

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет энергетический

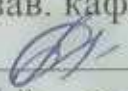
Кафедра автоматизации производственных процессов и электротехники

Направление подготовки 15.03.04 – Автоматизация технологических
процессов и производств

Направленность (профиль) образовательной программы – Автоматизация
технологических процессов и производств в энергетике

ДОПУСТИТЬ К ЗАЩИТЕ

и.о. зав. кафедрой

 О.В. Скрипко
« 24 » июня 2025 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Система управления мобильным роботом-манипулятором

Исполнитель

студент группы 141 об

 20.06.2025 В.А. Кузнецов
(подпись, дата)

Руководитель

профессор, д-р техн. наук

 21.06.2025 О.В. Скрипко
(подпись, дата)

Консультант по безопасности
и экологичности

доцент, канд. техн. наук

 19.06.2025 А.Б. Булгаков
(подпись, дата)

Нормоконтроль

профессор, д-р техн. наук

 23.06.2025 О.В. Скрипко
(подпись, дата)

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

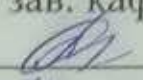
факультет Энергетический

Кафедра Автоматизации производственных процессов и электротехники

Направление подготовки 15.03.04 – Автоматизация технологических
процессов и производств

УТВЕРЖДАЮ

и.о. зав. кафедрой

 О.В. Скрипко

« 24 » июня 2025 г.

ЗАДАНИЕ

К выпускной квалификационной работе студента Кузнецова Валерия Александровича

1. Тема выпускной квалификационной работы: Система управления
мобильным роботом-манипулятором

(утверждена приказом от 29.04.2025 № 1104-уч)

2. Срок сдачи студентом законченной работы (проекта): 25.06.2025 г.

3. Исходные данные к выпускной квалификационной работе:

1) Рабочая документация к стенду САУ-робот;

2) Приказ об утверждении темы бакалаврской работы;

3) Материалы, собранные в ходе практики.

4. Содержание выпускной квалификационной работы (перечень
подлежащих разработке вопросов):

1) Описание объекта автоматизации;

2) Обзор технических средств автоматизации;

3) Разработка системы управления;

4) Разработка программного обеспечения;

5) Безопасность и экологичность

3. Перечень материалов приложения:

Лист 1: Схема системы управления манипулятора;

Лист 2: Структурная и кинематическая схемы манипулятора;

Лист 3: Принципиальная электрическая схема манипулятора;

Лист 4: Принципиальная схема управления мотор-колесами;

Лист 5: Пользовательский интерфейс;

Лист 6: Схема микроконтроллерной программы управления.

6. Консультанты по выпускной квалификационной работе (с указанием относящихся к ним разделов):

Безопасность и экологичность – Булгаков А.Б., доцент, канд. техн. наук

7. Дата выдачи задания: 10.04.2025 г.

8. Руководитель выпускной квалификационной работы: профессор кафедры АППиЭ, д-р техн. наук Скрипко Ольга Валерьевна

(фамилия, имя, отчество, должность, ученая степень, учёное звание)

Задание принял к исполнению (дата, подпись): 10.04.2025

РЕФЕРАТ

Выпускная работа содержит 81 с., 41 рисунок, 15 таблиц, 20 источников, 2 приложения.

МИКРОКОНТРОЛЛЕР, ШАГОВЫЙ ДВИГАТЕЛЬ, ДРАЙВЕР ШАГОВОГО ДВИГАТЕЛЯ, РОБОТ-МАНИПУЛЯТОР, АЛГОРИТМ, ПК, ШИМ, ИНТЕРФЕЙС, ОПТИЧЕСКАЯ РАЗВЯЗКА.

Объектом разработки является мобильный робот-манипулятор, используемый в образовательных целях.

Цель работы. Разработка системы управления роботом, осуществляющую автоматическое, дистанционное и программное управление, для использования в проектной деятельности студентов и проведения лабораторных работах по автоматизации.

Робот манипулятор обладает следующими конструктивными характеристиками: масса – 25 кг; размеры – 85 см х 50 см х 70 см (длина, ширина, высота). Техничко-эксплуатационные характеристики: напряжение питания 36 В; мощность мотор-колёс 250 Вт; максимальный ток шаговых двигателей 3 А; вылет стрелы 350 мм; угол поворота башни -180...+180 градусов; подъем стрелы 350 мм.

Разработанная система управления может быть использована для внедрения и в другие подобные манипуляторы, использующие в качестве электропривода шаговые двигатели.

Спроектированная система, представляет собой программно-аппаратный комплекс, который может быть положен в основу создания управляющей части роботов, используемых в технологических процессах, складах и в качестве сервисных или коллаборативных роботов.

Разработка программно-аппаратных комплексов управления роботами имеет важное значение при автоматизации технологических процессов и логистических процессов на складах.

СОДЕРЖАНИЕ

Введение	7
1 Описание мобильного робота-манипулятора	9
1.1 Технические характеристики	9
1.2 Используемый электропривод	11
1.2.1 Шаговые электродвигатели вертикального и горизонтального перемещения стрелы	12
1.2.2 Шаговый электродвигатель поворота башни	13
1.2.3 Сервоприводы схвата манипулятора	15
1.2.4 Мотор-колеса мобильного робота	15
1.3 Кинематическая схема манипулятора	16
2 Выбор электронных компонентов	18
2.1 Выбор драйверов шагового двигателя	18
2.1.1 Принцип действия ШИМ	18
2.1.2 Виды драйверов шаговых двигателей	19
2.1.3 Выбор драйверов шаговых двигателей	21
2.1.4 Подключение и настройка драйверов	24
2.2 Выбор датчиков конечного положения для манипулятора	27
2.3 Разработка платы опторазвязки	29
2.4 Выбор контроллера для управления манипулятором	33
2.5 Выбор понижающего преобразователя для 5 вольт	34
2.6 Выбор понижающего преобразователя для 12 вольт	38
2.7 Выбор камеры с передатчиком	39
3 Разработка системы управления	41
3.1 Подсистема управления манипулятором	41
3.2 Подсистема управления мотор-колесами	42
3.1 Система электропитания мобильного робота	43
3.2 Электрическая схема	45
3.3 Разработка логической схемы управления роботом	48

4 Разработка программного обеспечения робота-манипулятора	51
4.1 Микроконтроллерная программа управления мотор-колесами и протокол связи RPM	51
4.2 Микроконтроллерная программа управления манипулятором	52
4.3 Разработка компьютерной программа управления манипулятором мобильного робота	59
5 Безопасность и экологичность	63
5.1 Безопасность	63
5.1.1 Безопасность при работе с роботом	63
5.1.2 Меры безопасности при зарядке литий-ионных аккумуляторов	66
5.2 Экологичность	69
5.2.1 Подготовка аккумуляторов к утилизации	70
5.3 Чрезвычайные ситуации	71
Заключение	73
Библиографический список	74
Приложение 1. Техническое задание	76
Приложение 2. Код программы	78

СПИСОК СОКРАЩЕНИЙ

ШИМ – широтно-импульсная модуляция.

PPM – протокол связи для передачи сигнала по одному проводу

ESP32 – микроконтроллер, разработанный компанией Espressif Systems.

Arduino IDE – среда разработки программного кода для микроконтроллеров.

ШД – шаговый двигатель.

ШВП – шарико-винтовая передача.

ЧПУ – числовое программное управление.

CC-CV – constant current, constant voltage.

RC – радиоуправляемый.

АС – переменный ток.

DC – постоянный ток.

BLDC – бесщеточный двигатель постоянного тока.

ВВЕДЕНИЕ

Отечественный рынок робототехники развивается медленными темпами, и степень роботизации промышленности не высока, плотность промышленных роботов составляет 19 единиц на 10000 работников на 2025 год. Основная проблема, которая тормозит развитие робототехники, это отсутствие компонентной базы, относительно высокий срок окупаемости оборудования и недостаточный уровень научно-технических разработок.

Целесообразность разработки систем управления манипуляторами заключается в следующем. Манипуляторы и роботизированные комплексы позволяют механизировать технологические операции, что повышает производительность и качество продукции, а также снижает трудозатраты. Адаптация к динамичным условиям. Современные роботы-манипуляторы должны не только выполнять заранее заданные задачи, но и реагировать на изменения в окружающей среде. Снижение риска повреждений. Системы управления с обратной связью позволяют оператору контролировать усилие захвата, что предотвращает повреждение объектов.

Основанием и исходными данными для разработки темы связанной с системой учебным роботом-манипулятором стали работы при прохождении преддипломной и производственной практик, работы проводимые в рамках проектной деятельности, а также результаты выпускных квалификационных работ следующих студентов: Бендика Владимира, Потемкина Михаила и Бовы Дмитрия.

Новизна разработанной системы управления заключается в создании гибридного способа управления манипулятором. С помощью программного переключения, возможно управление роботом по сети Wi-Fi или радиоаппаратурой. Способ по сети Wi-Fi позволяет программировать траекторию движения рабочего органа или перемещать его в ручном режиме. Радиоаппаратура используется для ручного дистанционного режима.

1 ОПИСАНИЕ МОБИЛЬНОГО РОБОТА-МАНИПУЛЯТОРА

Робот-манипулятор может применяться в научных целях для ознакомления с системами автоматизации и получения практических навыков работы.

1.1 Технические характеристики

Мобильный робот на мотор-колесах может перемещаться по ровным поверхностям в любых направлениях. Манипулятор мобильного робота может перемещаться по трем осям: ось X – вылет стрелы, ось Y – подъем стрелы, ось Z – поворот башни и применять схват. В качестве привода для звеньев манипулятора используются гибридные шаговые двигатели и сервоприводы для схвата. Для ограничения движения применяются датчики конечного положения. Система управляется дистанционно с применением ручного управления. Внешний вид робота приведен на рисунке 1.1.



Рисунок 1.1 – Внешний вид мобильного робота – манипулятора

Для управления роботом будет использоваться пульт или программа на компьютере. Основные характеристики робота указаны в таблице 1.1.

Таблица 1.1 – Технические характеристики манипулятора

Параметр	Значение
Напряжение электропитания, В	36
Диапазон углов поворота башни	-180..180°
Максимальный вылет стрелы, мм	350
Максимальный подъем стрелы, мм	350
Скорость, км/ч	15-20
Масса, кг	25

Рама мобильного робота имеет форму прямоугольника и изготовлена из металлического профиля размером 20 мм. Под рамой располагаются крепления для колес, каждое из которых зафиксировано с помощью 4 болтов, диаметр болтов – 6 мм. Сверху рамы расположены шкаф управления колесами мобильного робота, и манипулятор со своей системой управления. Чертёж рамы представлен на рисунке 1.2.

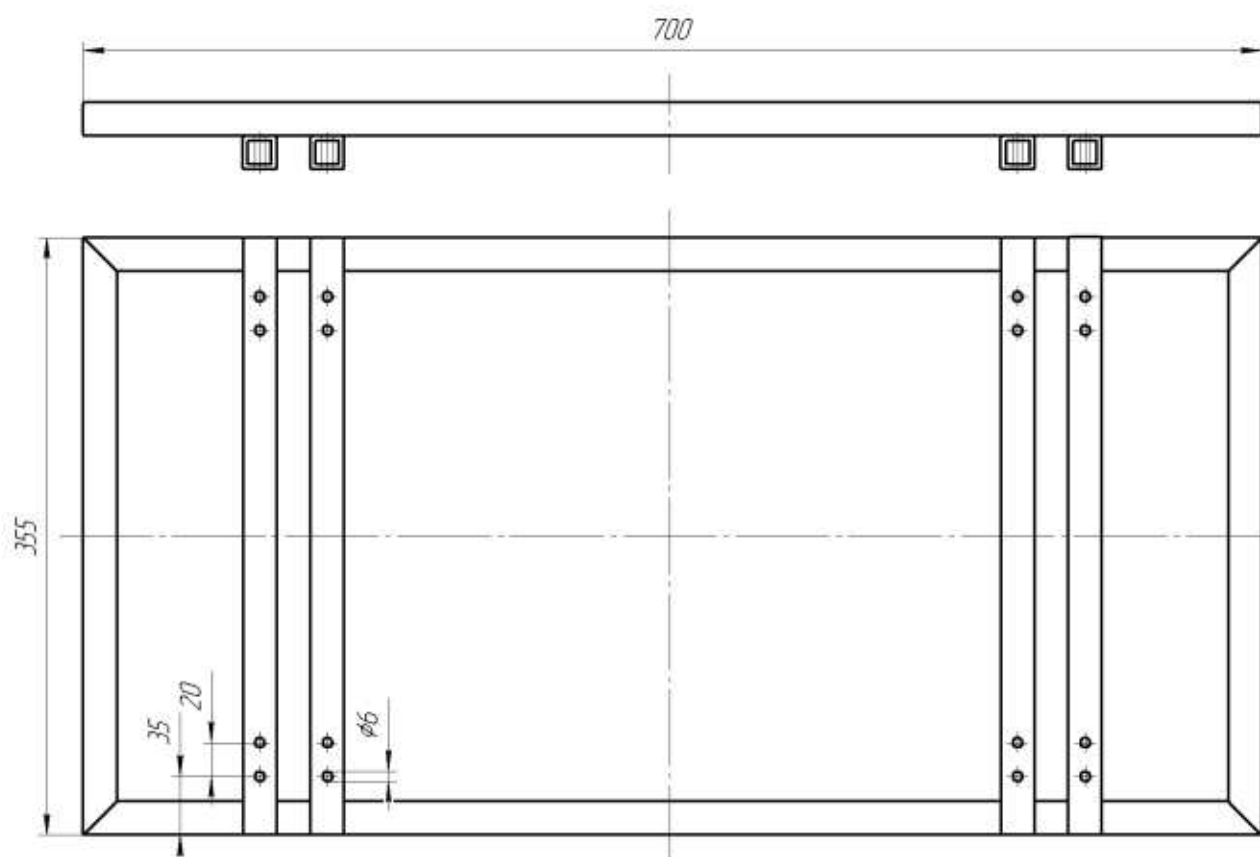


Рисунок 1.2 – Чертёж рамы мобильного робота

1.2 Используемый электропривод

В манипуляторе используются три шаговых двигателя и два сервопривода. Шаговый двигатель - это вращающийся электродвигатель с дискретными угловыми перемещениями ротора. Предшественником шагового двигателя является серводвигатель. Шаговые (импульсные) двигатели напрямую (без датчика обратной связи) преобразуют управляющий сигнал в виде последовательности импульсов в фиксированный угол поворота вала или линейное перемещение механизма, пропорциональное количеству импульсов. Такой привод обладает такими преимуществами, как снижение стоимости устройства (меньшее количество элементов) и повышенная точность за счет фиксации ротора шагового двигателя при отсутствии сигнальных импульсов. Недостаток привода с шаговым двигателем также очевиден: при сбое импульса дальнейшее отслеживание происходит с погрешностью в угле, пропорциональной количеству пропущенных импульсов. Конструкция шагового двигателя приведена на рисунке 1.3.

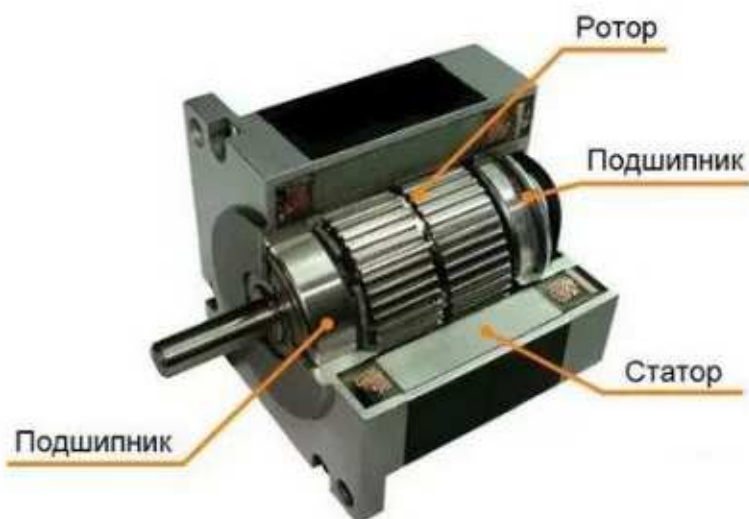


Рисунок 1.3 – Конструкция шагового двигателя

Шаговый двигатель, как и любой электродвигатель, состоит из вращающейся части (ротора) и неподвижной части (статора).

Определить, имеет ли ротор постоянные магниты или нет, можно, вращая обесточенный двигатель, если при вращении возникает удерживающий момент, то ротор выполнен на постоянных магнитах. Статор шагового двигателя имеет

сердечник с выступающими полюсами, которые обычно изготавливаются из многослойных штампованных листов электротехнической стали для уменьшения вихревых токов и выделения тепла. Статор шагового двигателя обычно имеет от двух до пяти фаз. Поскольку шаговый двигатель не рассчитан на непрерывное вращение, мощность в его параметрах не указана.

Шаговый двигатель является маломощным двигателем по сравнению с другими электродвигателями. Одним из определяющих параметров шагового двигателя является шаг ротора, то есть угол поворота ротора, соответствующий одному импульсу. Размер шага зависит от конструкции двигателя: количества обмоток, полюсов и зубьев. В зависимости от конструкции двигателя размер шага может варьироваться в диапазоне от 90 до 0,75 градусов. Определить принадлежность проводов к обмоткам можно с помощью мультиметра. Схема расположения обмоток в четырехпроводном шаговом двигателе приведена на рисунке 1.4.

1.2.1 Шаговые электродвигатели вертикального и горизонтального перемещения стрелы

Для осуществления вертикального и горизонтального перемещения стрелы применяются шаговые двигатели GD57STH56-2808A. Его основные характеристики приведены в таблице 1.2. Схема подключения обмоток привода изображена на рисунке 1.4. Габаритные размеры для монтажа двигателя приведены на рисунке 1.5.

Таблица 1.2 – Технические характеристики гибридного ШД GD57STH56

Параметр	Значение
Тип двигателя	2х-фазный ШД
Фланец	57 мм (NEMA 23)
Момент удержания	1,3 Нм
Ток фазы	3,0 А
Шаг	1,8° (200 шагов/об)
Диаметр вала	6,35 мм
Сопротивление обмотки	0,7 Ом
Масса	0,69 кг

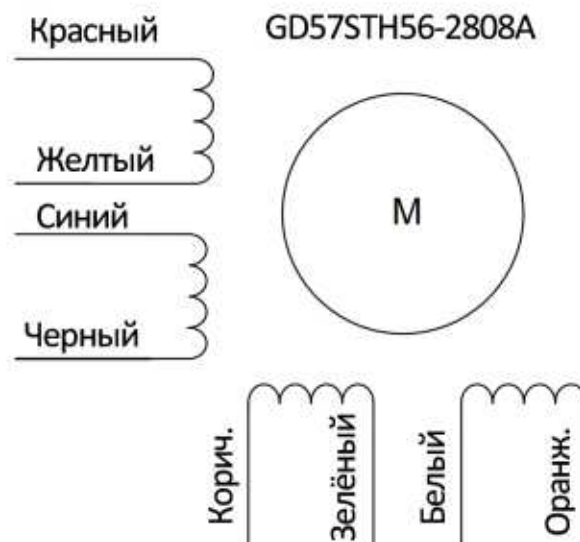


Рисунок 1.4 – Расположение обмоток GD57STH56-2808A

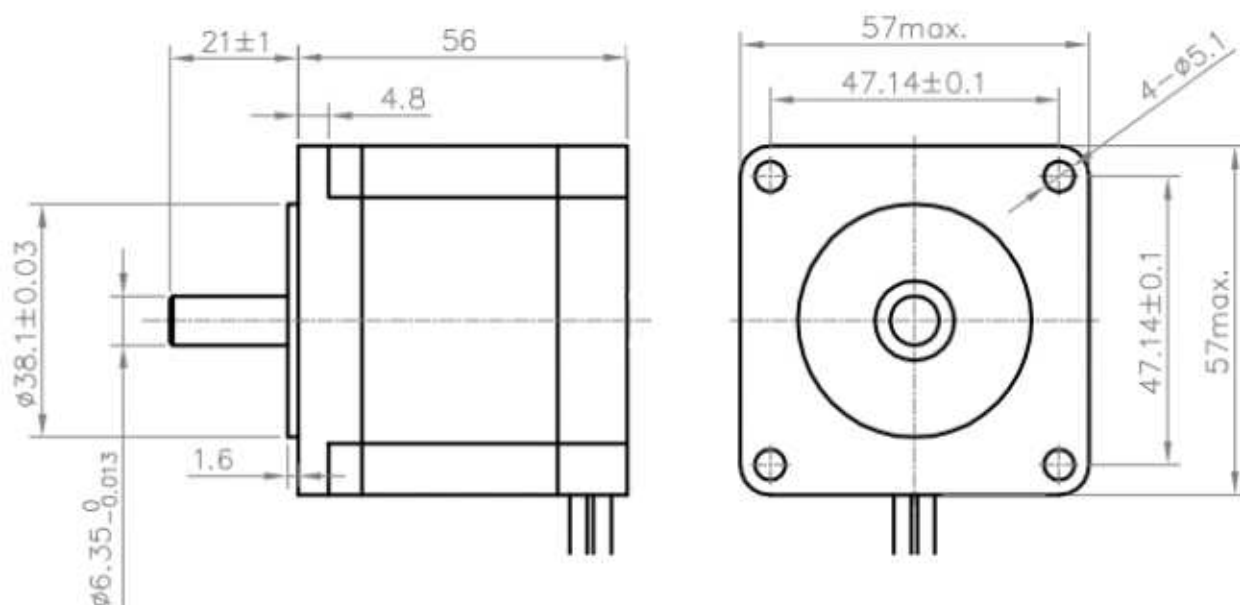


Рисунок 1.5 – Габаритные размеры GD57STH56-2808A

1.2.2 Шаговый электродвигатель поворота башни

Для осуществления поворота стрелы вокруг вертикальной оси применяется гибридный шаговый двигатель GD57STH76-2006A. Его основные характеристики приведены в таблице 1.3. Схема подключения обмоток привода изображена на рисунке 1.6. Габаритные размеры для монтажа двигателя приведены на рисунке 1.7.

