

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 кафедры)

**11.03.04 «Электроника и нанoeлектроника»**

(код и наименование направления подготовки, специальности)

**Электроника и робототехника**

(направленность (профиль))

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему «Управление макетом трехкоординатного манипулятора с помощью ПЛК»

Обучающийся

**Э.И. Зарифуллин**

(И.О. Фамилия)

(личная подпись)

Руководитель

**А.А. Шевцов**

(И.О. Фамилия)

(личная подпись)

Тольятти 2023

## Аннотация

В данной бакалаврской работе разработана система управления макетом трехкоординатного манипулятора с помощью ПЛК.

Бакалаврская работа выполнена на 65 страницах машинописного текста, 35 рисунков, а также графическая часть в объеме 7 листов А1.

Объектом исследования являются системы управления трёхкоординатным манипулятором с помощью человеко-машинного интерфейса.

Задачи работы заключались в разработке системы управления трехкоординатным манипулятором со схватом на основе ПЛК с использованием в качестве взаимодействия с обучаемым НМІ панели, создать проект для панели НМІ, позволяющий задавать путем нажатия кнопок на сенсорном экране панели направление движения каждой оси и величину перемещения вдоль этой оси на расстояние 0-10 см. Обеспечить постоянное электрическое соединение и взаимодействие между ПЛК и НМІ панелью. В состав системы управления включены необходимые управляемые источники питания и драйверы двигателей в соответствии с исходными данными.

Работа состоит из 4 глав, в которых решены выше представленные задачи.

Для реализации интерфейса оператора и взаимодействия с обучаемым была выбрана операторская НМІ панель Weintek MT8071iP, и использовалось программное обеспечение EasyBuilder Pro.

## Содержание

Введение.....	5
1 Обзор состояния рынка .....	7
1.1 Описание объекта управления.....	7
1.1.1 Описание конструкции манипулятора.....	7
1.1.2 Размеры манипулятора и рабочей области .....	9
1.1.3 Типы датчиков, используемых в манипуляторе.....	10
1.2 Обзор типовых решений для управления реальными промышленными манипуляторами.....	10
1.2.1 Примеры промышленных манипуляторов с описанием используемых способов управления.....	11
1.2.2 Анализ рынка управления манипуляторами и тренды развития данного направления.....	19
1.3 Обзор решений для построения макетов манипуляторов .....	20
2 Разработка системы управления.....	21
2.1 Выбор модели ПЛК.....	23
2.2 Определение и выбор необходимых датчиков .....	25
2.3 Разработка драйверов питания двигателей .....	26
2.4 Схема системы управления манипулятором.....	33
3 Разработка управляющей программы для ПЛК.....	35
3.1 Контроллер S7-1200.....	35
3.2 Питание схемы .....	40
4 Разработка управляющего проекта для HMI панели.....	43
4.1 Написание программы для HMI Weintek .....	43
4.2 Быстрое спаривание Weintek и ПЛК Siemens .....	45
4.3 Добавление устройства Modbus TCP .....	46
4.3.1 Драйвер ПЛК с абсолютной адресацией .....	48
4.3.2 Драйвер ПЛК с символьной адресацией .....	49
4.4 Создание управляющей программы.....	53

Заключение.....	59
Список используемой литературы и источников .....	60

## Введение

В настоящее время робототехника и автоматизация являются ключевыми направлениями развития промышленности. Технологии автоматизации процессов производства и управления роботизированными системами уже не являются чем-то недостижимым и экзотическим. Однако, внедрение роботизированных систем и манипуляторов в промышленность требует серьезных знаний и опыта, а также высококвалифицированных специалистов. Применение роботизированных систем и манипуляторов позволяет повысить производительность, качество и эффективность производства, а также уменьшить трудозатраты и риски для работников. Одно из наиболее распространенных устройств, используемых в робототехнике и промышленности, предназначенных для перемещения объектов в трехмерном пространстве – это трехкоординатный манипулятор. Манипуляторы используются в различных сферах промышленности, таких как автомобильная, электронная, пищевая, фармацевтическая и другие. Они могут использоваться для перемещения деталей, сборки, покраски, сварки, обработки и других задач.

В данной работе рассматривается управление макетом трехкоординатного манипулятора с помощью ПЛК.

В работе будут рассмотрены основные аспекты проектирования системы управления, выбор необходимых компонентов, а также разработка управляющей программы и управляющего проекта для НМІ панели. Целью работы является создание функционирующего макета трехкоординатного манипулятора, который будет управляться с помощью ПЛК на основе исходных данных.

Исходя из вышесказанного, в настоящей работе ставятся следующие задачи:

- разработать систему управления трехкоординатным манипулятором со схватом на основе ПЛК с использованием в качестве взаимодействия с обучаемым НМІ панели;

- создать проект для панели НМІ, позволяющий задавать путем экранной клавиатуры направление движения каждой оси и величину перемещения вдоль этой оси на расстояние 0-10 см;
- обеспечить постоянное электрическое соединение и взаимодействие между ПЛК и НМІ панелью;
- в состав системы управления включить необходимые управляемые источники питания и драйверы двигателей в соответствии со следующими исходными данными:
  - используемый ПЛК: серия Siemens 1200, выходное напряжение +24В;
  - количество осей управления: 3;
  - наличие конечных выключателей на каждой оси, шт: 1;
  - напряжение питания управляемых двигателей на осях перемещения, В: 9;
  - встроенный в двигатели энкодер: импульсный, напряжение питания энкодера 9В, выход энкодера – открытый коллектор;
  - максимальный потребляемый двигателями осей ток: 1А;
  - напряжение питания ПЛК: 24В;
  - устройство взаимодействия с обучаемым: НМІ панель Weintek MT8071iP;
  - источник питающего напряжения всей системы: сетевое напряжение 220В 50Гц.

Таким образом, данная работа будет полезна для инженеров и специалистов в области автоматизации, а также для всех, кто интересуется техническими решениями в промышленности.

## **1 Обзор состояния рынка**

В данном разделе мы провели обзор типовых решений для управления реальными промышленными манипуляторами, а также обзор решений для построения макетов манипуляторов.

Мы также описали существующие на рынке решения для построения макетов манипуляторов, их конструкцию и возможности. Были описаны компоненты, используемые в этих решениях, а также датчики и ПЛК, используемые для управления макетами манипуляторов.

В ходе обзора состояния рынка мы также рассмотрели существующие решения для управления манипуляторами, используемые в промышленности, их конструкцию, а также возможности по интеграции в систему управления. Были описаны преимущества и недостатки различных решений.

### **1.1 Описание объекта управления**

В данном разделе рассматриваются основные характеристики манипулятора, которые могут повлиять на выбор оптимального решения при разработке системы управления.

#### **1.1.1 Описание конструкции манипулятора**

Типовой трехкоординатный манипулятор состоит из трех основных частей: базы, горизонтального балансира и вертикального подъемника. База манипулятора является его основной опорной частью и обычно имеет форму круглой или прямоугольной платформы, которая крепится к рабочей поверхности.

Горизонтальный балансир крепится к базе и обеспечивает манипулятору движение в горизонтальной плоскости. Он может иметь различную конструкцию в зависимости от типа манипулятора, но обычно представляет собой конструкцию, состоящую из нескольких секций, которые могут

двигаться относительно друг друга. Горизонтальный балансир также обычно имеет крепления для крепления рабочего инструмента.

Вертикальный подъемник крепится к горизонтальному балансиру и обеспечивает манипулятору движение в вертикальной плоскости. Он может быть реализован в виде различных механизмов, таких как гидравлический или пневматический цилиндр, или механический механизм, использующийся для перемещения рабочего инструмента вверх и вниз.

Кроме того, в конструкцию манипулятора могут быть включены различные виды датчиков, такие как энкодеры, датчики положения, датчики нагрузки и другие, которые необходимы для обратной связи и контроля движения манипулятора в реальном времени.

Таким образом, конструкция манипулятора является важной характеристикой объекта управления, которая должна быть учтена при разработке системы управления макетом трехкоординатного манипулятора.

Для примера, рассмотрим конструкцию трехкоординатного манипулятора типа SCARA (Selective Compliance Assembly Robot Arm), который широко используется в промышленности для манипулирования предметами с высокой точностью. В этом типе манипулятора первые две оси имеют стандартную структуру "плечо-локоть", а третья ось расположена перпендикулярно к первым двум осям и представляет собой подъемник.

Пример конструкции SCARA манипулятора можно увидеть на рисунке 1 [1]:





Рисунок 1 – Пример SCARA-манипулятора

Источник: [1].

### **1.1.2 Размеры манипулятора и рабочей области**

Для разработки системы управления манипулятором необходимо знать размеры манипулятора и его рабочей области. Размеры манипулятора могут зависеть от его конструкции и назначения.

Рабочая область манипулятора – это область в пространстве, которую он может охватить. Для данного проекта мы будем ограничены рабочей областью 10 см x 10 см x 10 см, что соответствует размерам манипулятора и заданным условиям проекта. В данном проекте мы будем использовать манипулятор с размерами 20 см x 20 см x 20 см (длина, ширина, высота).

Эти размеры важны для определения возможностей манипулятора и разработки системы управления, которая обеспечит нужное перемещение в пределах рабочей области. Они также могут влиять на выбор компонентов, используемых при разработке манипулятора и системы управления.

### **1.1.3 Типы датчиков, используемых в манипуляторе**

Для связи энкодеров с HMI панелью Weintek MT8071iP можно использовать различные интерфейсы связи, такие как RS485, RS232, USB и Ethernet.

Наиболее распространенным интерфейсом является RS485, который позволяет передавать данные на большие расстояния (до 1,2 км) и поддерживает множественное подключение устройств к одной линии. Для использования интерфейса RS485 требуется специальный модуль, подключенный к порту коммуникации на HMI панели.

Также можно использовать интерфейс RS232, который обеспечивает более надежную связь на коротких расстояниях (до 15 метров), но не поддерживает множественное подключение.

Для беспроводной связи можно использовать интерфейс Bluetooth, который позволяет передавать данные на расстояние до 10 метров. Для этого требуется специальный модуль, установленный на HMI панель и на энкодерах.

В нашем проекте мы будем использовать интерфейс Ethernet, так как он обеспечивает достаточно быструю и надежную связь на большие расстояния.

## **1.2 Обзор типовых решений для управления реальными промышленными манипуляторами**

Механическая конструкция манипулятора обычно состоит из трех соединенных между собой суставов (осей), которые позволяют ему перемещаться в трехмерном пространстве. Каждый сустав может быть управляемым посредством одного или нескольких приводов.

Электроника управления манипулятором может быть выполнена на базе программируемых логических контроллеров (PLC), микроконтроллеров (MCU) или персональных компьютеров (PC). Электроника управления обычно отвечает за управление двигателями, обработку сигналов датчиков и обеспечение связи с интерфейсом управления.

Программное обеспечение управления манипулятором может быть написано на разных языках программирования, таких как C++, Python, Matlab и другие. Программное обеспечение обычно позволяет оператору задавать желаемое положение манипулятора, контролировать его движение, мониторить работу датчиков и т.д.

Датчики обратной связи обычно используются для измерения углов поворота каждого сустава, а также для измерения перемещения манипулятора в пространстве. Эти данные могут быть использованы для обеспечения точности управления и контроля за движением манипулятора.

Приводы манипулятора, как правило, могут быть осуществлены различными способами, например, сервоприводами, шаговыми двигателями или гидравлическими приводами. Каждый привод обеспечивает управление одним или несколькими суставами манипулятора.

### **1.2.1 Примеры промышленных манипуляторов с описанием используемых способов управления**

Промышленные манипуляторы используются в различных отраслях, включая производство автомобилей, электронику, пищевую и медицинскую промышленность, а также в логистике и складской логистике. В зависимости от задачи, которую необходимо решить, манипуляторы могут иметь разные конфигурации и использовать разные способы управления.

Примеры промышленных манипуляторов:

ABB IRB 1200 (рисунок 2) – многофункциональный робот-манипулятор, который может использоваться для погрузочно-разгрузочных операций, монтажа, обработки и сварки. Управление происходит с помощью промышленного контроллера ABB, который осуществляет управление силами и моментами, применяемыми в процессе работы манипулятора [7].

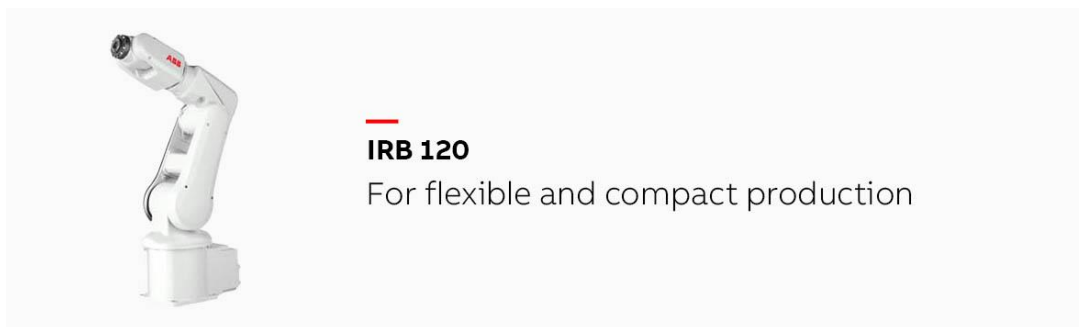


Рисунок 2 – Робот ABB IRB 1200

FANUC M-20iA – гибкий манипулятор, предназначенный для работы в ограниченном пространстве (рисунок 3). Он может выполнять операции сборки, паллетизации и обработки поверхности. Управление происходит с помощью контроллера FANUC, который обеспечивает точное и плавное управление движениями манипулятора [8].



