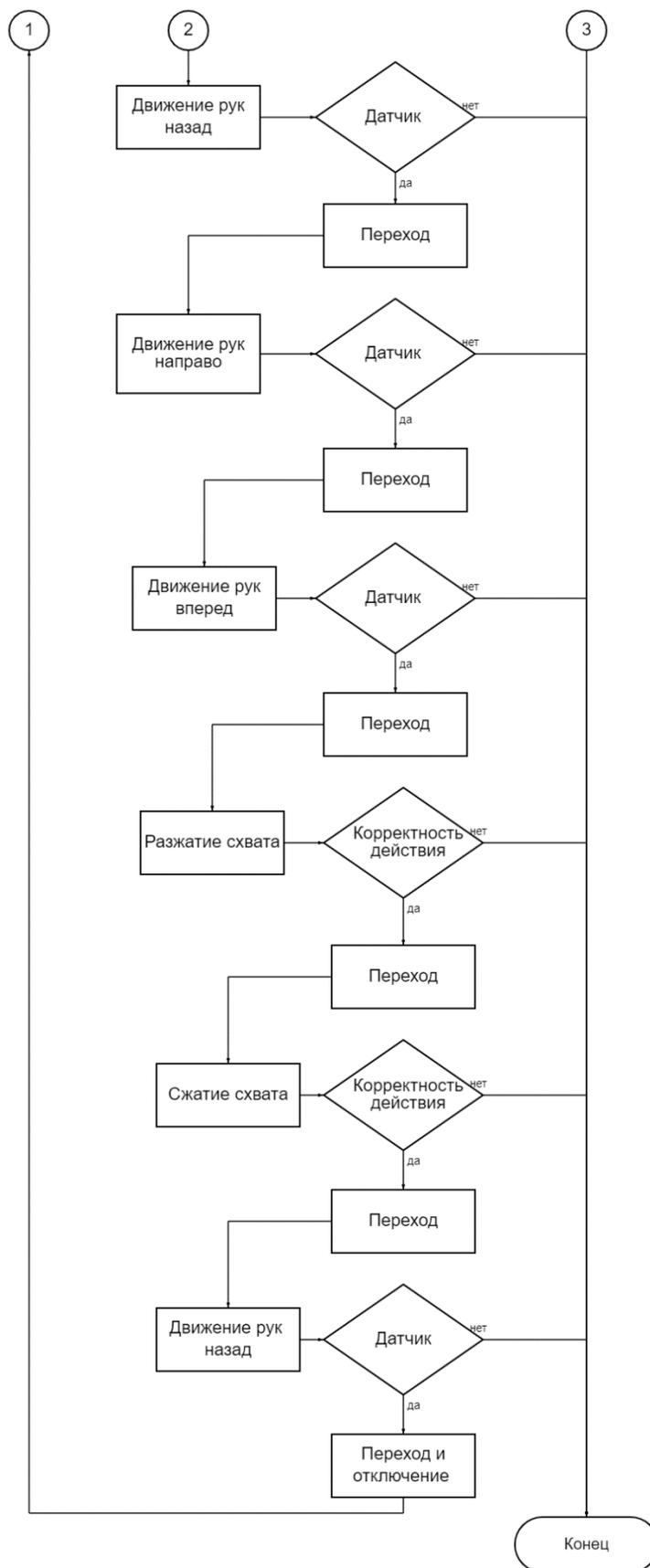


Рисунок 49 – Блок схема действий, выполняемых манипулятором (продолжение)

Данный алгоритм работы осуществляет простейшие транспортные функции и переносит объект манипулирования из начальной в конечную точку.

### **5.5 Второй алгоритм работы**

Различия первого и второго алгоритмов работы заключаются в выборе тары, в которую в конечном итоге будет перемещен объект манипулирования и рабочего схвата для выполнения действий. Выборка производится при помощи цветовой маркировки, нанесенной на деталь. Порядок действий, обрабатываемый комплексом в процессе выполнения второго алгоритма идентичен действиям, представленным в пункте 6.1. Как было отмечено выше, будет различаться лишь тара, в которую нужно транспортировать объект манипулирования, для чего в порядок выполнения действий манипулятора были внесены соответствующие изменения. Программный код, реализующий выполнение второго алгоритма для server устройства представлен в приложении Б. Client устройство осуществляет контроль, обработку и передачу информации при помощи блока данных, используемого для реализации связи по протоколу Modbus. На рисунках 50 и 51 представлена упрощенная блок схема алгоритма выполнения действий промышленным роботом манипулятором в ходе выполнения первого алгоритма работы.