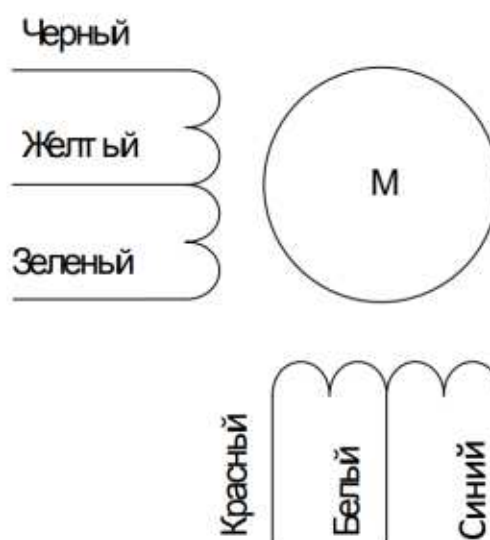


Рисунок 1.6 – Расположение обмоток GD57STH76-2006A

Таблица 1.3 – Технические характеристики ШД GD57STH76-2006A

Параметр	Значение
Тип двигателя	2х-фазный ШД
Фланец	57 мм (NEMA 23)
Момент удержания	1,8 Нм
Ток фазы	4,0 А
Длина корпуса	76 мм
Шаг	1,8° (200 шагов/об)
Диаметр вала	6,35 мм
Сопротивление обмотки	0,8 Ом
Масса	1,055 кг

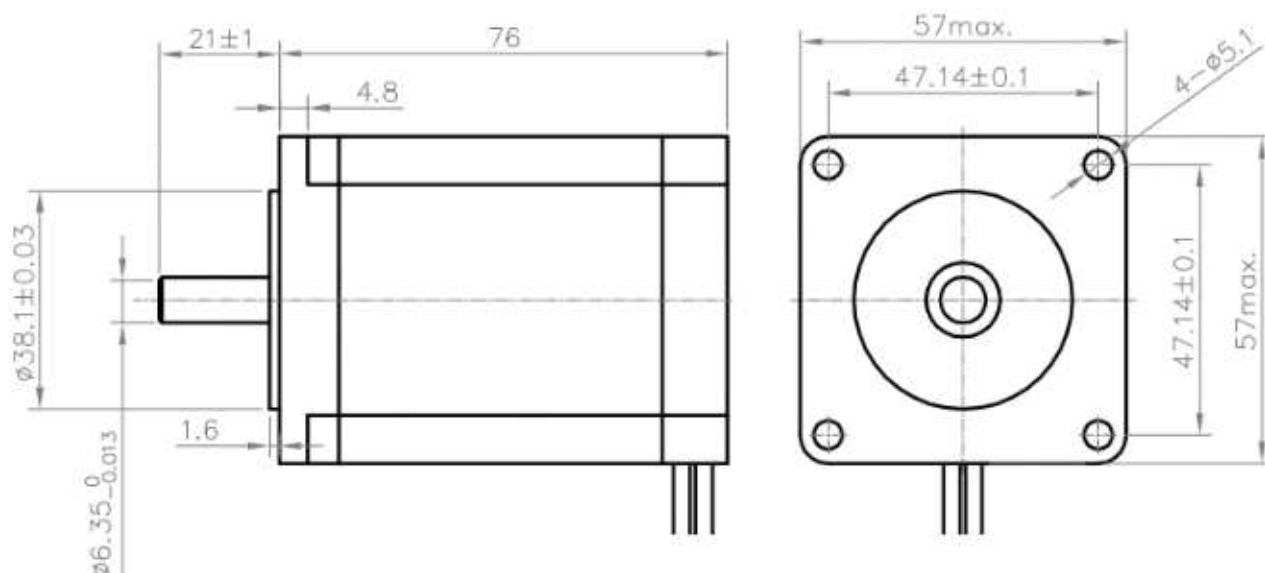


Рисунок 1.7 – Габаритные размеры GD57STH76-2006A

1.2.3 Сервоприводы схвата манипулятора

За применение схвата манипулятора отвечают два сервопривода MG996R. Сервопривод MG996R с механизмом из металла и пластика используется для поворота разных механизмов, открытия/закрытия заслонок и в качестве привода для суставов при создании разнообразных роботов, обладает неплохим крутящим моментом до 10 кг·см. Механизм сервопривод производится из металлических деталей, что обеспечивает долговечность и надежность работы. Привод может применяться в роботизированных системах, для перемещения задвижек. Сервопривод MG996R подключается с помощью трех проводов: Коричневый - GND (земля) - подключается к минусу источника питания. Красный - VCC (питание) - подключается к плюсу источника питания 5 - 6 В. Оранжевый - S (сигнал) - подключается к управляющему аналоговому выводу контроллера. Не следует использовать питание для сервопривода доступное с платы микроконтроллера, из-за больших токов микроконтроллер может выйти из строя, следует использовать отдельный источник питания. Остальные характеристики привода приведены в таблице 1.4.

Таблица 1.4 – Характеристики сервопривода MG996R

Параметр	Значение
Рабочее напряжение:	4,8 ~ 6 В
Усилие:	10 кг/см
Скорость:	0.2 с/60о (при 4.8 V)
Рабочая температура:	0 - 55C
Разъём:	JR
Диапазон вращения:	180 гр.
Вес:	55 г
Размер:	40.7 x 19.7 x 42.9 мм

1.2.4 Мотор-колеса мобильного робота

На раме мобильного робота установлены четыре мотор-колеса. Основное преимущество мотор-колес заключается в их эргономичности, мотор-колеса

легко фиксируются на раме и не имеют лишних выступающих элементов. Чертёж колеса представлен на рисунке 1.8. Характеристики приведены в таблице 1.5.

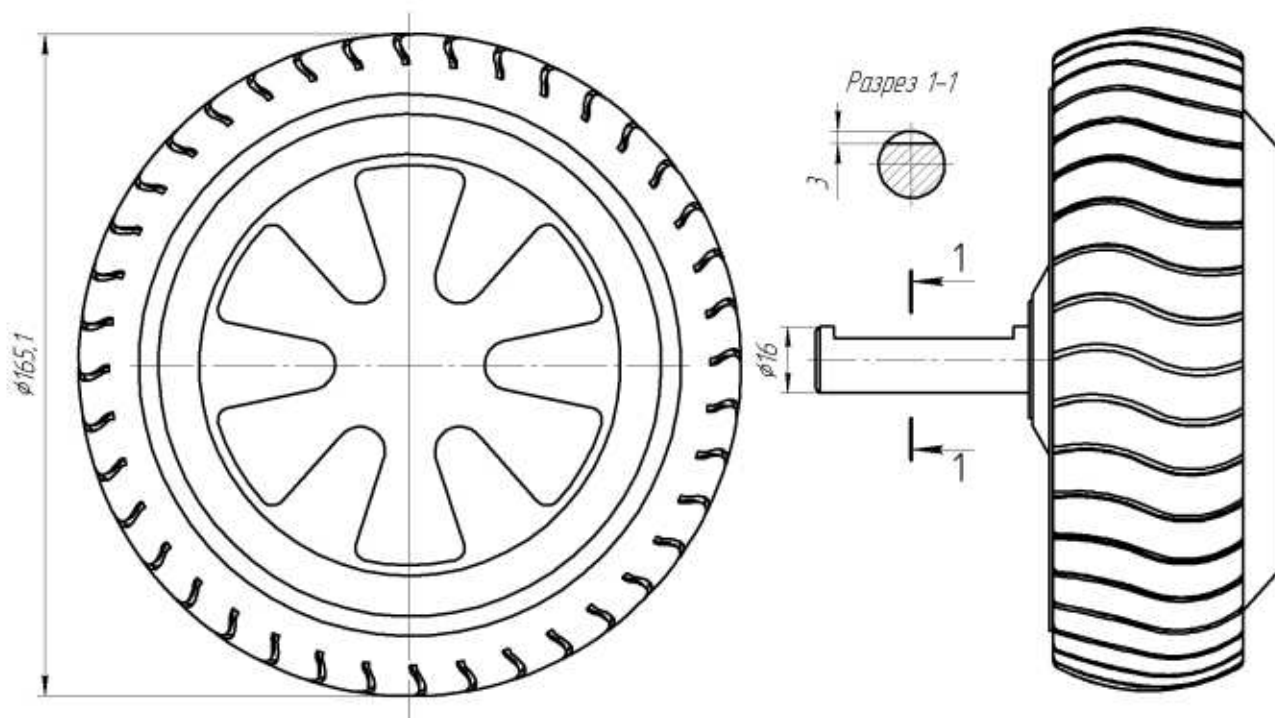


Рисунок 1.8 – Чертёж мотор-колеса мобильного робота

Таблица 1.5 – Характеристики мотор-колёс

Параметр	Значение
Максимальная мощность	350 Вт
Крутящий момент	10-120 Нм
Рабочий ток	<16 А
Номинальное напряжение	36 В
Масса	2,4 кг
КПД	90 %

1.3 Кинематическая схема манипулятора

Ниже представлена кинематическая схема манипулятора с применением винтов с трапецеидальной резьбой, шаг резьбы - 4 мм. Схема приведена на рисунке 1.9. Спецификация к схеме приведена в таблице 1.6.

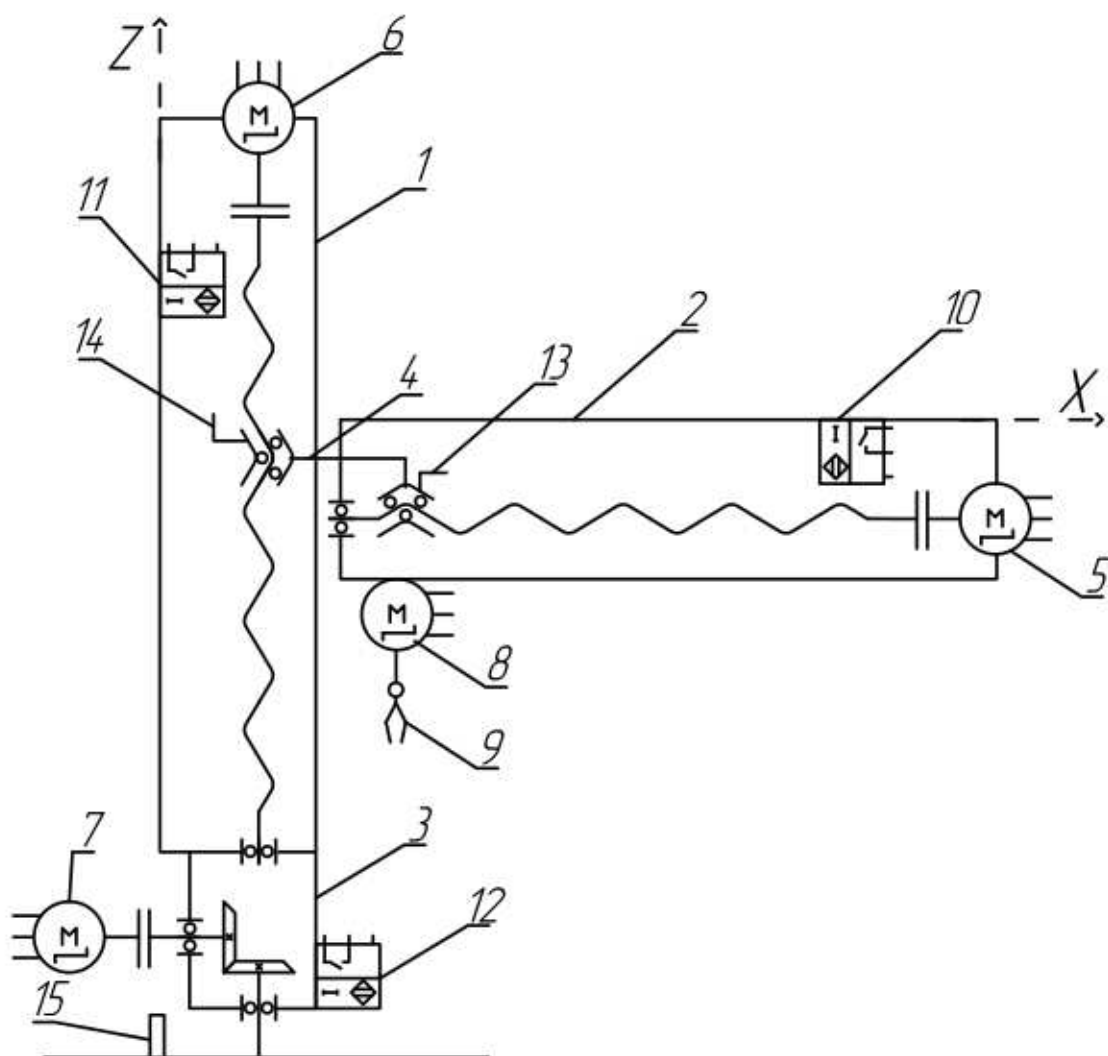


Рисунок 1.9- Кинематическая схема манипулятора

Таблица 1.6 – Спецификация к кинематической схеме манипулятора

1	Башня, обеспечивает движение вдоль оси Z (по вертикали)
2	Рабочая стрела, обеспечивает движение вдоль оси X
3	Опорно-поворотное устройство, вращение башни вокруг оси Z
4	Каретка ШВП осей X и Z
5	ШД (GD57STH56-2909A), привод вылета стрелы
6	ШД (GD57STH56-2909A) привод подъема стрелы
7	ШД (GD57STH56-2006A) привод поворота стрелы
8	Сервопривод MG996R
9	Схват
10,11,12	Индуктивный датчик (LJ8A3-2-Z/AX) конечного положения по осям X, Y, Z соответственно
13,14,15	Флажок конечного положения по осям X, Y, Z соответственно

2 ВЫБОР ЭЛЕКТРОННЫХ КОМПОНЕНТОВ

В данном разделе будет описан выбор необходимых компонентов для разработки системы управления.

2.1 Выбор драйверов шагового двигателя

Шаговый двигатель – двигатель со сложной схемой управления, которому требуется специальное электронное устройство – драйвер шагового двигателя. Драйвер шагового двигателя получает на входе логические сигналы STEP/DIR, которые, как правило, представлены высоким и низким уровнем опорного напряжения 5 В, и в соответствии с полученными сигналами изменяет ток в обмотках двигателя, заставляя вал поворачиваться в соответствующем направлении на заданный угол. Сигналы STEP/DIR генерируются ЧПУ-контроллером или персональным компьютером, на котором работает программа управления.

Задача драйвера – изменять ток в обмотках как можно более эффективно, а поскольку индуктивность обмоток и ротор гибридного шагового двигателя постоянно вмешиваются в этот процесс, то драйверы весьма отличаются друг от друга своими характеристиками и качеством получаемого движения. Ток, протекающий в обмотках, определяет движение ротора: величина тока задает крутящий момент, его динамика влияет на равномерность и т.п.

2.1.1 Принцип действия ШИМ

ШИМ (шиотно-импульсная модуляция) – это современный метод управления мощностью подаваемой нагрузке, он заключается в изменении продолжительности импульса при постоянной частоте их следования. Принцип ШИМ стал основным для электронных устройств, в которых требуется поддержание и регулировка на заданном уровне выходных параметров. Этот метод также применяется для изменения скорости вращения двигателей. Наглядный принцип действия ШИМ показан на рисунке 2.1.

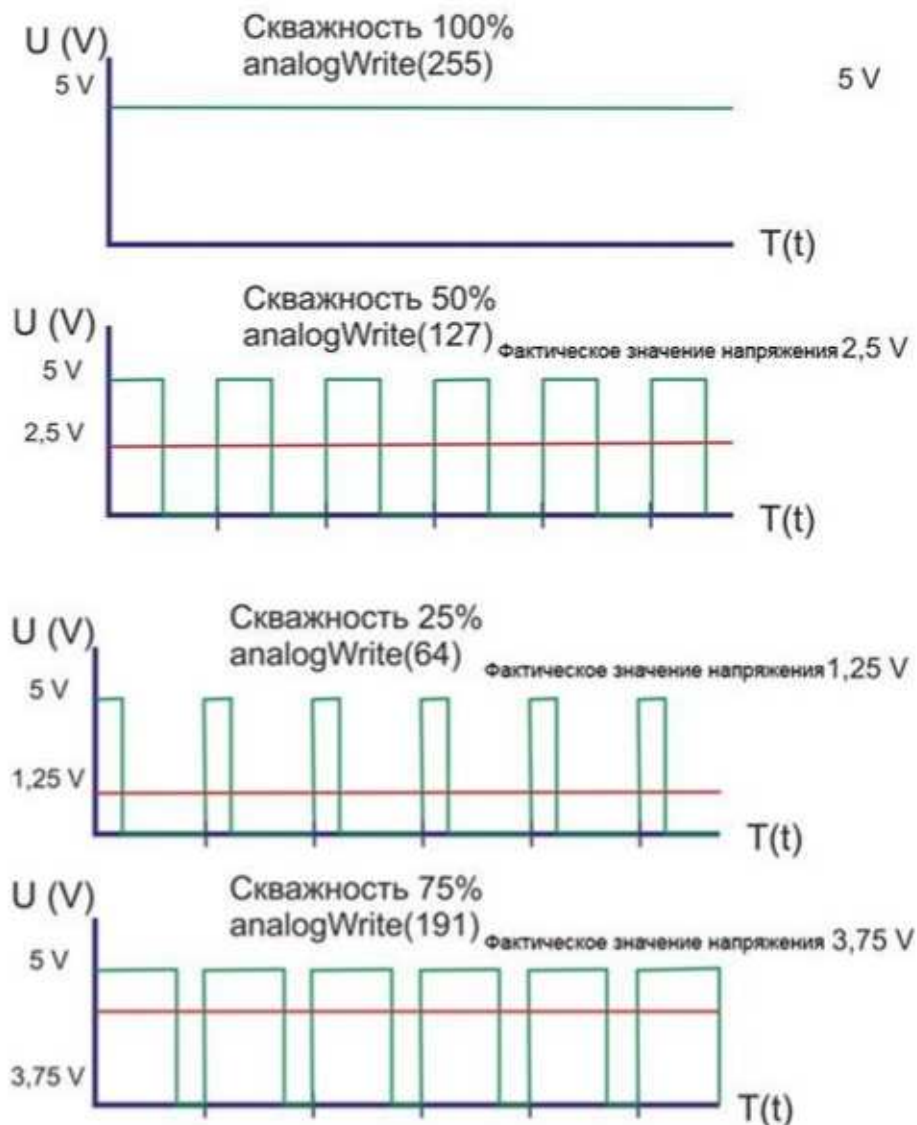


Рисунок 2.1 – Принцип действия ШИМ

Широтно-импульсная модуляция – основные параметры:

- 1) Период тактирования – промежутки времени, через которые подаются импульсы.
- 2) Длительность импульса – время пока подается сигнал.
- 3) Скважность – соотношение длины импульса к импульсному периоду тактования.

2.1.2 Виды драйверов шаговых двигателей

Драйверы делятся по способу подачи тока в обмотки на несколько видов:

1) Драйверы постоянного напряжения

Эти драйверы подают постоянный уровень напряжения поочередно на обмотки, результирующий ток зависит от сопротивления обмотки, а на высоких

скоростях – и от индуктивности. Эти драйверы крайне неэффективны, и могут быть использованы только на очень малых скоростях.

2) Двухуровневые драйверы

В драйверах этого типа ток в обмотке сперва поднимается до нужного уровня с помощью высокого напряжения, затем источник высокого напряжения отключается, и нужная сила тока поддерживается источником малого напряжения. Такие драйверы достаточно эффективны, помимо прочего они снижают нагрев двигателей, и их все еще можно иногда встретить в высококлассном оборудовании. Однако, такие драйверы поддерживают только режим шага и полушага.

3) Драйверы с ШИМ

На текущий момент ШИМ-драйверы шаговых двигателей наиболее популярны, практически все драйверы на рынке – этого типа. Эти драйверы подают на обмотку шагового мотора ШИМ-сигнал очень высокого напряжения, которое отсекается по достижению током необходимого уровня. Величина силы тока, по которой происходит отсечка, задается либо потенциометром, либо DIP-переключателем, иногда эта величина программируется с помощью специального ПО. Эти драйверы достаточно интеллектуальны, и снабжены множеством дополнительных функций, поддерживают разные деления шага, что позволяет увеличить дискретность позиционирования и плавность хода. Однако, ШИМ-драйверы также весьма сильно отличаются друг от друга. Помимо таких характеристик, как питающее напряжение и максимальный ток обмотки, у них отличается частота ШИМ. Лучше, если частота драйвера будет более 20 кГц, и вообще, чем она больше – тем лучше. Частота ниже 20 кГц ухудшает ходовые характеристики двигателей и попадает в слышимый диапазон, шаговые моторы начинают издавать неприятный писк. Драйверы шаговых двигателей вслед за самими двигателями делятся на униполярные и биполярные. Начинающим станкостроителям настоятельно рекомендуем не экспериментировать с приводами, а выбрать те, по которым можно получить максимальный объем технической поддержки, информации и для которых продукты на рынке

представлены наиболее широко. Такими являются драйверы биполярных гибридных шаговых двигателей.

2.1.3 Выбор драйверов шаговых двигателей

Основной параметр при выборе драйвера – это сила тока, которую может обеспечивать драйвер. Как правило она регулируется в достаточно широком диапазоне. Но драйвер следует выбирать такой, который может выдать максимальную силу тока на 15-40% больше чем фазный ток шагового двигателя. С одной стороны, это даст запас на случай, если потребуется получить больший момент от мотора, с другой стороны не будет излишней. Производители зачастую выпускают драйверы, подходящие к определенному виду/размеру двигателей. Слишком мощный драйвер может вызывать излишнее вибрации.

Второй не менее важный фактор – напряжение питания. Его влияние весьма многогранно – напряжение питания влияет в основном на динамику, вибрации, нагрев двигателя и драйвера. Обычно максимальное напряжение питания драйвера примерно равно максимальному току I , умноженному на 8-10.

Наличие опторазвязанных входов. Практически во всех драйверах и контроллерах, выпускаемых на заводах, тем более брендовых, опторазвязка стоит обязательно, ведь драйвер – устройство силовой электроники, и пробой ключа может привести к мощному импульсу на кабелях, по которым подаются управляющие сигналы, и выгоранию дорогостоящего ЧПУ-контроллера.

Наличие механизмов подавления резонанса. Резонанс шагового двигателя – явление, которое проявляется всегда, разница только в резонансной частоте, которая прежде всего зависит от момента инерции нагрузки, напряжения питания драйвера и установленной силы тока фазы мотора. При возникновении резонанса шаговый двигатель начинает вибрировать и терять крутящий момент, вплоть до полной остановки вала. Для подавления резонанса используется микрошаг и встроенные алгоритмы компенсации резонанса. Колеблющийся в резонансе ротор шагового двигателя порождает микроколебания ЭДС индукции в обмотках, и по их характеру и амплитуде драйвер определяет, есть ли резонанс и насколько он силен. В зависимости от полученных данных драйвер несколько

смещает шаги двигателя во времени относительно друг друга – такая искусственная неравномерность нивелирует резонанс.

Протокольная часть. Надо убедиться, что драйвер работает по нужному протоколу, а уровни входных сигналов совместимы с требуемыми логическими уровнями. Подавляющее число драйверов работает по протоколу STEP/DIR/ENABLE и совместимо с уровнем сигналов 0..5 В.

Наличие защитных функций. Среди них защита от превышения питающего напряжения, тока обмоток (в т.ч. от короткого замыкания обмоток), от переполюсовки питающего напряжения, от неправильного подключения фаз шагового мотора.

Наличие микрошаговых режимов. Сейчас практически в каждом драйвере есть множество микрошаговых режимов. Микрошаг полезен не только повышением дискретности позиционирования или точности, а тем, чем больше деление шага, тем плавней движение вала мотора и меньше резонанс. Соответственно, при прочих равных условиях стоит использовать больший делитель шага.

TB6560 - драйвер управления двухфазными шаговыми двигателями выполнен на специализированном чипе Toshiba TB6560AHQ с питанием 10В – 35В постоянного напряжения, предназначен для использования с двигателями типа NEMA17 – NEMA23 с максимальным током фазы до 3А. Широко используется в ЧПУ системах. Вид платы представлен на рисунке 2.2.

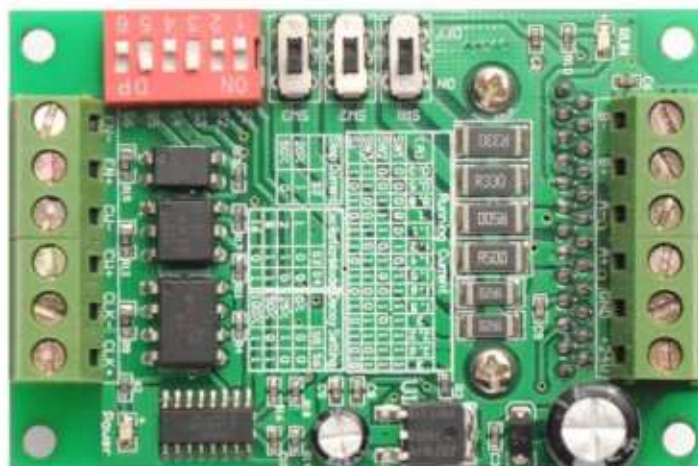


Рисунок 2.2 – Драйвер ШД TB6560

Особенности:

- напряжение питания 10В – 35В постоянного напряжения;
- оптоизолированные входы сигналов управления;
- делитель шага (микрошаг) – 1,2,8,16;
- установка максимального тока – 14 ступеней.