Рисунок 3 – Робот FANUC M-20iA

KUKA KR AGILUS – компактный робот-манипулятор, который может использоваться для операций сборки, сварки, монтажа и погрузки (рисунок 4). Он обладает высокой скоростью и точностью перемещения, а также защитой от внешних воздействий. Управление происходит с помощью контроллера KUKA, который позволяет осуществлять программное управление манипулятором и его интеграцию с другими системами [9].



Рисунок 4 – KUKA KR AGILUS

Universal Robots UR10e – легкий и компактный манипулятор, предназначенный для выполнения операций сборки, погрузки и разгрузки, а также работы со сложными формами (рисунок 5). Управление происходит с помощью контроллера Universal Robots, который обеспечивает гибкое и быстрое управление манипулятором с помощью технологии Teach & Go [10].



Рисунок 5 – Робот Universal Robots UR10e

В каждом из этих примеров используется свой способ управления, который был выбран в соответствии с поставленной задачей. Однако эти роботы достаточно большие, в то время как в настоящей работе мы будем использовать робота с рабочим полем 10 см, что означает нецелесообразность подбора крупного робота, а, следовательно, и сравнения его с «большими братьями», а потому и обозреть правильнее роботы поменьше.

Mecademic Меса500 - 6 осей, рабочая нагрузка 0,5 кг, масса 4 кг, размеры 190x190x324 мм.; управление осуществляется с помощью промышленного контроллера и программного обеспечения Mecademic Robot Control Software [11].

Mecademic Меса500 - это робот-манипулятор, который использует собственную систему управления, разработанную компанией Mecademic. Она основана на реальном времени и выполнена на базе FPGA (программируемой матрицы).

Система управления Меса500 имеет высокую точность и надежность, а также обладает быстрым временем отклика и высокой скоростью перемещения

манипулятора. Она также поддерживает функции обработки сигналов и связи с внешними устройствами. Все программы для управления Meca500 создаются с использованием специализированного программного обеспечения Mecademic RoboStudio, которое позволяет легко и удобно создавать программы движения и выполнять задачи автоматизации. Mecademic Meca500 оснащен собственной электроникой, включающей в себя встроенный компьютер, контроллер двигателя, интерфейсы для коммуникации с внешними устройствами, а также систему безопасности, включающую защиту от перегрузок, обрыва цепи и т.д. Внутренняя электроника идет в комплекте с манипулятором и управляется программным обеспечением, которое также предоставляется производителем.

Franka Emika Panda - 7 осей, рабочая нагрузка 3 кг, масса 18 кг, размеры 82x82x61 мм.; управление осуществляется с помощью встроенной системы управления, которая использует технологию сенсорных обратных связей и методы машинного обучения [12].

Franka Emika Panda - это роботизированное устройство с семью степенями свободы и чувствительностью к силе, созданное для сотрудничества с людьми в общих рабочих пространствах. Система управления роботом основана на ROS (Robot Operating System) и поддерживает такие операционные системы, как Ubuntu и Windows.

Электроника робота состоит из:

- компьютера управления: высокопроизводительный процессор Intel i7, оперативная память 16 ГБ, накопитель SSD емкостью 512 ГБ;
- контроллеров двигателей: Franka Emika Panda использует 7 собственных контроллеров двигателей, каждый из которых управляет соответствующим двигателем в работе.
- интерфейс сенсоров: робот оснащен датчиками силы и момента на каждом звене, а также встроенными датчиками положения и скорости в каждом моторе;

- коммуникационный интерфейс: робот подключается к ПК через Ethernet или Wi-Fi, используя ROS для обмена сообщениями между компьютером и роботом;
- интерфейс с пользователем: для управления роботом можно использовать как программное обеспечение (например, через ROS), так и программы, созданные специально для управления Panda, такие как PandaMove или PandaROS.

Franka Emika Panda имеет открытую архитектуру и может быть дополнительно оборудован различными дополнительными датчиками и актуаторами, чтобы соответствовать специфическим потребностям пользователя.

Universal Robots UR3 - 6 осей, рабочая нагрузка 3 кг, масса 11 кг, размеры 118x118x280 мм.; управление осуществляется с помощью программного обеспечения Polyscope, которое позволяет программировать робота непосредственно на его сенсорном экране [13].

Архитектура системы управления:

Контроллер: UR3 оснащен контроллером, который управляет всей системой. Он основан на процессоре ARM и работает на базе операционной системы Linux. Контроллер обрабатывает информацию от всех датчиков и принимает решения о движении робота.

Сенсоры: UR3 имеет несколько типов датчиков, включая энкодеры, датчики тока, датчики силы и момента, датчики касания и датчики безопасности. Эти датчики используются для определения положения и ориентации робота, контроля силы и момента, а также для обеспечения безопасности взаимодействия робота с человеком.

Интерфейсы: UR3 имеет несколько интерфейсов, включая USB, Ethernet и RS-485, которые позволяют подключать дополнительное оборудование и управлять роботом удаленно.

Программное обеспечение: UR3 поставляется с программным обеспечением Polyscope, которое предоставляет пользовательский интерфейс



для управления роботом и создания программ движения. Полскоуп также позволяет программировать робота на языках Python и C++.

Epson T3 - 6 осей, рабочая нагрузка 3 кг, масса 20 кг, размеры 260x300x345 мм.; управление осуществляется с помощью программного обеспечения Epson RC+ 7.0, которое позволяет программировать робота на языке исходного кода, а также использовать графический интерфейс для создания программ [14].

Архитектура системы управления Epson T3 основана на встроенном контроллере RC700-A, который обеспечивает управление всеми функциями робота, включая движение, обработку сигналов и связь с другими устройствами. Контроллер оснащен процессором ARM, а также имеет различные входы-выходы для связи с внешними устройствами.

Электроника Epson T3 включает в себя многофункциональную плату ввода-вывода, которая обеспечивает управление цифровыми и аналоговыми сигналами, а также плату связи, которая обеспечивает связь между контроллером и другими устройствами в системе автоматизации. Также в работе используются моторы, датчики и энкодеры для обеспечения точности и надежности движения.

ABB IRB 120 - 6 осей, рабочая нагрузка 3 кг, масса 25 кг, размеры 220x220x595 мм.; управление осуществляется с помощью контроллера IRC5 Compact, который позволяет программировать робота на языке исходного кода или с использованием графического интерфейса [15].

Архитектура системы управления роботом ABB IRB 120 включает в себя контроллер IRC5 Compact, который является центральным устройством управления и связи между компонентами системы.

Контроллер IRC5 Compact оснащен микропроцессором, оперативной памятью и жестким диском, а также сетевыми интерфейсами для обмена данными с другими устройствами. Он также содержит программное обеспечение, необходимое для программирования, настройки и управления роботом.

KUKA LBR iiwa - 7 осей, рабочая нагрузка 7 кг, масса 23 кг, размеры 570x350x520 мм.; управление осуществляется с помощью контроллера KUKA Sunrise Cabinet, который позволяет программировать робота на языке исходного кода или с использованием графического интерфейса [16].

Электроника KUKA LBR iiwa состоит из вычислительного блока, контроллера и устройства ввода-вывода. Вычислительный блок использует процессор Intel Core i7 и операционную систему Linux, что позволяет выполнять сложные вычисления и обработку данных. Контроллер управляет движением манипулятора и обрабатывает данные с сенсоров. Устройство ввода-вывода предназначено для подключения внешних устройств, таких как камеры или сенсоры.

KUKA LBR iiwa также имеет встроенную систему безопасности, которая позволяет ему безопасно работать в окружении людей. Он также может использоваться с различными программными платформами, такими как ROS, MATLAB и Simulink, что облегчает его интеграцию в различные приложения.

Yamaha YK400XR - 6 осей, рабочая нагрузка 3 кг, масса 14 кг, размеры 200x200x400 мм.; управление осуществляется с помощью контроллера RCX240, который позволяет программировать робота на языке исходного кода или с использованием графического интерфейса [17].

Он оснащен контроллером YRC1000, который имеет процессор на 1,4 ГГц и 4 Гб оперативной памяти, а также различные входы/выходы, поддержку Ethernet и USB для связи с внешними устройствами.

Робот имеет максимальную нагрузку в 6 кг, рабочую область равную 562 мм по оси X, 762 мм по оси Y и 190 мм по оси Z, а также максимальную скорость перемещения до 8 000 мм/с и точность позиционирования до 0,03 мм.

Yamaha YK400XR использует технологию инвертированного редуктора, которая позволяет сократить высоту робота до 980 мм, а также улучшить точность и надежность его работы.

Из рассмотренных «маленьких» промышленных роботов можно сделать вывод, что различные способы управления используются в зависимости от

конкретной задачи и требований к роботу. Некоторые роботы, такие как Mecademic Meca500, используют динамическое управление, которое позволяет им работать с высокой точностью и скоростью. Другие роботы, такие как Universal Robots UR3, используют технологию коллаборативного управления, что делает их безопасными для работы рядом с людьми. Однако, все роботы имеют одну общую цель - улучшение производительности и эффективности в различных промышленных задачах.

### **1.2.2 Анализ рынка управления манипуляторами и тренды развития данного направления**

Рынок управления манипуляторами является довольно конкурентным и динамичным, поскольку постоянно появляются новые технологии и инновации в данной области. Разработка новых методов управления роботизированными системами и манипуляторами становится все более актуальной задачей, поскольку это позволяет улучшить эффективность работы, повысить безопасность, снизить затраты на обслуживание и улучшить управляемость манипуляторов.

Одним из ключевых трендов в области управления манипуляторами является разработка новых методов программирования и управления роботизированными системами, в том числе методов искусственного интеллекта и машинного обучения. Это позволяет улучшить гибкость и автономность манипуляторов, а также повысить качество и точность работы.

Также наблюдается увеличение популярности сенсорных систем и систем обработки данных, которые позволяют манипуляторам более точно выполнять задачи и реагировать на изменения внешней среды.

Еще одним трендом является повышение доступности и универсальности манипуляторов. Сейчас на рынке представлены различные типы и модели манипуляторов, от больших и сложных промышленных роботов до компактных и доступных манипуляторов для образовательных и научных целей.

Также стоит отметить увеличение использования мобильных и коллаборативных роботов, которые могут работать рядом с людьми и выполнять задачи совместно с ними.

В целом, рынок управления манипуляторами представляет собой динамичное направление, которое продолжает развиваться и совершенствоваться, с постоянным появлением новых методов, технологий и моделей манипуляторов.

### **1.3 Обзор решений для построения макетов манипуляторов**

Для построения макетов манипуляторов используются различные инструменты и технологии. Наиболее распространенными являются.

CAD-системы (Computer-Aided Design) - программные средства, которые позволяют проектировать и моделировать объекты в трехмерном пространстве. Для построения макетов манипуляторов могут использоваться такие CAD-системы, как SolidWorks, Autodesk Inventor, CATIA и другие.

3D-принтеры - устройства, которые создают физические объекты из цифровых моделей. Для построения макетов манипуляторов могут использоваться 3D-принтеры различных типов и технологий, таких как FDM, SLA, SLS и другие.

Робототехнические наборы - комплекты, содержащие все необходимые компоненты для сборки манипулятора, включая моторы, контроллеры, датчики и другие элементы. Такие наборы могут быть полезны для быстрой сборки прототипов и экспериментов.

Онлайн-сервисы – в последнее время появилось много онлайн-сервисов, которые позволяют создавать трехмерные модели и заказывать их печать на 3D-принтерах удаленно.

Все эти инструменты и технологии имеют свои преимущества и недостатки, и выбор того или иного подхода зависит от конкретной задачи и возможностей разработчика.

## 2 Разработка системы управления

Система динамического управления манипулятором формируется в соответствии со следующей функциональной схемой (рисунок 6):

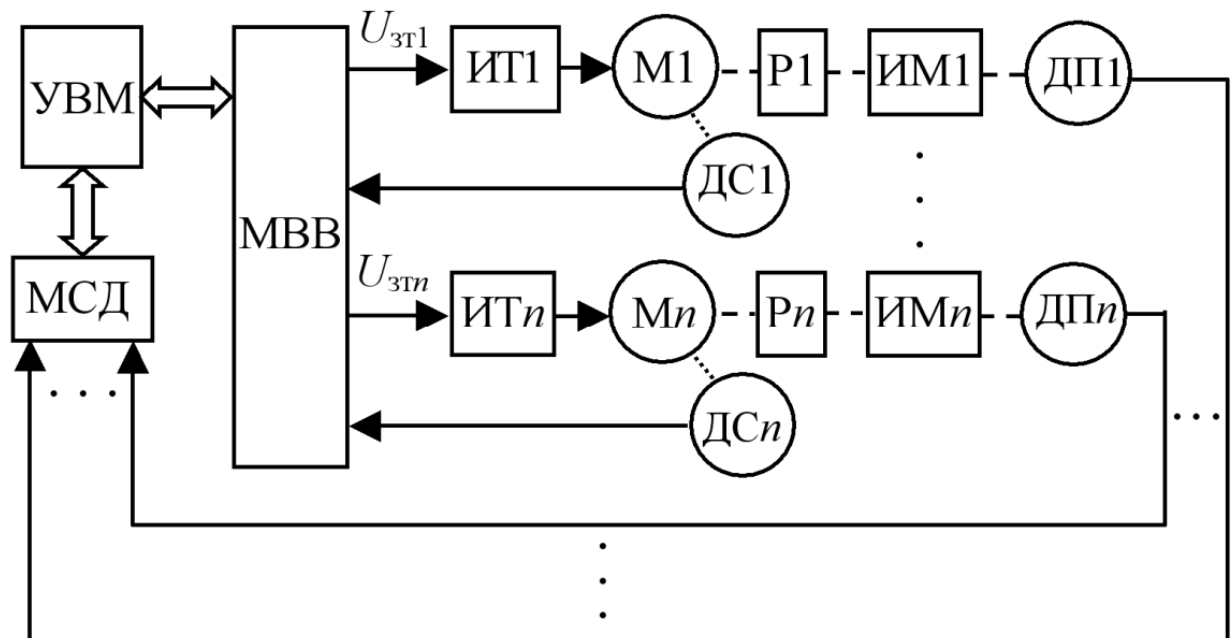


Рисунок 6 – Функциональная схема системы управления

На данном рисунке:

- УВМ – управляющая вычислительная машина;
- МСД – модуль связи с датчиками;
- МВВ – модуль ввода-вывода;
- ИТ1÷ИТn – источники тока координат робота,  $n = 1, 2, 3$ ;
- М1÷Мn – исполнительные двигатели;
- ДС1÷ДСn – датчики скоростей;
- Р1÷Рn – редукторы (при необходимости);
- ИМ1÷ИМn – исполнительные механизмы координат робота;
- ДП1÷ДПn – датчики перемещений.