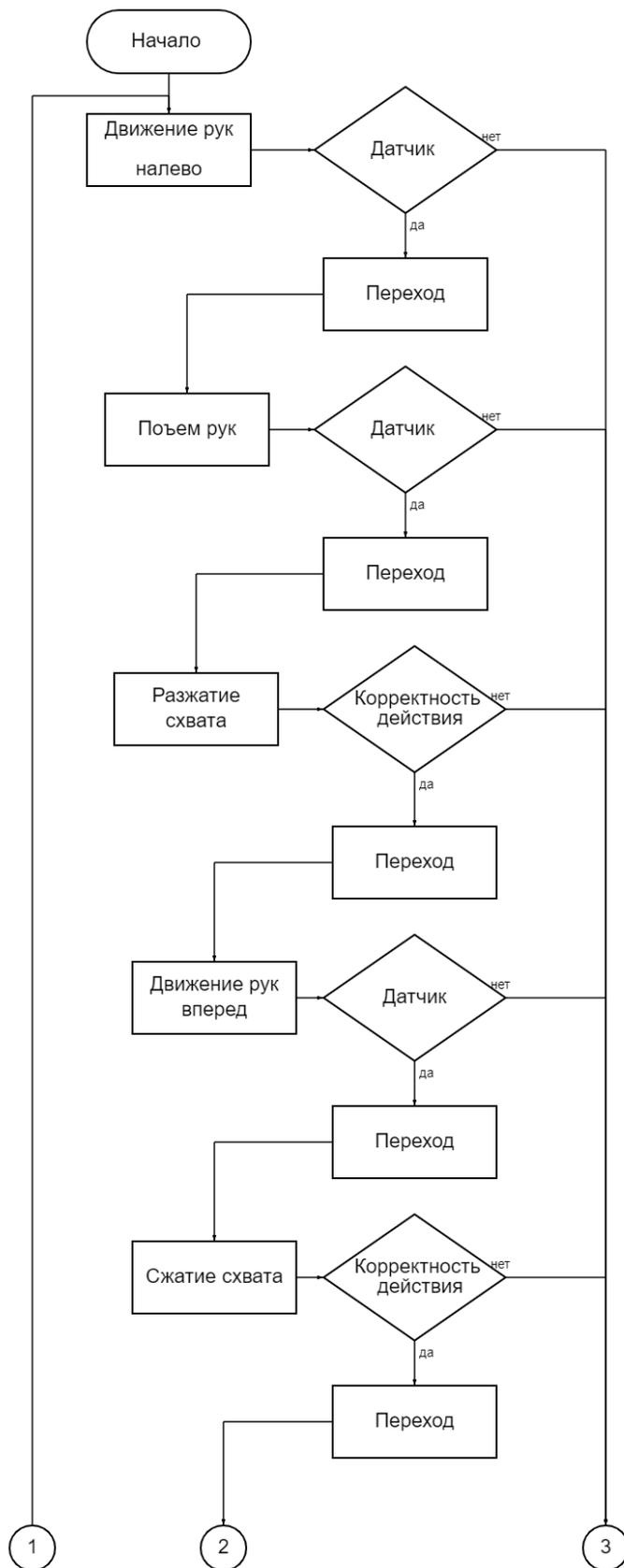


Рисунок 50 – Блок схема действий, выполняемых манипулятором

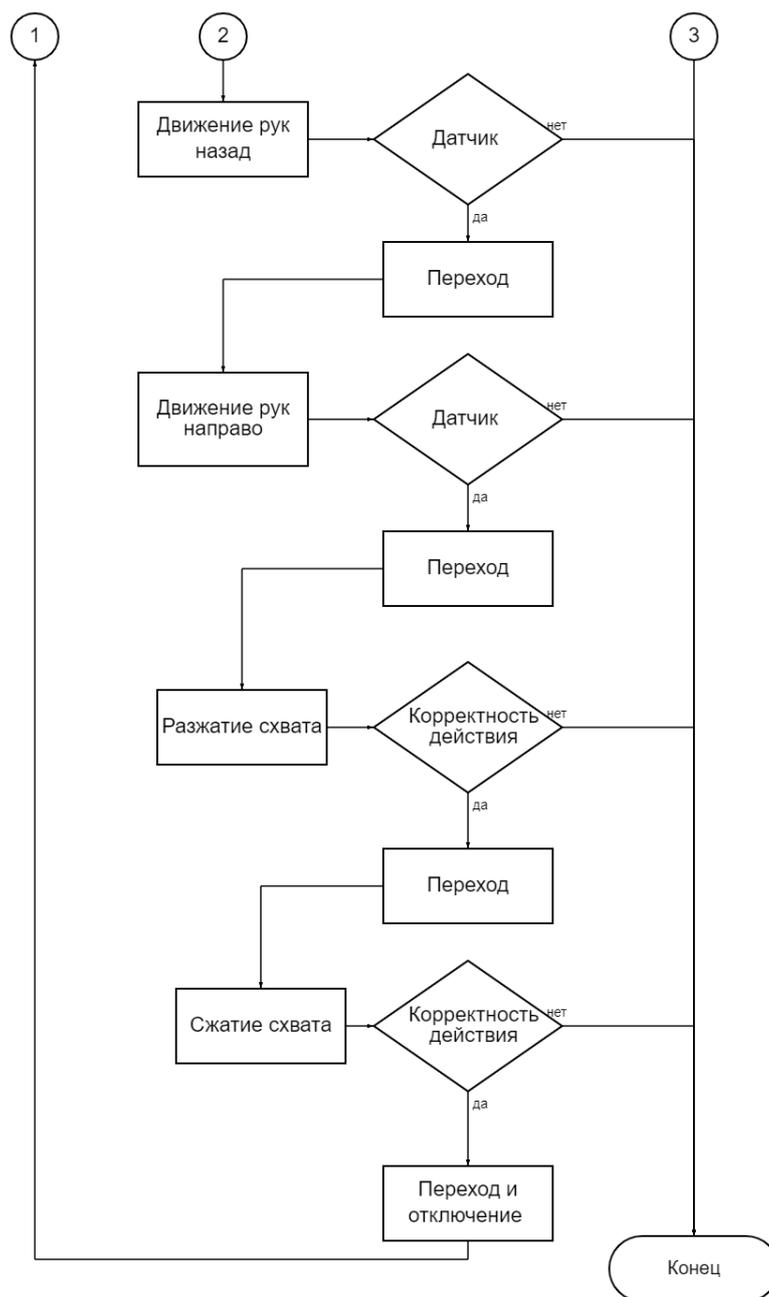


Рисунок 51 – Блок схема действий, выполняемых манипулятором (продолжение)

Данный алгоритм работы осуществляет простейшие транспортные функции и переносит объект манипулирования из начальной в конечную точку, но с измененными условиями, по сравнению с первым алгоритмом

## 5.6 Демонстрационный режим работы

Данный режим работы манипулятора активизируется, когда интерфейсный модуль не подключен к программируемому логическому контроллеру Siemens S7-1200 и предназначен для показательной работы манипулятора и демонстрации его двигательных способностей. Программный код, реализующий выполнение демонстрационного режима представлен в приложении В. На рисунках 52 и 53 представлена упрощенная блок схема одного из возможных алгоритмов выполнения действий промышленным роботом манипулятором в ходе выполнения показательного цикла работы.

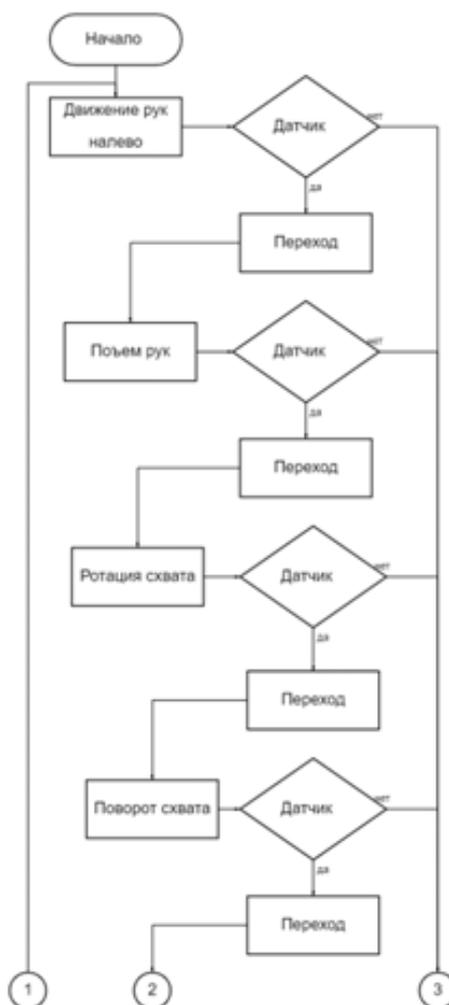


Рисунок 52 – Блок схема действий, выполняемых манипулятором

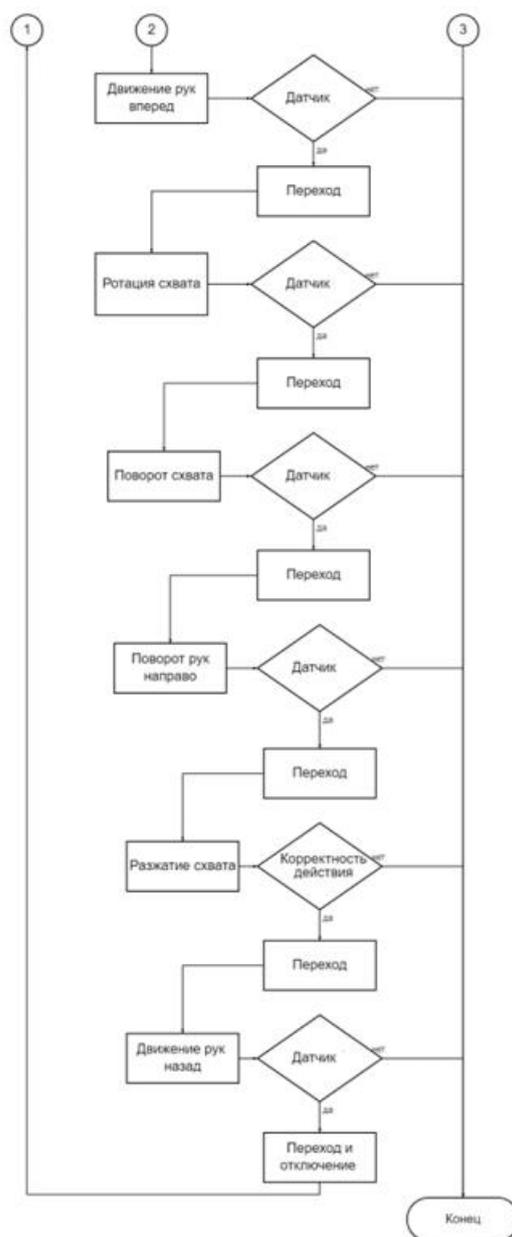


Рисунок 53 – Блок схема действий, выполняемых манипулятором (продолжение)

Данный режим работы демонстрирует кроссплатформенность интерфейсного модуля, осуществляя контроль за перемещением манипулятора при помощи микроконтроллера.

## 5.7 Режим ручного управления

В данном режиме работы управление манипулятором ведется при помощи панели оператора или кнопок, интегрированных в интерфейсный модуль. Режим ручного управления запускается при помощи нажатия кнопки Пуск и завершается при нажатии кнопки Стоп, а все действия совершаются манипулятором только при нажатии соответствующей кнопки. Программный код, реализующий выполнение режима ручного управления для server устройства представлен в приложении Г. Программный код для client устройства представлен в приложении Д. На рисунках 54 и 55 представлена упрощенная блок схема ручного режима работы.