Назначение всех разъёмов платы драйвера приведена в таблице 2.1.

Таблица 2.1 – Маркировка и назначение разъемов драйвера TB6560

Маркировка	Описание
CLK+, CLK-	Входы тактового сигнала
CW+, CW-	Входы управления направлением вращения
EN+, EN-	Входы разрешения работы
+24V, GND	Входы питания
A+, A-	Выход управления обмоткой А
B+, B-	Выход управления обмоткой В

Переключатели на плате служат для настройки её работы.

Переключателями S3, S4 отвечают за установку делителя шага (микрошага), в соответствии с таблицей 2.2.

Таблица 2.2 – Установка микрошага на плате TB6560

Делитель	S3	S4
1	ОТКЛ	ОТКЛ
2	ВКЛ	ОТКЛ
8	ВКЛ	ВКЛ
16	ОТКЛ	ВКЛ

Переключатель S2 отвечает за установку удерживающего тока в режиме покоя, в соответствии с таблицей 2.3.

Таблица 2.3 – Установка тока в режиме покоя на плате TB6560

Значение	S2
20%	ВКЛ
50%	ОТКЛ

Переключатели SW1-SW3 и S1 устанавливают выходной ток для вращения двигателя в соответствии с таблицей 2.4.

Таблица 2.4 – Установка тока в режиме вращения на плате TB6560

Выходной ток (A)														
(A)	0,3	0,5	0,8	1,0	1,1	1,2	1,4	1,5	1,6	1,9	2,0	2,2	2,6	3,0
SW1	0	0	0	0	0	1	0	1	1	1	1	1	1	1
SW2	0	0	1	1	1	1	1	0	0	1	0	1	1	1
SW3	1	1	0	0	1	0	1	1	0	0	1	1	0	1
S1	1	0	1	1	1	1	0	1	0	1	0	1	0	0

В драйвере есть возможность настроить форму дискретных импульсов для формирования сглаженного сигнала, близкого к синусоиде. Это необходимо из-за различий в параметрах двигателей и режимах их работы. Параметр «Decay» определяет наклон горизонтальной части импульса после переднего фронта. Для прямоугольного импульса значение «Decay» равно 0 %, а для треугольного импульса – 100 %. Эта функция полезна для выбора оптимального режима работы шагового привода и помогает улучшить работу двигателя, снизить шум и вибрации. Переключатели S5, S6 изменяют это значение в соответствии с таблицей 2.5.

Таблица 2.5 – Затухание дискретных импульсов на плате TB6560

Decay	S5	S6
0%	откл	откл
25%	вкл	откл
50%	откл	вкл
100%	вкл	вкл

2.1.4 Подключение и настройка драйверов

Для управления драйвером необходимо подавать импульсы нужной частоты от контроллера. Чтобы двигатель совершил один полный шаг на его обмотки поступает серия сигналов от драйвера. Большинство драйверов позволяют разделить полный шаг двигателя на несколько микрошагов. Выбор микрошага устанавливается согласно таблице в инструкции к драйверу. Если для целого поворота ротора двигателя в режиме 1 микрошаг на полный шаг требуется 200 импульсов, то в режиме 4 микрошага на полный шаг, потребуется уже 800 тактов.

Чем больше микрошагов в полном шаге, тем точнее и плавнее

поворачивается ротор шагового двигателя, но для поддержания той же скорости, требуется увеличивать частоту следования тактовых импульсов. Силовые выводы используются для подачи напряжения питания шагового двигателя и подключения его обмоток. Далее представлена схема подключения драйвера (рисунок 2.3).

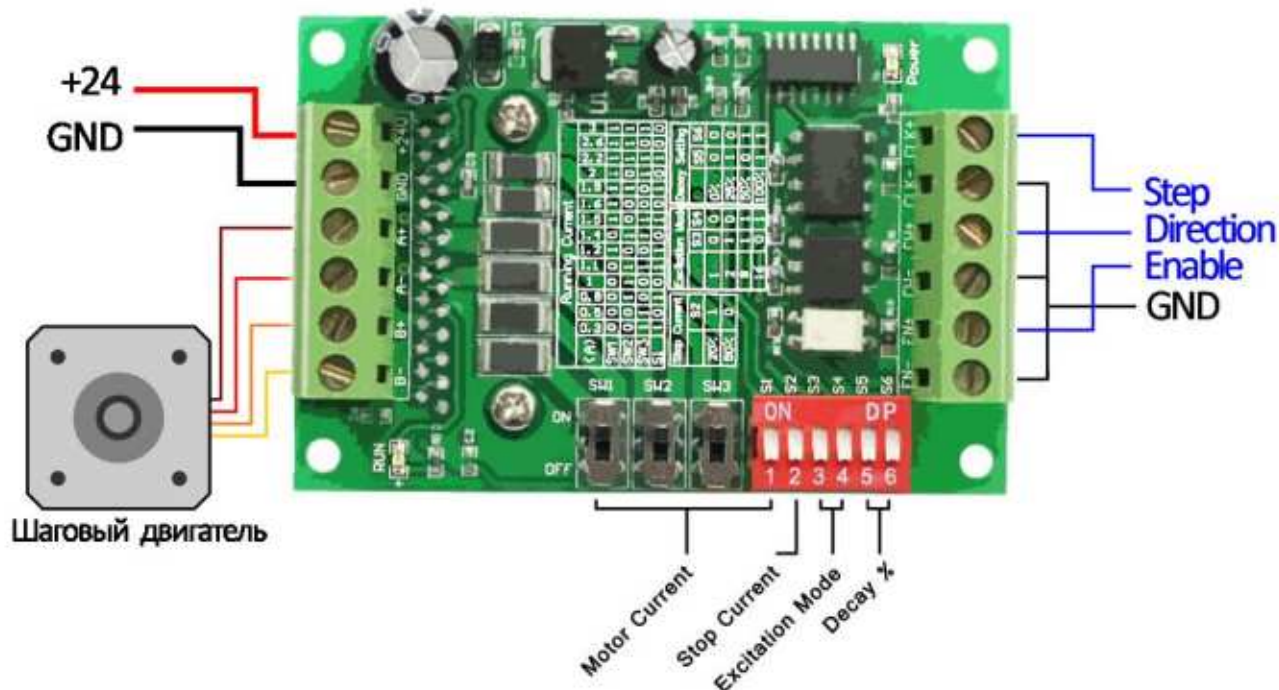


Рисунок 2.3 – Схема подключения драйвера

Силовые выводы драйвера:

- входы «vcc», «gnd» / «+v», «gnd» / «ac+», «ac-» - предназначены для получения напряжения питания шагового двигателя;
- выводы «a+» и «a-» - предназначены для подключения первой обмотки шагового двигателя;
- выводы «b+» и «b-» - предназначены для подключения второй обмотки шагового двигателя;
- подключение обмоток двигателя к драйверу зависит от количества выводов у двигателя.

Сигналы управления STEP/DIR (PUL/DIR)

Вход драйвера «STEP» (он же «PULSE») - предназначен для получения тактовых импульсов. За один импульс ротор двигателя поворачивается на один микрошаг. Вход может работать по фронту или спаду импульса. Чем выше

частота импульсов, тем выше скорость вращения ротора.

Вход драйвера «DIR» – предназначен для выбора направления вращения двигателя («0» – в одну сторону, «1» – в другую сторону). Смена состояния вывода «DIR» должна осуществляться при отсутствии импульсов на выводе «STEP».

Вход драйвера «ENABLE» – разрешает работу двигателя. У большинства драйверов данный вход является инверсным, работа двигателя разрешена при отсутствии напряжения на входе. Некоторые драйверы позволяют вообще не подключать этот вход. Если работа двигателя запрещена, то его обмотки электрически отключаются и вал двигателя не удерживается. Двигатель отключён если на входе «ENABLE» есть напряжение.

Сигналы на входах «STEP» и «DIR» игнорируются драйвером. Вал двигателя освобождён.

Вал поворачивается на один микрошаг с каждым импульсом на входе «STEP», при условии, что на входе «ENABLE» нет напряжения.

Направление поворота вала зависит от состояния на входе «DIR».

Вал двигателя удерживается в неподвижном состоянии если на входе «ENABLE» нет напряжения и на вход «STEP» не подаются импульсы.

Сигналы управления CW/CCW

Вход драйвера «CW» – предназначен для получения тактовых импульсов. За один импульс ротор двигателя поворачивается на один микрошаг. Вход может работать по фронту или спаду импульса. Чем выше частота импульсов, тем выше скорость вращения ротора.

Вход драйвера «CCW» – выполняет те же действия что и вход «CW», но ротор двигателя поворачивается в другую сторону.

Вход драйвера «ENABLE» – разрешает работу двигателя. У большинства драйверов данный вход является инверсным, работа двигателя разрешена при отсутствии напряжения на входе. Некоторые драйверы позволяют вообще не подключать этот вход. Если работа двигателя запрещена, то его обмотки электрически отключаются и вал двигателя не удерживается.

2.2 Выбор датчиков конечного положения для манипулятора

Датчик конечного положения необходим, чтобы предотвратить удары тележки стрелы об корпус. Сигналы индуктивного датчика используются контроллером, чтобы вовремя остановить привод. Был выбран индуктивный датчик LJ8A3-2-Z/AX.

Концевой бесконтактный активный датчик индуктивного типа. Металлический корпус с резьбой M8x1,0 для монтажа в отверстие. Схема датчика показана на рисунке 2.4. Основные характеристики датчика приведены в таблице 2.6.

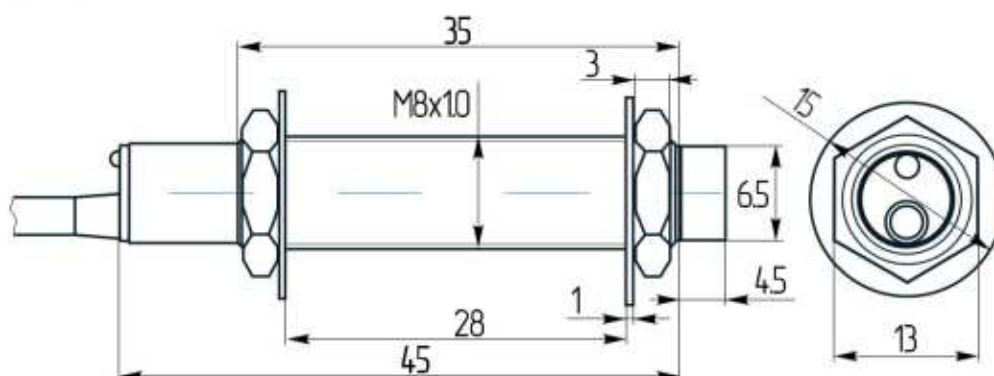


Рисунок 2.4 – Схема датчика LJ8A3-2-Z/AX

Тип NPN, NO (нормально замкнутый - при срабатывании выход отсоединяется от минуса питания, нормально разомкнутый - при срабатывании выход соединяется с минусом питания).

Таблица 2.6 – Технические характеристики датчика LJ8A3-2-Z/AX

Параметр	Значение
Вид	Индуктивный
Тип	NPN, NC
Напряжение	6-36 В
Частота срабатывания	До 500 Гц
Дистанция	2 мм
Материал реагирования	Сталь, алюминий медь, свинец, латунь, бронза
Количество проводов	3

Датчики серий SN, LJ и GB это герметичные индуктивные бесконтактные выключатели, реагирующие на приближения металлов (стали, алюминия, меди, бронзы, свинца и пр.). Принцип действия датчика: внесение в высокочастотное электромагнитное поле, создаваемое датчиком объекта, проводящего электрический ток, вызывает потери энергии на возникающие в нем вихревые токи. Величина потерь зависит от расстояния между объектом и датчиком. Бесконтактные выключатели, как следует из названия, работают в двух режимах («есть объект/нет объекта»), открывая или закрывая выходной транзистор (рисунок 2.5).

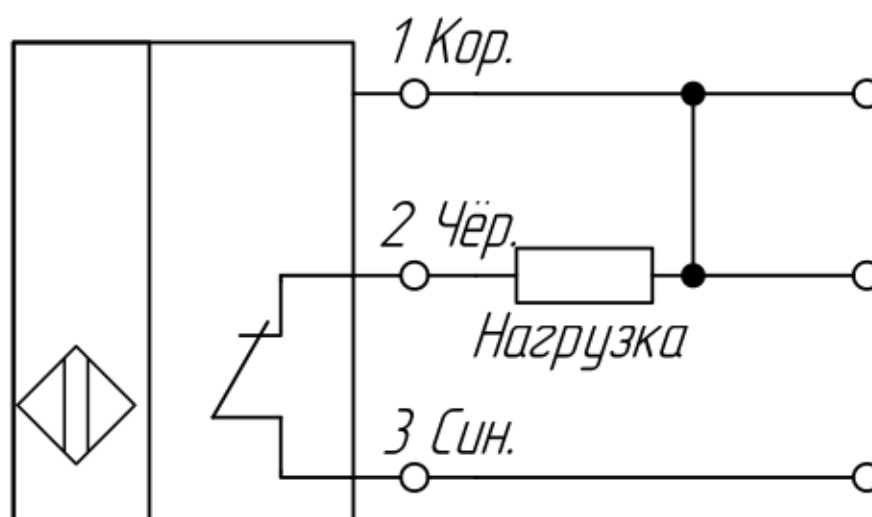


Рисунок 2.5– Схема подключения датчика LJ8A3-2-Z/AX

2.2.1 Расчет токоограничивающего резистора и делителя напряжения

Микроконтроллер выдерживает на своих входах до 12 мА, выход источника питания 5 В. Рассчитаем сопротивление резистора по закону Ома.

$$R_{огр} = \frac{U_{пит}}{I} \quad (2.1)$$

где $U_{пит}$ – напряжения источника питания;

I – максимально допустимый ток.

$$R_{огр} = \frac{5}{0,012} = 417 \text{ Ом} \quad (2.2)$$

Подберем существующий резистор большего сопротивления – 560 Ом. Чтобы снизить напряжение на входе микроконтроллера, нужно установить второй резистор для делителя напряжения номиналом в два раза больше этого. Если возьмем существующий резистор на 1,1 кОм, получим следующий результат:

$$U_{\text{мк}} = \frac{R_{\text{дел}}}{R_{\text{дел}} + R_{\text{огр}}} * U_{\text{пит}} \quad (2.3)$$

где $U_{\text{мк}}$ – напряжения логического уровня для микроконтроллера;
 $R_{\text{дел}}$ – сопротивление резистора для делителя напряжения.

$$U_{\text{мк}} = \frac{1100}{1100 + 560} * 5 = 3,31 \text{ В}$$

Напряжение на входах микроконтроллера должно быть не больше 3,3 В. Поэтому необходимо подобрать резистор другого сопротивления. Лучше использовать сопротивление, чтобы уровень напряжения был примерно на 5% меньше чем 3,3 В. Возьмем резистор 900 Ом:

$$U_{\text{мк}} = \frac{900}{900 + 560} * 5 = 3,08 \text{ В}$$

Такая конфигурация резисторов подходит, микроконтроллер сможет зафиксировать логический уровень единицы и не сгорит.

Однако, после проведения опыта, оказалось, что мощности на выходе датчика недостаточно, чтобы создать нагрузку на делителе. Поэтому такая схема оказалась неприменимой.

Чтобы подключить датчики к микроконтроллеру и сохранить нужные уровни напряжения для высокого и низкого логических уровней, нужно было разработать схему с оптической развязкой.

2.3 Разработка платы опторазвязки

Для реализации платы были выбраны подходящие оптроны РС817. И резисторы для подтяжки к плюсу питания от микроконтроллера номиналом в 5кОм. В дальнейшем их нужно было расположить согласно разработанной схеме на одной печатной плате для удобного монтажа. Схема подключения датчиков

вместе с платой показана на рисунке 2.6. Нормально закрытые Индуктивные датчики имеют обозначение на схеме BB1..BB3. Провод с положительным знаком идет к источнику питания 12В, специально выбранному для питания данных датчиков. Остальные два провода попадают на плату с оптической развязкой. К каждому датчику подключена своя оптопара, а на выходе транзистор формирует сигнал микроконтроллеру. Такой подход позволяет использовать источники питания с разными напряжениями.

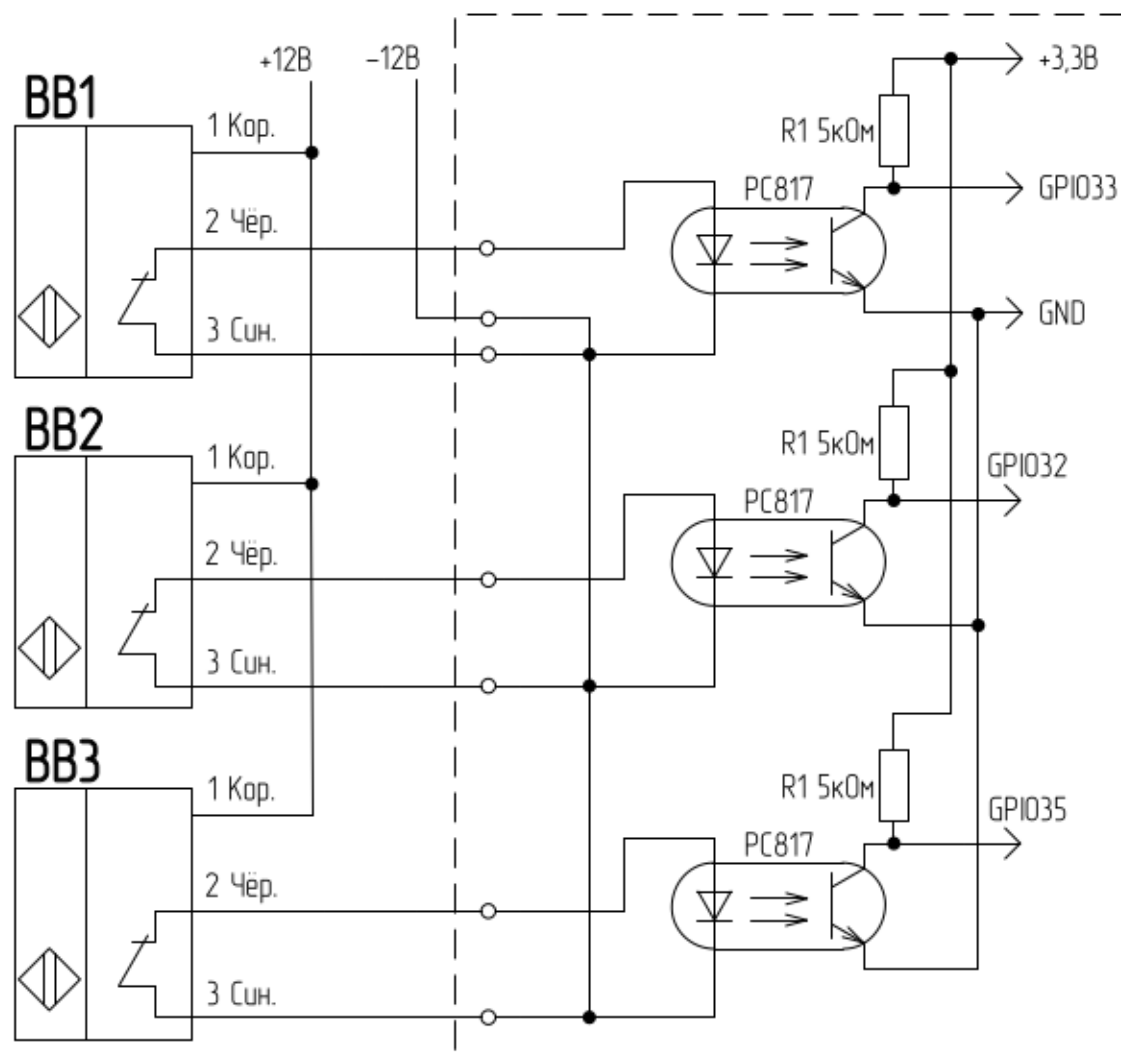


Рисунок 2.6 – Схема подключения датчиков с помощью платы опторазвязки
Данная схема была реализована в среде Proteus, рисунок 2.7.

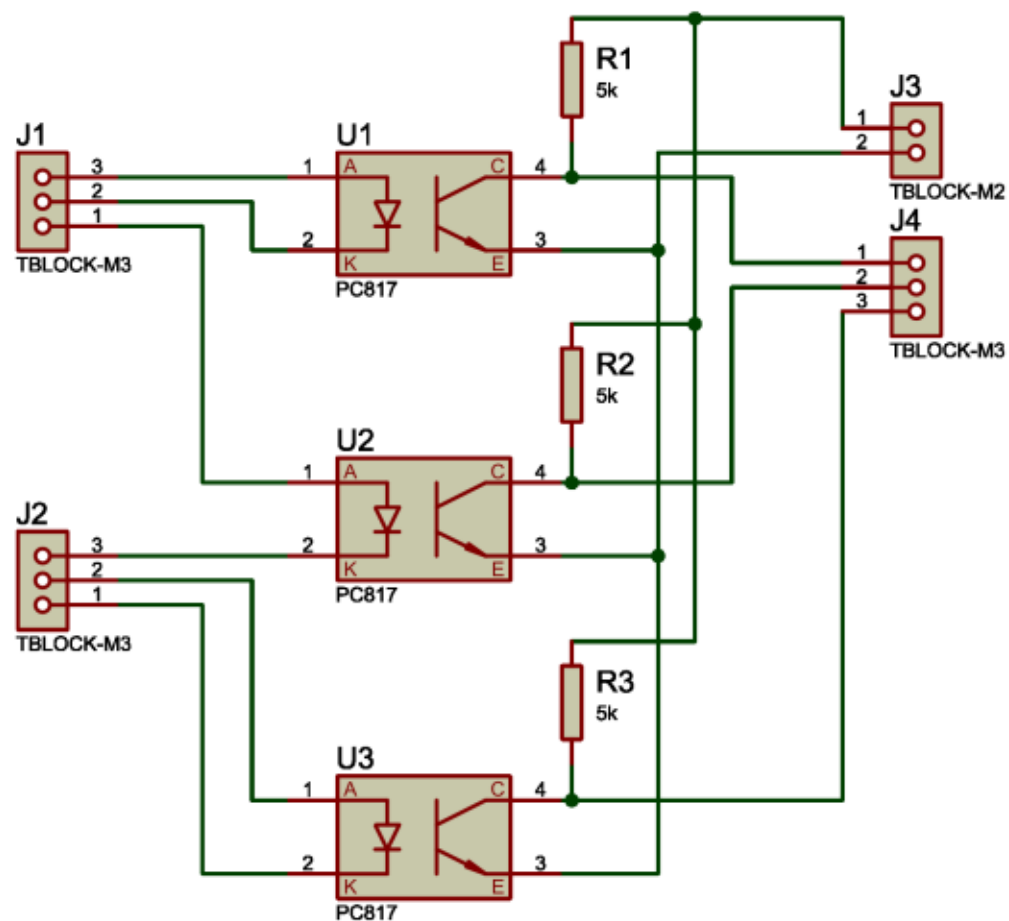


Рисунок 2.7 – Схема опторазвязки в Proteus

Чтобы перенести спроектированную плату на текстолит, применялась технология лазерно-утюжной печати. Подготовленная к печати плата изображена на рисунке 2.8. Здесь плата нарочно отзеркалена. Но здесь была допущена ошибка при экспорте в Proteus'a, на печати останутся не только дорожки, но и контуры от корпусов деталей. После травления платы их нужно будет убрать. Плата с дорожками показана на рисунке 2.9.