Системы управления манипуляторами могут быть построены на основе различных принципов. Вот некоторые из них:

- программное управление системы управления манипуляторами, основанные на программном управлении, являются наиболее распространенным и простым вариантом. В этом случае оператор задает команды через пользовательский интерфейс, которые затем выполняются манипулятором. Программное управление обычно используется в промышленности и лабораторных условиях, где манипуляторы выполняют стандартные задачи, требующие минимальной адаптации к изменяющейся окружающей среде;
- управление с использованием сенсоров системы управления манипуляторами, использующие датчики, предоставляют манипулятору обратную связь от окружающей среды. Это позволяет манипулятору реагировать на изменения в окружающей среде и адаптироваться к ним. Например, датчики могут помочь манипулятору избегать препятствий или корректировать движение, чтобы обеспечить более точную и надежную работу;
- управление на основе моделирования системы управления манипуляторами, основанные на математическом моделировании, используют модель манипулятора для управления его движением. Оператор может точно задать требуемое движение манипулятора в соответствии с моделью. Такой принцип управления позволяет достичь более точного управления, так как он учитывает физические ограничения манипулятора и его динамику;
- гибридное управление системы управления манипуляторами, использующие гибридный подход, комбинируют несколько подходов к управлению манипулятором. Это позволяет использовать наилучшие аспекты каждого подхода и достигать наилучших результатов. Например, система может использовать управление на основе моделирования для задания общего движения манипулятора и

управление на основе сенсоров для адаптации к изменениям в окружающей среде;

- автоматическое управление: системы управления манипуляторами, полностью автоматизированные и не требующие вмешательства оператора, могут быть запрограммированы для выполнения определенных задач и процессов без необходимости присутствия мешков с костями;
- управление на основе ИИ: современные системы управления манипуляторами все чаще используют технологии искусственного интеллекта для улучшения точности управления и адаптации к изменяющейся окружающей среде. [20]

Выбор принципа управления манипулятором зависит от требований и целей конкретного случая. Например, в некоторых случаях может быть достаточно использовать простое программное управление, а в других случаях может потребоваться гибридное управление, комбинирующее несколько подходов. В нашем случае мы будем использовать гибридное управление, то есть программная составляющая, получающая данные от оператора и датчиков обратной связи.

## **2.1 Выбор модели ПЛК**

ПЛК выбран в соответствии со следующими исходными данными:

- используемый ПЛК: серия Siemens 1200;
- выходное напряжение +24В;
- количество осей управления: 3
- наличие конечных выключателей на каждой оси, шт: 1
- напряжение питания управляемых двигателей на осях перемещения, В: 9.

SIMATIC S7-1200 - это новое семейство микроконтроллеров, разработанных компанией Siemens для решения различных задач

автоматизации на низком уровне. Эти контроллеры имеют модульную конструкцию и предназначены для широкого спектра применений. Они способны работать в реальном времени и поддерживают обмен данными через сети Industrial Ethernet/PROFINET и PtP (точка-точка) соединения.

Программируемые контроллеры S7-1200 имеют компактные пластиковые корпуса с защитой IP20 и могут быть установлены на стандартную 35-миллиметровую профильную шину DIN или монтажную плату. Они работают в широком диапазоне температур от 0 до +50 °С. Контроллеры могут обрабатывать от 10 до 284 дискретных и от 2 до 51 аналоговых каналов ввода-вывода. При сравнимых конфигурациях ввода-вывода с предыдущей моделью S7-200, контроллеры S7-1200 занимают на 35% меньше монтажного объема.

Центральный процессор (CPU) программного контроллера S7-1200 может быть расширен с помощью коммуникационных модулей (CM), сигнальных модулей (SM) и сигнальных плат (SB) для подключения дискретных и аналоговых сигналов. Кроме того, вместе с ними используются 4-канальный коммутатор Industrial Ethernet (CSM 1277) и модуль блока питания (PM 1207) [7]. На рисунке 7 показан Siemens S1200.



Рисунок 7 – Siemens S1200



Все центральные процессоры обладают высокой производительностью и обеспечивают поддержку широкого набора функций:

- программирование на языках LAD и FBD, исчерпывающий набор команд;
- высокое быстродействие, время выполнения логической операции не превышает 0.1 мкс;
- встроенная загружаемая память объемом до 2 Мбайт, расширяемая картой памяти емкостью до 24 Мбайт;
- рабочая память емкостью до 50 Кбайт;
- энергонезависимая память емкостью 2 Кбайт для необслуживаемого сохранения данных при перебоях в питании контроллера;
- встроенные дискретные входы универсального назначения, позволяющие вводить потенциальные или импульсные сигналы.

## **2.2 Определение и выбор необходимых датчиков**

Руководствуясь приведёнными в техническом задании исходными данными по двигателям и обратной связи, можно заключить, что выбор датчиков сводится к выбору концевых выключателей для останова работы двигателей на осях.

Концевой выключатель - это электромеханический аппарат для цепей управления, приводимый в действие непосредственным механическим воздействием механизма или части машины.

Представляет собой устройство механизированного управления путём изменения электрических соединений между своими внутренними выводами. Примеры концевых выключателей приведены на рисунке 8.



Рисунок 8 – Концевые выключатели

Информация о датчиках обратной связи для двигателей приведена в следующем пункте, так как датчики встроены непосредственно в саму конструкцию двигателей.

### 2.3 Разработка драйверов питания двигателей

В техническом задании по исполнительным механизмам приведены следующие исходные данные:

- количество осей управления: 3;
- наличие концевых выключателей на каждой оси, шт: 1;
- напряжение питания управляемых двигателей на осях перемещения, В: 9;
- встроенный в двигатели энкодер: импульсный, напряжение питания энкодера 9В, выход энкодера – открытый коллектор;
- максимальный потребляемый двигателями осей ток: 1А.

Выход с открытым коллектором (Open Collector) предоставляет возможность получать сигнал с уровнем напряжения, определяемым не напряжением питания энкодера, а напряжением дополнительного источника питания. Для этого необходимо подключить внешний резистор между цепями

питания ( $U_{пит}$ ) и выходом. Значение сопротивления внешнего резистора зависит от величины напряжения дополнительного источника питания. Тип выхода с открытым коллектором также позволяет подключать несколько энкодеров к одному счётному входу (рисунок 9).

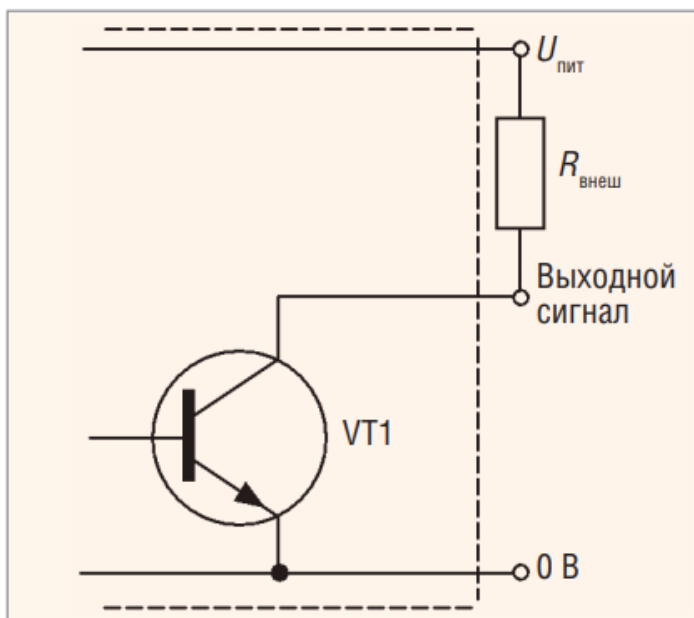


Рисунок 9 – Тип выхода с открытым коллектором

Основная проблема при подключении к программируемому логическому контроллеру (ПЛК) заключается в согласовании частоты, с которой энкодер выдает импульсы ( $f_1$ ), и частоты опроса входов контроллера ( $f_2$ ). Частота опроса ( $f_2$ ) должна быть выше частоты импульсов ( $f_1$ ) и достаточно высокой для корректного подсчета импульсов. Значение частоты  $f_2$  ограничено длительностью рабочего цикла ПЛК, который включает фазы чтения входов, выполнения программы пользователя и записи значений выходов. Время исполнения программы пользователя зависит от содержания программы и вносит наибольший вклад в общее время рабочего цикла ПЛК. Если не учитывать специализированные энкодеры с повышенной точностью, то частота импульсов, выдаваемых фотоэлектрическими инкрементными энкодерами, может варьироваться от 100 до 2500 импульсов и более за один оборот [5].

Подсчет скорости вращения электродвигателя может потребовать обработки нескольких десятков тысяч оборотов в секунду, что превышает типичное время рабочего цикла ПЛК. Поэтому использование стандартных функциональных блоков счётчиков внутри программы пользователя может быть затруднительным или нарушать рабочий цикл ПЛК.

Для решения этой проблемы можно использовать блоки высокоскоростных счётчиков (HSC), которые позволяют счетчику работать независимо от основной программы и контролировать его выходное значение при помощи специальной подпрограммы обработки прерываний. Например, ПЛК Siemens S7-1200 позволяют использовать до 6 таких счетчиков.

Пример программного блока CTUD показан на рисунке 10.

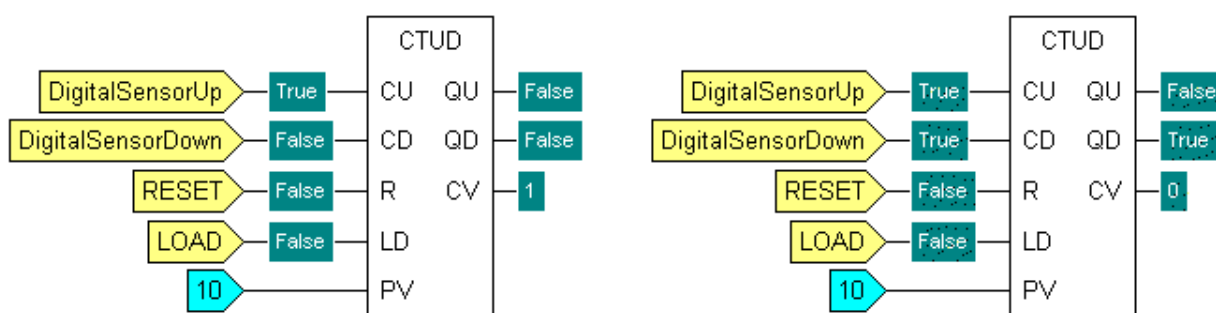


Рисунок 10 – Пример программного блока CTUD

Для решения этой проблемы можно использовать блоки высокоскоростных счётчиков (HSC), которые позволяют счетчику работать независимо от основной программы и контролировать его выходное значение при помощи специальной подпрограммы обработки прерываний. Например, ПЛК Siemens S7-1200 позволяют использовать до 6 таких счетчиков. [6]

При самостоятельной разработке важно выбрать компоненты, подходящие по току, напряжению и прочим параметрам, а также соответствующие типу двигателей. Проект платы можно выбрать из изобилия

материалов в открытом доступе, что может уберечь от множества проблем, связанных с собственной разработкой. Пример платы управления двигателем приведён на рисунке 11.

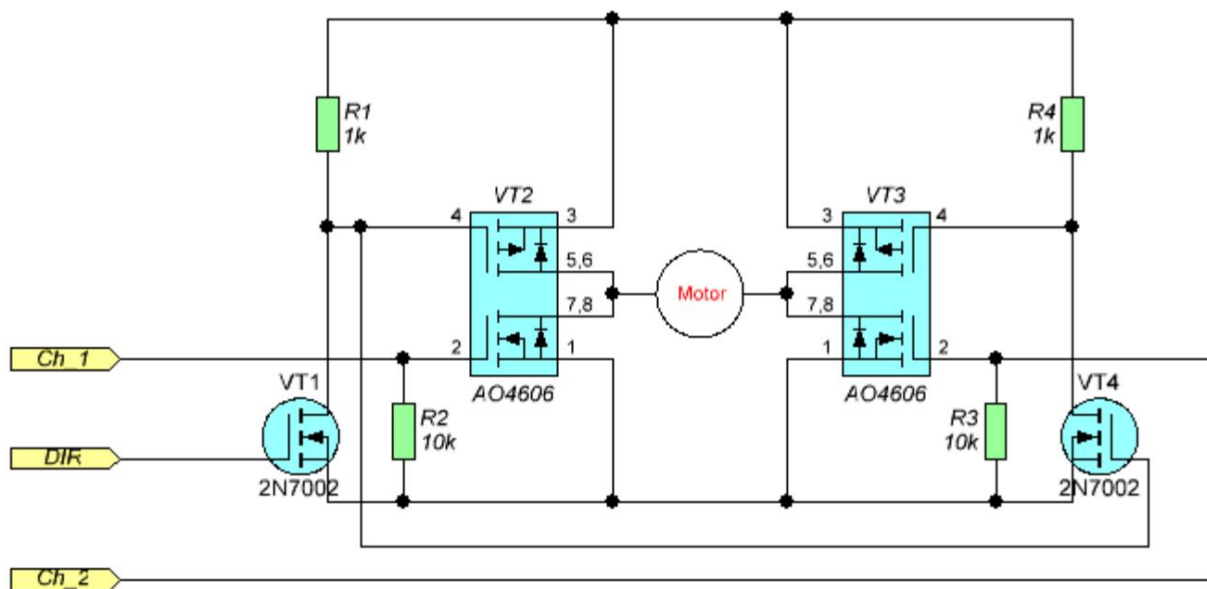


Рисунок 11 – Плата драйвера управления двигателем на МОП-транзисторах

В данной схеме отсутствует защита от сквозного тока, поэтому такую защиту придется предусматривать программно.