Рисунок 54 – Блок схема действий, выполняемых манипулятором

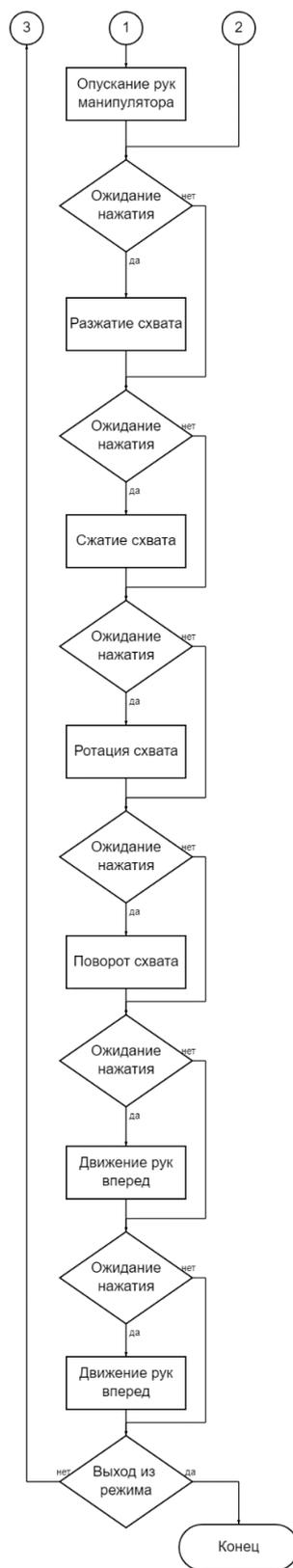


Рисунок 55 – Блок схема действий, выполняемых манипулятором  
(продолжение)

## Выводы по главе

В данной главе было рассмотрено устройство управления транспортным роботизированным комплексом, принцип его функционирования и основные параметры, рассмотрены разработанные режимы работы транспортного роботизированного комплекса на базе ПЛК Siemens S7-1200. Для их реализации, в соответствии с поставленными задачами и организацией рабочего пространства, были подобраны оптимальные последовательности движений манипулятора, а также составлены блок-схемы алгоритмов для них. Исследован и реализован способ, позволяющий сократить временные затраты на выполнение пользовательского кода со стороны server устройства. Программный код для server устройства разрабатывался в среде программирования Arduino IDE, а для client устройства в среде разработки TIA Portal. А на основе данной главы были составлены блок-схемы алгоритмов работы программ client и server устройств, входящих в состав транспортного роботизированного комплекса.

## **6 Методические рекомендации по работе с лабораторным стендом**

Завершающим этапом выполнения выпускной квалификационной работы необходимо было разработать методические рекомендации по работе с разработанным лабораторным стендом. На основе которых, в последующем, будут составлены методические пособия для выполнения лабораторных работ в рамках проведения занятий среди студентов старших курсов направления подготовки Электроника и робототехника, обучающихся по программе бакалавриата.

При разработке лабораторного стенда были учтены требования электрической безопасности для студентов и первичные меры защиты оборудования от случаев короткого замыкания. Таким образом, разъемы штыревого типа расположены лишь на тех элементах комплекса, к которым напрямую не подводится напряжение питания, все контакты и соединения изолированы. В целях предотвращения травмоопасных ситуаций при работе с лабораторным стендом, студентам запрещается находиться внутри рабочей зоны манипулятора в процессе его функционирования. Рабочая зона ограждена операторскими местами, за которыми работают студенты. Подключение составных частей комплекса к интерфейсному модулю должно осуществляться только при выключенном напряжении питания. Рабочее давление компрессора, передаваемое в блок распределения воздуха манипулятора МП-11 не должно превышать 3 атмосфер, номинальное рабочее давление составляет порядка 2,5 атмосфер. Настройка и проверка электрических соединений, а также обслуживание комплекса производится исключительно при отключенном канале подачи воздуха и напряжении питания. После окончания работы с транспортным роботизированным комплексом, необходимо стравить воздух из компрессора, убедиться в том, что все электропневмоклапана закрыты, после чего обесточить комплекс.

Разработанный пример лабораторного практикума ориентирован как на студентов, только знакомящихся с программируемыми логическими контроллерами и системами автоматизации производственных процессов, так и на студентов, уже ознакомленных с данными основами на других дисциплинах. Для ознакомления с принципами работы ПЛК и обработкой дискретных сигналов была спроектирована и собрана панель переключателей и индикации, представленная на рисунках 56.

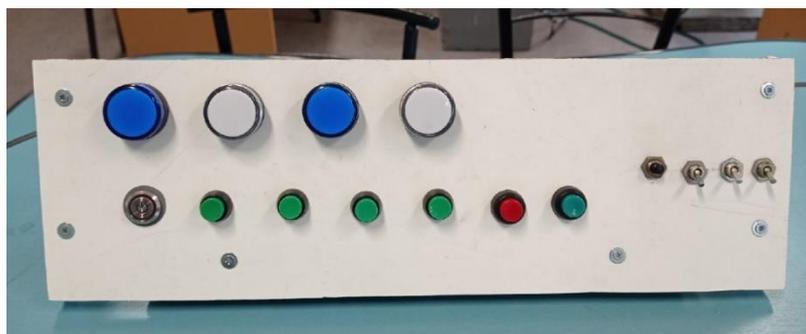


Рисунок 56 – Панель переключателей и индикации

Примеры лабораторных работ, которые могут быть использованы при последующем создании методических материалов для студентов, представлены в приложении Е.

#### Выводы по главе

В данной главе были рассмотрены методические рекомендации, основные требования, предъявляемые к безопасности и работе с разработанным лабораторным комплексом, а также разработаны и представлены примеры лабораторных работ, которые в дальнейшем могут быть включены в образовательный процесс в рамках лабораторного практикума по дисциплинам, связанным с автоматизацией производственных процессов.

## **Заключение**

В ходе выполнения выпускной квалификационной работы были выполнены все поставленные задачи. В рамках проекта был разработан учебный лабораторный стенд, реализованный в виде транспортного роботизированного комплекса на базе ПЛК Siemens S7-1200 и ленточного транспортера.

Для данного комплекса были разработаны все необходимые электронные и электрические связи и подобраны аппаратные платформы и компонентная база. Реализовано несколько режимов работы комплекса и все необходимое программное обеспечение, а также составлены методические рекомендации по работе со стендом, включающие в себя требования безопасности при работе с роботизированным комплексом и примеры лабораторных работ, которые могут быть включены в образовательный план по дисциплинам, связанным с автоматизацией технологических процессов.

Была составлена структурная схема транспортного роботизированного комплекса на базе ПЛК Siemens, схема электрическая соединений и подключения промышленного пневматического робота манипулятора МП-11, перечень элементов, входящих в состав данной схемы представлен в приложении Ж.

Составлена схема электрическая соединений интерфейсного модуля и схема электрическая соединений разъемов интерфейсного модуля, перечни элементов, представленных в данных схемах приведены в приложениях И и К соответственно.

Для каждого из режимов работы были составлены блок-схемы алгоритмов функционирования программы и блок-схемы алгоритмов движения манипулятора. Разработанный транспортный роботизированный комплекс, при дальнейшей механической и программной настройке и отладке может быть использован в качестве учебно-лабораторного стенда при проведении плановых занятий по дисциплинам, связанным с автоматизацией технологических процессов и программируемыми контроллерами.

## Список используемой литературы

1. Белов, А.В. «Микроконтроллеры AVR: от азов программирования до создания практических устройств» / А.В. Белов. - СПб.: Наука и техника, 2016. - 544 с.
2. Белов А.В. «Разработка устройств на микроконтроллерах AVR: шагаем от "чайника" до профи» : Книга / А.В. Белов. - СПб.: Наука и техника, 2013. - 528 с.
3. Бьерн Страуструп. «Язык программирования С++». - 369 с
4. Вагнер Б. «С#. Эффективное программирование. 50 рекомендаций по усовершенствованию программирования на С#» / Б. Вагнер; Пер. с англ. М. Горелик. - М.: Лори, 2013. - 256 с.
5. Водовозов А.М. «Микроконтроллеры для систем автоматики»: Учебное пособие / А.М. Водовозов. - Вологда: ВоГТУ, 2002. - 123 с.
6. ГОСТ 30097-93 «Роботы промышленные. Системы координат и направления движений». – Введён 01.01.1996 г.
7. Евстифеев А.В. «Микроконтроллеры AVR семейства Tiny фирмы ATMEЛ»: Руководство пользователя / А.В. Евстифеев. - М.: ДМК, 2015. - 426 с.
8. Евстифеев А.В. «Микроконтроллеры AVR семейств Mega»: Руководство пользователя / А.В. Евстифеев. - М.: ДМК, 2015. - 588 с.
9. Евстифеев А.В. «Микроконтроллеры AVR семейства Tiny»: Руководство пользователя / А.В. Евстифеев. - М.: Додэка XXI, 2007. - 432 с.
10. Евстифеев А.В. «Микроконтроллеры AVR семейства Classic фирмы ATMEЛ» / А.В. Евстифеев. - М.: ДМК, 2015. - 286 с.
11. Кравцов А.Г. «Изучение конструкций манипуляторов промышленных роботов РБ-241 и МП 11.01» / А.Г. Кравцов. – Оренбург: ОГУ, 2013 – 57 с.
12. Магда Ю.С. «Программирование и отладка С/С++ приложений микроконтроллеров ARM» / Ю.С. Магда. - М.: ДМК Пресс, 2012. - 168 с.