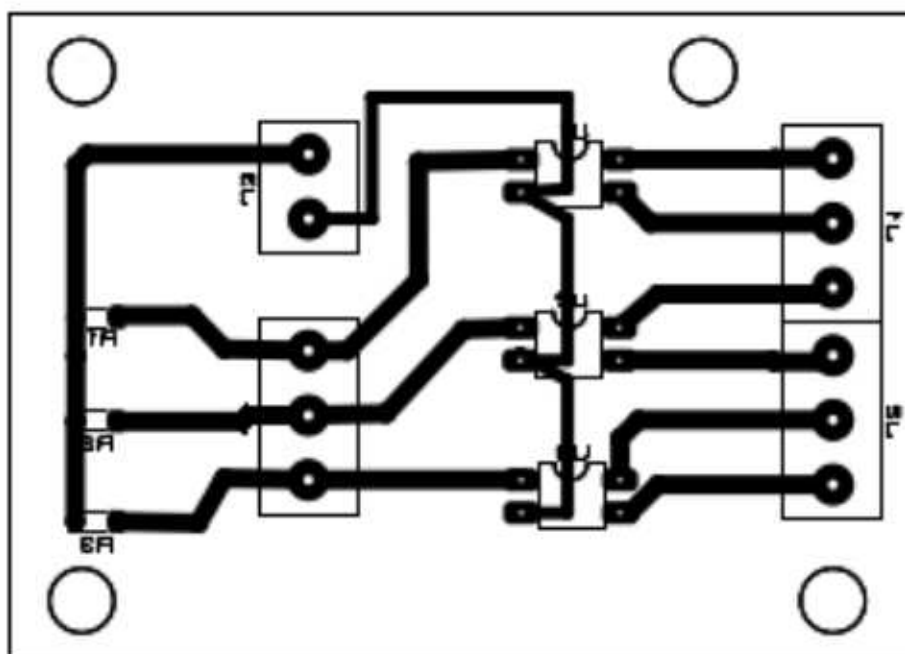


Рисунок 2.8 – Схема подготовленная для печати

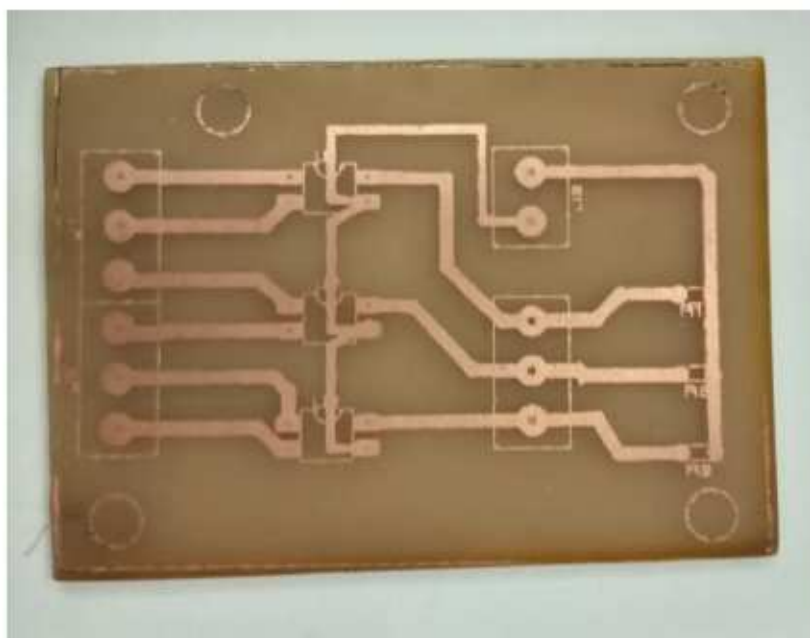


Рисунок 2.9 – Плата с дорожками после травления

Замыкающие контуры от корпусов необходимо убрать, например соскрести ножом. После чего просверливаются отверстия для монтажа клеммников на плате и самой платы. Затем можно приступить к пайке деталей. Готовая плата показана на рисунке 2.10.

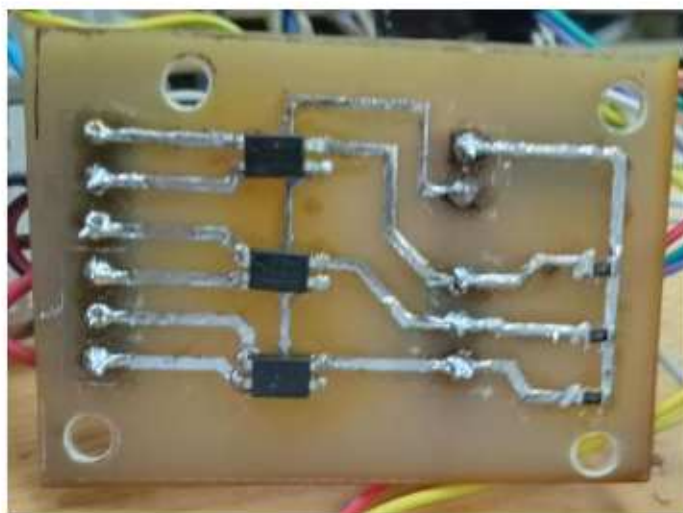


Рисунок 2.10 – Готовая плата оптической развязки

Особенностью этой платы является то, что все дорожки проходят на ней лишь с одной стороны, поэтому чтобы установить клеммники их нужно было размещать с обратной стороны. Хотя плата и выглядит привлекательно, она исправна выполняет свои функции.

2.4 Выбор контроллера для управления манипулятором

В качестве контроллера для управления манипулятором была выбрана микроконтроллерная плата Esp32-WROOM DevKit-V1. В плате используется микрочип ESP32-D0WDQ6. Основные характеристики микроконтроллера приведены в таблице 2.7.

Таблица 2.7 – Характеристики микроконтроллера ESP32

Параметр	Значение
1	2
Частота беспроводной передачи	2,4 ГГц
Тактовая частота	До 240 МГц
Flash-память	448 КБ
Внешняя Flash-память	4 МБ
Выводы общего назначения	25 ввода-вывода(GPIO)
Входов с АЦП	15
Разрядность АЦП	12
Контактов с ЦАП	2

1	2
Разрядность ЦАП	8 бит
Напряжение логических уровней	3,3 В
Максимальный ток на вводах/выводах	12 мА
Максимальный выходной ток вывода 3V3	1 А
Входное напряжение Vin	5-14 В

Внешний вид микроконтроллерной платы с обозначением её выводов представлен на рисунке 2.11. Плата имеет совсем небольшие размеры. В верхней части платы расположена антенна для сетей Wi-Fi и Bluetooth. Особенностью данной платы является выходное напряжение на её выходах – 3.3 В. Этого не всегда достаточно для запитывания периферийных устройств. Поэтому к данной плате дальше будет подобран преобразователь для 5 В. Он же будет выдавать напряжение для питания самой платы.

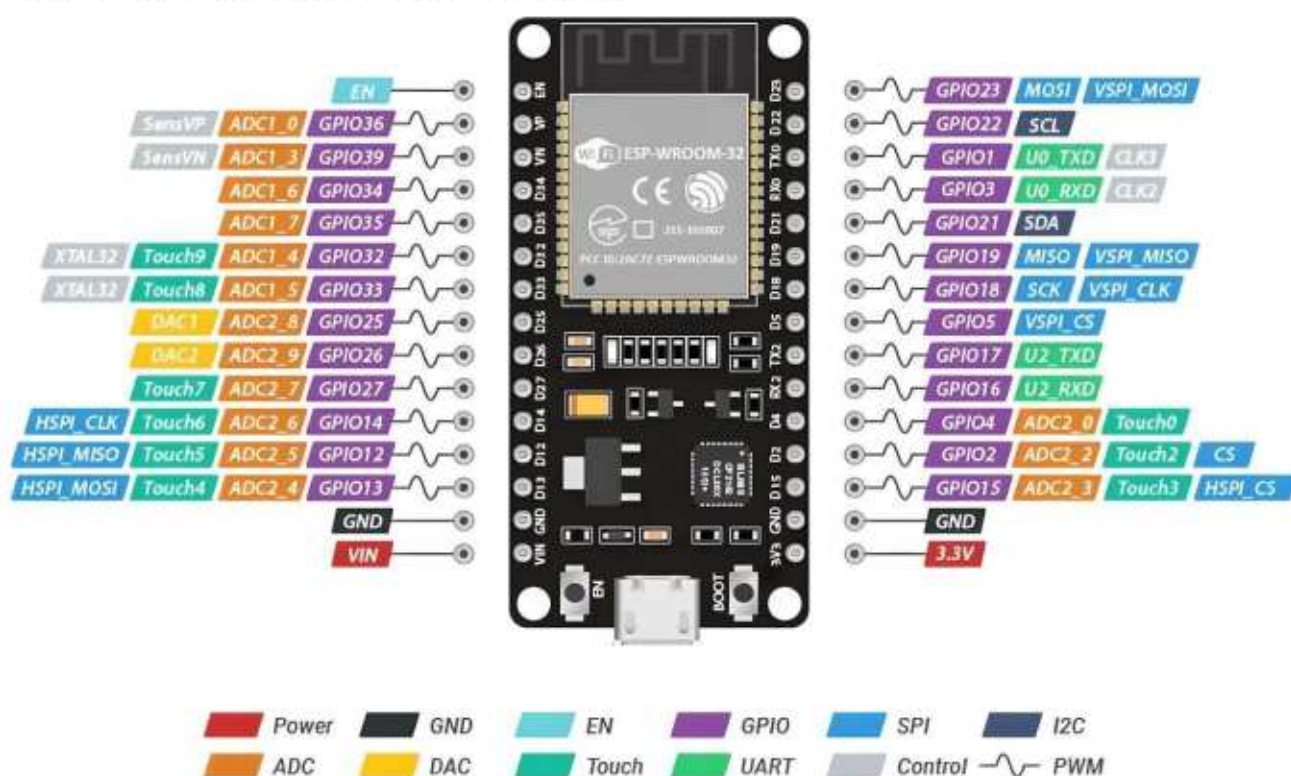


Рисунок 2.11 – Плата с чипом ESP32

2.5 Выбор понижающего преобразователя для 5 вольт

Для питания микроконтроллерной платы и сервоприводов манипулятора

был выбран понижающий регулируемый DC-DC преобразователь HW-319 с вольтметром. Графическое описание преобразователя приведено на рисунке 2.12.

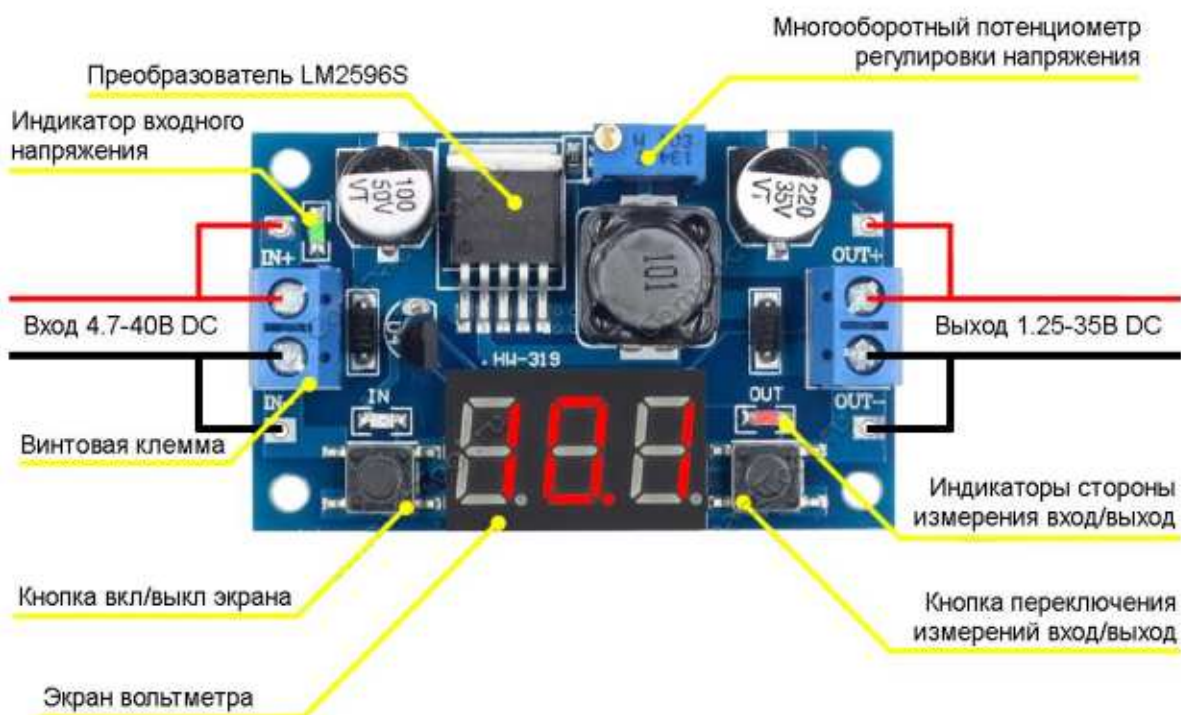


Рисунок 2.12 – преобразователь HW-319

Понижающий регулятор напряжения HW-319 представляет собой устройство на базе микросхемы LM2596S, предназначенное для интеграции в различные электронные схемы. Его основная функция – стабилизация и понижение напряжения. Устройство работает в широком диапазоне: входное напряжение от 4.7 В до 40 В, выходное – от 1.25В до 35В. При напряжении ниже 4.5 В происходит автоматическое отключение.

Преобразователь HW-319 не имеет встроенного регулируемого ограничителя по току нагрузки. В нормальных условиях (~25 °C) эксплуатации рекомендованный ток на выходе, без установки на чип LM2596S дополнительного активного или пассивного воздушного охлаждения, не должен превышать 1-1,5 А. Для нагрузок с током до 3 А – должно быть установлено охлаждение. Полный список характеристик приведен в таблице 2.8.

Таблица 2.8 – Характеристики преобразователя HW-319

Параметр	Значение
Энергопотребление в режиме ожидания	Менее 20 мА
Входное напряжение	4,7 В ~ 40 В DC
Блокируемое напряжение	менее 4.5 В
Выходное напряжение	1.25 ~ 35 В DC
Ток нагрузки	до 2 А (3 А с радиатором)
Падение напряжения (+25°C)	0,7 ± 0,3 В
Чип преобразователя	LM2596S
Коэффициент эффективности (КПД)	макс.: 93 %
Частота преобразования ШИМ	150 ± 15% кГц
Встроенные функции	вольтметр
Шаг измерения	0,1 В
Погрешность	± 0.5В
Способ настройки	вращение подстроечного винта многооборотного потенциометра
Количество оборотов потенциометра	25 ± 3
Защита от	переплюсовки на входе
Рабочая температура	-25 °C ~ +75 °C
Размеры	56,2 x 35,5 x 11,8 мм
Вес	20 г

Особенностью данной модели является наличие встроенного цифрового вольтметра, который позволяет отслеживать как входное, так и выходное напряжение. Вольтметр оснащен трехсимвольным светодиодным индикатором красного цвета с высокой яркостью, что обеспечивает хорошую видимость даже при ярком освещении. Погрешность измерений составляет ±0,5 В.

Понижающий преобразователь имеет 3 светодиодных индикатора.

Зеленый светодиод показывает о наличии напряжения на входе преобразователя. Светодиоды красного цвета IN и OUT определяют входную/выходную сторону регулятора, в текущий момент измеряемую вольтметром. Кнопки цифрового управления платой расположены по бокам семи-сегментного экрана, первая со стороны клемм входного напряжения, вторая – со стороны выходного напряжения. Левая кнопка ON/OFF рядом с клеммами входа включает и выключает экран, правая кнопка IN/OUT рядом с клеммами выхода переключает показания вольтметра между входным (IN) и выходным (OUT) напряжением. При повторном включении сохраняются настройки отображения для последних изменений.

Подключение и регулировка напряжения на выходе преобразователя

Контакты источника исходного напряжения и тока зажимаются в винтовом клеммнике группы входных контактов IN+/IN-. Цепь для потребителя подключается с другой стороны OUT+/OUT-. Установка требуемого значения на выходе преобразователя осуществляется регулировкой подстроечного винта, переменного резистора. Направление вращения по часовой стрелке - увеличение уровня выходного напряжения, против часовой стрелки - уменьшение. На весь диапазон регулируемого напряжения, от минимального до максимального уровня, потребуется примерно 25 ± 3 полных оборотов подстроечного винта. Регулировка напряжения на выходе преобразователя должна выполняться обязательно без нагрузки.

Во время проведения регулировки преобразователя HW-319 следует брать во внимание процент эффективности преобразования (КПД LM2596S), который может принимать большее или меньшее значение в зависимости от условий пользования. Из-за небольшого падения напряжения в плате HW-319 на $0.7 \pm 0.3\text{В}$, рекомендуется сохранять разницу напряжений между входом и выходом преобразователя не меньше 1-1.5 В.

Преобразователь выполняет стабилизацию напряжения на выходе (выравнивает на заданном уровне) до тех самых пор, пока уровень входного напряжения остаётся выше выходного порога. Когда напряжение на входе

опускается ниже установленной потенциометром отметки, напряжение на выходе пропорционально снижается. Значение входного напряжения в понижающем преобразователе определяет верхний порог допустимой регулировки.

2.6 Выбор понижающего преобразователя для 12 вольт

Для стабильной работы датчиков потребовался дополнительный преобразователь напряжения, так как индуктивные датчики рассчитаны на напряжение 6 – 36 В. Был выбран понижающий DC-DC преобразователь LM2596S. Такой преобразователь обладает совсем небольшими размерами, его внешний вид представлен на рисунке 2.13.

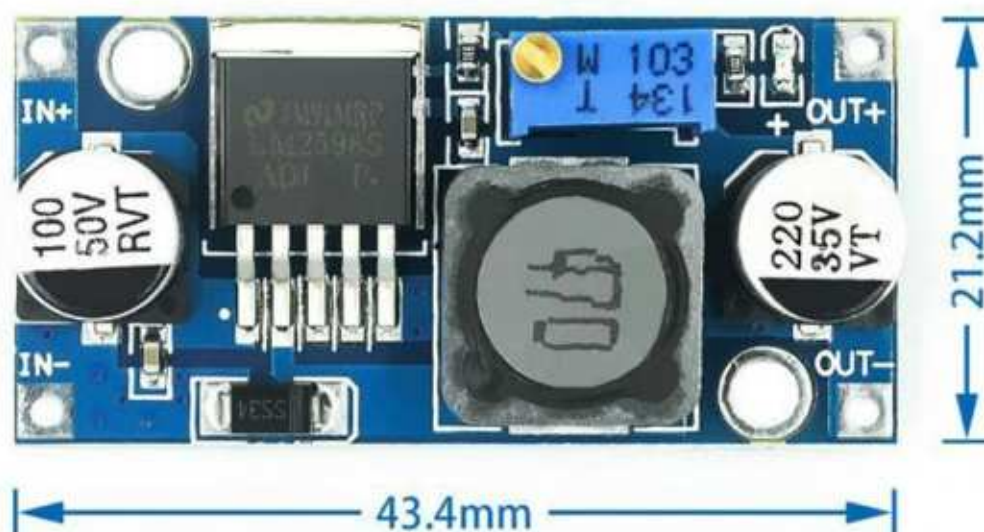


Рисунок 2.13 – Преобразователь LM2596S

Понижающий DC-DC преобразователь LM2596 позволяет получить стабилизированное напряжение постоянного тока на выходе меньше, чем напряжение постоянного тока, поданное на его вход. LM2596S имеет высокий КПД, меньше нагревается по сравнению с модулями на линейных стабилизаторах. Такой преобразователь может применяться в широком спектре устройств. Его характеристики приведены в таблице 2.9.

Таблица 2.9 – Характеристики преобразователя LM2596S

Параметр	Значение
Входное напряжение	3,2 В – 46 В
Выходное напряжение	1,25-35 В
Выходной ток	2 А
Потребление без нагрузки	6 мА
КПД	92 %
Минимально-возможная разница между входным и выходным напряжением	1,5 В

2.7 Выбор камеры с передатчиком

Для получения видео данных о состоянии робота было принято решение подобрать подходящую видеокамеру и установить её на башне манипулятора.

Была выбрана камера LST S2 OSD 5.8G 40CH AIO FPV Camera (аббревиатура FPV означает вид от первого лица, от англ. - First Person View). Внешний вид камеры показан рисунке 2.14.

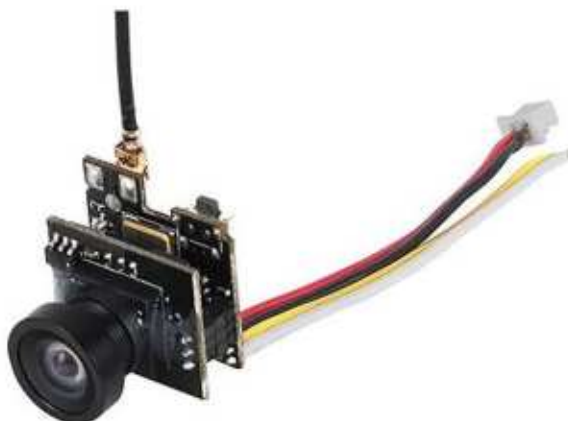


Рисунок 2.14 – камера FPV LST S2 OSD

Камерой с РЧ-передатчиком, функцией наложения данных телеметрии, изначально предназначена для установки в модели летательных аппаратов с дистанционным управлением. Видеоданные передаются камерой LST-S2-OSD на экран FPV-очков или пульта ДУ, позволяет оператору в реальном времени отслеживать состояние системы. Характеристики камеры с передатчиком в

таблице 2.10.

Таблица 2.10 – Характеристики камеры

Параметр	Значение
Питание:	3 ~ 5,5В постоянного тока DC
Потребление тока	450 мА
Частотный диапазон передачи	5.8 ГГц (5,645 ~ 5,945)
Каналы	40 (5 полос x 8 частот)
Угол зрения	150° по горизонтали, 170° по вертикали
Видеосигнал	аналоговый, 800TVL HD
Размеры	16 x 16 x 18 мм

Для монтажа камеры на робота был разработан корпус, с который эта камера отлично входит и теперь может быть прикручена с помощью винта в отведенное ей место на специальном кронштейне, в виде изогнутой стальной полоски с отверстиями для ввинчивания винтов. Корпус для камеры был разработан в программе для моделирования «Blender» и распечатан на 3D-принтер.

3 РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ

Всю систему правления можно поделить на две подсистемы: управление манипулятором и управление мотор колесами.

3.1 Подсистема управления манипулятором

На рисунке 3.1 представлена разработанная схема подсистемы управления манипулятором. На схеме приведены:

- 1) микроконтроллер ESP32;
- 2) источник питания 24В;
- 3) преобразователь 24-5В;
- 4) шаговые двигатели;
- 5) драйверы для шаговых двигателей;
- 6) сервоприводы захвата.