Либо можно рассмотреть готовые драйверы двигателей, предварительно подобрав их по характеристикам.

Так же часто используемым решением для управления подобными двигателями является драйвер 1298N (Рисунок 12).

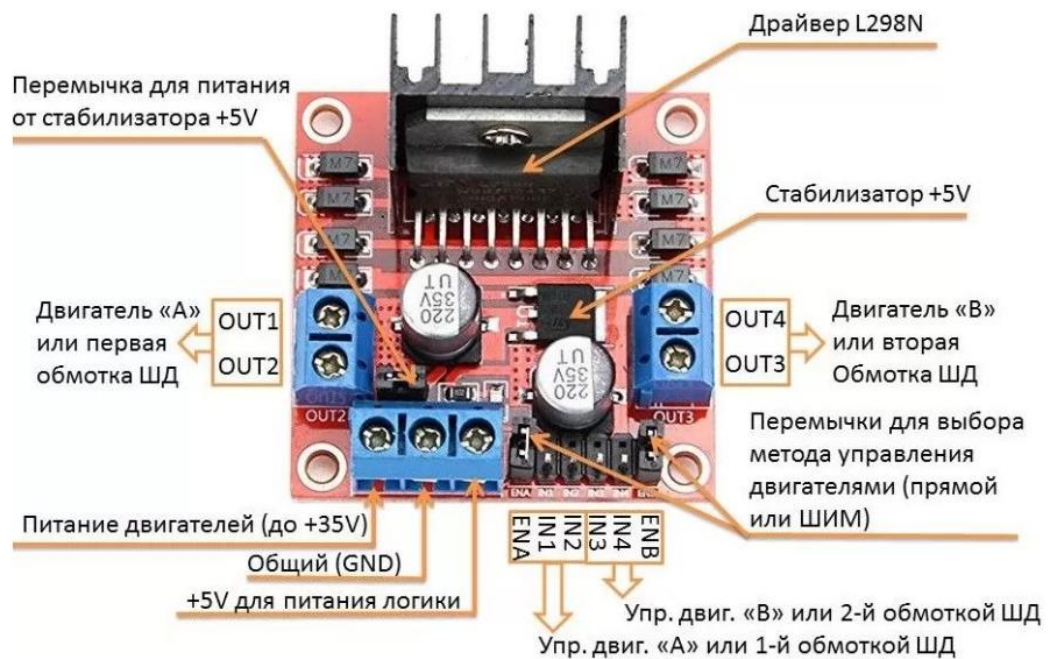


Рисунок 12 – Драйвер l298N

- OUT1 и OUT2 – разъёмы для подключения «первого щётчного двигателя или первой обмотки шагового двигателя»;
- OUT3 и OUT4 – разъёмы для подключения второго щётчного двигателя или второй обмотки шагового двигателя;
- VSS – вход для питания двигателей (максимальный уровень +35V);
- GND – общий провод;
- Vs – вход для питания логики +5V. Через него непосредственно запитывается сама микросхема L298N. Есть ещё второй способ питания, при котором 5V для L298N берётся от встроенного в модуль стабилизатора напряжения. В таком случае на разъём подаётся только питание для двигателей (Vss), контакт Vs остаётся не подключенным, а на плате устанавливается переключка питания от стабилизатора, который ограничит питающее моторы напряжение до приемлемых 5V.
- IN1, IN2 – контакты управления первым щётчным двигателем или первой обмоткой шагового двигателя.

- IN3, IN4 – контакты управления вторым щётчным двигателем или второй обмоткой шагового двигателя.
- ENA, ENB – контакты для активации/деактивации первого и второго двигателей или соответствующих обмоток ШД. Подача логической единицы на эти контакты разрешает вращение двигателей, а логический ноль – запрещает. Для изменения скорости вращения щётчных моторов на эти контакты подаётся ШИМ-сигнал. Для работы с шаговым двигателем, как правило, на эти контакты ставят переключки, обеспечивающие постоянную подтяжку к +5V.» [21]

Итого, для управления 3 двигателями постоянного тока мы можем использовать 2 драйвера L298N в режиме H-моста, подключив питание двигателей на выводы OUT1,2 или OUT3,4, а выводы энкодера подключить напрямую к выводам контроллера.

На рисунке 13 показана электрическая схема модуля L298N.

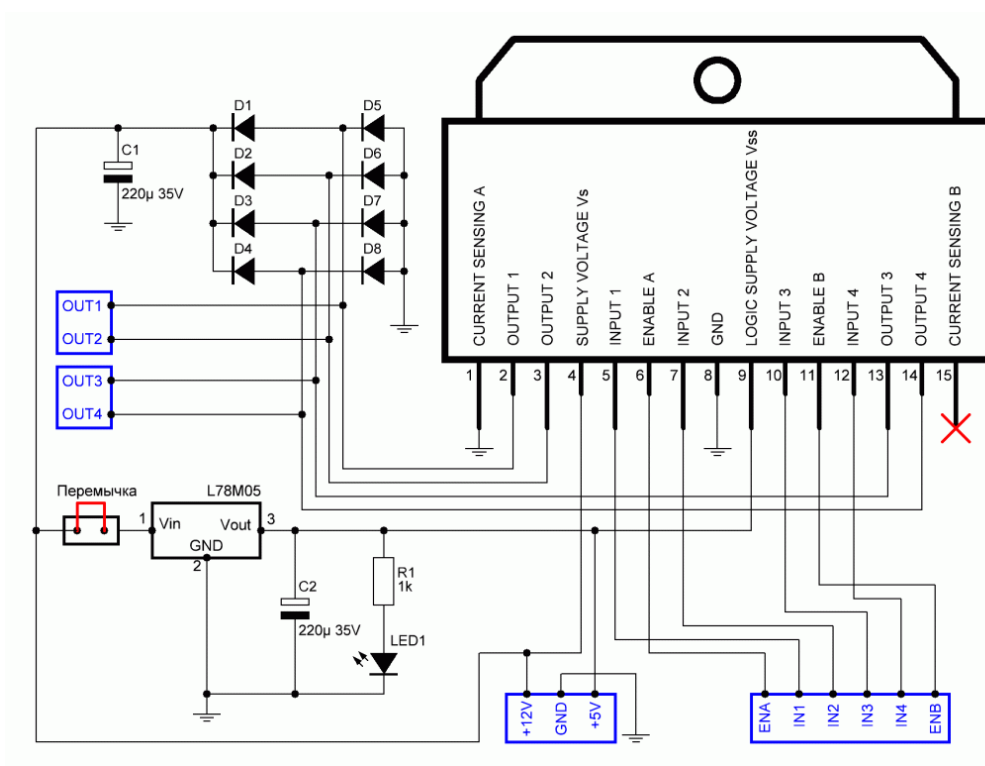


Рисунок 13 – Электрическая схема модуля L298N

Модуль управления манипулятором использует микросхему L298N, которая содержит два полноценных Н-моста. Каждый Н-мост состоит из четырех транзисторных ключей и обмотки двигателя в центре. Это позволяет менять полярность в обмотке и направление вращения двигателя, чередуя пары открытых и закрытых ключей. Этот процесс проиллюстрирован на рисунке 14.

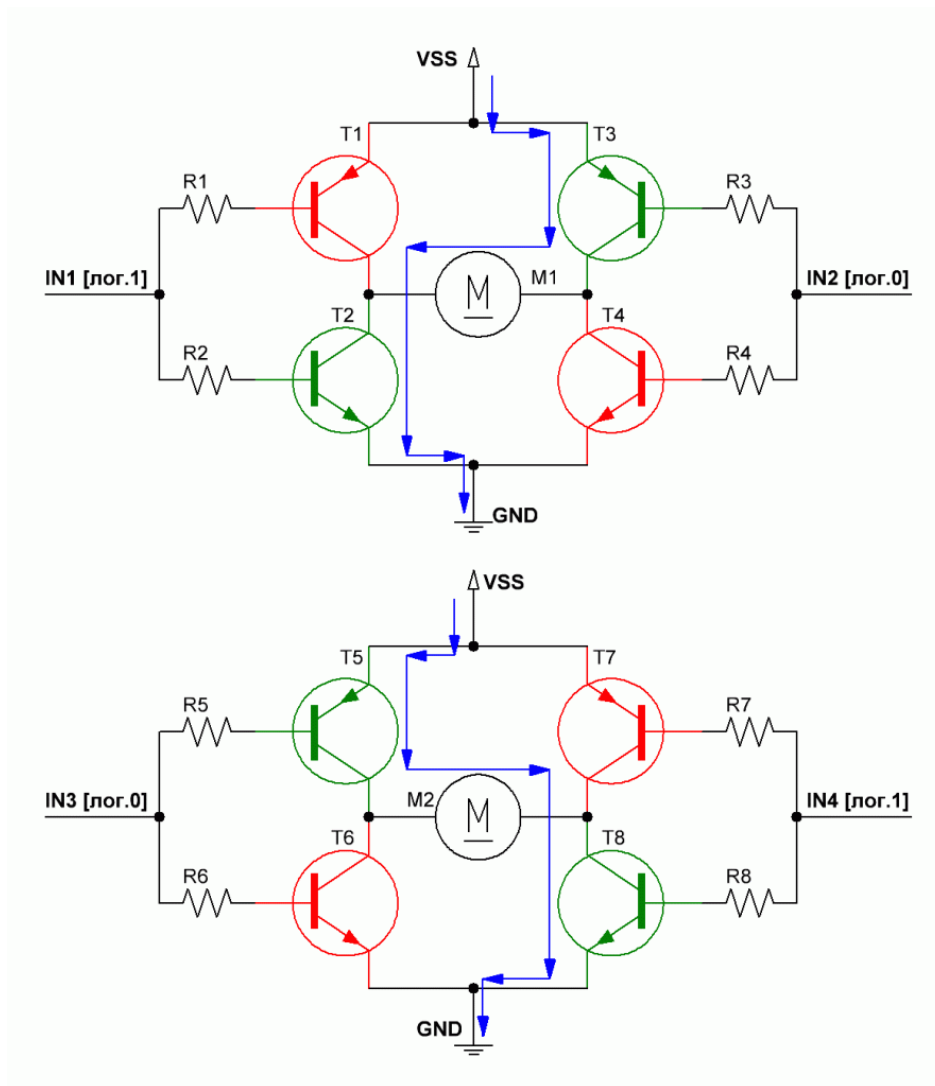


Рисунок 14 – Транзисторные мосты Н-типа

На данном рисунке представлены две схемы транзисторных мостов Н-типа. В первом случае на один из входов (IN1) поступает логическая единица, а на другой вход (IN2) - логический ноль. Так как транзисторы имеют разную проводимость, при таком входном сигнале транзисторы T1 и T4 останутся закрытыми, а через T2 и T3 потечет ток, проходящий через обмотку двигателя



и приводящий его в движение в определенном направлении. На второй схеме вход IN3 установлен в логический ноль, а вход IN4 - в логическую единицу, что приводит к обратному направлению тока и, как следствие, вращению второго двигателя в противоположную сторону.

## **2.4 Схема системы управления манипулятором**

Система управления манипулятором состоит из нескольких основных компонентов, которые описаны ниже.

Программируемый логический контроллер (ПЛК) - это центральный элемент системы управления, который обрабатывает информацию, полученную от датчиков и других устройств, и отправляет сигналы на исполнительные механизмы. ПЛК часто используется в промышленных системах управления, так как он обеспечивает надежную и точную работу.

Человеко-машинный интерфейс (HMI): Это устройство, обеспечивающее взаимодействие оператора с системой управления манипулятором. HMI предоставляет пользователю удобный интерфейс для ввода команд, мониторинга и отображения информации о состоянии манипулятора, а также для отслеживания процессов и выполнения настроек.

Сетевой коммутатор (Switch) - это устройство, которое обеспечивает связь между компонентами системы управления, включая ПЛК, драйверы и датчики. Он позволяет передавать данные между устройствами в реальном времени и обеспечивает стабильную и надежную связь.

Блок питания (БП) - это устройство, которое обеспечивает питание всей системы управления и исполнительных механизмов. Он должен быть достаточно мощным, чтобы обеспечить достаточную мощность для всех устройств, подключенных к системе управления.

Драйверы - это устройства, которые управляют двигателями манипулятора. Они преобразуют сигналы от ПЛК в управляющие сигналы для двигателей, чтобы они могли перемещаться в нужном направлении и с нужной

скоростью. Двигатели - это исполнительные механизмы манипулятора, которые обеспечивают движение и манипуляцию объектами. Двигатели должны быть достаточно мощными, чтобы обеспечивать требуемую нагрузку и скорость.