13. Магда Ю.С. «Программирование и отладка C/C++ приложений для микроконтроллеров ARM» / Ю.С. Магда. - М.: ДМК, 2014. - 168 с.
14. Магда Ю.С. «Современные микроконтроллеры. Архитектура, программирование, разработка устройств» / Ю.С. Магда. - М.: ДМК, 2013. - 228 с.
15. «Промышленные роботы-манипуляторы их применение в современном мире» // Информio [Электронный ресурс]. URL: <https://www.informio.ru/publications/id5024/Promyshlennye-roboty-manipuljatory-ih-primenenie-v-sovremennom-mire?> (дата обращения: 19.06.2022).
16. Прокопенко В.С. «Программирование микроконтроллеров ATMEЛ на языке С» / В.С. Прокопенко. - СПб.: Корона-Век, 2015. - 320 с.
17. «Подробное описание протокола Modbus TCP с примерами команд» // IPC2U [Электронный ресурс]. URL: <https://ipc2u.ru/articles/prostye-resheniya/modbus-tcp/?> (дата обращения: 19.06.2022).
18. «Программируемые контроллеры S7-1200» [Электронный ресурс]. URL: [https://www.saa.su/Document/PLC/Simatic/cpu\\_S7-1200\\_2013.pdf](https://www.saa.su/Document/PLC/Simatic/cpu_S7-1200_2013.pdf) (дата обращения: 19.06.2022).
19. «Роботы-манипуляторы – виды и особенности применения» // VECTORUS [Электронный ресурс]. URL: <https://vektor.us.ru/blog/robot-manipulyator.html?> (дата обращения: 19.06.2022).
20. С. Монк. «Программируем Arduino. Профессиональная работа со скетчами» — СПб.: Питер, 2017.
21. Схиртладзе, А.Г. «Оборудование машиностроительных предприятий» [Электронный ресурс] / А.Г. Схиртладзе, В.И. Выходец, Н.И. Никифоров. — Электрон. текстовые дан. — Волгоград: РПК «Политехник», 2021. URL: [доступа: http://bookre.org/reader?file=807789](http://bookre.org/reader?file=807789) (дата обращения 15.03.2022).

22. Тарабарин, В.Б. «Промышленные роботы и манипуляторы» [Электронный ресурс] / В.Б. Тарабарин. — Электрон. текстовые дан. — Москва: МГТУ им. Н.Э.Баумана, 2021. URL: [http://cncnc.ru/documentation/theory\\_of\\_mechanismus\\_and\\_machines/lect\\_19.htm](http://cncnc.ru/documentation/theory_of_mechanismus_and_machines/lect_19.htm) (дата обращения 11.03.2022).

23. «Arduino oficial store» [Электронный ресурс] // «Arduino Mega» URL: <https://store.arduino.cc/usa/mega-2560-r3> (дата обращения: 20.02.2021).

24. «Arduino IDE» [Электронный ресурс]. «Download the Arduino IDE» URL: <https://www.arduino.cc/en/main/software> (дата обращения: 20.02.2021).

25. Industry Support Siemens» [Электронный ресурс]. URL: <https://support.industry.siemens.com/cs/start?lc=en-WW> (дата обращения: 19.06.2022).

26. IEC 61131-3. «Programmable controllers – Part 3: Programming languages», - 2013 у.

27. James A. Langbridge. «Arduino Sketches. Tools and Techniques for Programming Wizardry», 2015. 480p. ISBN 9781118919606

28. «MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b» [Электронныйресурс]. URL: [https://modbus.org/docs/Modbus\\_Messaging\\_Implementation\\_Guide\\_V1\\_0b.pdf](https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf) (дата обращения: 20.02.2021).

29. «NTRODUCTION TO MODBUS TCP/IP» [Электронный ресурс]. URL: [https://www.prosoft-technology.com/kb/assets/intro\\_modbustcp.pdf](https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf) (дата обращения: 19.06.2022).

30. «Runtime программирование промышленного робота на RCML» [Электронный ресурс] / — Электрон. текстовые дан. — 2021. — URL: <https://habr.com/ru/post/283226/> (дата обращения 15.03.2022).

## Приложение А

### Программный код первого алгоритма работы

```
// 1 алгоритм работы
void algorithmNUM1() // Проверка алгоритмов работы
{
int registr = modbusTCPServer.holdingRegisterRead(0x00);
registr = ((registr << 8) | (registr >> 8));
registr1 = 0;
    DDRC = 0b11111100;
    DDRL = 1;
    DDRG = 1;
    DDRH = 0;
    DDRD = 0;
    DDRA = 0;
    PORTC = 0;
    PORTG = 0;
    PORTL = 0;
    PORTH = 0;
    PORTD = 0;
    PORTA = 0;
    ADCSRA |= (1<<ADPS2);
    ADCSRA &=~ ((1<<ADPS1)|(1<<ADPS0));
    byte I = PINA;
    byte I1 = PIND;
    byte I2 = PINC;
    byte I3 = PINH;
int ColorCoding = analogRead(A0);
int induct1 = ((1 << PD1) & I1);
int induct2 = ((1 << PD0) & I1);
    while(1)
    {
        if (!induct1)
        {
            if (printMark > 350 && printMark < 500)// 1 Режим работы по цветовой
маркировке
```

## Продолжение Приложения А

```
{
    registr1 = 1;
    PORTL |= (1 << 6); // Запуск конвейера
}
else
{
    break;
}
}

registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPserver.holdingRegisterWrite(0x00, registr1);

if (bitRead(registr, 1) == true && induct2) // Чтение бита режима работы
{
    PORTL &= ~(1 << 6); // Остановка конвейера
    PORTG |= (1 << 6); // Движение рук налево 5 клапан 40 вывод
}

if ((1 << PH0) & I3) // Чтение датчика поворота налево 17 вывод
{

    registr1 = 8; // Запись управляющего бита
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPserver.holdingRegisterWrite(0x00, registr1);

if (bitRead(registr, 8) == true) // Чтение управляющего бита
{

    PORTL |= (1 << 4); // Подъем рук 8 клапан 46 вывод

}
```

## Продолжение Приложения А

```
if ((1 << PD0) & I1) // Чтение датчика подъема 21 вывод
{

    registr1 = 16; // Запись управляющего бита
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);

if (bitRead(registr, 16) == true) // Чтение управляющего бита
{

    PORTC |= (1 << 7); // Разжатие схвата 11 клапан 39 вывод
    registr1 = 32;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);
delay (3000);

if (bitRead(registr, 32) == true) // Чтение управляющего бита
{

    PORTC |= (1 << 4); // Движение рук вперед 2 клапан 34 вывод

}

if ((1 << PA3) & I) // Чтение датчика движения рук 25 вывод
{

    registr1 = 64; // Запись управляющего бита
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);
```

## Продолжение Приложения А

```
if (bitRead(registr, 64) == true) // Чтение бита срабатывания датчика
{

    PORTC &= ~(1 << 7); // Сжатие схвата
    delay (3000);
    PORTC &= ~(1 << 5); // Выключить 2 клапан
    PORTC |= (1 << 2); // Движение рук назад 1 клапан 32 вывод
    registr1 = 128;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);

if ((1 << PA5) & I) // Чтение датчика движения рук 27 вывод
{

    registr1 = 256; // Запись управляющего бита
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);

if (bitRead(registr, 256) == true) // Чтение бита срабатывания датчика
{
    PORTG &= ~(1 << 6); // Выключить 5 клапан
    PORTL |= (1 << 0); // Поворот рук направо 6 клапан 42 вывод
}

if ((1 << PD2) & I) // датчик поворота направо
{
    registr1 = 512;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);
```