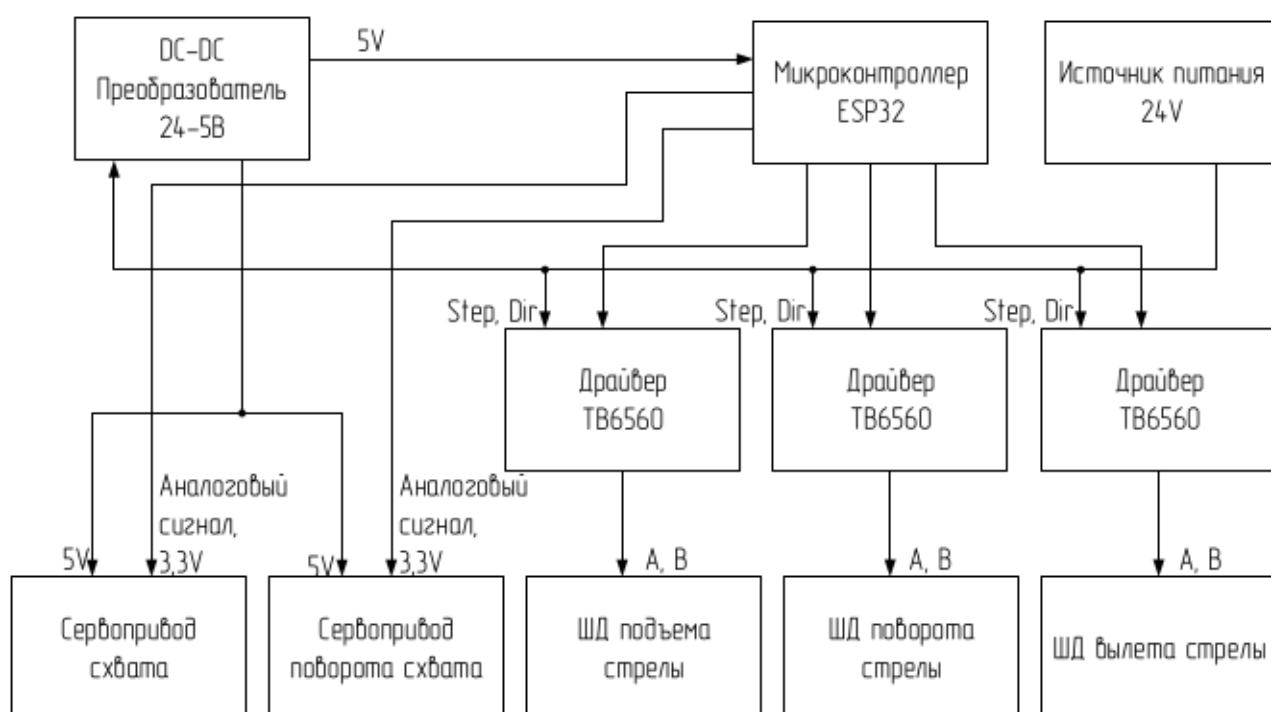


Рисунок 3.1 – Структурная схема подсистемы управления манипулятором

В данной системе предусмотрены три шаговых двигателя для изменения положения манипулятора по осям X, Y, Z. С помощью захвата, робот может перемещать грузы в любое доступное место. Рука захвата работает на двух сервоприводах, где один отвечает за сжатие схвата, а другой за поворот руки на угол до 90 градусов в обе стороны в плоскости стрелы манипулятора.

Драйверы, указанные в схеме, необходимы для того, чтобы приводить в движение шаговые двигатели. Плата драйвера, в отличие от микроконтроллера, может выдавать нужные токи и напряжения, необходимые для создания момента на шаговом двигателе.

Микроконтроллер ESP32, управляет всей подсистемой по установленной программе, команды управления пользователь задает через интерфейс на персональном компьютере.

Для питания платы с микроконтроллером и модулей сервоприводов используется преобразователь HW-319 DC-DC. Преобразователь настроен на 5В.

Для повышения надежности системы были установлены датчики конечного положения по трём осям, в которых применяются шаговые двигатели. Схема с применением датчиков показана на рисунке 3.2.

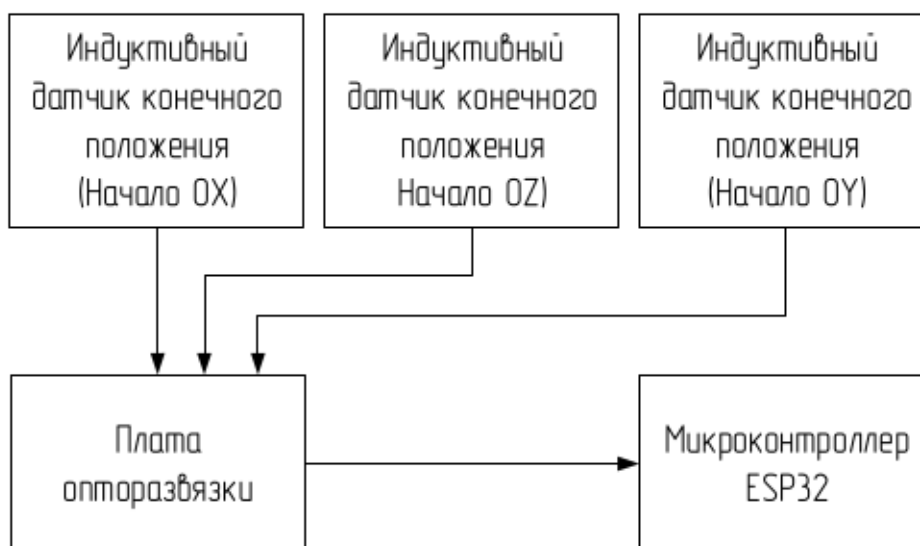


Рисунок 3.2 – Структурная схема блока датчиков

Концевые индуктивные датчики LJ8A3-2-Z/AX должны оповестить микроконтроллер об приближении к концу рабочей области звеньев манипулятора, в случае пропуска шагов по каким-либо причинам. По сигналу индуктивного датчика контроллер должен оставить движение чтобы предотвратить аварийную ситуацию.

3.2 Подсистема управления мотор-колесами

Для управления мотор-колесами в мобильном роботе используются два драйвера STM32. Они выполняют функцию коммутации обмоток моторов,

чтобы колёса могли вращаться с определённой скоростью в заданном направлении. Структурная схема управления колёсами показана на рисунке 3.3.

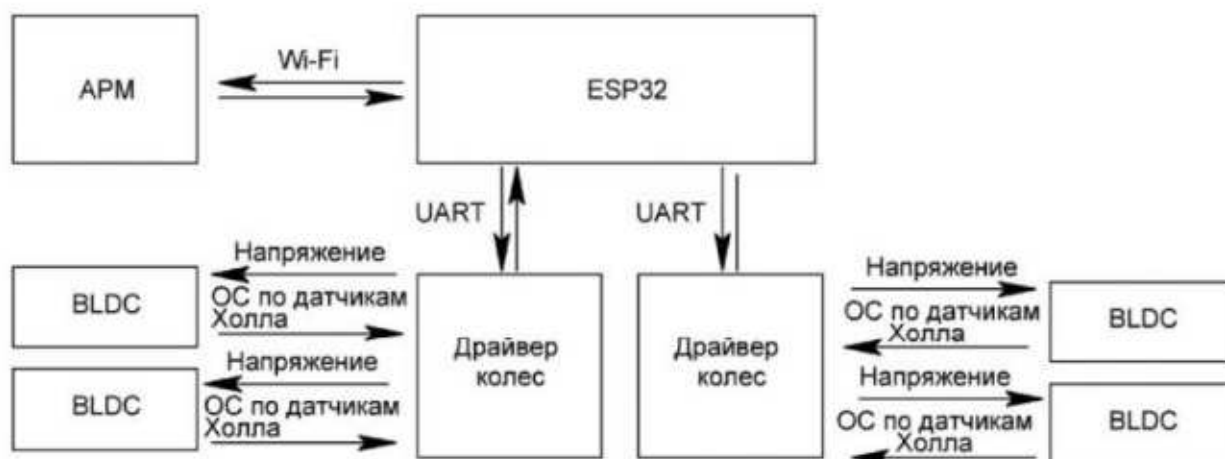


Рисунок 3.3 – Структурная схема управления мотор-колесами

3.1 Система электропитания мобильного робота

В процессе разработки в проекте развилась сложная схема электропитания, с применением множества источников:

- 220 В AC;
- 36 В DC;
- 24 В DC;
- 12 В DC;
- 5 В DC.

Поэтому была разработана дополнительная схема, рисунок 3.4. Её основной смысл наглядно продемонстрировать взаимосвязь электропитания системы.

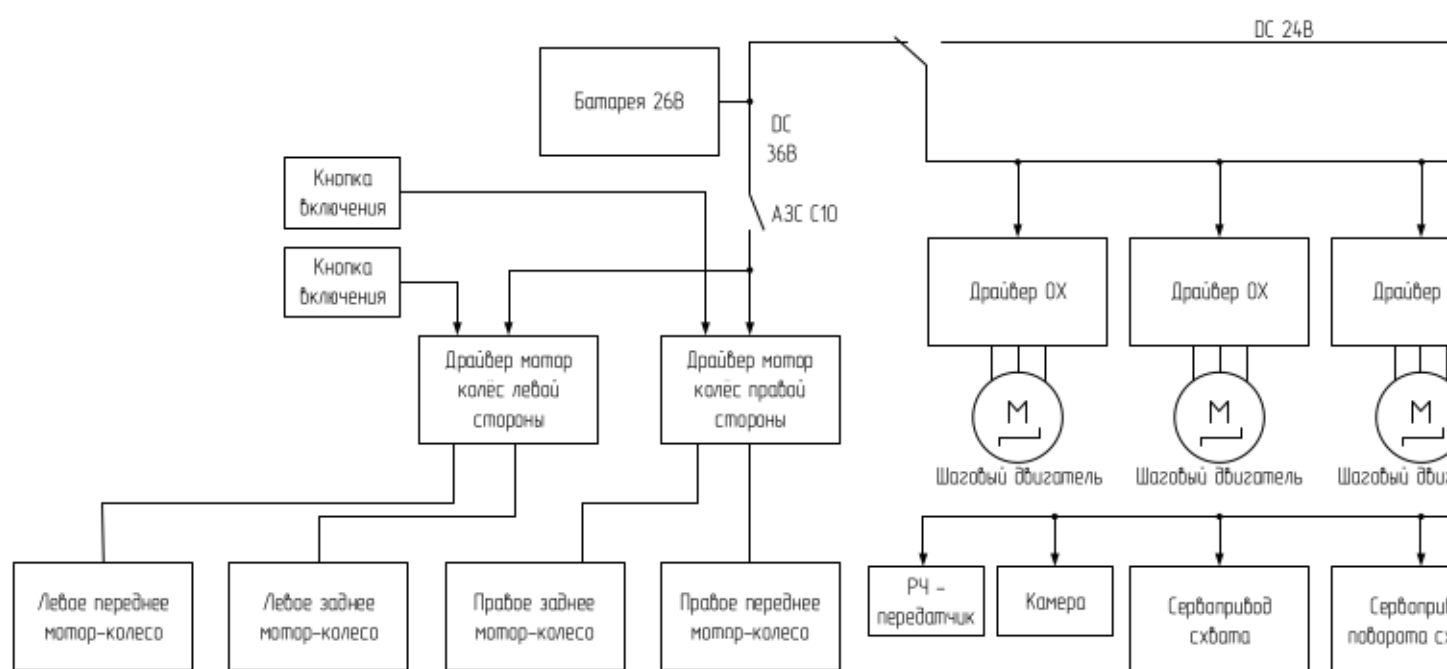


Рисунок 3.4 – Структурная схема электропитания системы

3.2 Электрическая схема

В данном разделе будет описан принцип действия отдельных участков схемы подсистемы управления манипулятором. На рисунке 3.5 приведена схема подключения драйверов к микроконтроллеру ESP32.

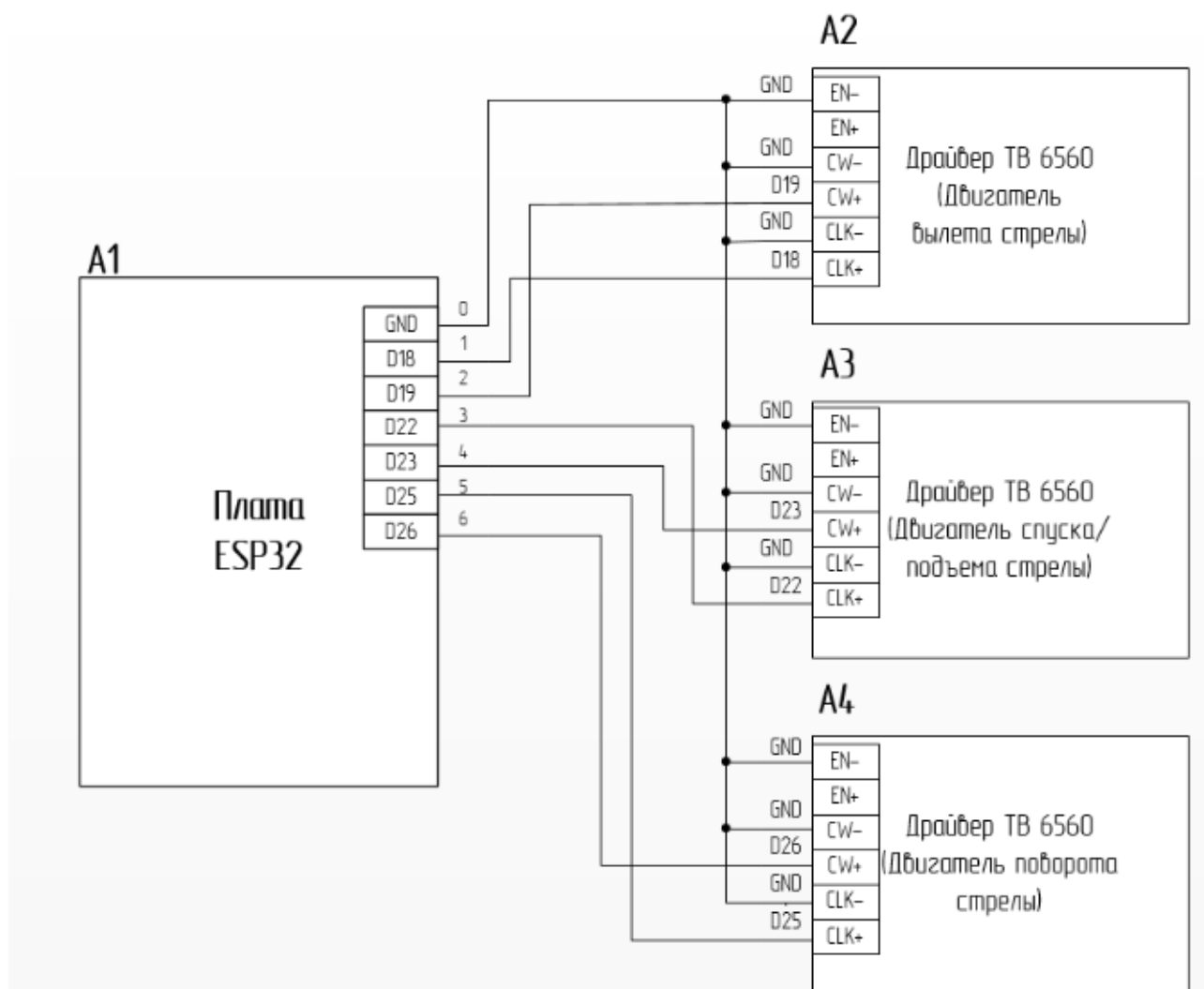


Рисунок 3.5 – Схема подключения драйверов к микроконтроллеру

На схеме представлен принцип подключения драйверов к микроконтроллеру с указанием всех задействованных входов, входы EN+ на плате драйверов не задействуются в проекте, так как при подаче напряжения на эти входы двигатель удерживает вал в неподвижном положении. Манипулятор робота не предусматривает очень большие моменты, к которым эта функция могла бы быть применима.

На рисунке 3.6 показана схема подключения сервоприводов. Первый сервопривод отвечает за сжатие схвата и управляется по сигнальному проводу,

подключенному к пину D2 микроконтроллера, настроенному на аналоговый выход. Второй сервопривод разворачивает схват на угол от 0 до 180 градусов, он подключается аналогично, но только в пину D4. Питание сервоприводы получают от регулируемого преобразователя, выход которого настроен на 5В.

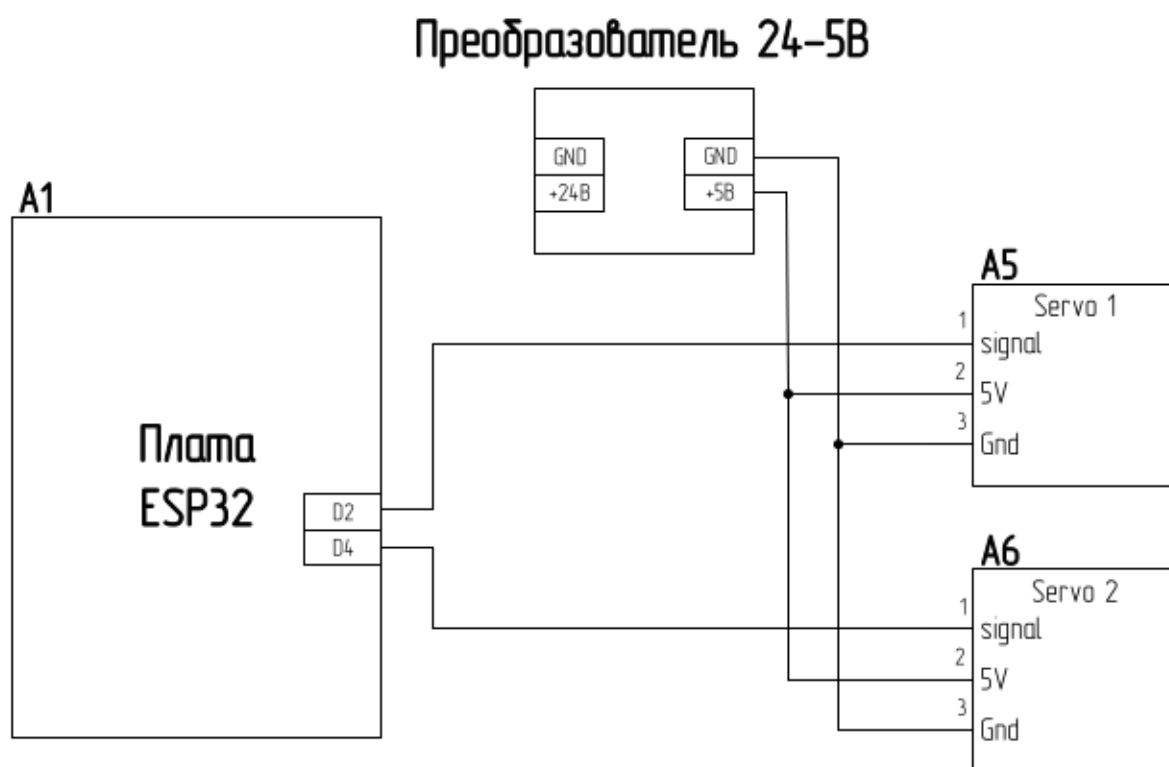


Рисунок 3.6 – Схема подключения сервоприводов манипулятора

На рисунке 3.7 представлена схема подключения шагового двигателя к драйверу, отвечающие за перемещение манипулятора по оси OX. Соединение осуществляется через переходной разъем, поэтому цветность проводов, идущих от драйвера и шагового двигателя отличаются. Шаговый двигатель GD57STH56-2808A имеет внутри четыре обмотки, в проекте используется последовательное соединение обмоток в пары, как показано на схеме.

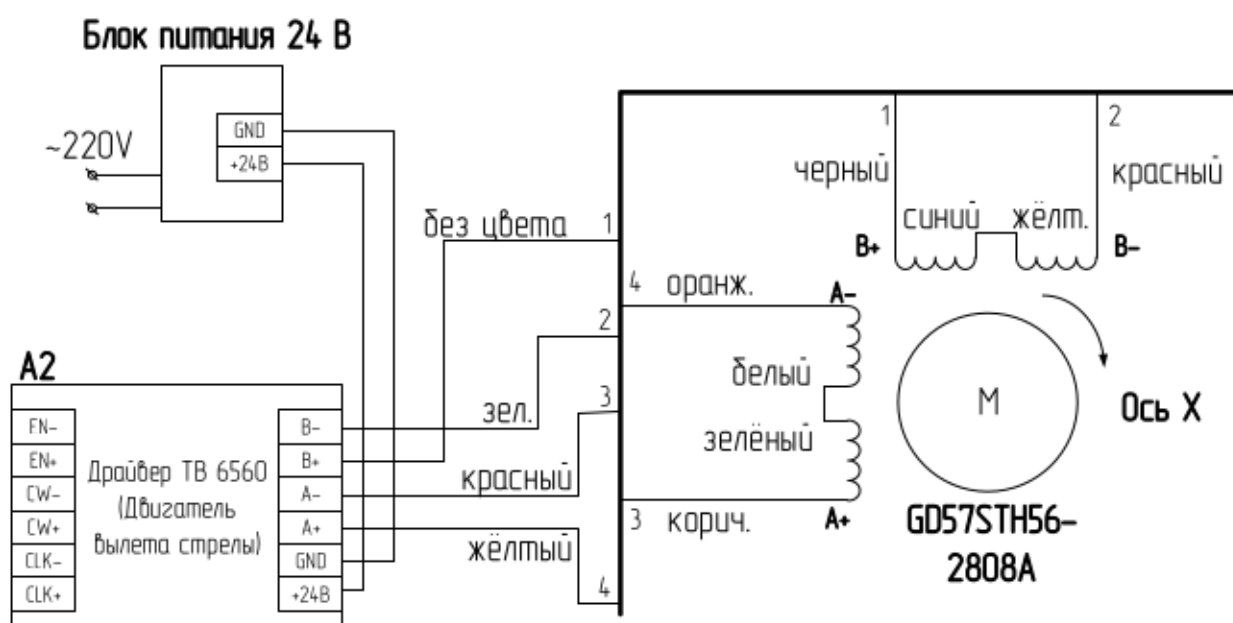


Рисунок 3.7 – Схема подключения драйвера к шаговому двигателю OX

На рисунке 3.8 представлена аналогичная схема подключения шагового двигателя и драйвера, за исключением пары проводников, которые спаянны в переходном разъеме по-другому.