Датчики ОС - это устройства, которые измеряют положение, скорость, ускорение и другие параметры движения манипулятора. Они отправляют сигналы на ПЛК, чтобы обеспечить точное управление движением.

Соединительные кабели - это кабели, которые соединяют компоненты системы управления, включая ПЛК, драйверы и датчики.

Структурно систему управления можно увидеть на рисунке 15.

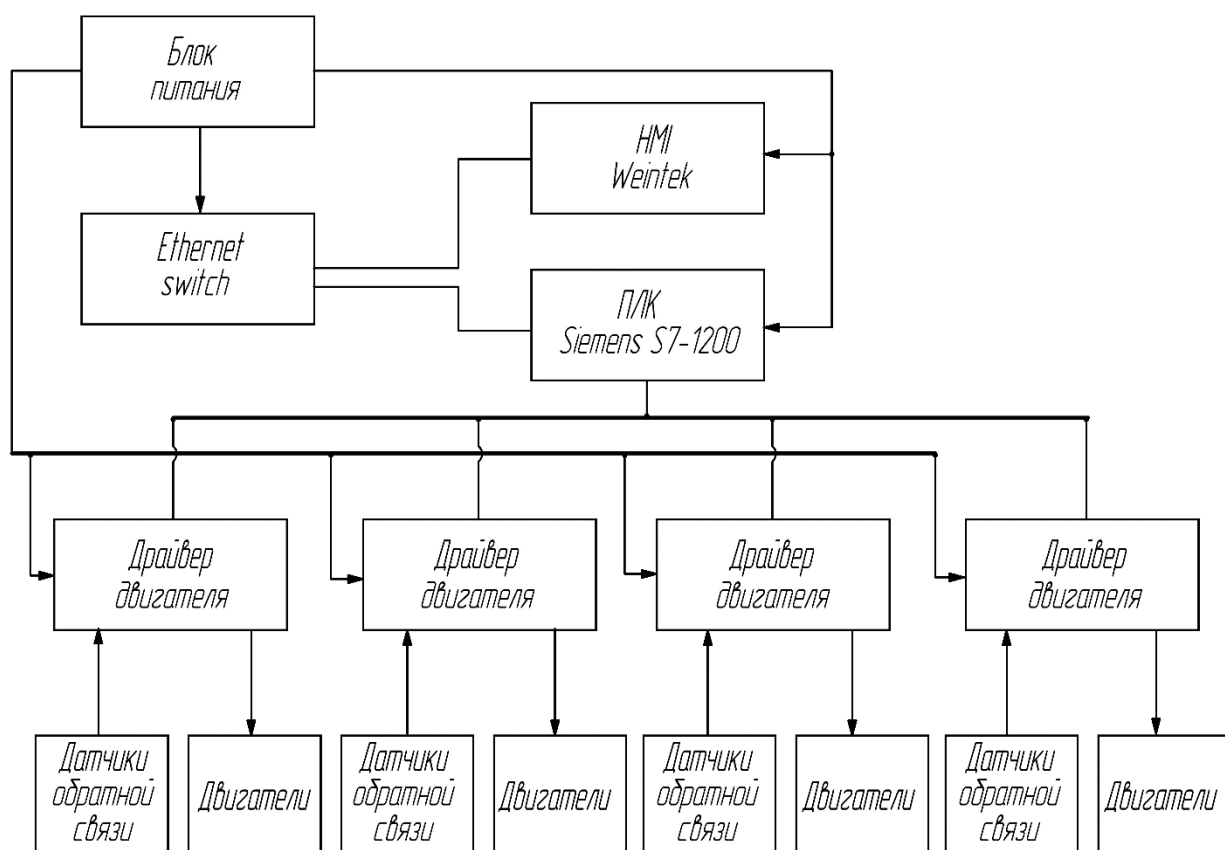


Рисунок 15 – Структурная организация архитектуры системы

Таким образом рассмотрена структурная организация архитектуры системы.

### 3 Разработка управляющей программы для ПЛК

В настоящей работе используется программируемый логический контроллер S7-1200.

#### 3.1 Контроллер S7-1200

Программируемые контроллеры S7-1200 предназначены для решения логических задач, выполнения математических операций и создания систем автоматизации. Они могут использоваться в качестве замены релейно-контактных систем и для построения сложных автоматизированных структур. На рисунке 16 показан пример программируемого логического контроллера S7-1200.



Рисунок 16 – внешний вид программируемого котроллера S7-1200

Контроллеры Siemens S7-1200 предлагают разные версии центральных процессоров, которые выбираются в зависимости от потребностей разработчика. Различные процессоры, такие как CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C и другие, могут быть установлены на базе контроллера S7-

1200. Они отличаются по производительности, объему памяти, а также количеству встроенных аналоговых и дискретных входов и выходов. В нашем случае мы будем использовать контроллер с CPU 1212C, так как эта версия уже имеет необходимое количество входов и выходов для подключения концевиков. Использовать мы будем версию с релейными выходами, так как нам необходимо снимать сигналы с концевых выключателей, а туда их надо подавать непрерывно с выходов контроллера (или с питания, или с соседних входов), а реле позволяют делать это наиболее удобным и малозатратным способом. Кат.номер контроллера 6ES7 212-1BE40-0XB0.

Концевые выключатели, в количестве 3 штук, согласно техническому заданию, можно подключить напрямую к выводам контроллера. Таким образом, концевик будет выполнять роль переключки между цифровым выходом и цифровым входом на контроллере, в случае, когда он нажат.

Так как для подключения 2 драйверов двигателей нам необходимо как минимум 12 дискретных выходов и 4 дискретных входов, нам необходим дополнительный сигнальный модуль ввода/вывода – SM1222. Этот модуль имеет на борту 16 выходов с замыкающими контактами реле: постоянного напряжения: 5...30 В, переменного напряжения: 5 ... 250 В, ток - 2 А, итого: 4 группы по 4 выхода.

Сигнальные модули устанавливаются справа от центрального процессора CPU 1212C, CPU 1214C, CPU 1215C или CPU 1217C. Центральный процессор CPU 1211C интерфейса подключения сигнальных модулей не имеет

Таким образом, имея релейные выходы 5 В, данный модуль способен управлять драйверами L298N. Его кат.номер будет отражён в спецификации: 6ES7 222-1NN32-0XB0 [21]. Данный модуль представлен на рисунке 17.

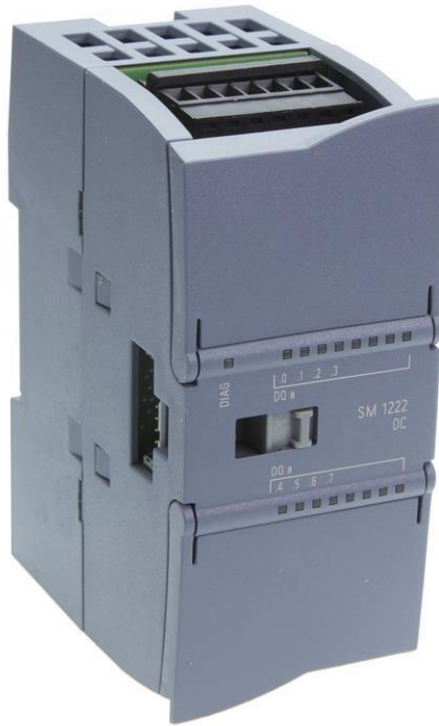


Рисунок 17 – Дополнительный модуль ввода/вывода  
6ES7 222-1BH32-0XB0

Общение контроллеров с внешним миром происходит через коммуникационные модули (Рисунок 18).



Рисунок 18 – Коммуникационный модуль

Для написания программы для Siemens S7-1200 нужно использовать специализированное программное обеспечение TIA Portal.

Шаги для написания программы:

- создать проект и выбрать необходимый тип контроллера Siemens S7-1200;
- добавить необходимые блоки данных (DB) и программные блоки (OB, FC, FB);
- написать логику программы на языке программирования Ladder, FBD, SCL, STL или GRAPH;
- привязать адреса блоков данных и переменных к графическим элементам пользовательского интерфейса;
- отладить программу в режиме симуляции, проверив ее работу без реального оборудования;
- загрузить программу на контроллер через Ethernet-соединение.

Важно учитывать особенности работы Siemens S7-1200, такие как ограничения на использование памяти, принципы адресации и коммуникации с устройствами в сети.

В нашем проекте будет целесообразнее осуществлять управление, экспортируя данные по датчикам обратной связи в панель НМІ, и импортируя оттуда управляющие воздействия, чтобы не писать программу управления дважды на разных типах контроллеров, а просто связав их в единую систему.

Для того, чтобы управлять двигателями при помощи НМІ через ПЛК, необходимо разметить адресную область, а также прописать программные теги и блоки данных. Для выделения адресного пространства необходимо определиться с количеством управляющих и информационных воздействий на объект управления.

Исходя из исходных данных, из элементов управления мы имеем:

- 3 двигателя постоянного тока, подключенные через драйвер L298N, 3 вывода в каждом;
- 3 энкодера, совмещенных с двигателями, 1 вывод в каждом;
- 3 концевых выключателя.

Итого мы имеем 9 выводов подключения типа DO и 6 выводов подключения типа DI. На рисунке 19 представлена схема подключения внешних цепей к ПЛК.

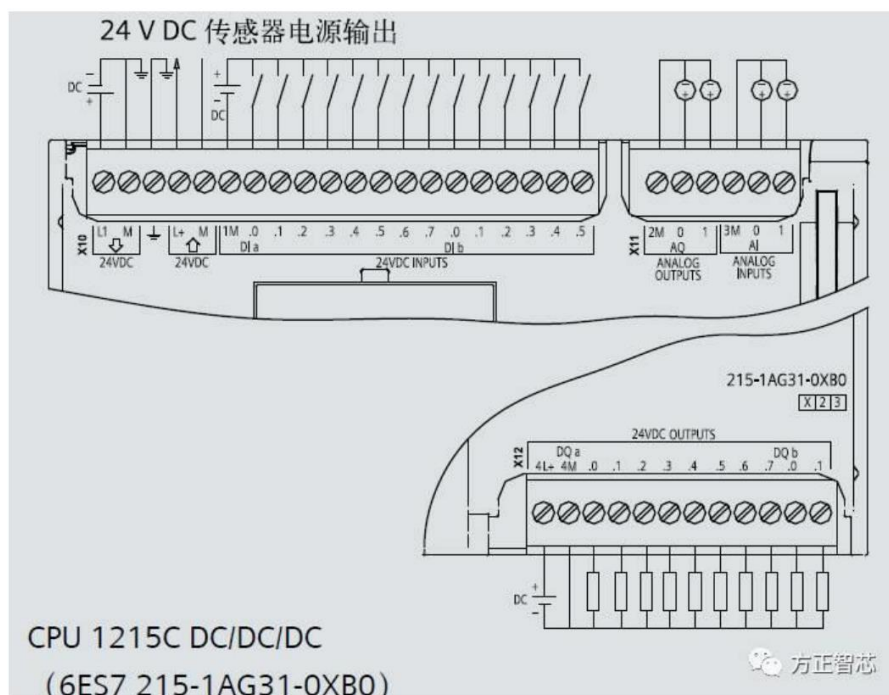


Рисунок 19 – Схема подключения внешних цепей к ПЛК

Обозначим адреса данных и занесём сигнальные линии и соответствующие им теги в таблицу 1.

Таблица 1 – Таблица локальных тегов для сигналов управления

Тип сигнала	Тег
Выходной сигнал на двигатель 1	Q0.1
Выходной сигнал на двигатель 2	Q0.2
Выходной сигнал на двигатель 3	Q0.3
Входной сигнал от энкодера 1 двигателя	I0.1
Входной сигнал от энкодера 4 двигателя	I0.2
Входной сигнал от энкодера 3 двигателя	I0.3
Входной сигнал от концевика на 1 оси манипулятора	I0.4
Входной сигнал от концевика на 2 оси манипулятора	I0.5
Входной сигнал от концевика на 3 оси манипулятора	I0.6

Таким образом рассмотрена локальных тегов для сигналов управления.

### 3.2 Питание схемы

Питание схемы производится двумя разными блоками питания.



Блок питания Siemens Sitop \_\_\_\_\_ используется для питания контроллера S7-1200 1212C и всех модулей, которые присоединены к контроллеру. Этот блок питания имеет кат.номер и показан на рисунке 20.



Рисунок 20 – Блок питания PM1207 =24В

Ниже приведен его обзор и характеристики:

- стабилизированный блок питания для программируемых контроллеров SIMATIC S7-1200.
- компактный пластиковый корпус формата модулей S7-1200 шириной 70 мм.
- входное напряжение  $\sim 120/230$  В с автоматической настройкой на уровень входного напряжения.
- выходное напряжение =24 В, номинальный ток нагрузки 2.5 А.
- защита от коротких замыканий в цепи нагрузки.

Это блок питания, рекомендуемый компанией Siemens для успешного функционирования всей системы.

Модуль РМ 1207 не имеет интерфейса для подключения к внутренней шине контроллера, поэтому он должен монтироваться в крайней левой или крайней правой позиции по отношению к модулям контроллера.

Блок питания HZY-009 используется в схеме для питания двигателей постоянного тока напряжением 9В, и для питания драйверов для этих двигателей. Используя переключку (о ней говорилось ранее), мы можем питать драйверы L298N напряжением от 5 до 35 Вольт. Ниже на рисунке 21 показан регулируемый блок питания HZY-009.



Рисунок 21 – Блок питания HZY-009

На рисунке 21 был рассмотрен блок питания HZY-009.