## Продолжение Приложения А

```
if (bitRead(registr, 512) == true) // Чтение бита срабатывания датчика
{

    PORTC |= (1 << 4); // Движение рук вперед

}

if ((1 << PA3) & I) // Чтение датчика движения рук 25 вывод
{

    registr1 = 1024;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPserver.holdingRegisterWrite(0x00, registr1);

if (bitRead(registr, 1024) == true) // Чтение бита срабатывания датчика
{

    PORTC |= (1 << 7); // Разжатие схвата 11 клапан 37 вывод
    registr1 = 2048;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPserver.holdingRegisterWrite(0x00, registr1);
delay (3000);

if (bitRead(registr, 2048) == true) // Чтение бита срабатывания датчика
{
    PORTC &= ~(1 << 7); // Сжатие схвата
    delay (3000);
    PORTC &= ~(1 << 4); // Выключить 2 клапан
    PORTC |= (1 << 2); // Включить 1 клапан
}
```

## Продолжение Приложения А

```
if ((1 << PA5) & I) // Чтение датчика движения рук 27 вывод
{

    registr1 = 2048;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPServer.holdingRegisterWrite(0x00, registr1);

if (bitRead(registr, 2048) == true) // Чтение бита срабатывания датчика
{
    PORTL &= ~(1 << 0);
    registr1 = 0;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPServer.holdingRegisterWrite(0x00, registr1);
break;
}
return 0;
}
```

## Приложение Б

### Программный код второго алгоритма работы

```
// 2 алгоритм работы
void algorithmNUM2() // Проверка алгоритмов работы
{
int registr = modbusTCPServer.holdingRegisterRead(0x00);
registr = ((registr << 8) | (registr >> 8));
registr1 = 0;
    DDRC = 0b11111100;
    DDRL = 1;
    DDRG = 1;
    DDRH = 0;
    DDRD = 0;
    DDRA = 0;
    PORTC = 0;
    PORTG = 0;
    PORTL = 0;
    PORTH = 0;
    PORTD = 0;
    PORTA = 0;
    ADCSRA |= (1<<ADPS2);
    ADCSRA &=~ ((1<<ADPS1)|(1<<ADPS0));
    byte I = PINA;
    byte I1 = PIND;
    byte I2 = PINC;
    byte I3 = PINH;
int ColorCoding = analogRead(A0);
int induct1 = ((1 << PD1) & I1);
int induct2 = ((1 << PD0) & I1);

    while(1)
    {
        if (!induct1)
        {
```

## Продолжение Приложения Б

```
if (ColorCoding > 500 && ColorCoding < 750)// 1 Режим работы по цветовой  
маркировке
```

```
{  
    registr1 = 2;  
    PORTL |= (1 << 6); // Запуск конвейера  
}  
else  
{  
    break;  
}  
}
```

```
    registr1 = ((registr1 << 8) | (registr1 >> 8));  
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);
```

```
if (bitRead(registr, 2) == true && induct2) // Чтение бита режима работы  
{  
    PORTL &= ~(1 << 6); // Остановка конвейера  
    PORTG |= (1 << 6); // Движение рук налево 5 клапан 40 вывод  
}
```

```
if ((1 << PH0) & I3) // Чтение датчика поворота налево 17 вывод  
{  
  
    registr1 = 8; // Запись управляющего бита  
}
```

```
registr1 = ((registr1 << 8) | (registr1 >> 8));  
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);
```

```
if (bitRead(registr, 8) == true) // Чтение управляющего бита  
{
```

```
    PORTL |= (1 << 4); // Подъем рук 8 клапан 46 вывод
```

## Продолжение Приложения Б

```
}

if ((1 << PD0) & I1) // Чтение датчика подъема 21 вывод
{

    registr1 = 16; // Запись управляющего бита
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);

if (bitRead(registr, 16) == true) // Чтение управляющего бита
{

    PORTC |= (1 << 7); // Разжатие схвата 11 клапан 39 вывод
    registr1 = 32;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);
delay (3000);

if (bitRead(registr, 32) == true) // Чтение управляющего бита
{

    PORTC |= (1 << 4); // Движение рук вперед 2 клапан 34 вывод

}

if ((1 << PA3) & I) // Чтение датчика движения рук 25 вывод
{

    registr1 = 64; // Запись управляющего бита
}
}
```

## Продолжение Приложения Б

```
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPserver.holdingRegisterWrite(0x00, registr1);

if (bitRead(registr, 64) == true) // Чтение бита срабатывания датчика
{

    PORTC &= ~(1 << 7); // Сжатие схвата
    delay (3000);
    PORTC &= ~(1 << 5); // Выключить 2 клапан
    PORTC |= (1 << 2); // Движение рук назад 1 клапан 32 вывод
}

if ((1 << PA5) & I) // Чтение датчика движения рук 27 вывод
{

    registr1 = 256; // Запись управляющего бита
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPserver.holdingRegisterWrite(0x00, registr1);

if (bitRead(registr, 256) == true) // Чтение бита срабатывания датчика
{
    PORTG &= ~(1 << 6); // Выключить 5 клапан
    PORTL |= (1 << 0); // Поворот рук направо 6 клапан 42 вывод
}

if ((1 << PD2) & I) // датчик поворота направо
{
    registr1 = 512;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPserver.holdingRegisterWrite(0x00, registr1);
```

## Продолжение Приложения Б

```
if (bitRead(registr, 1024) == true) // Чтение бита срабатывания датчика
{
    PORTC |= (1 << 7); // Разжатие схвата 11 клапан 37 вывод
    registr1 = 2048;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);
delay (3000);

if (bitRead(registr, 2048) == true) // Чтение бита срабатывания датчика
{
    PORTC &= ~(1 << 7); // Сжатие схвата
    delay (3000);
    registr1 = 2048;
}

if (bitRead(registr, 2048) == true) // Чтение бита срабатывания датчика
{
    PORTL &= ~(1 << 0);
    registr1 = 0;
}
registr1 = ((registr1 << 8) | (registr1 >> 8));
modbusTCPSTCServer.holdingRegisterWrite(0x00, registr1);
break;
}
return 0;
}
```

## Приложение В

### Программный код демонстрационного режима

```
if (Ethernet.hardwareStatus() == EthernetNoHardware) {
  Serial.println("Ethernet shield was not found. Sorry, can't run without hardware. :(");
  while (true) {
    delay(1); // do nothing, no point running without Ethernet hardware
  }
}
if (Ethernet.linkStatus() == LinkOFF) {
  Serial.println("Ethernet cable is not connected.");
  DDRC = 0b11111100;
  DDRL = 1;
  DDRG = 1;
  DDRH = 0;
  DDRD = 0;
  DDRA = 0;
  PORTC = 0;
  PORTG = 0;
  PORTL = 0;
  PORTH = 0;
  PORTD = 0;
  PORTA = 0;
  byte I = PINA;
  byte I1 = PIND;
  byte I2 = PINC;
  byte I3 = PINH;

  // Демонстрационный режим работы
  while(1)
  {

    PORTG |= (1 << 6); // Поворот налево 5 клапан 40 вывод

    if ((1 << PH0) & I3)// 1 датчик (поворот налево)D17
```

## Продолжение Приложения В

```
{
  PORTL |= (1 << 4); // Подъем рук 8 клапан 46 вывод
}

if((1 << PD0) & I1)// 3 датчик (подъем рук)D21
{
  PORTC |= (1 << 6); // Движение схвата 3 клапан 36 вывод
}

if((1 << PA7) & I)// 7 датчик (движение схвата)D29
{

  PORTC |= (1 << 3); // Поворот схвата 9 клапан 33 вывод
}

if((1 << PH1) & I3)// 9 датчик (поворота схвата)D16
{
  PORTC |= (1 << 3); // Движение рук вперед 2 клапан 34 вывод
}

if((1 << PA3) & I)// 5 датчик (движение рук вперед)D25
{
  PORTC &= ~(1 << 6); // Выключить 3 клапан
  PORTD |= (1 << 0); // Движение схвата 4 клапан 38 вывод
}

if((1 << PC6) & I2)// 8 датчик (датчик движения схвата)D31
{
  PORTC &= ~(1 << 3); // Выключить 9 клапан
  PORTC |= (1 << 5); // Поворот схвата 10 клапан 35 вывод
}

if ((1 << PD3) & I1)// 10 датчик (поворот схвата)D18
```