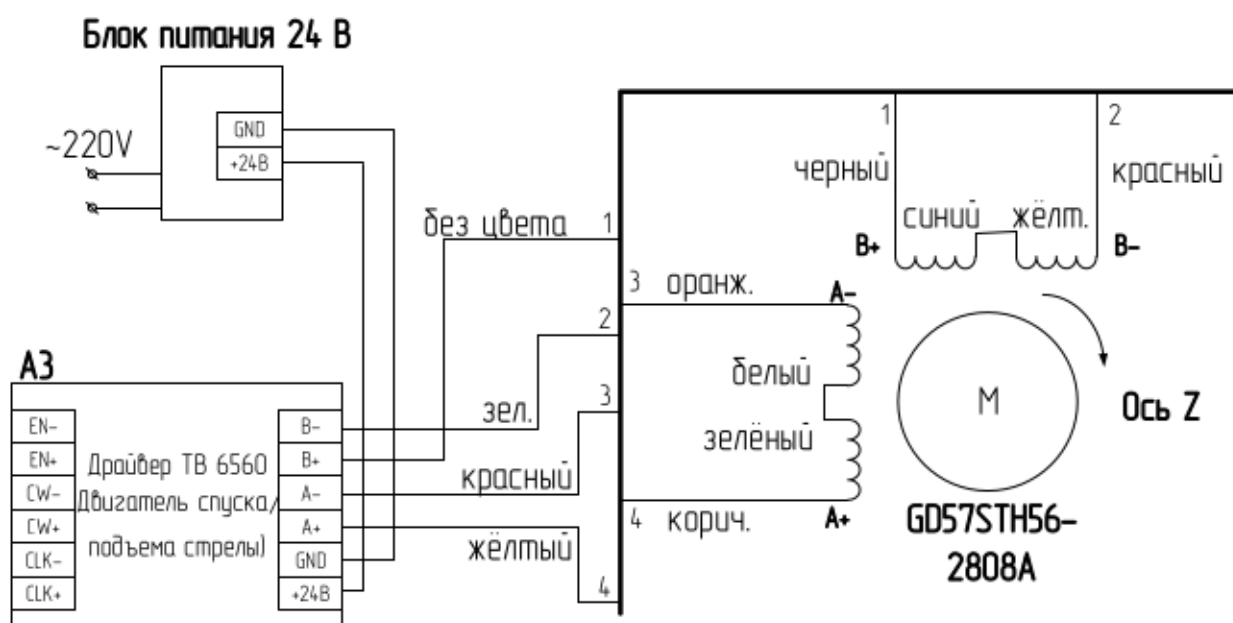


Рисунок 3.8 – Схема подключения драйвера к шаговому двигателю OZ

На рисунке 3.9 показана схема подключения двигателя и драйвера, отвечающих за поворот башни манипулятора, в этой схеме применяется другой шаговый двигатель, с другой последовательностью подключения обмоток.

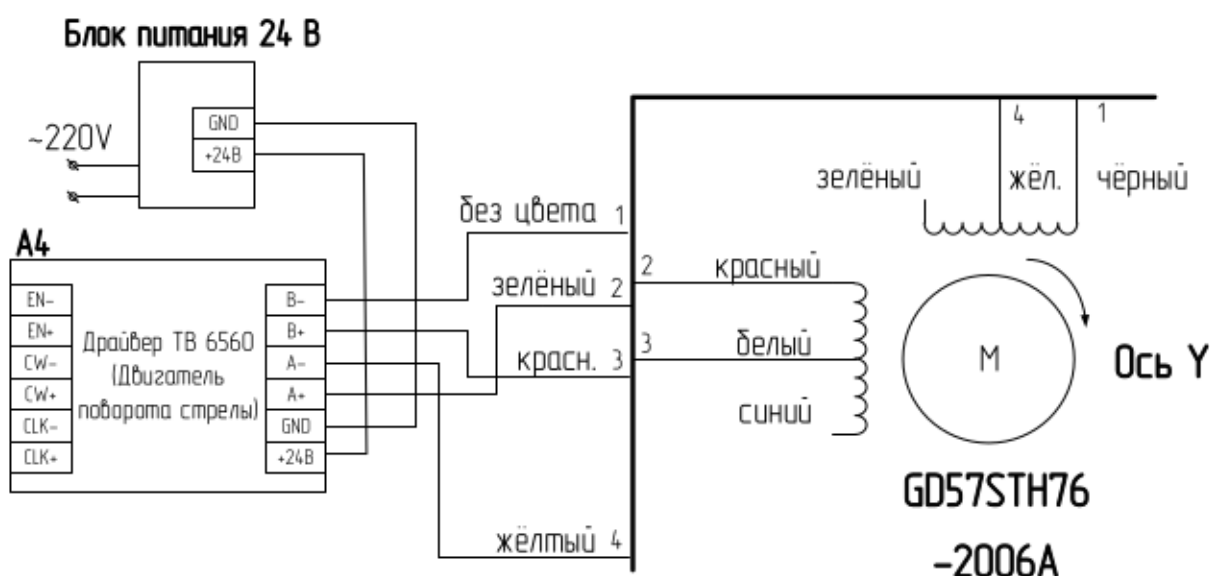


Рисунок 3.9 – Схема подключения драйвера к шаговому двигателю OY

3.3 Разработка логической схемы управления роботом

Управление роботом может осуществляться с помощью пульта. Чтобы управлять роботом, применяя при этом только один пульт, предусмотрено несколько режимов:

- 1) Режим передвижения (схема управления на рисунке 3.11);
- 2) Режим управления манипулятором (схема управления на рисунке 3.12);
- 3) Режим программного управления.

Переключение режимов осуществляется тумблером SWC. Всего на пульте 4 переключателя: SWA, SWB, SWC, SWD. Из всех четырех только SWC может переключаться в трех положениях, остальные могут занимать только два положения. Справа от экрана пульта есть кнопка Power, она включает и выключает пульт. При включении пульта все стики и переключатели должны находиться в своем начальном положении, иначе пульт не будет работать.

Пульт с назначением всех тумблеров показан на рисунке 3.10.



Рисунок 3.10 – Назначение переключателей на RC-пульте

Всего для управления роботом задействуется 7 каналов связи: 4 канала для двух стиков и 3 канала для переключателей. Пульт должен быть настроен на протокол PPM.

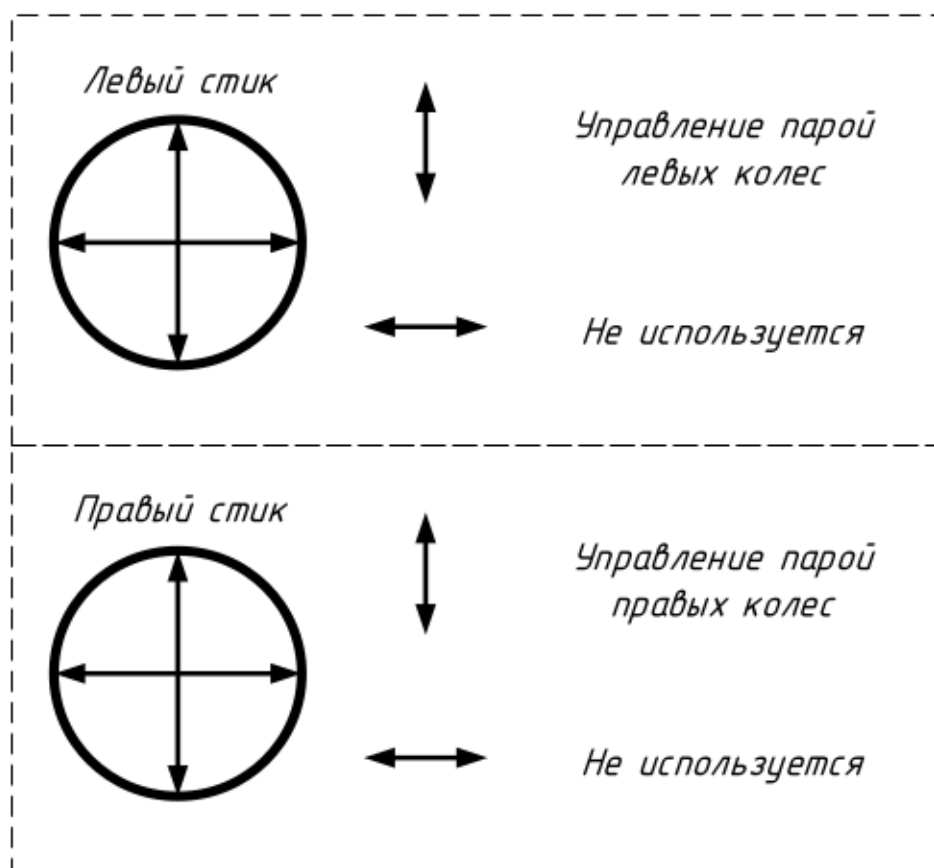


Рисунок 3.11 – Схема управления передвижением робота

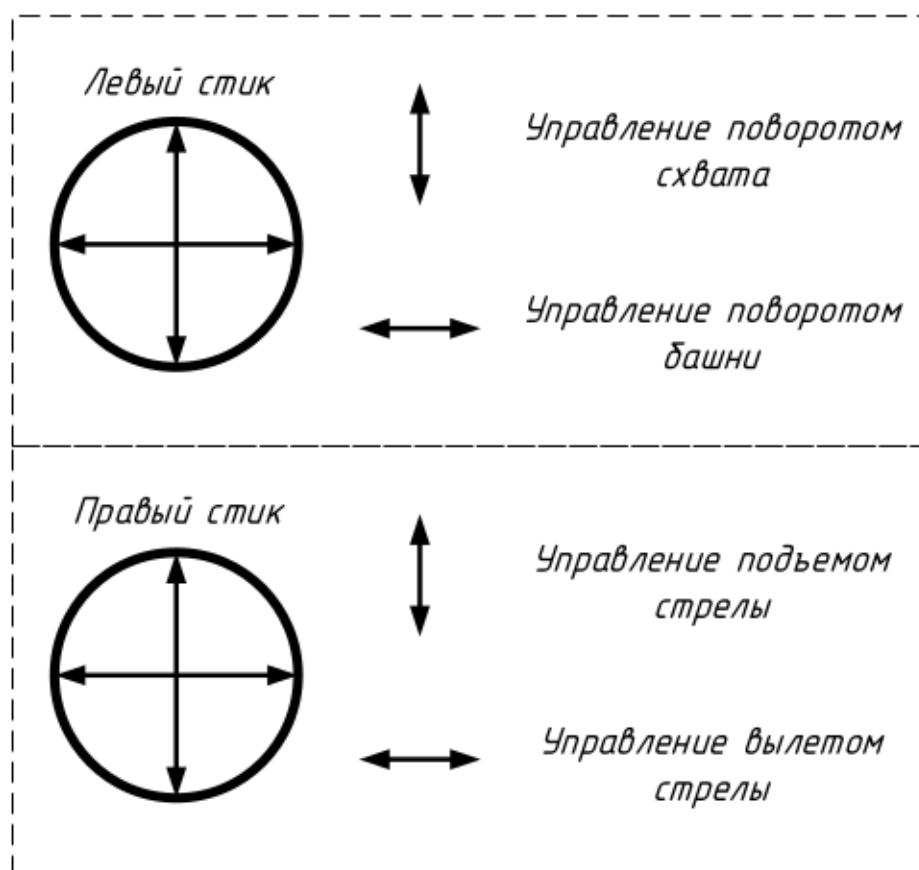


Рисунок 3.12 – Схема управления манипулятором

4 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОБИЛЬНОГО РОБОТА-МАНИПУЛЯТОРА

Каждая из двух подсистем управления работает по собственной программе, независимо от другой.

4.1 Микроконтроллерная программа управления мотор-колесами и протокол связи PPM

Программа для управления BLDC моторами, используемых в качестве привода колёс мобильного робота, включает в себя следующее:

1. Задание скорости вращения мотора;
2. Задание направления вращения колёс;
3. Различные режимы для работы моторов: режим постоянной скорости, постоянной мощности;
4. Защита оборудования от короткого замыкания, перегрева и перегрузки;
5. Возможность интеграции с другими системами.

Для ручного управления мотор-колесами применяется RC-пульт. Такие пульта широко используется для различных радиоуправляемых систем. Для передачи сигналов пультом используется метод PPM. Сигнал PPM представляет собой последовательность импульсов постоянной длительности, с определенным периодом следования – временным промежутком между импульсами. Длительностью этих промежутков кодируются передаваемые значения. Преимущество данного метода над PPM заключается в основном в количестве проводов, которые необходимо подключать к микроконтроллеру. Для PPM это один провод для каждого канала, а для PWM всего один проводник для всех каналов сразу.

Для микроконтроллера реализована программа для считывания данных из этих импульсов. Чтобы считать переданное значение по прерываниям вызывается функция. Она реализует считывание периодов следования импульсов в микросекундах.

```

////////////////////////////////PPM Interrupt////////////////////////////////////////
void IRAM_ATTR onPPMInterrupt() {
    uint32_t now = micros();
    uint32_t duration = now - lastTime;
    lastTime = now;

    if (duration >= SYNC_GAP) {
        // Sync pulse detected, reset channel index
        currentChannel = 0;
    } else if (currentChannel < NUM_CHANNELS) {
        ppmValues[currentChannel] = duration;
        currentChannel++;
    }
}

```

В основном цикле программы полученные импульсы обрабатываются в команды для драйверов.

4.2 Микроконтроллерная программа управления манипулятором

Для микроконтроллера ESP32 была разработана программа управление в среде Arduino IDE. Далее будет приведено описание алгоритма.

```

//////////////// Настройка выводов //////////////////////////////////////////
#define STEPx 18
#define DIRx 19
#define STEPz 22
#define DIRz 23
#define STEPy 25
#define DIRy 26
#define Grab 2
#define GrabTurn 4
#define SensX_s 33
#define SensZ_s 32
#define SensY_s 35
//Настройки PPM
#define PPM_PIN 15 // Pin connected to the PPM signal
#define NUM_CHANNELS 8 // The number of PPM channels you expect
#define SYNC_GAP 3000 // Minimum pulse length in microseconds that
signals a frame reset
#define PPM_Speed 1000 // Скорость шаговых двигателей при управлении с
пульта

```

В этом разделе объявлены выводы, используемые для управления. На каждый шаговый двигатель используется два вывода. Для сервоприводов используется по одному аналоговому выводу. Протокол PPM реализует общение всего лишь по одному выводу с прерываниями.

```

//////////////// Переменные //////////////////////////////////////////
Thread serialThread = Thread();
AccelStepper stepperX(1, STEPx, DIRx);
AccelStepper stepperZ(1, STEPz, DIRz);
AccelStepper stepperY(1, STEPy, DIRy);

```

```

Servo grabber;
Servo grabberTurn;
bool enable = true;
bool XSensor = false; bool ZSensor = false; bool YSensor = false;
int mod = 0; // Режим
//Переменные для PPM
volatile uint16_t ppmValues[NUM_CHANNELS] = { 0 };
volatile uint8_t currentChannel = 0;
volatile uint32_t lastTime = 0;
hw_timer_t* timer = NULL; // Timer for accurate pulse width
measurement
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;

```

Для удобства и практичности кода реальные объекты управления определены как программные. Здесь описаны три шаговых двигателя и два сервопривода.

```

////////// Настройки точки доступа ESP32 //////////
const char* ssid = "E32 Manipulator"; // Имя точки доступа
const char* password = "12345678"; // Пароль (минимум 8 символов)
WiFiUDP udp;
unsigned int localPort = 1234; // Порт UDP-сервера
String reply;

```

Программа управления связывается с компьютером по сети Wi-Fi. Для этого в программе настраивается точка доступа беспроводной сети. В этом разделе представлены константы и переменные отвечающие за настройку точки доступа. Переменная строка “reply” – хранит в себе ответ на запрос клиента.

```

//////////PPM Interrupt//////////
void IRAM_ATTR onPPMInterrupt() {
    uint32_t now = micros();
    uint32_t duration = now - lastTime;
    lastTime = now;

    if (duration >= SYNC_GAP) {
        // Sync pulse detected, reset channel index
        currentChannel = 0;
    } else if (currentChannel < NUM_CHANNELS) {
        ppmValues[currentChannel] = duration;
        currentChannel++;
    }
}

```

Функция чтения PPM сигнала по прерыванию, здесь последовательно считывается длительность сигналов для всех каналов в момент прерывания.

```

////////// setup //////////
void setup() {
    grabber.setPeriodHertz(50); // standard 50 hz servo
    grabber.attach(Grab, 500, 2500); // attaches the servo on pin 2 to
the servo object
    grabberTurn.setPeriodHertz(50);
}

```

```

grabberTurn.attach(GrabTurn, 500, 2500);
Serial.begin(115200);
// Создаем точку доступа
WiFi.softAP(ssid, password);
// Запускаем UDP-сервер
udp.begin(localPort);

pinMode(STEPx, OUTPUT);
pinMode(DIRx, OUTPUT);
pinMode(STEPz, OUTPUT);
pinMode(DIRz, OUTPUT);
pinMode(STEPy, OUTPUT);
pinMode(DIRy, OUTPUT);
pinMode(SensX_s, INPUT);
pinMode(SensZ_s, INPUT);
pinMode(SensY_s, INPUT);
pinMode(PPM_PIN, INPUT); //Вход для PPM
attachInterrupt(digitalPinToInterrupt(PPM_PIN), onPPMInterrupt,
RISING);

stepperX.setMaxSpeed(10000.0);
stepperX.setAcceleration(10000.0);
stepperY.setMaxSpeed(10000.0);
stepperY.setAcceleration(10000.0);
stepperZ.setMaxSpeed(10000.0);
stepperZ.setAcceleration(10000.0);

serialThread.onRun(serialRead);
serialThread.setInterval(100);
}

```

В разделе первоначальной настройки код выполняется единожды при подключении микроконтроллера к питанию. Здесь применяются настройки для шаговых двигателей, сервоприводов и точки доступа, а также формируется поток «serialThread», который в последствии будет периодически вызывать функцию «serialRead» с интервалом 100 мс.

```

//////////loop//////////////////////////////////////////
void loop() {
  //Датчики
  //OX
  if (digitalRead(SensX_s) == HIGH and not(XSensor)) { //Если сработал датчик
    stepperX.stop(); //Немедленно остановиться
    XSensor = true;
    stepperX.setCurrentPosition(0); //Обнулить положение
    stepperX.moveTo(0);
  }
  else if (digitalRead(SensX_s) == LOW) XSensor = false;
  //OZ
  if (digitalRead(SensZ_s) == HIGH and not(ZSensor)) {
    stepperZ.stop(); //Немедленно остановиться
    ZSensor = true;
  }
}

```

```

        stepperZ.setCurrentPosition(0); //Обнулить положение
        stepperZ.moveTo(0);
    }
    else if (digitalRead(SensZ_s) == LOW) ZSensor = false;
    //OY
    if (digitalRead(SensY_s) == HIGH and not(YSensor)) {
        stepperY.stop(); //Немедленно остановиться
        YSensor = true;
        stepperY.setCurrentPosition(0); //Обнулить положение
        stepperY.moveTo(0);
    }
    else if (digitalRead(SensY_s) == LOW) YSensor = false;

    //Проверка команд с пульта
    PPMRead();
    //Шаговые двигатели
    if (enable) {
        stepperX.run(); stepperZ.run(); stepperY.run();
    }
    //Проверка команд
    if (mod == 3 or mod == 0) {
        if (serialThread.shouldRun()) serialThread.run();
    }
}
}

```

Основной цикл представляет из себя сравнительно небольшой раздел кода, потому что основная часть программы описана функциями. Однако в этом цикле осуществляется управление шаговыми двигателями, поскольку для этого необходимо формировать быстро-сменяющийся поток импульсов. За это отвечает библиотечная функция «run()», также в основном цикле обрабатывается каждый датчик, и в случае срабатывания останавливает движение нужного шагового двигателя.

```

////////// Чтение команд //////////////////////////////////////////
void serialRead() {
    byte packetBuffer[255]; // Буфер для входящих данных
    int packetSize = udp.parsePacket();
    if (packetSize) {
        // Читаем данные
        int len = udp.read(packetBuffer, 255);
        if (len > 0) {
            packetBuffer[len] = 0; // Нуль-терминатор
        }
        String message = (char*)packetBuffer; //Полученное сообщение
        // Обработка команд
        len = message.length();
        char data = message[len-1];
        message[len-1] = '\0';
        //Проверка сообщения
        //
        if (data == 'd') {
            stepperX.stop(); stepperZ.stop(); stepperY.stop();
        }
    }
}

```

```

        enable = false;
        reply ="Ok";
    }
    else if (data == 'e') {
        enable = true;
        reply ="Ok";
    }
    else if (data == 'r') {
        stepperX.setCurrentPosition(0);
        stepperZ.setCurrentPosition(0);
        stepperY.setCurrentPosition(0);
        reply ="Ok";
    }
    else if (data == 'x') {
        long target = message.toInt();
        if (target >= 0 or (target < 0 and not(XSensor)))
stepperX.moveTo(-target*16*50);
        reply ="Ok";
    }
    else if (data == 'z') {
        long target = message.toInt();
        if (target >= 0 or (target < 0 and not(ZSensor)))
stepperZ.moveTo(-target*16*50);
        reply ="Ok";
    }
    else if (data == 'y') {
        long target = message.toInt();
        if (target >= 0 or (target < 0 and not(YSensor)))
stepperY.moveTo(target*16*5000/90);
        reply ="Ok";
    }
    else if (data == 'g') {
        long target = message.toInt();
        grabber.write(target);
        reply ="Ok";
    }
    else if (data == 't') {
        long target = message.toInt();
        grabberTurn.write(target);
        reply ="Ok";
    }
    else if (data == 'q') {
        reply = String(stepperX.currentPosition());
    }
    // Отправляем ответ
    udp.beginPacket(udp.remoteIP(), udp.remotePort());
    udp.print(reply);
    udp.endPacket();
}
}

```

В самом большом разделе программы проверяются и обрабатываются сообщения от персонального компьютера, строка с входящим сообщением преобразуется в числовое значение сигнала управления и букву, отвечающую за

применение конкретного объекта. После выполнения команды формируется ответ и отправляется на компьютер. На рисунке 4.1 представлена схема применения сети Wi-Fi для управления манипулятором.

```
//////////PPM Read//////////////////////////////////////////
void PPMRead() {
    static uint32_t lastPrintTime = 0;
    uint32_t currentTime = millis();

    if (currentTime - lastPrintTime >= 100) { // Print every 100 ms
        lastPrintTime = currentTime;

        portENTER_CRITICAL(&timerMux); // Enter critical section
        // Вывод отладочной информации для пульта
        //Serial.print("PPM values: ");
        //for (int i = 0; i < NUM_CHANNELS; i++) {
        //    Serial.print(ppmValues[i]);
        //    Serial.print(" ");
        //}
        //Serial.println();

        //Переключение режимов на пульте, переключатель SWD
        if (ppmValues[4] < 1900) {
            mod = -1;
            stepperX.stop(); stepperZ.stop(); stepperY.stop();
            stepperX.move(0); stepperY.move(0); stepperZ.move(0);
        }
        else if (ppmValues[6] > 1900) mod = 3;
        else if (ppmValues[6] > 1300 and ppmValues[6] < 1600) mod = 2;
        else if (ppmValues[6] > 900 and ppmValues[6] < 1100) mod = 1;
        else mod = 0;
        //Управление пультом
        if (mod == 2) { //Если включен режим управления манипулятор с
помощью пульта
            // OX
            if (ppmValues[0] > 1900 and not(XSensor)){ // Стик вправо
                stepperX.moveTo(230000); // Переместиться в крайнее положение
            }
            else if (ppmValues[0] > 800 and ppmValues[0] < 1100){ // Стик
влево
                stepperX.moveTo(-230000); // Переместиться в крайнее положение
            }
            else {stepperX.move(0);} // Остановиться в текущем положении
            // OZ
            if (ppmValues[1] > 1900 and not(ZSensor)){
                stepperZ.moveTo(220000);
            }
            else if (ppmValues[1] > 800 and ppmValues[1] < 1100){
                stepperZ.moveTo(-220000);
            }
            else {stepperZ.move(0);}
            // OY
            if (ppmValues[3] > 1900 and not(YSensor)){
                stepperY.moveTo(80000);
            }
            else if (ppmValues[3] > 800 and ppmValues[3] < 1100){
```


реализованный в проекте. Для обмена данными между собой, клиент и сервер должны находиться в одной сети Wi-Fi, в качестве сервера в таких системах обычно выступает ESP32 с встроенной антенной и модулем беспроводной связи или отдельный модуль беспроводной связи подключенный к контроллеру. Для подключения к одной сети, создается точка доступа Wi-Fi средствами микроконтроллера ESP32.