## **4 Разработка управляющего проекта для HMI панели**

Панели Weintek поддерживают следующие интерфейсы:

- RS-232;
- RS-485 (2-проводной, 4-проводной);
- CAN bus (некоторые модели, например, сMT3072X, сMT3092X, сMT3102X и т.д.);
- USB Host(используется для подключения, например, считывателя штрих-кодов);
- Ethernet.

Список поддерживаемых протоколов у панелей Weintek весьма большой (можно скачать описания на сайте Weintek). Общение панели с внешним миром осуществляется через интерфейс Ethernet.

### **4.1 Написание программы для HMI Weintek**

Для создания управляющей программы используется программное обеспечение EasyBuilder Pro, которое можно скачать с официального сайта производителя.

Распишем краткое руководство по действиям, необходимым для создания программы в среде EasyBuilder Pro по шагам:

- создать новый проект в EasyBuilder Pro и выбрать модель панели (рисунок 22).

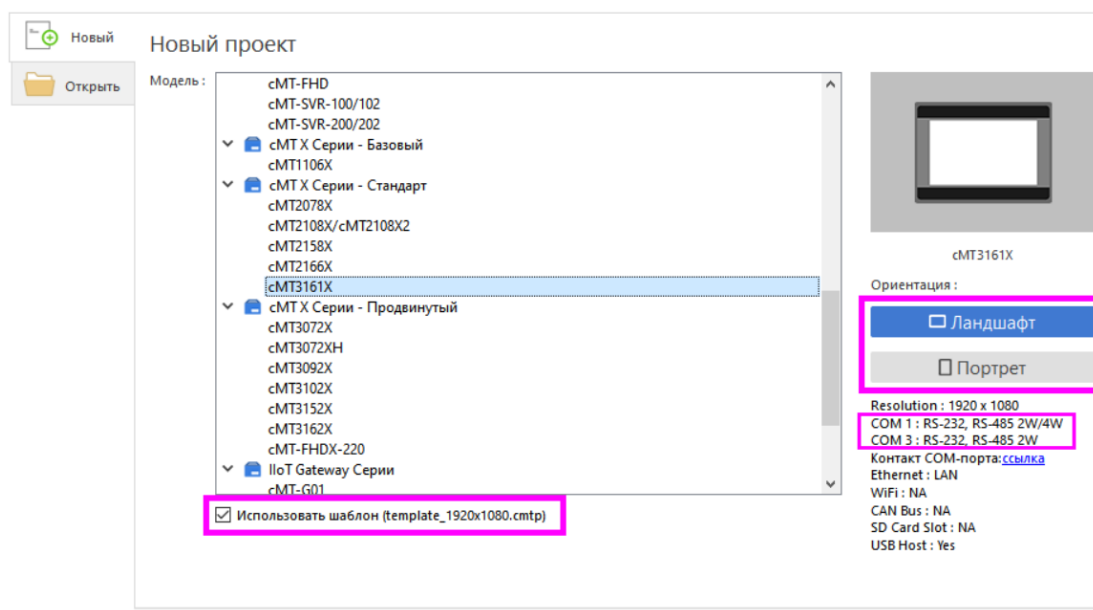


Рисунок 22 – Создание проекта

- добавить различные графические элементы на экран, такие как кнопки, переключатели, текстовые поля и графики. Это позволяет создать удобный и интуитивно понятный интерфейс для пользователей;
- настроить свойства элементов, такие как цвет, размер, положение, текст и т.д. Это помогает сделать интерфейс более привлекательным и удобным для использования;
- привязать элементы к адресам блоков данных (DB) ПЛК. Для этого надо создать теги в EasyBuilder Pro и связать их с адресами блоков данных ПЛК. Это позволяет читать и записывать значения нашего ПЛК непосредственно через панель оператора Weintek;
- написать логику программы с помощью встроенного языка программирования, называемого "Script". Язык "Script" очень похож на язык программирования C и позволяет выполнять различные операции, такие как чтение и запись значений, обработка данных и т.д. Мы использовали язык "Script" для написания обработчиков событий

и другой логики, которая позволяет управлять нашим ПЛК с помощью панели оператора Weintek.

- сохранить программу на панели оператора Weintek и загрузить ее на панель через Ethernet или USB-порт. Мы выбрали Ethernet-подключение, так как оно было более удобным для нас.
- проверить работу программы на нашей панели оператора и внести необходимые изменения в программу, если это было необходимо.

## **4.2 Быстрое спаривание Weintek и ПЛК Siemens**

Для добавления программно-логического контроллера (ПЛК) с абсолютной адресацией в систему, необходимо выбрать соответствующий драйвер Siemens S7-1200/Siemens S7-1500 (Absolute Addressing) (Ethernet) от компании Siemens AG и указать IP-адрес устройства и номер порта. После этого можно начать привязывать адреса блоков данных (DB) к графическим объектам с помощью библиотеки адресных меток или импортировать теги из самого ПЛК. При импорте тегов из TIA Portal важно отключить оптимизированный доступ к используемым блокам и разрешить доступ с помощью сообщений PUT/GET от удаленного устройства.

Если используется символьная адресация, то необходимо выбрать драйвер Siemens S7-1200 (Symbolic Addressing) (Ethernet) от компании Siemens AG. В этом случае также можно импортировать теги из ПЛК, но вместо создания адресных меток, теги будут привязываться напрямую к объектам. При выборке данных в символьной адресации необходимо создавать массив.

При первом входе и создании проекта, программа предлагает выбрать модель, для которой создается проект [8].

После выбора панели для создания проекта появляется окно системных настроек, где можно выбрать к чему будет подключена панель и каким способом. Для подключения к Siemens S7-1200 можно использовать два способа: абсолютную и символьную адресацию.

### 4.3 Добавление устройства Modbus TCP

Для добавления устройства, которое может обмениваться данными с панелью управления по протоколу Modbus TCP через интерфейс Ethernet, необходимо выполнить следующие шаги:

- откройте окно "Системные настройки" на панели управления;
- нажмите на кнопку "Новое устройство/сервер";
- в группе драйверов "Modbus IDA" выберите драйвер "Modbus TCP";
- продолжите настройку параметров устройства, таких как IP-адрес и номер порта, чтобы установить соединение с устройством, поддерживающим Modbus TCP.

Таким образом, после добавления устройства с выбранным драйвером Modbus TCP, панель управления будет готова к обмену данными с этим устройством по указанному протоколу и интерфейсу Ethernet.

Говоря про Modbus, нельзя обойти стороной такие понятия как «регистры Modbus», «функции Modbus» (Таблица 2).

Таблица 2 – Регистры Modbus

Bit/Word	Device type	Format	Range	Memo
B	0x	DDDDD	1 ~ 65535	Output bit
B	1x	DDDDD	1 ~ 65535	Input bit (read only)
B	0x_single_Bit	DDDDD	1 ~ 65535	
B	1x_single_Bit	DDDDD	1 ~ 65535	
B	3x_bit	DDDDDdd	100 ~ 6553515	Input Register bit(read only)
B	4x_bit	DDDDDdd	100 ~ 6553515	Output Register bit
B	6x_bit	DDDDDdd	100 ~ 6553515	Output Register bit
B	0x_multi_coils	DDDDD	1 ~ 65535	Write multiple coils
W	3x	DDDDD	1 ~ 65535	Input Register
W	4x	DDDDD	1 ~ 65535	Output Register
DW	5x	DDDDD	1 ~ 65535	4x double word swap
W	6x	DDDDD	1 ~ 65535	4x single word write

Самое полное описание драйвера содержится в соответствующем PDF файле на официальном сайте либо в справочном элементе EasyBuilder Pro.

Настройка соединения между устройством-клиентом (PLC Siemens S7-1200) и управляющими командами производится в среде разработки TIA Portal V15.1. Для написания программного кода, отвечающего за протокол связи и систему управления, используется язык программирования Ladder (язык лестничной логики) [26].

Основные команды, относящиеся к режиму работы промышленного протокола связи Modbus TCP/IP, включают следующие:

- MB\_MODE: содержит информацию о режиме работы протокола Modbus;
- MB\_DATA\_ADDR: указывает начальный адрес памяти, где происходит поиск информационного сообщения;
- MB\_DATA\_LEN: определяет длину информации, которую необходимо обработать устройству-клиенту [25].

Эти команды позволяют настроить соединение и управление данными между устройством-клиентом и другими устройствами, использующими протокол Modbus TCP/IP.

Для осуществления чтения и записи данных в блоке "Чтение/запись" необходимо выбрать устройство "Modbus TCP/IP" из выпадающего списка и указать адрес соответствующего типа регистра, согласно документации на ПЛК. Если используется драйвер MODBUS RTU и на линии несколько устройств, нужно указать адрес станции в поле номера регистра или через системный регистр. Важно учитывать, что один регистр Modbus занимает 16 бит, что соответствует одному машинному слову (WORD) и его формат может быть различным (например, 16 без знака, 16 бит со знаком). Чтобы определить формат передаваемых данных, необходимо в окне свойств объекта "Числовой" на вкладке "Формат данных" выбрать соответствующий формат из выпадающего списка.

Если в формате данных выбрать 32 битный формат, то для передачи переменных типа float, следует считать 2 регистра подряд на стороне ПЛК. Это более удобно, чем конфигурация переменных в соседние регистры с учетом адресации.

Однако, работать с адресами регистров может быть неудобно, особенно, когда их больше 4-5. В этом случае сложно запомнить, за каким номером закреплен какой технологический параметр, особенно, если используется 32 битные переменные и нужно учитывать адресацию +1 регистр для следующей переменной. Чтобы облегчить этот процесс, EasyBuilder Pro предоставляет библиотеку адресных меток, которые позволяют использовать символьные имена переменных. Теги, привязанные в проекте, будут отмечены красным шрифтом в Библиотеке адресных меток.

Для ПЛК S7-1200/S7-1500 можно выделить два варианта адресации:

- абсолютная (блоки DB);
- символьная (по именам тегов).

#### **4.3.1 Драйвер ПЛК с абсолютной адресацией**

Формально уже можно переходить к окну и привязывать адреса тегов м DB-блоков к графическим объектам. Однако это весьма неудобно (как и работать с регистрами Modbus).

Можно создать адресные метки в Библиотеке адресных меток.

Можно импортировать теги из проекта или из окнтроллера по сети (если в данный момент есть доступ по сети к ПЛК). После добавления устройства Siemens становятся доступны 2 кнопки для импорта тегов:

- импорт тегов (из проекта, см. описание драйвера в PDF [9]);
- импорт тегов прямо из ПЛК («Получить информацию о тегах»).

Теги в проекте в TIA Protal располагаются в двух местах:

- PLC tags (так называемые "плоские" теги);
- Program blocks (блоки DB).

Импорт PLC tags происходит посредством их экспорта в файл \*.XLSX.



Импорт тегов для ПЛК Siemens в проекте EasyBuilder Pro из файлов предлагает два варианта:

- импорт из \*.scl, \*.db, \*.xlsx файлов;
- из проекта TAI Portal (\*.ap\* файл).

Импорт тегов из проекта TAI Portal (.ap файл) удобен тем, что позволяет избежать траты времени на экспорт необходимых данных. Вместо этого можно взять проект целиком и выбрать в EasyBuilder Pro только нужные теги из нужных блоков для импорта. После выбора необходимых тегов для импорта, менеджер импорта предлагает пользователю создать адресные метки, которые будут являться "плоскими" и разделенными точкой и использоваться в дальнейшем, или отказаться от создания меток и хранить теги в древовидной структуре, которая будет доступна только во время привязки тегов.

#### 4.3.2 Драйвер ПЛК с символьной адресацией

В случае драйвера с символьной адресацией адресных меток не создается. Импортированные теги можно увидеть только в момент привязки тега в объекте (рисунок 23).

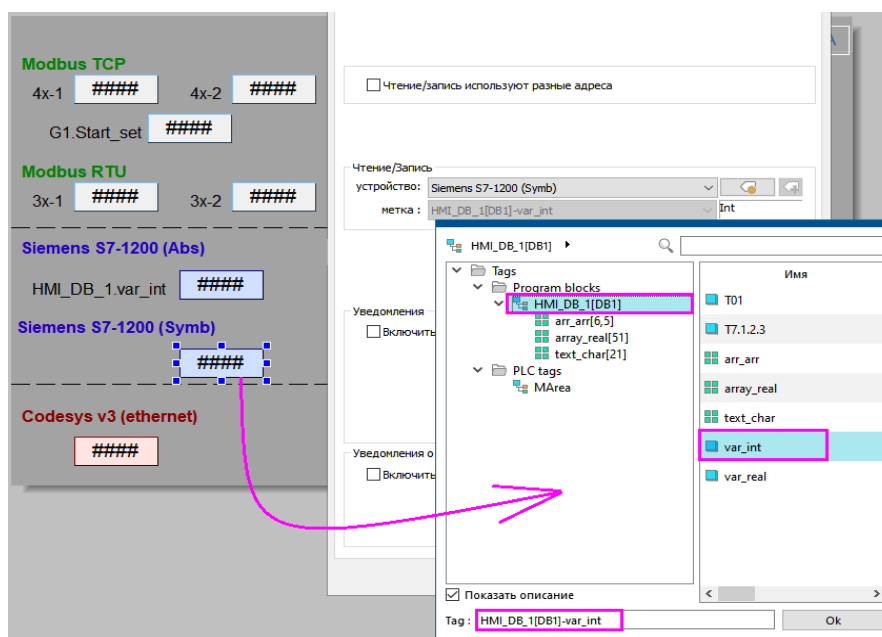


Рисунок 23 – Импорт тегов с символьной адресацией

Для «Выборки данных» в случае драйвера с символьной адресацией нужно использовать массив.