## Продолжение Приложения В

```
{
  PORTG &= ~(1 << 6); // Выключить 5 клапан
  delay(3000);
  PORTL &= ~(1 << 4); // Выключить подъем рук 8 клапан 46 вывод
}

if ((1 << PD3) & I1) // Датчик опускания манипулятора
{
  PORTL |= (1 << 0); // Поворот рук направо 6 клапан 42 вывод
}

if ((1 << PD2) & I1) // 2 датчик (поворот направо) D19
{
  PORTG |= (1 << 5); // Разжатие схвата 12 клапан 39 вывод
  delay(3000);
  PORTG &= ~(1 << 5); // Сжатие схвата 12 клапан 39 вывод
}

PORTC &= ~(1 << 3); // Выключить 2 клапан
PORTC |= (1 << 2); // Движение рук назад 1 клапан 32 вывод
delay(3000);
PORTC = 0;
PORTG = 0;
PORTL = 0;
PORTH = 0;
PORTD = 0;
PORTA = 0;
}
}
```

## Приложение Г

### Программный код режима ручного управления

```
void algorithmNUM3() // Проверка алгоритмов работы
{
int registr = modbusTCPServer.holdingRegisterRead(0x00);
registr = ((registr << 8) | (registr >> 8));
registr1 = 0; // возможно надо будет убрать
    DDRC = 0b11111100;
    DDRL = 1;
    DDRG = 1;
    DDRH = 0;
    DDRD = 0;
    DDRA = 0;
    DDRJ = 0;
    DDRB|= (0 << 2)|(0 << 3)|(0 << 1);// Индуктивные датчики
    // установка пределителя 16 код 100
    ADCSRA |= (1<<ADPS2);
    ADCSRA &=~ ((1<<ADPS1)|(1<<ADPS0));
    // Маски чтения битов
    byte I = PINA;
    byte I1 = PIND;
    byte I2 = PINC;
    byte I3 = PINH;
    byte I4 = PINB;
    byte I5 = PINJ;
    // Индуктивные датчики
    int induct1 = ((1 << PB5) & I4);
    int induct2 = ((1 << PB6) & I4);

while(1)
{
    if ((1 << PB7) & I4)// Чтение кнопки старт
    {
        if ((1 << PJ0) & I5)// Чтение кнопки Стоп
```

## Продолжение Приложения Г

```
{
  while(1)
  {
    ;
  }
}
else{

  if ((1 << PJ0) & I5)// Чтение кнопки Сброс
  {
    registr1 = 0;
    PORTC = 0;
    PORTG = 0;
    PORTL = 0;
    PORTH = 0;
    PORTD = 0;
    PORTA = 0;
    PORTJ = 0;
  }
  registr1 = ((registr1 << 8) | (registr1 >> 8));
  modbusTCPServer.holdingRegisterWrite(0x00, registr1);

  if (induct1)// Срабатывания датчика, идентифицирующего наличие детали
  {
    registr1 = 4;
    PORTL |= (1 << 6); // Запуск конвейера
  }

  registr1 = ((registr1 << 8) | (registr1 >> 8));
  modbusTCPServer.holdingRegisterWrite(0x00, registr1);

  if (bitRead(registr, 4) == true && induct2) // Чтение бита режима работы
  {
```

## Продолжение Приложения Г

```
PORTL &= ~(1 << 6); // Остановка конвейера
}

// Работа манипулятора в ручном режиме управления
if (bitRead(registr, 8) == true) // Чтение бита режима работы
{
    PORTL &= ~(1 << 1);
    PORTG |= (1 << 7); // Движение рук налево
}

if (bitRead(registr, 16) == true) // Чтение бита режима работы
{
    PORTG &= ~(1 << 7);
    PORTL |= (1 << 1); // Движение рук направо
}

if (bitRead(registr, 32) == true) // Чтение бита режима работы
{
    PORTL |= (1 << 5); // Подъем рук
}

if (bitRead(registr, 64) == true) // Чтение бита режима работы
{
    PORTL &= ~(1 << 3); // Опускание рук
}

if (bitRead(registr, 128) == true) // Чтение бита режима работы
{
    PORTC &= ~(1 << 3);
    PORTC |= (1 << 5); // Движение рук вперед
}

if (bitRead(registr, 256) == true) // Чтение бита режима работы
```

## Продолжение Приложения Г

```
{
    PORTC &= ~(1 << 5);
    PORTC |= (1 << 3); // Движение рук назад
}

if (bitRead(registr, 512) == true) // Чтение бита режима работы
{
    PORTC |= (1 << 7); // Разжатие нижнего схвата
}

if (bitRead(registr, 1024) == true) // Чтение бита режима работы
{
    PORTC &= ~(1 << 7); // Сжатие нижнего схвата
}

if (bitRead(registr, 2048) == true) // Чтение бита режима работы
{
    PORTC &= ~(1 << 5);
    PORTC |= (1 << 3); // Вращение нижнего схвата по часовой стрелке
}

if (bitRead(registr, 4096) == true) // Чтение бита режима работы
{
    PORTC &= ~(1 << 3);
    PORTC |= (1 << 5); // Вращение нижнего схвата против часовой стрелки
}

if (bitRead(registr, 8192) == true) // Чтение бита режима работы
{
    PORTG |= (1 << 5); // Ржатие верхнего схвата
}

if (bitRead(registr, 16384) == true) // Чтение бита режима работы
{
    PORTG &= ~(1 << 5); // Сжатие верхнего схвата
```