Настройка точки доступа в Arduino IDE:

```
const char* ssid = "E32 Manipulator"; // Имя точки доступа
const char* password = "12345678"; // Пароль (минимум 8 символов)
unsigned int localPort = 1234; // Порт UDP-сервера
```

Для создания точки доступа необходимо указать несколько параметров, как это сделано выше.

SSID (Service Set Identifier) – имя, для идентификации беспроводной локальной сети среди множества других, оно может включать в себя до 32 символов.

Password – пароль, который вводит пользователь при попытке подключения к сети.

LocalPort – порт UDP-сервера, который задействуется для передачи данных в этой сети, такой же порт используется в программе-интерфейсе на локальной машине пользователя.

4.3 Компьютерная программа управления манипулятором мобильного робота

Окно разработанного интерфейса для программы управления с помощью компьютера показано на рисунке 4.2.

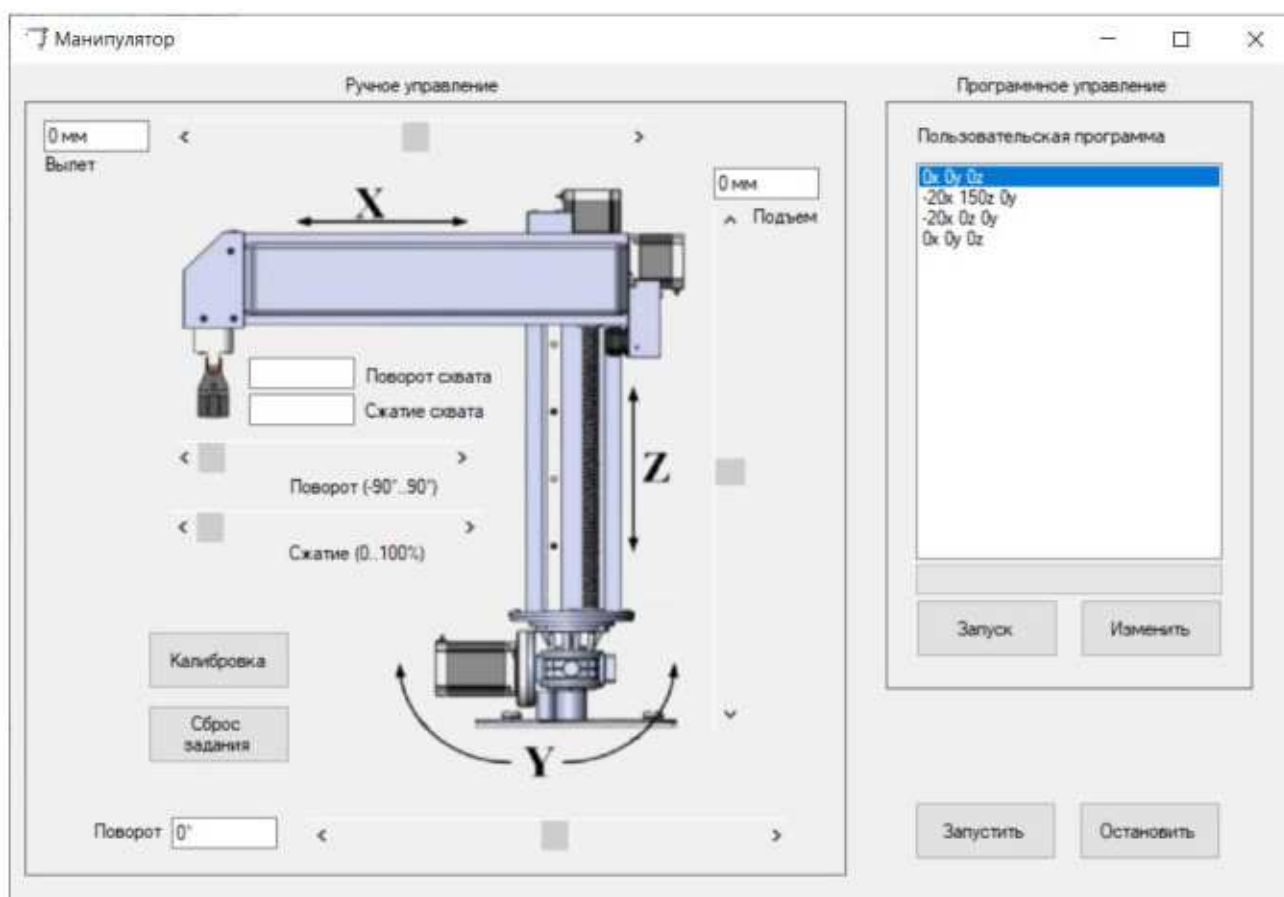


Рисунок 4.2 – Окно программы управления манипулятором

Перед запуском программы компьютер должен быть подключен к Wi-Fi сети «E32 Manipulator».

В окне программы есть две основных панели, панель ручного управления манипулятором и панель программного управления. Пользуясь инструментами первой панели можно устанавливать положения звеньев манипулятора в мм, для вылета и подъема стрелы, и в градусах для поворота башни манипулятора и поворота схвата. Сжатие захвата указывается в процентах. Кнопка «Калибровка» отправит манипулятор в исходное положение, которое определено расположением концевых датчиков. Кнопка «Сброс задания» сбросит переменные о положении шаговых двигателей манипулятора в нули, как внутри программы на ПК, так и в микроконтроллере на борту мобильного робота. При чем такой же сброс должен выполняться при окончании калибровки.

После изменения значений, или нажатия кнопок, команда будет передана микроконтроллеру по беспроводной сети.

Вторая панель окна программы предоставляет пользователю возможность

задать собственную программу управления. Команда отправляются роботу последовательно, при чем все команды, указанные в одной строке отправляются почти одновременно, а команды в следующей строке будут отправлены спустя некоторый период ожидания, за время которого робот должен выполнить предыдущие. Каждая команда состоит из числа и буквы. Числа могут быть как положительные, так и отрицательные, они соответствуют тем же единицам измерения, как и при ручном управлении манипулятором. Буквы, используемые в командах:

1. x – перемещение по OX в мм (вылет стрелы);
2. z – перемещение по OZ в мм (подъем стрелы);
3. y – поворот по OY в градусах (поворот башни);
4. g – сжатие схвата в процентах;
5. t – поворот схвата в градусах.

Чтобы изменить пользовательскую программу, нужно её в текстовом документе формата (.txt), после выбрать кнопку «Изменить», и указать путь к нужному файлу. Буквы следует указывать в нижнем регистре, без пробелов, как это записано в образце по умолчанию. Чтобы запустить пользовательскую программу следует нажать кнопку «Запустить». После чего, если все было сделано правильно, робот начнет выполнять программу. Во время выполнения пользовательского алгоритма, остальная часть окна будет заблокирована, чтобы исключить ненужное вмешательство, однако в окне программы предусмотрены графические элементы, которые дадут понять пользователю, что программа в ходе выполнения и не зависла.

Кнопки «Запустить» и «Остановить» не ни к одной из панелей в окне, и не блокируются при выполнении программы. Так, при нажатии кнопки «Остановить», роботу будет немедленно передана соответствующая команда, и он остановится при выполнении любых действий, пользовательская программа также будет приостановлена. Дальнейшая работа не будет возобновлена, пока пользователь не нажмет кнопку «Запустить».

Вид окна программы во время выполнения алгоритма пользовательской программы показан на рисунке 4.3.

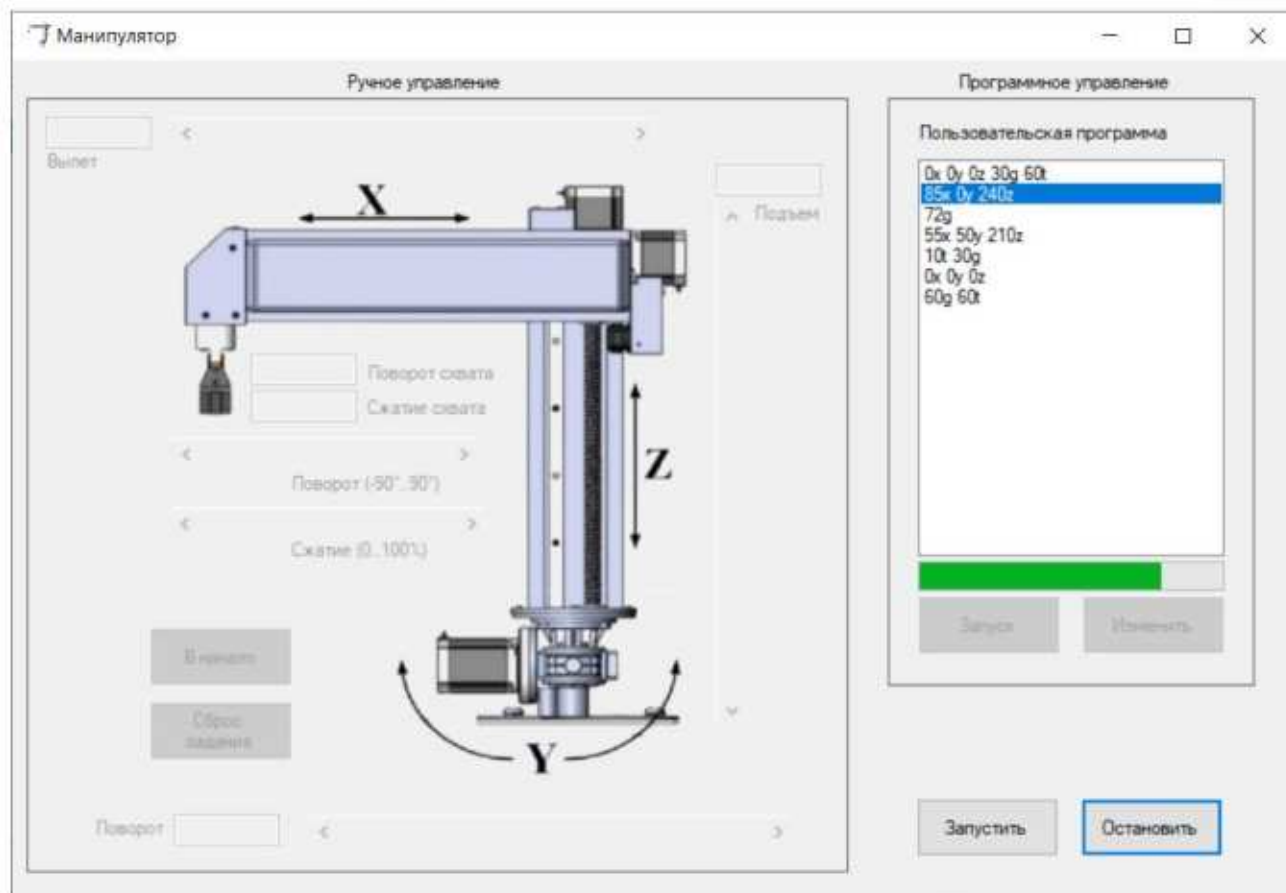


Рисунок 4.3 – Вид приложения в запущенном режиме

Полный текст разработанной программы управления представлен в приложении 2.

5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

Мобильный робот-манипулятор может представлять целый ряд различных опасностей, обычно возникающих из-за неправильных условий эксплуатации. Робот на мотор-колёсах может столкнуться с людьми или объектами. Манипулятор на роботе имеет много подвижных частей, если между ними окажется рука человека, то он может её травмировать. Также в роботе содержатся химические источники тока, которые при неправильной эксплуатации могут привести к их самовозгоранию или взрыву. Чтобы не оказывать негативного воздействия на окружающую среду важно обратить внимание на утилизацию литий-ионных аккумуляторов, остальные части робота не так опасны для природы.

5.1 Безопасность

5.1.1 Безопасность при работе с роботом

Перед применением робота, следует проверить заряд аккумулятора. При недостаточном напряжении робот не будет работать ни в одном из режимов.

Необходимо убедиться, что ничего не будет мешать движению робота. Заблокированные посторонними предметами оси звеньев манипулятора приведут к пропуску импульсов шаговыми двигателями, что в свою очередь может привести к непредвиденной аварийной ситуации. Для предотвращения вероятности возникновения таких ситуаций необходимо, после запуска системы, произвести калибровку манипулятора, нажатием кнопки “Калибровка” в окне компьютерной программы для управления манипулятором.

Для безопасности и удобства применения оператор должен стоять позади робота. Зона, в которой планируется передвижение робота должна быть огорожена, или допускается выбрать сопровождающего, который следуя перед роботом предупредит угрозу возможного столкновения. Также следует заранее оповестить всех присутствующих рядом лиц об возможном, внезапном появлении подвижного робота.

Необходимо избегать любых столкновений. Робот имеет массу свыше 20

кг, Поэтому, при столкновении на своей максимальной скорости может нанести сильный удар. Следует проявлять особенную осторожность при движении в узких коридорах. Чтобы уменьшить последствия случайных столкновений, спереди и сзади робота предусмотрены мягкие бамперы. Однако, если стрела манипулятора выпущена более чем на половину или повернута не по направлению движения, робот может столкнуться ею с высокими препятствиями или стенами.

При использовании манипулятора следует помнить об осторожности, чтобы случайно ничего не опрокинуть, сломать или раздавить. Углы вращения башни манипулятора ограничены в пределах от -180 до +180 градусов, чтобы избежать наматывания проводов, идущих от шкафа управления к башне. В манипуляторе предусмотрены датчики конечного положения, они остановят движение манипулятора в том случае, если он дойдет до края рабочей области.

При работе с роботом с помощью пульта, можно выключить его во время движения, за отключение отвечает левый крайний переключатель. В таком случае робот немедленно остановится. Пульт управления показан на рисунке 5.1. Кроме того предусмотрен автоматический выключатель, установленный на корпусе робота, позволяющий обесточить автономную систему электропитания.

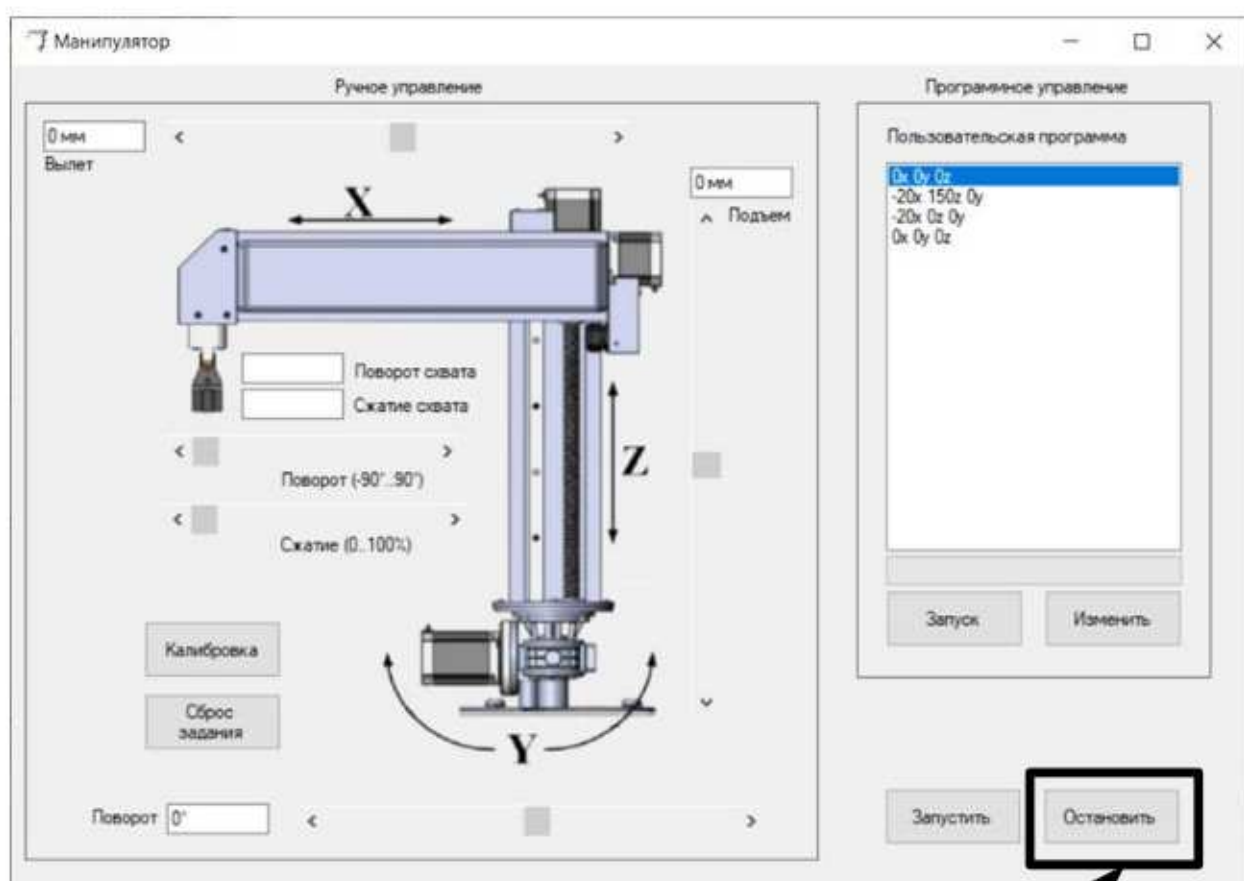


Рисунок 5.1 – Пульт для управления роботом

Управление роботом можно осуществлять с помощью программы запущенной на компьютере. Связь между микроконтроллером и компьютером осуществляется с применением сети Wi-Fi. Для чрезвычайных случаев в программе предусмотрена кнопка для остановки манипулятора. Окно компьютерной программы показано на рисунке 5.2. Если беспроводное Wi-Fi соединение случайно будет разорвано, например если робот уехал слишком далеко, или кто-то закрыл программу на компьютере, то манипулятор выполнит только последнюю опрарвленную ему команду и после чего остановится.

Применение вышеупомянутых средств для остановки движения робота никак не сказываются на работоспособности всей системы, поэтому их можно применять при первом же возникновении любых рисков.

В окне компьютерной программы помимо средств управления, присутствуют простые элементы визуализации, их назначение – предоставлять пользователю информацию об этапах выполнения программы и о том, что программа не зависла.



Кнопка остановки манипулятора

Рисунок 5.2 – Окно программы управления манипулятором

Если по каким-то причинам персональному компьютеру не удалось связаться с микроконтроллером на борту робота, то появится сообщение с соответствующим текстом и предложит перезапустить программу, чтобы повторить попытку подключения.

Оси, по которым перемещается манипулятор имеют программные ограничения.

5.1.2 Меры безопасности при зарядке литий-ионных аккумуляторов

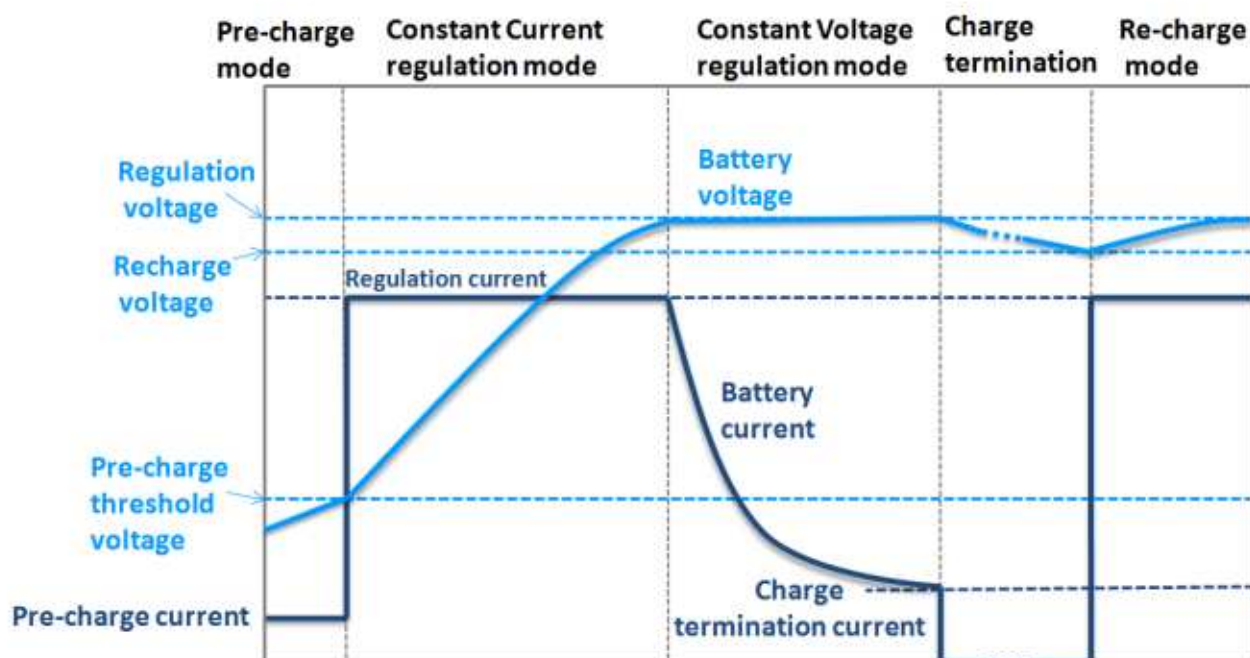
Литий-ионные аккумуляторы не взрываются и не загораются при нормальной эксплуатации благодаря строгим мерам предосторожности и использованию автоматических электронных систем контроля. Ключевым элементом безопасности является применение «умных» зарядных устройств, которые обеспечивают правильную работу аккумуляторов.

Процесс зарядки литий-ионных аккумуляторов включает несколько этапов

и требует строгого контроля параметров. Важно, чтобы заряд прекращался вовремя, так как любая ошибка может привести к взрыву или возгоранию. Поэтому ответственность за правильную зарядку не может быть доверена пользователю – все необходимые функции должны выполняться автоматически.

Наиболее распространённым и рекомендованным производителями методом зарядки является алгоритм CC-CV. Этот метод предполагает первоначальную стабилизацию тока, а затем, при достижении определённого напряжения, его стабилизацию на этом уровне. Такой подход наиболее близок к традиционному методу зарядки свинцово-кислотных аккумуляторов и обеспечивает безопасную и эффективную зарядку литий-ионных аккумуляторов. График применения алгоритма CC-CV наглядно представлен на рисунке 5.3.

Для большинства стандартных литий-ионных аккумуляторов напряжение перехода от стадии CC к стадии CV при комнатной температуре 4,20 В. Даже небольшое превышение этого напряжения вызывает быстрое сокращение срока службы аккумулятора, а более значительное превышение приводит к возгораниям и взрывам. Требуемая точность установки порогового напряжения для стандартных аккумуляторов составляет ± 50 мВ, а у «высоковольтных» еще выше, пропорционально их напряжению, вплоть до ± 5 мВ. Однако, пониженное напряжение приводит лишь к снижению доступной емкости, а вот повышение напряжения недопустимо ни при каких случаях.



Battery charger operation modes

Рисунок 5.3 – График алгоритма CC-CV

Для безопасной зарядки литий-ионных аккумуляторов используется многоступенчатый процесс. Стандартный ток зарядки составляет $0,5\ C$, где C – ёмкость аккумулятора в ампер-часах. Большинство аккумуляторов допускают зарядку токами до $1\ C$ без ущерба, а некоторые модели могут поддерживать более высокие токи при условии контроля температуры.