После добавления устройства переходим к привязыванию переменных устройства к графическим объектам.

Затем переходим во вкладку "Объекты" и добавляем на экран битовую кнопку и битовый индикатор. Через библиотеку изображений выбираем для них подходящую форму и цвет. Размеры получившихся графических элементов составляют 95x90 пикселей для битовой кнопки и 105x50 пикселей для битового индикатора. Для каждого состояния на индикаторе добавим метку с соответствующим цветом. Чтобы проверить работу, установим для кнопки и индикатора один и тот же локальный адрес - LB 1.

Для чтения данных с интерфейсного модуля в рамках данного проекта был использован режим работы под номером 103, который называется "holding register read". Этот режим позволяет считывать информационные сообщения, которые хранятся в области памяти с адресацией от 0 до 65535. Количество считываемых сообщений может быть до 125, и каждое сообщение имеет размер 2 байта.

Для записи управляющих команд в интерфейсный модуль был использован режим работы под номером 106, который называется "holding register write". Этот режим позволяет записывать одно информационное сообщение размером 2 байта в область памяти с адресацией от 0 до 65535.

Таким образом, эти режимы работы позволяют осуществлять обмен данными между системой управления и интерфейсным модулем, используя специфические адреса памяти и определенные форматы сообщений.

На рисунках 24-25 представлены программные блоки, реализующие функции чтения и записи информации соответственно.

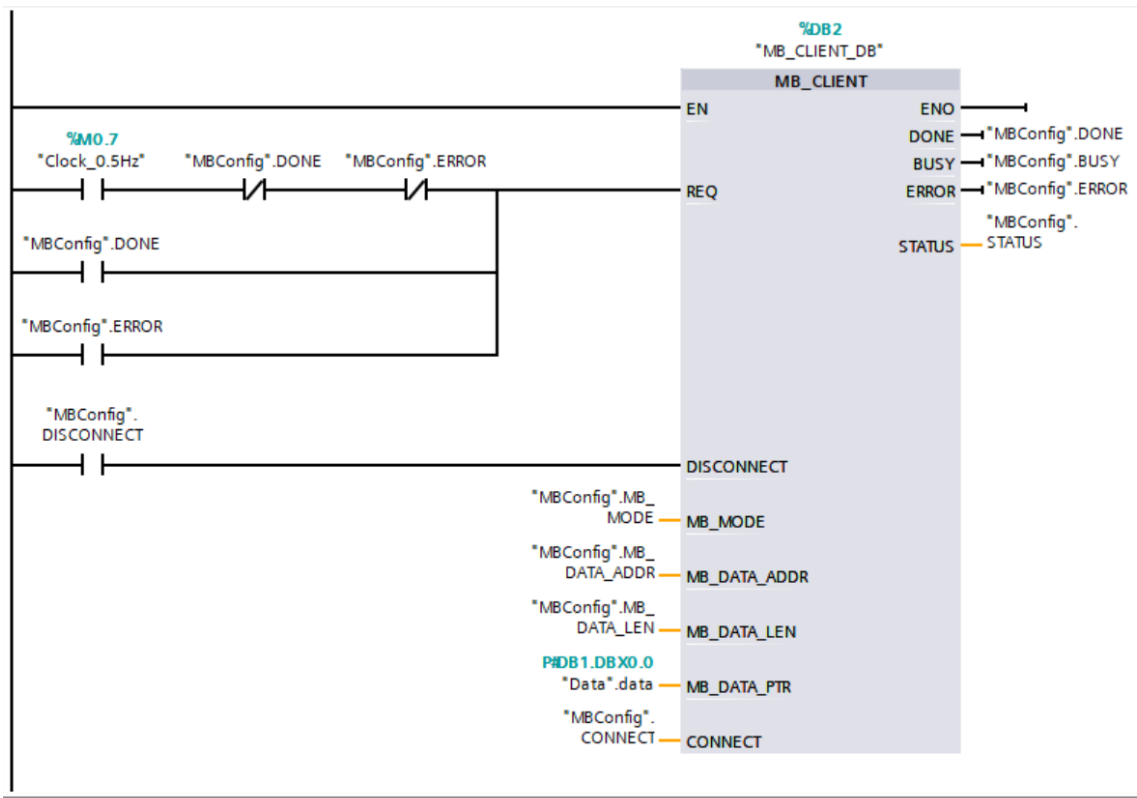


Рисунок 24 – Чтение данных по протоколу Modbus TCP

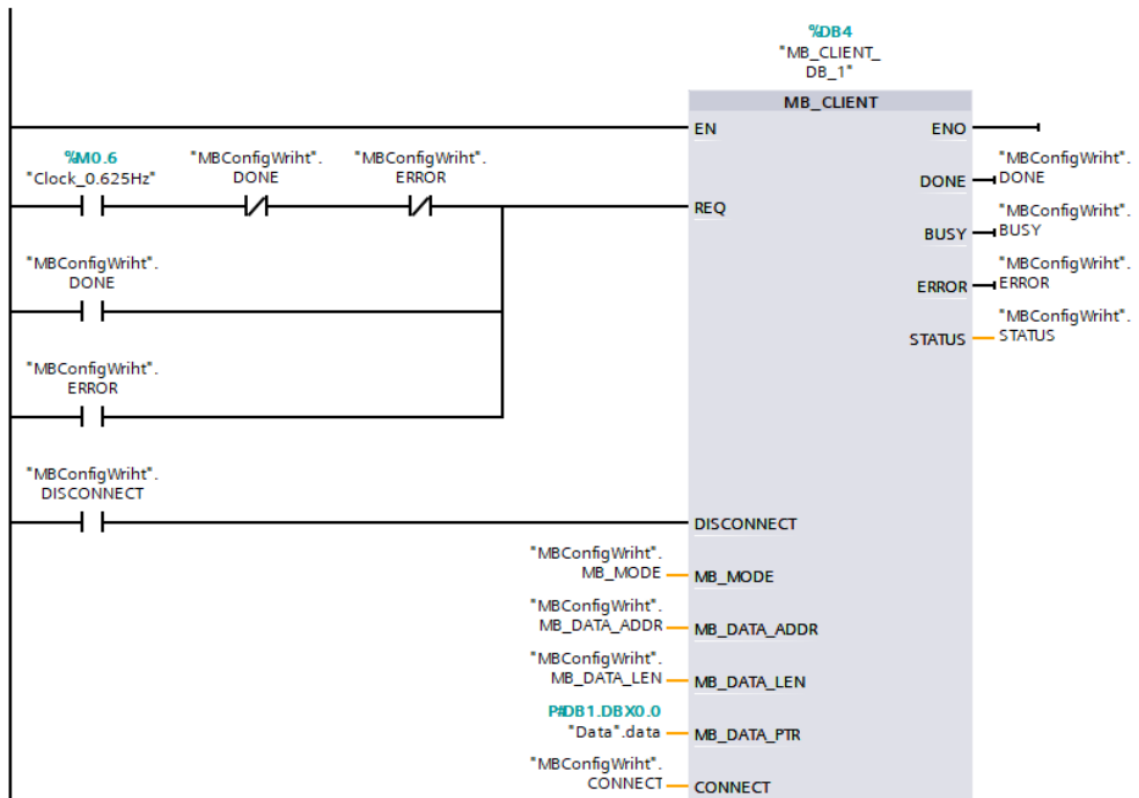


Рисунок 25 – Запись данных по протоколу Modbus TCP

Кроме того, для каждого программного блока требуется создать соответствующий блок данных, который содержит настройки соединения и режимов работы. В этом блоке данных необходимо указать ID устройства-клиента, IP-адрес устройства-сервера для подключения, а также другую необходимую информацию о режиме работы, типе протокола связи и портах подключения. На рисунках 26 и 27 показаны настройки программных блоков, отвечающих соответственно за чтение и запись данных.

При создании этих блоков данных нужно указать все необходимые параметры, чтобы обеспечить правильное соединение и обмен данными между устройством-клиентом и устройством-сервером (рисунки 26-27).

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

Рисунок 26 – Настройка блока чтения информации

Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
Static								
REQ	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
DISCONNECT	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
MB_MODE	USInt	106	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
ERROR	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
MB_DATA_ADDR	UDInt	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
MB_DATA_LEN	UInt	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CONNECT	TCON_IP_V4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Interfaceld	HW_ANY	64	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	HWIdentifier of IE-interface submodule
ID	CONN_OUC	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	connection reference / identifier
ConnectionType	Byte	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	type of connection: 11=TCP/IP, 19=UDP (17=TCP)
ActiveEstablished	Bool	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	active/passive connection establishment
RemoteAddress	IP_V4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	remote IP address (IPv4)
ADDR	Array[1..4] of Byte		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IPv4 address
ADDR[1]	Byte	192	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IPv4 address
ADDR[2]	Byte	168	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IPv4 address
ADDR[3]	Byte	31	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IPv4 address
ADDR[4]	Byte	240	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IPv4 address
RemotePort	UInt	502	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	remote UDP/TCP port number
LocalPort	UInt	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	local UDP/TCP port number
DONE	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
BUSY	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
STATUS	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Рисунок 27 – Настройка блока записи информации

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

#### 4.4 Создание управляющей программы

Для того чтобы кнопки выполняли требуемую функцию при нажатии необходимо указать кнопкам адресацию, которая использовались в TIA Portal (рисунок 28).

Так же указываем эти адреса для индикаторов.

Вид созданного блока данных с адресами показан на рисунке ниже.

Tutor > PLC > PLC tags > DI [2]									
DI									
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Monitor value	
1	data[0]	Bool	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
2	data[1]	Bool	0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
3	data[2]	Bool	0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
4	data[3]	Bool	0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
5	data[4]	Bool	0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
6	data[5]	Bool	0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
7	data[6]	Bool	0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
8	data[7]	Bool	0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
9	data[8]	Bool	1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
10	data[9]	Bool	1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
11	data[10]	Bool	1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	

Рисунок 28 – Теги в TIA Portal

Экран ручного управления манипулятором должен выглядеть примерно, как на рисунке 29.



Рисунок 29 – Панель оператора

Для того, чтобы кнопки выполняли требуемые функции при нажатии, необходимо указать адреса, которые были использованы в TIA Portal. Эти адреса также должны быть указаны на индикаторах.

При нажатии на кнопку в панели с адресом, соответствующим блоку данных в TIA Portal, состояние переключается с "false" на "true", как показано на рисунке ниже. Первые три блока данных не требуют назначения адресов, так как они зарезервированы для системы (рисунок 30).

Tutor ▶ PLC ▶ PLC tags ▶ DI [2]

DI

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Monitor value
1	data[0]	Bool	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
2	data[1]	Bool	0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
3	data[2]	Bool	0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
4	data[3]	Bool	0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> TRUE
5	data[4]	Bool	0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
6	data[5]	Bool	0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
7	data[6]	Bool	0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
8	data[7]	Bool	0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
9	data[8]	Bool	1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
10	data[9]	Bool	1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE

Рисунок 30 – Изменение состояния выхода при нажатии на кнопку

Для считывания состояния адресов мы будем использовать функцию GetData. Структура функции выглядит следующим образом: GetData(устройство, тип адреса, адрес считывания, количество данных). Для установки нового состояния мы будем использовать функцию SetData. Структура функции выглядит следующим образом: SetData(устройство, тип адреса, адрес записи, количество данных).

«В итоге получаем следующую программу:

```

short Stop, j // Устанавливаем тип данных
GetData(Stop, "Local HMI", LB, 31, 1) //Проверяем состояния по
указанному адресу
if Stop == 1 then //Условия запуска макроса
j = 0 //Проверка всех адресов в
Siemens SetData(j, "Siemens", M, 555.3, 1)
SetData(j, "Siemens", M, 555.4, 1)
SetData(j, "Siemens", M, 555.5, 1)
SetData(j, "Siemens", M, 555.6, 1)
SetData(j, "Siemens", M, 555.7, 1)
SetData(j, "Siemens", M, 555.8, 1)

```

*// Управление каждым движением манипулятора в отдельности по заданной очереди*

*i = 1*

*SetData(i, "Local HMI", LB, 3, 1)*

*SetData(i, "Local HMI", LB, 4, 1)*

*SetData(i, "Local HMI", LB, 5, 1)*

*SetData(i, "Local HMI", LB, 6, 1)*

*SetData(i, "Local HMI", LB, 7, 1)*

*SetData(i, "Local HMI", LB, 8, 1)*

*end if» [7]*

Данная программа осуществляет проверку адресации двигателей и их движений, а также очередное управление движением звеньев манипулятора. Ниже приведена блок-схема очередного управления манипулятором.

В нашем случае кнопки управления манипулятором расположены на панели HMI (показано ранее), а автоматический алгоритм работы осуществляет очередное управление манипулятором путём последовательного «нажатия» соответствующих кнопок на панели оператора по заданному алгоритму. Алгоритм задан так, чтобы манипулятор перемещал свою стрелу со схватом в заданные координаты, введённые пользователем, в цикле, пока не закончатся введённые пользователем данные.

Здесь всё управление происходит внутри панели HMI, а ПЛК Siemens S7-1200 играет роль передатчика данных в тегах на нужные вводы/выводы. Для передачи используется протокол Modbus TCP по сети Ethernet через коммутатор (на рисунке 31 обозначен как Ethernet switch), порядок работы с которым показан ранее.