## Продолжение Приложения Г

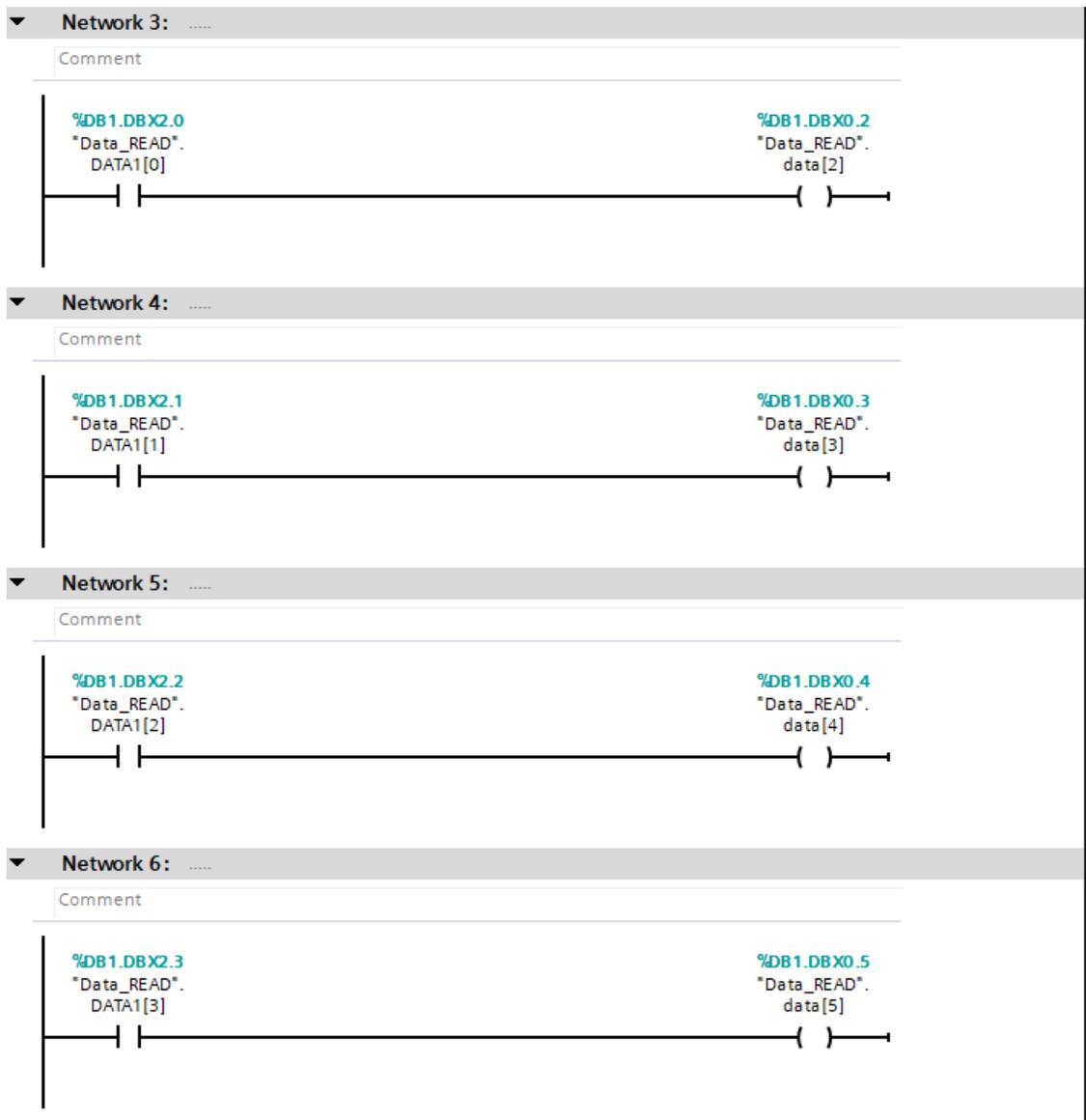
```
}

if (bitRead(registr, 32768) == true) // Чтение бита режима работы
{
    PORTD &= ~(1 << 0);
    PORTC |= (1 << 6); // Ротация схвата влево
}

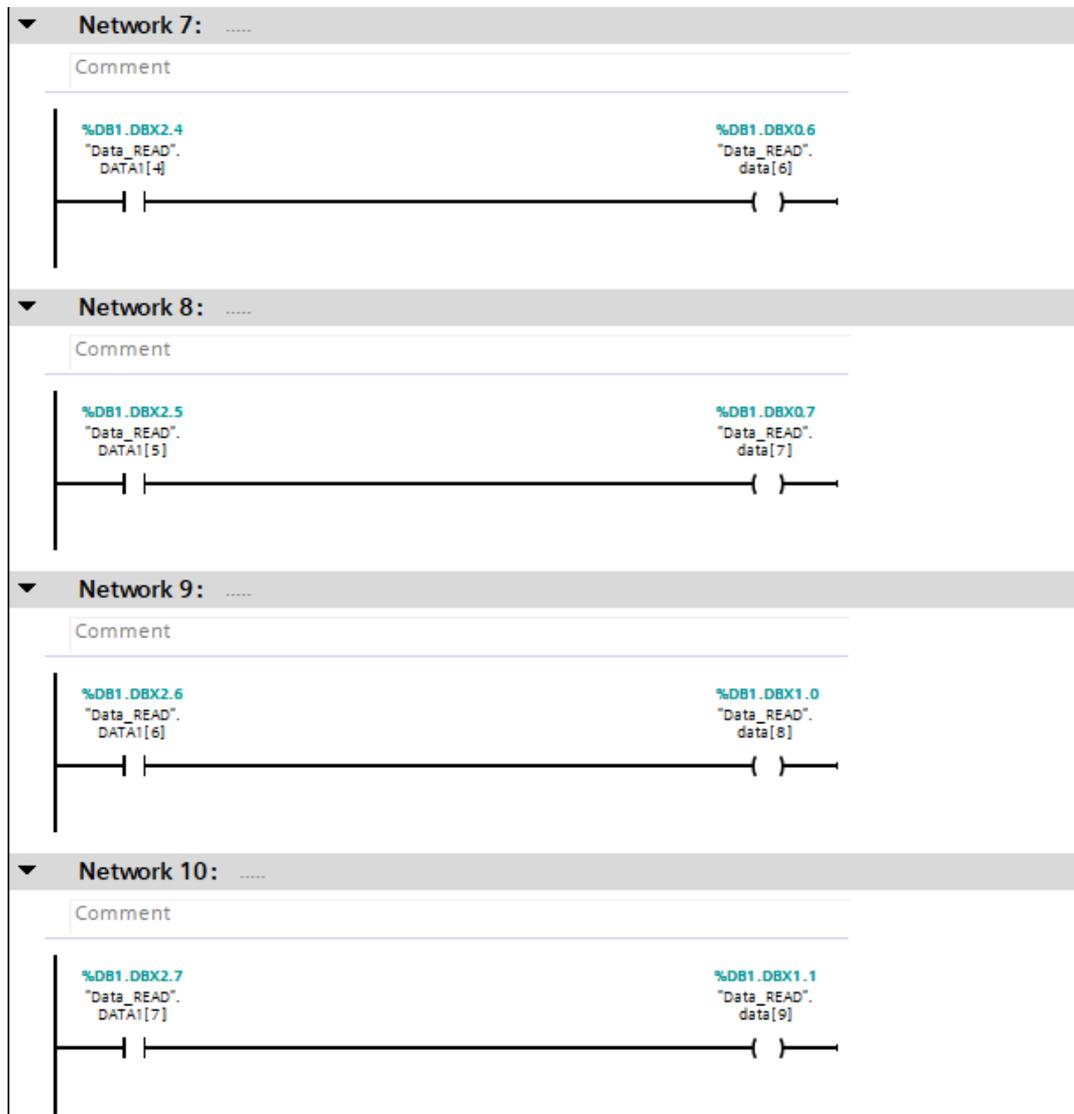
if (bitRead(registr, 16384) == true) // Чтение бита режима работы
{
    PORTC &= ~(1 << 6);
    PORTD |= (1 << 0); // Ротация схвата вправо
}
}
}
else {
    break;
}
}
}
```

# Приложение Д

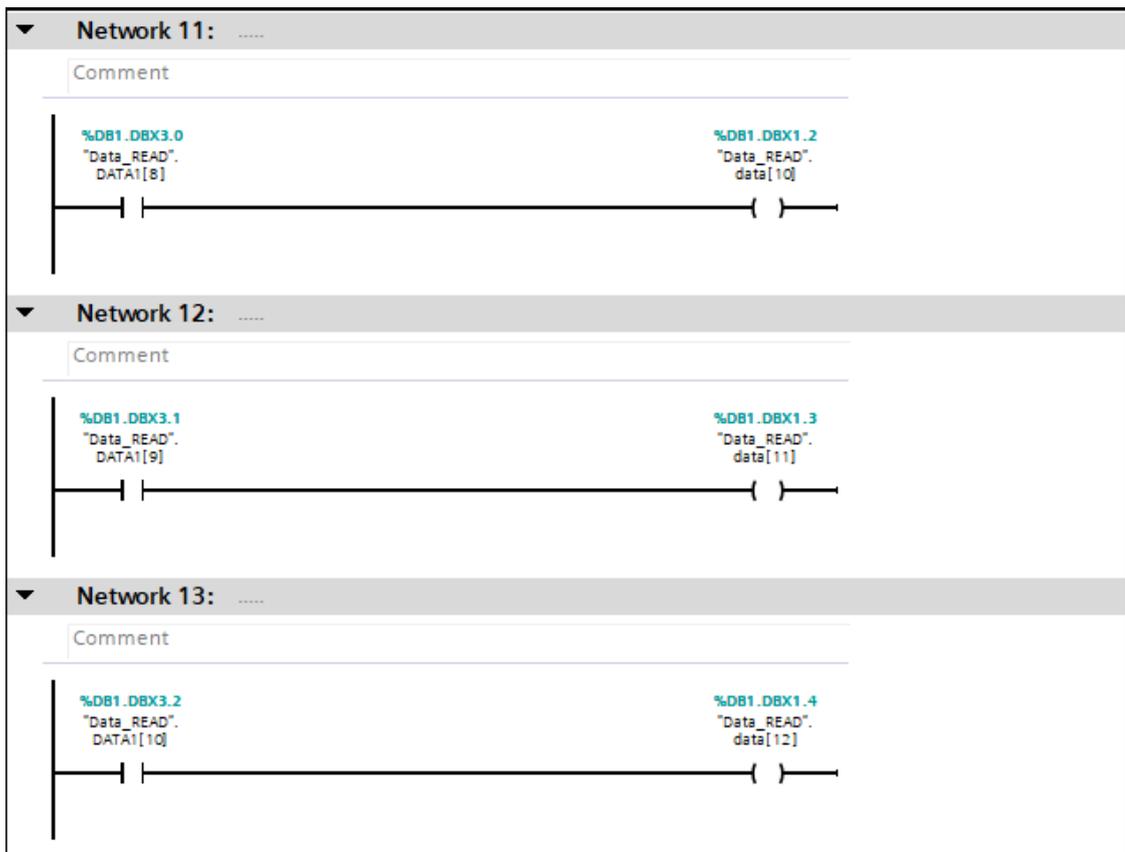
## Программный код client устройства



## Продолжение Приложения Д



## Продолжение Приложения Д



## Приложение Е

### Примеры лабораторных работ

#### Лабораторная работа № 1

Первая лабораторная работа «Изучение ПЛК Siemens S7-1200. Ввод, вывод и обработка дискретных данных» направлена на студентов, не имеющих до этого опыта работы с ПЛК. Для обеспечения удобства выполнения данной работы была спроектирована и собрана панель переключателей и индикации.

Цель работы: изучить основы подключения ПЛК и периферийных устройств. Получить первичные навыки в передаче и обработке дискретных сигналов (управление дискретными входами и выходами). Ознакомиться со средой разработки TIA Portal.

Необходимый методический материал: информация о среде разработки TIA Portal, сведения о ПЛК Siemens S7-1200, схема подключения и некоторые параметры.

Лабораторно-техническое оснащение: панель переключателей и индикации, конвейер.

В ходе выполнения данной лабораторной работы студентам необходимо подсоединить панель к ПЛК Siemens и с помощью кнопок, подключенных к дискретным входам контроллера коммутировать индикаторные лампы, подключенные к дискретным выходам. Реализовать разные варианты, например, обеспечить включение или выключение конкретных лампы при нажатии определенных кнопкой, включение лампы с задержкой, мигание, включение и выключение одной лампы с разных кнопок.

Во второй части лабораторной работы необходимо дополнительно подключить к ПЛК Siemens конвейер и добиться его запуска по указанному в варианте алгоритму. Например, запуск конвейера осуществляется по нажатию кнопки и получению сигнала с первого индуктивного датчика. При выполнении данного условия на панели загорается индикаторная лампа и через какое-то время лента конвейера приходит в движение, продолжая его до момента детектирования детали вторым индуктивным датчиком. После выполнения данного условия первая индикаторная лампа гаснет и загорается другая.

Благодаря кроссплатформенности интерфейсного модуля, который будет использоваться в более поздних лабораторных работах, все задания этой и последующих работ можно выполнять и на ПЛК ОВЕН, а некоторые и вовсе только с применением микроконтроллера, являющегося, при отключенном ПЛК, устройством управления интерфейсного модуля и всего комплекса.

## Продолжение Приложения Е

### Лабораторная работа № 2

Вторая лабораторная работа “Подключение панели оператора Weintek. HMI интерфейс” ориентирована на знакомство студентов с разработкой простейших человеко-машинных интерфейсов, позволяющих производить мониторинг и управление различными процессами.

Цель работы: познакомиться с принципами работы панели оператора, ее функциями и возможностями. Ознакомиться со средой разработки Easy Builder PRO. Научиться подключать панель оператора и устанавливать связь с ПЛК Siemens S7-1200.

Необходимый методический материал: информация о подключении панели оператора и установки связи, информация о среде разработки Easy Builder PRO (создание проекта, подключение, конфигурирование устройств и инструменты разработки). Краткая информация о связи ПЛК Siemens S7-1200 с панелью оператора и передаче данных.

Лабораторно-техническое оснащение: панель оператора Weintek, ПЛК Siemens S7-1200.

В ходе выполнения данной лабораторной работы студентам необходимо реализовать управление ПЛК с помощью панели оператора. Сделать HMI интерфейс с отображением на панели состояния входов и выходов и возможностью передавать данные в ПЛК.

## Продолжение Приложения Е

### Лабораторная работа № 3

Третья лабораторная работа Промышленный протокол связи Modbus TCP/IP направлена на изучение одного из популярных протоколов связи и возможностей его реализации.

Необходимые методические материалы: общая информация о протоколе TCP/IP и его реализации в TIA Portal.

Лабораторно-техническое оснащение: ПЛК Siemens S7-1200, панель оператора/панель переключателей и индикации.

В ходе выполнения данной лабораторной работы студентам необходимо реализовать связь между двумя ПЛК Siemens S7-1200 по протоколу Modbus TCP/IP. Например, к первому ПЛК, имеющему статус server, подключить кнопки с панели оператора или панели переключателей и индикации. Вторым ПЛК, имеющим статус client, опрашивать дискретные входа первого ПЛК и в соответствии с этим коммутировать дискретные выходы, после коммутации дискретных выходов 2 ПЛК, отправлять команду на коммутацию дискретных выходов 1 ПЛК.

## Продолжение Приложения Е

### Лабораторная работа № 4

Четвертая лабораторная работа “ Система Traceability на производстве” ориентирована на изучения студентами принципов использования и реализации системы прослеживаемости на производстве.

Цель работы: познакомиться с системой Traceability на производстве и примерами ее реализации.

Необходимый методический материал: информация о системе Traceability, ее назначении и примерах реализации. Информация о подключении и установке связи сканера штрих-кодов с панелью оператора. Информация о работе с удаленными базами данных и установке связи с ними.

Лабораторно-техническое оснащение: панель оператора Weintek, ПЛК Siemens S7-1200, сканер штрих-кодов.

В ходе выполнения данной лабораторной работы студентам необходимо реализовать систему Traceability. Обеспечить вывод информации со штрих-кодов на панель оператора (таблица данных, включающая в себя порядковый номер сканирования, время, дату и саму информацию со штрих-кода/qr кода), отобразить количество битов считанной информации и дополнительно в отдельное окно значение считанной информации, а также обеспечить передачу данных в ПЛК или облачное хранилище, например, базу данных MySQL.

## Продолжение Приложения Е

### Лабораторная работа № 5

Пятая лабораторная работа “Управление транспортным робототехническим комплексом” ориентирована на закрепление полученных навыков, получение первичных компетенций в области управления автоматизированными комплексами, а также изучение реализации промышленного протокола связи Modbus TCP/IP в среде разработки Arduino IDE.

Цель работы: применить ранее полученные знания и получить навыки в реализации системы управления роботизированным комплексом.

Необходимые методические материалы: Схемы электрические соединения и подключений всех составных частей лабораторного комплекса, данные о манипуляторе МП-11, данные о связи ПЛК с Arduino по протоколу Modbus TCP/IP.

Лабораторно-техническое оснащение: панель оператора Weintek, интерфейсный модуль, сканер штрих-кодов, промышленный манипулятор МП-11, конвейер.

В ходе выполнения данной лабораторной работы студентам необходимо реализовать циклическую систему управления транспортным робототехническим комплексом по заданному техническому заданию, имитирующему определенный вид транспортной функции, сортировки или набор необходимых действий.



## Приложения И

### Перечень элементов к схеме электрической соединений интерфейсного модуля

<i>Поз. обозначение</i>	<i>Наименование</i>	<i>Кол-во</i>	<i>Примечание</i>
<i>Модули</i>			
A1,A2	<i>Релейный модуль 8-и канальный JQC-3FF-S-Z</i>	2	
A3	<i>Сетевой модуль Ethernet shield W5500</i>	1	
A4	<i>Плата Arduino Mega 2560 PRO</i>	1	
A5	<i>Блок питания SIMATIC PM 1507</i>	1	
A6,A7	<i>Регулируемый преобразователь напряжения LM2596</i>	2	
A8,A9	<i>Оптический изолятор DTS-1R8P</i>	2	
A10	<i>Блок служебных кнопок Ф 12 мм</i>	1	
<i>Разъемы</i>			
X1-X4	<i>Разъем DB 37-F</i>	4	
<b>22-110304.45/09.344.03 ПЭ4</b>			
<b>Интерфейсный модуль перечень элементов</b>			
		<i>Стадия</i>	<i>Масса</i>
		<i>Масштаб</i>	
		<i>Лист 1</i>	<i>Листов 1</i>
<b>ТГУ зр. ЭЛБ-1802а</b>			

<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Изм.</i>	<i>Кол.уч.</i>
<i>Разработал</i>	<i>Лист</i>
<i>Проверил</i>	<i>№ док.</i>
<i>Т.контр.</i>	<i>Подп.</i>
<i>Исполн.</i>	<i>Дата</i>
<i>Утвердил</i>	