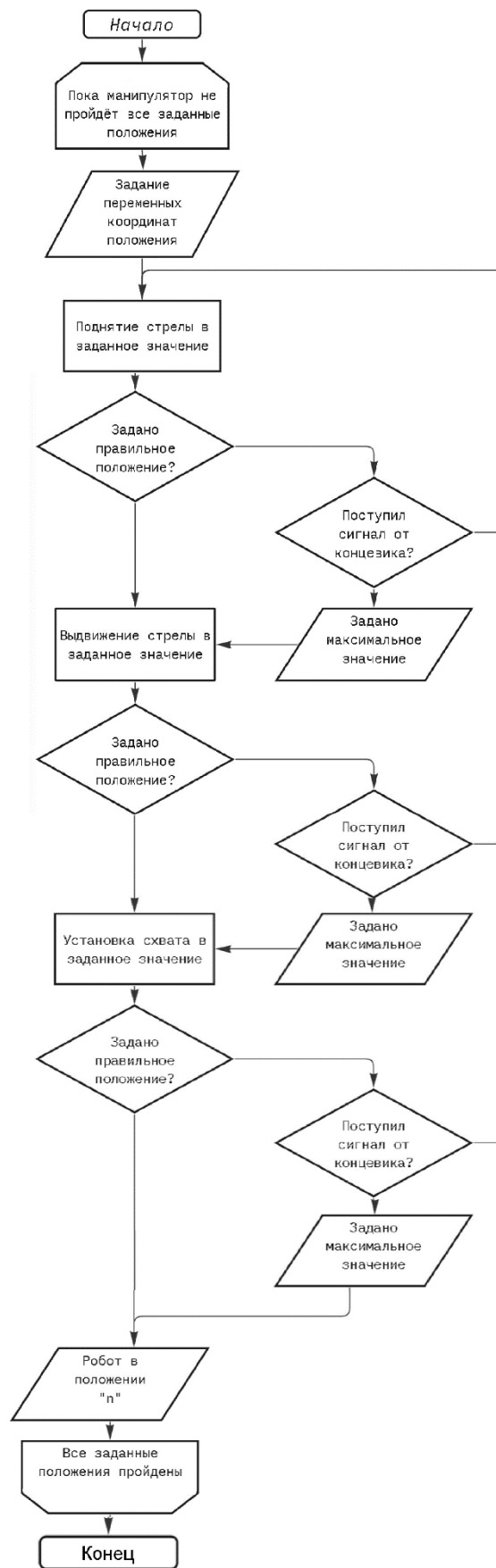


Рисунок 31 – Блок-схема алгоритма работы

Структурно система выглядит, как показано на рисунке 32.

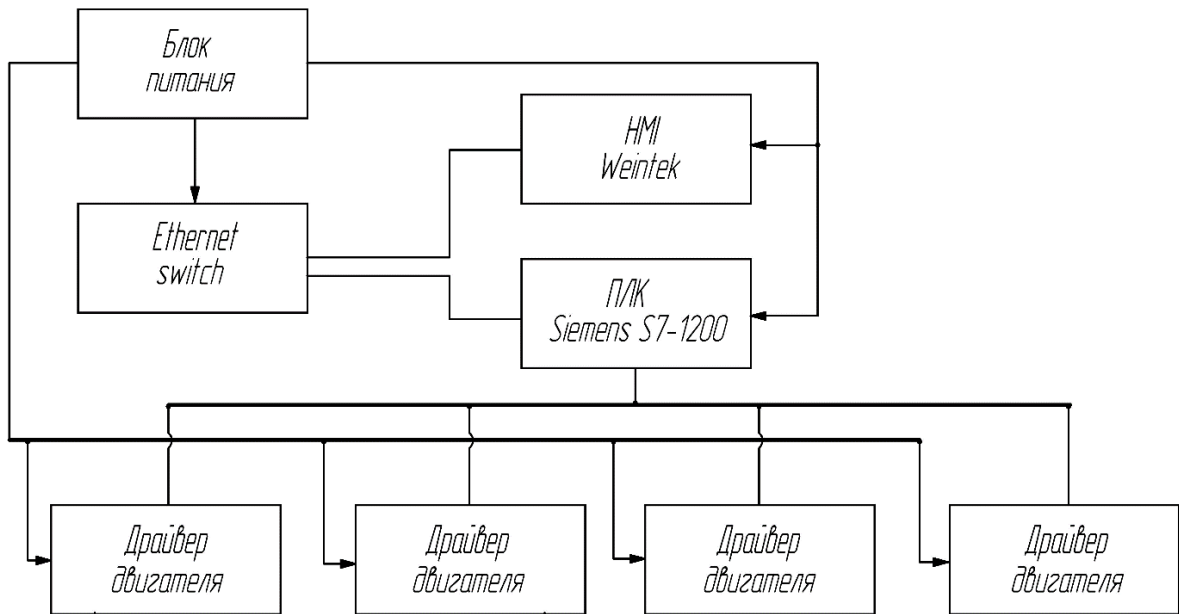


Рисунок 32 – Структура системы управления манипулятором

Соединение всех элементов системы стоит производить по схеме ниже (рисунок 33).

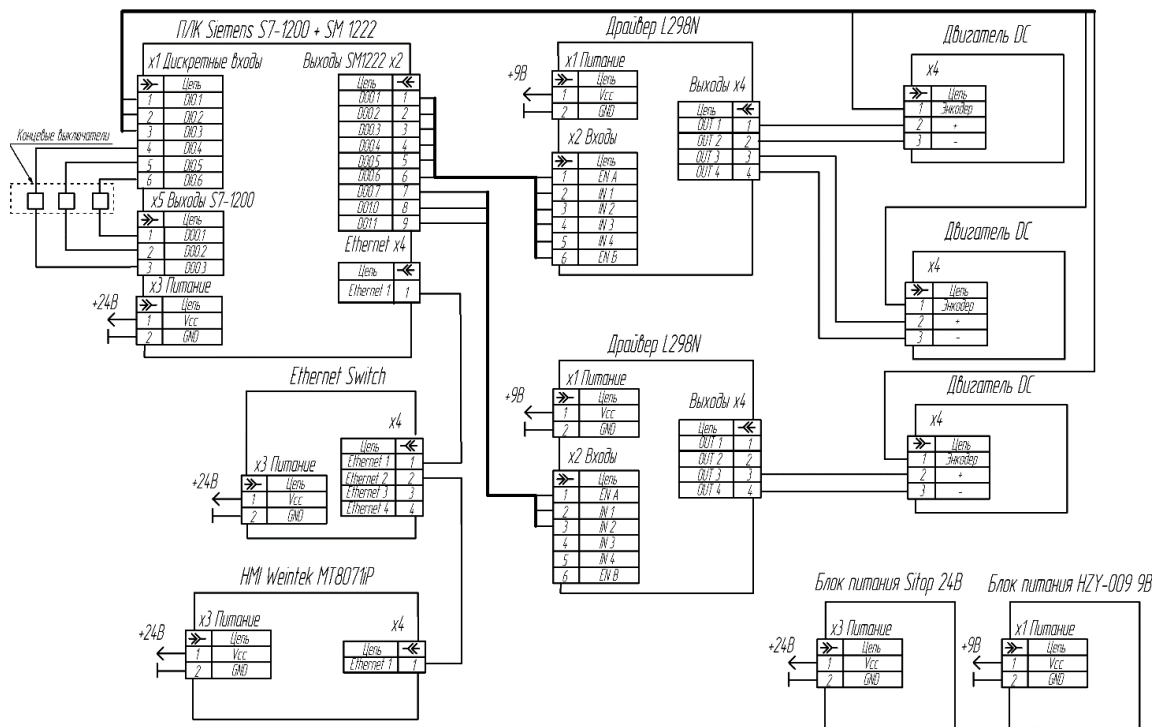


Рисунок 33 – Схема принципиальная электрическая

## Заключение

Целью выпускной квалификационной работы было создание единой системы интерфейса оператора для лабораторного стенда, которая будет отвечать за контроль и управление трехкоординатным манипулятором.

В процессе выполнения выпускной квалификационной работы были изучены вариации операторских панелей, произведена связь между программируемым логическим контроллером Siemens S7-1200 и панелью оператора Weintek Weintek MT8071iP. В ходе разработки программы были созданы алгоритмы управления трехкоординатным манипулятором. Так же для работы студентов было подробно разобрано программное обеспечение EasyBuilder Pro с описанием интерфейсных особенностей программы, для того чтобы студенты в ходе лабораторных работ могли изменять сценарии работы манипулятора.

В рамках работы были выполнены нижеследующие задачи.

- анализ требований и функциональных возможностей системы. Был проведен обзор существующих интерфейсов управления манипуляторами, изучены основные принципы и подходы к разработке подобных систем;
- проектирование интерфейса. Был разработан дизайн интерфейса;
- разработка программного обеспечения. Был разработан программный код, обеспечивающий взаимодействие с манипулятором и обработку команд оператора. В процессе работы были использованы современные методы программирования и инструменты разработки;

Результаты работы могут быть использованы в различных областях, где требуется управление трехкоординатными манипуляторами, таких как промышленность, медицина, робототехника и другие.

## Список используемой литературы и источников

1. "Robotics: Modelling, Planning and Control" от Bruno Siciliano и Oussama Khatib.
2. "Introduction to Robotics: Mechanics and Control" от John J. Craig.
3. "Robot Mechanisms and Mechanical Devices Illustrated" от Paul E. Sandin и Oliver R. Rye.
4. Выбор мотора для робопрокта // Роботшоп URL: <https://www.robotshop.com/community/blog/show/choosing-right-motor-for-robotics-project/> (дата обращения: 24.04.2023).  
<https://www.sciencedirect.com/topics/engineering/electric-motors-for-robotics>
5. 37D Metal Gearmotors // Pololu URL: <https://www.pololu.com/category/116/37d-metal-garmotors> (дата обращения: 24.04.2023). <https://www.siemens-pro.ru/components/s7-1200.htm>
6. Как подключить ПЛК к панели Weintek // РусАвтоматика URL: <https://www.rusavtomatika.com/articles/kak-podklyuchit-plk-k-paneli-weintek-/> (дата обращения: 19.04.2023).
7. Siemens S7-1200/S7-1500 (Absolute Addressing) (Ethernet) // РусАвтоматика URL: [https://www.rusavtomatika.com/upload\\_files/drivers/Siemens\\_S7\\_1200\\_S7\\_1500\\_Absolute\\_Addressng\\_Ethernet.pdf](https://www.rusavtomatika.com/upload_files/drivers/Siemens_S7_1200_S7_1500_Absolute_Addressng_Ethernet.pdf) (дата обращения: 19.03.2023).
8. Александров А.А. Электротехнические чертежи и схемы / Александров К.К., Кузьмина Е.Г.- М.: Энергоатомиздат, 1990. - 288с.
9. Промышленная электроника: Учебник для вузов / Горбачев Г.Н., Чаплыгин Е.Е.; Под ред. В.А. Лабунцова. – М.: Энергоатомиздат, 1988. – 320 с.: ил.
10. Электронные промышленные устройства: Учеб. для студ. вузов спец. «Пром. электрон.»/ В.И. Васильев, Ю.М. Гусев, В.Н. Миронов и др. – М.: Высш. шк., 1988. – 303 с.: ил.

11. The world's smallest, most compact, and precise industrial robots // mecademic URL: <https://www.mecademic.com/> (дата обращения: 19.03.2023).

12. Precise industrial robots // franka URL: <https://www.franka.de/panda/> (дата обращения: 09.03.2023).

13. THE UR3e COMPACT TABLE-TOP ROBOT // Universal Robots URL: <https://www.universal-robots.com/products/ur3-robot/> (дата обращения: 09.03.2023).

14. Robotics Solutions // epson URL: <https://corporate.epson/en/products/robot-systems.html> (дата обращения: 27.03.2023).

15. Articulated Robots // ABB URL: <https://new.abb.com/products/robotics/robots/articulated-robots/irb-120> (дата обращения: 27.03.2023).

16. Robotics System // Kuka URL: <https://www.kuka.com/en-us/products/robotics-systems/lbr-iiwa> (дата обращения: 27.02.2023).

17. Motor Robotics // yamaha URL: <https://yamaha-motor-robotics.de/> (дата обращения: 18.03.2023).

18. Глибин Е.С. Программирование электронных устройств : электронное учеб. пособие / Е.С. Глибин, А.В. Прядилов. – Тольятти : Изд-во ТГУ, 2014.: 1 оптический диск

19. Дискретные модули ввода/вывода [Электронный ресурс]: - режим доступа: <http://progressavtomatika.ru/katalog-tovarov/diskretnye-moduli-vvoda-vyvoda-siemens-simatic-s7-1200-sm1221-sm1222-sm1223.html> (дата обращения: 05.05.2023)

20. Z. Zhang, F. Xiao and S. Zhu, "A hybrid control approach for robot manipulators based on task space and joint space control," 2020 4th IEEE International Conference on Control Science and Systems Engineering (ICCSSE), Chengdu, China, 2020, pp. 527-531, doi: 10.1109/ICCSSE49887.2020.9199122.

21. SG90 9g Micro Servo [Электронный ресурс]: документация – режим доступа: <http://www.micropik.com/PDF/SG90Servo.pdf>

22. Интерфейс  $I^2C$  и Arduino [Электронный ресурс]: статья. – режим доступа: <http://soltau.ru/index.php/arduino/item/371-interfejs-i2c-i-arduino>

23. Digital LED Pixels: Instructions for use and a characterization of their properties, 2015, Pete R. Jones, Sara E. Garcia, Marko Nardini.

24. Servo signal processing for flying height control in hard disk drives, 2011, Uwe Boettcher, Christopher A. Lacey, Hui Li, Kensuke Amemiya, Raymond A. de Callafon, Frank E. Talke.

25. Simulation of multibody systems with servo constraints through optimal control, 2016, R. Altmann, J. Heiland.

26. Bluetooth in Intelligent Transportation Systems: A Survey, 2014, M. R. Friesen, R. D. McLeod.