Важно учитывать степень разрядки аккумулятора. Если напряжение на его выводах падает ниже $3\ В$, применять стандартный метод зарядки нельзя. В этом случае необходима предварительная зарядка (precharge) малым током $0,05\text{--}0,1\ C$ до достижения напряжения $3\ В$.

Категорически запрещается заряжать аккумуляторы, напряжение которых опустилось ниже $2,5\ В$, так как это приводит к значительной потере ёмкости и создаёт риск возгорания из-за возможной металлизации лития. Особенно чувствительны к глубокому разряду высоковольтные аккумуляторы, их нельзя разряжать ниже $2,75\ В$.

На стадии постоянного напряжения (CV) ток зарядки снижается по экспоненте. Зарядка на этом этапе должна автоматически прекращаться, когда ток падает до $0,05\text{--}0,1\ C$.

Для реализации такого сложного алгоритма зарядки рекомендуется использовать специализированные микросхемы-контроллеры, которые обеспечивают точное управление процессом и повышают безопасность эксплуатации аккумуляторов.

Из всего вышесказанного можно выделить основные правила.

1. Не допускать полной разрядки аккумуляторов.
2. Заряжать аккумуляторы только специальными умными зарядными устройствами.
3. Хранить аккумуляторы в частично разряженном состоянии.

Особое внимание нужно уделить обращению с повреждёнными аккумуляторами. Такие элементы могут протекать и выделять токсичные вещества, представляя пожарную опасность. При работе с ними необходимо использовать защитные перчатки и очки. Повреждённые аккумуляторы следует отдельно изолировать и упаковать, например, в пластиковый контейнер, чтобы предотвратить утечку содержимого и контакт с другими предметами. Эти меры помогут обеспечить безопасность при транспортировке и последующей утилизации аккумуляторов.

5.2 Экологичность

Литий-ионные аккумуляторы считаются опасными отходами из-за содержащихся в них активных химических веществ и риска возгорания, поэтому их утилизация обязательна и должна проводиться в соответствии с нормативными требованиями [3].

Для безопасной утилизации таких аккумуляторов необходимо сдавать их в специализированные пункты приёма и переработки. Существует несколько основных вариантов, куда можно отнести отработанные аккумуляторы:

Рециркуляционные центры. В большинстве городов имеются специальные центры, принимающие отработанные литий-ионные аккумуляторы. Они оснащены всем необходимым оборудованием для безопасного хранения, транспортировки и переработки аккумуляторов, что позволяет извлечь ценные материалы и минимизировать вред для окружающей

среды.

Экологические инициативы и программы. Различные общественные и частные организации реализуют программы по сбору и утилизации аккумуляторов. Часто такие программы включают временные пункты сбора и акции по приёму опасных отходов, где можно сдать отработанные аккумуляторы.

Муниципальные пункты сбора. Многие города предлагают свои программы утилизации, включающие специальные пункты приёма опасных отходов, таких как батареи и аккумуляторы. Эти пункты обычно располагаются на территории коммунальных служб или специализированных предприятий по утилизации.

Экологические мероприятия. Местные власти и экологические организации регулярно проводят акции и мероприятия по сбору отработанных аккумуляторов. Информацию о таких событиях можно найти на официальных сайтах муниципальных органов или в социальных сетях.

Сдача отработанных литий-ионных аккумуляторов в указанные пункты приёма – это важный шаг для их безопасной утилизации. Независимо от того, выберете ли вы рециркуляционный центр, магазин электроники, крупную розничную сеть или муниципальный пункт, главное – убедиться, что аккумуляторы будут утилизированы правильно. Это поможет сохранить окружающую среду и здоровье людей.

5.2.1 Подготовка аккумуляторов к утилизации

Для безопасной сдачи литий-ионных аккумуляторов на утилизацию необходимо выполнить ряд подготовительных действий.

Прежде всего, важно изолировать контакты аккумуляторов, чтобы избежать короткого замыкания и возгорания. Это можно сделать при помощи изоляционной ленты, обмотанной вокруг контактов.

Далее следует правильно организовать хранение и транспортировку аккумуляторов. Их необходимо поместить в прочную и надёжную упаковку, которая обеспечит сохранность при перевозке. Подойдут оригинальные

заводские коробки или специализированные контейнеры для опасных отходов.

5.3 Чрезвычайные ситуации

Если во время эксплуатации случится возгорание, то действовать необходимо согласно инструкции по пожарной безопасности.

Основная эксплуатация работа производилась в 402 аудитории 6 корпуса АМГУ. На всех этажах установлены специальные планы эвакуации, которые легко можно заметить. Эти планы показывают, как действовать в случае пожара и где находятся важные средства безопасности. На таких схемах можно найти:

- расположение огнетушителей;
- места установки пожарных кранов;
- где находятся телефоны для вызова помощи;
- эвакуационные пути и выходы.

Такая система обозначений создана по правилам, которые прописаны – СП 112.13330.2011 «Пожарная безопасность зданий и сооружений» [1]. План эвакуации для 4 этажа 6 корпуса показан на рисунке 5.4.



Рисунок 5.4 – План эвакуации для 4 этажа 6 корпуса

Согласно нормативным документам, для тушения электроустановок применяются следующие типы огнетушителей:

Порошковые огнетушители. Они эффективно подавляют пламя, но после

применения требуется тщательная очистка оборудования из-за возможного повреждения от порошка.

Углекислотные огнетушители. Они наиболее подходят для таких случаев, так как содержат газ, который не проводит электричество и не оставляет следов после применения. Важно, чтобы содержание паров воды в углекислотном составе было менее 0,006 %, а дальность струи достигала не менее трёх метров.

Использование других типов огнетушителей, содержащих воду или другие проводящие вещества, категорически запрещено.

Перед применением углекислотного огнетушителя необходимо принять меры предосторожности, например, защитить органы дыхания, так как углекислый газ может быть опасен для дыхания [5].

При пожаре

1. Сообщить в Единую службу спасения (пожарную охрану) и дежурному учебному корпусу, общежитию, иному служебному помещению университета (в зависимости от места пожара);

2. Немедленно покинуть горящее здание (в холодный период и в случае атмосферных осадков – в учебный корпус или иное здание, наиболее удаленное от пожара).

3. При незначительном возгорании попытаться с помощью огнетушителей, воды из ПК (пожарных кранов) потушить или локализовать возгорание до прибытия пожарной команды.

Телефоны:

Единая служба МЧС -01; МТС-010; Билайн -002; Мегафон- 010

Пожарная охрана -01; МТС-010; Билайн -001; Мегафон- 010

Полиция -03; МТС-020; Билайн -002; Мегафон- 020

Скорая помощь -03; МТС-030; Билайн -003; Мегафон- 030

Условия хранения также играют важную роль. До момента сдачи аккумуляторы нужно держать в прохладном и сухом месте, вдали от источников тепла и влаги. Это поможет сохранить их целостность и снизит риск самовозгорания.

ЗАКЛЮЧЕНИЕ

В процессе разработки была создана система управления манипулятором мобильного робота, проведена ревизия контроллеров шаговых двигателей и концевых выключателей. Составлены новые принципиальные электрические схемы: управления мотор-колёсами, шаговыми двигателями, сервоприводами и индуктивными датчиками.

Разработаны алгоритмы и код управления с помощью радиоаппаратуры, сети Wi-Fi и простейший протокол связи. Разработка системы управления роботом-манипулятором оказалась комплексной задачей, требующей учета множества факторов и применения современных технологий. Реализация такой системы позволяет создавать эффективные решения для различных областей применения.

В данной работе была применена следующая техническая база и программные средства:

- учебный робот-манипулятор УР-4;
- микроконтроллерная плата ESP32 с встроенным модулем беспроводной связи;
- драйверы шаговых двигателей;
- шаговые двигатели;
- сервоприводы;
- среда Proteus 8.10;
- среда программирования Arduino IDE;
- среда разработки Visual Studio 2022, WinForms.

Поставленные задачи на разработку системы управления выполнены полностью, написано все необходимое программное обеспечение и проведены тестовые испытания.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Куликов, С.Е. Средства автоматизации и управления работа-манипулятора / С.Е. Куликов, А.В. Качалов, Е.И. Богатырев // «САУ-РОБОТ». 2015. – С. 64.
2. Giden Electronics [Электронный ресурс]: Шаговые двигатели –. Режим доступа : <https://shop.giden.ru/shagovye-dvigateli>. – 01.04.2025.
3. Mypractic.ru: Уроки программирования Ардуино [Электронный ресурс]: ОБОРУДОВАНИЕ ТЕХНОЛОГИИ РАЗРАБОТКИ – Режим доступа : <http://mypractic.ru/uroki-programmirovaniya-arduino-navigaciya-po-urokam> – 24.10.2024.
4. Purelogic.ru: Плата коммутации драйверов ШД/СД [Электронный ресурс]: Purelogic R&D – Режим доступа : <https://purelogic.ru> – 01.04.2025.
5. Библиотекарь.Ру: Основы робототехники [Электронный ресурс]: bibliotekar.ru. Режим доступа : <https://www.bibliotekar.ru> – 01.04.2025.
6. Электропривод: Гибридные шаговые двигатели [Электронный ресурс]: electoprivod.ru. Режим доступа : <https://electoprivod.ru> – 31.03.2025.
7. Драйверы шаговых двигателей [Электронный ресурс]: tehnologi.ru. Режим доступа : <https://cnc-tehnologi.ru/mnogokanalnye-drajvery-shagovykh-dvigatелеj-dlya-chpu> – 28.03.2025.
8. СП 112.13330.2011 «Пожарная безопасность зданий и сооружений».
9. Федеральный закон от 24 июня 1998 г. N 89-ФЗ «Об отходах производства и потребления».
10. Рыбалёв, А.Н. Теория автоматического управления: курсовое проектирование: уч. Пособие /А.Н. Рыбалёв. – Благовещенск: Амурский гос. университет. – 2004. – С. 144.

11. Рыбалев, А. Н. Имитационное моделирование АСУ ТП. уч. Пособие / А.Н. Рыбалёв. – Благовещенск: Амурский гос. университет. – 2019. – С. 408.
12. ГОСТ 12.1.007-76 «Вредные вещества».
13. ГОСТ 12.1.019-79 “ССБТ. Электробезопасность
14. СП 9.13130.2009 “Свод правил. Техника пожарная. Огнетушители. Требования к эксплуатации”
15. Нестеров, А.Л. Проектирование АСУТП Книга 1. – 2006. – С. 544.
16. Нестеров, А.Л. Проектирование АСУТП Книга 2. – 2009. – С. 944.
17. Зорин, А.Ю. Условные графические обозначения на схемах, – 2007. – С. 40.
18. ГОСТ 21.404–85. Обозначения условные приборов и средств автоматизации в схемах. – М.: Изд-во стандартов, – 1986. – С 8.
19. РМ 4-51-90. Принципы компоновки щитов и пультов управления. – М.: Изд-во стандартов. 1990. – С 28.
20. Ряд резисторов E24 (asenergi.com). [Электронный ресурс]. – Режим доступа : https://asenergi.com/pdf/rezistory/ryad_rezistorov_e24.pdf?ysclid=ladv3zcm2270_0190818.

ПРИЛОЖЕНИЕ 1

Техническое задание

1 Введение

Данное ТЗ распространяется на исследование мобильного робота-манипулятора, разработка системы управления на основе МК, используя протокол STEP/DIR для работы с шаговыми двигателями и создание программного обеспечения для управления роботом-манипулятором с помощью команд пользователя.

Заказчик: Федеральное государственное бюджетное образовательное учреждение высшего образования Амурский государственный университет (ФГБОУ ВО АмГУ).

Исполнитель: Кузнецов В. А.

Система разрабатывается на основании следующих документов:

ФГОС направления подготовки бакалавров 15.03.04 автоматизации технологических процессов и производств;

Учебный план направления подготовки бакалавров 15.03.04 автоматизации технологических процессов и производств;

Плановые сроки начала и окончания работ по созданию системы:

Начало: 07.10.2024

Окончание: 02.04.2025

2 Характеристики объекта автоматизации

Объектом автоматизации является робот-манипулятор на мобильном электрошасси. На осях робота концевые индуктивные датчики, на каждой оси по два датчика.

Основными компонентами системы управления являются: контроллер, драйверы шаговых двигателей, концевые датчики, персональный компьютер.

Контроллер используется для подачи сигналов на драйверы шаговых двигателей. Также контроллер опрашивает датчики, ведет обмен данными с ПК, отправляя компьютеру информацию от пользователя и концевых выключателей.

Еще одной функцией является передача числа шагов для шагового двигателя в протокол STEP/DIR, на выходе которого осуществляется подача сигналов на драйвер ШД.

Драйвер на основании полученных сигналов от контроллера осуществляет коммутацию обмоток двигателей, которые приводят в движение робота.

3 Цели создания системы

Разработка новой системы управления с помощью микроконтроллера. Совмещение работы микроконтроллера и компьютера для последующего управления ШД и считывания состояния датчиков.

Разработка программного обеспечения для управления роботом-манипулятором, начиная с приведения алгоритмов работы программы и разработки исходного текста, завершая тестированием на практике разработанного ПО.

Исследования имеющейся системы управления роботом-манипулятором приводится для изучения принципов управления ШД и обработки сигналов с датчиков.

4 Функции и задачи создаваемой АСУ

Основными компонентами системы управления являются: контроллер, драйверы шаговых двигателей, концевые датчики, энкодеры, персональный компьютер.

Задачи проекта:

1. Создание структурной схемы работы системы МК и ПК;
2. Разработка программы управления шаговыми двигателями;
3. Разработка приложения для обмена данными между ПК и МК;
4. Разработка опытного стенда для тестирования функций ШД.

ПРИЛОЖЕНИЕ 2

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO.Ports;
using System.Linq;
using System.Net.Sockets;
using System.Net;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Text.RegularExpressions;
using System.IO;

namespace WindowsFormsApp1
{
    public partial class Main : Form
    {
        // Подключаемся к точке доступа ESP32 (обычно 192.168.4.1)
        const string esp32ApIp = "192.168.4.1";
        const int serverPort = 1234;
        static UdpClient udpClient;

        bool enable = true;
        public Main() //Запуск//
        {
            udpClient = new UdpClient();
            // Настройка клиента (опционально - можно указать локальный порт)
            udpClient.Client.Bind(new IPEndPoint(IPAddress.Any, 0));

            Console.WriteLine($"UDP Client started. Connecting to ESP32 AP at {esp32ApIp}");
            InitializeComponent();
        }

        public static void SendToArduino(int data, string word) //Функция отправки
команды//
        {
            string message = data.ToString() + word;
            try
            {
                // Отправка сообщения
                byte[] dataB = Encoding.ASCII.GetBytes(message);
                udpClient.Send(dataB, dataB.Length, esp32ApIp, serverPort);

                //Получение ответа(с таймаутом 2 секунды)
                udpClient.Client.ReceiveTimeout = 2000;
                IPEndPoint remoteEP = new IPEndPoint(IPAddress.Any, 0);
                byte[] receivedData;

                try
                {
                    receivedData = udpClient.Receive(ref remoteEP);
                    string response = Encoding.ASCII.GetString(receivedData);
                    Console.WriteLine($"Response: {response}");
                }
                catch (SocketException ex) when (ex.SocketErrorCode ==
SocketError.TimedOut)
            }
        }
    }
}
```

Продолжение ПРИЛОЖЕНИЯ 2

```
{
    Console.WriteLine("No response from ESP32 (timeout)");
    MessageBoxButtons buttons = MessageBoxButtons.YesNo;
    DialogResult result;
    result = MessageBox.Show("Нет ответа от микроконтроллера. Закрыть
программу?", "Ошибка", buttons);
    if (result == System.Windows.Forms.DialogResult.Yes)
Application.Exit();
}
}
catch (Exception ex)
{
    Console.WriteLine($"Error: {ex.Message}");
}
}

private void ScrollBarX_ValueChanged(object sender, EventArgs e)
{
    int x = ScrollBarX.Value;
    textBoxX.Text = Convert.ToString(x + " мм");
    SendToArduino(x, "x");
}

private void ScrollBarZ_ValueChanged(object sender, EventArgs e)
{
    int z = ScrollBarZ.Value;
    textBoxZ.Text = Convert.ToString(z + " мм");
    SendToArduino(z, "z");
}

private void ScrollBarY_ValueChanged(object sender, EventArgs e)
{
    int y = ScrollBarY.Value;
    textBoxY.Text = Convert.ToString(y + "°");
    SendToArduino(y, "y");
}

private void ScrollBarG_ValueChanged(object sender, EventArgs e)
{
    int g = ScrollBarG.Value;
    textBoxG.Text = Convert.ToString(g * 100 / 70);
    SendToArduino(g, "g");
}

private void ScrollBarT_ValueChanged(object sender, EventArgs e)
{
    int t = ScrollBarT.Value;
    textBoxT.Text = Convert.ToString(t);
    //SendToArduino(t, "t");
}

private void StartButton_Click(object sender, EventArgs e)
{
    StartButton.Enabled = false; //Заблокировать кнопку Изменить
    EditButton.Enabled = false; //Заблокировать кнопку Изменить
    panel1.Enabled = false; //Заблокировать панель
    Start_cnc();
}

private void DisableButton_Click(object sender, EventArgs e)
```

Продолжение ПРИЛОЖЕНИЯ 2

```

{
    enable = false;
    SendToArduino(0, "d"); //Команда откл
}

private void EnableButton_Click(object sender, EventArgs e)
{
    SendToArduino(0, "e"); //Команда вкл
    enable = true;
}

private void ResetButton_Click(object sender, EventArgs e)
{
    SendToArduino(0, "r"); //Команда сброс
    ScrollBarX.Value = 0; //Сброс положений ползунков
    ScrollBarZ.Value = 0;
    ScrollBarY.Value = 0;
}

private void EditButton_Click(object sender, EventArgs e) //Кнопка Изменить
{
    string filePath = ""; // Переменная путь к подгружаемому файлу
    OpenFileDialog Dialog = new OpenFileDialog // Создание нового диалога выбора
    файла
    {
        Title = "Выберите файл", // Заголовок диалога
        Filter = "Текстовые файлы (*.txt)|*.txt" // Фильтр файлов в диалоге
    };
    if (Dialog.ShowDialog() == DialogResult.OK) //Действие после нажатие кнопки
    подтверждения в диалоге
    {
        filePath = Dialog.FileName; // Сохранить путь к файлу
        Dialog.Dispose(); //Очистить память от диалога
        //Переписывание окна команд
        if (filePath != "")
        {
            using (StreamReader sr = new StreamReader(filePath))
            {
                {
                    string line;
                    CommandListBox.Items.Clear();
                    while ((line = sr.ReadLine()) != null)
                        CommandListBox.Items.Add(line);
                }
            }
        }
    }
}

async void Start_cnc()
{
    string test, commandWord, inputString;
    int commandInt, delay, xOld = 0, yOld = 0, zOld = 0;
    for (int i = 0; i < CommandListBox.Items.Count; i++)
    {
        CommandListBox.SelectedIndex = i; //Выделить выполняемую команду
        inputString = CommandListBox.SelectedItem.ToString();
        string pattern = @"-?\d+[xyzgt]"; // \d - цифра, + - один или несколько
        символов, [xyzgt] - символы xyzgt
        MatchCollection matches = Regex.Matches(inputString, pattern);
        //Обработка строки с командами
        delay = 0;
        foreach (Match match in matches)
        {
            test = match.Value;

```

Продолжение ПРИЛОЖЕНИЯ 2

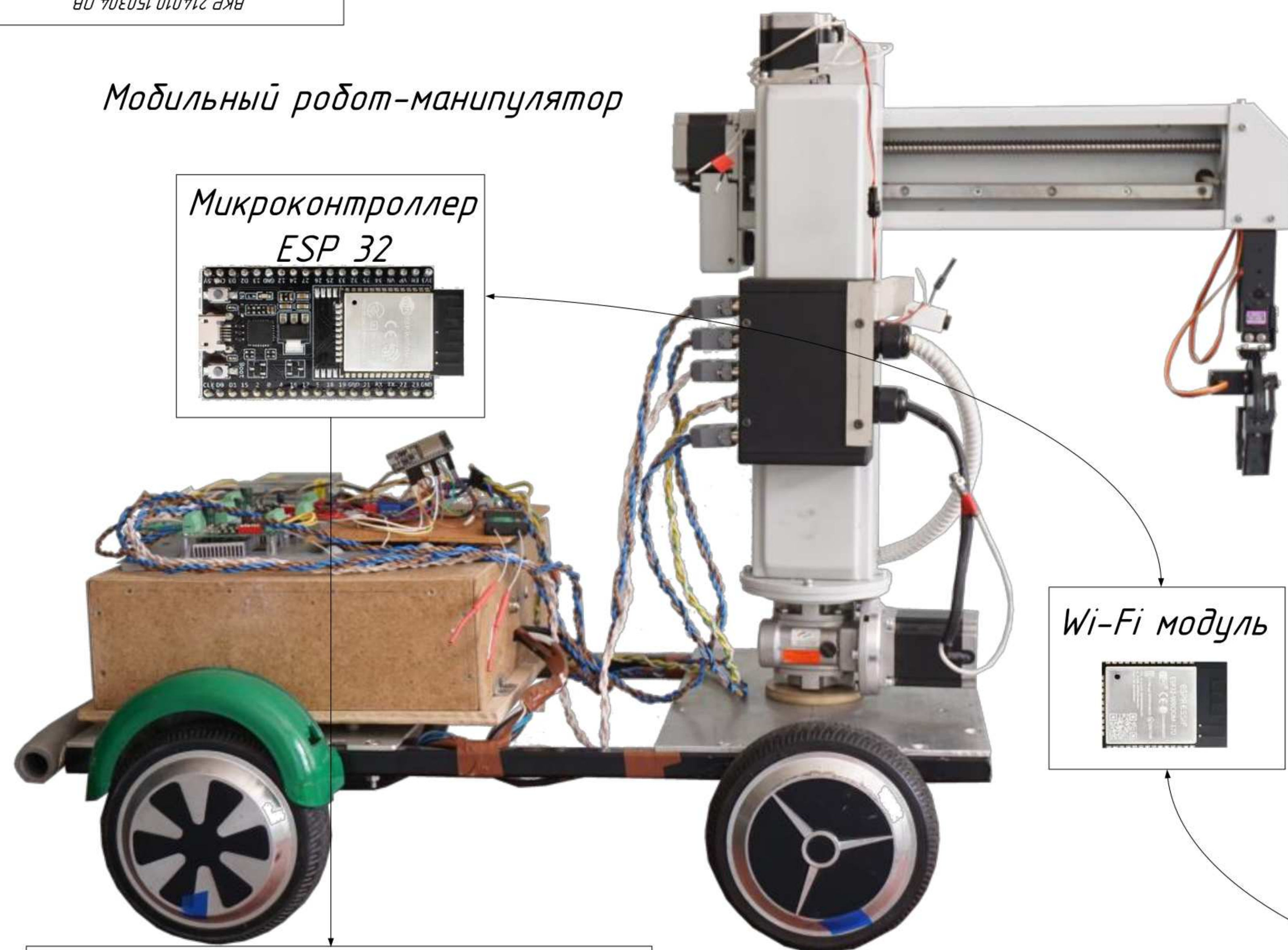
```

commandWord = test[test.Length - 1].ToString();
test = test.Remove(test.Length - 1);
commandInt = int.Parse(test);
//SendToArduino(commandInt, commandWord); //Отправка команды
if (commandWord == "x")
{
    delay = Math.Max(delay, Math.Abs(xOld - commandInt)); //Значение
задержки перед отправкой следующей строки с командами
    xOld = commandInt;
}
else if (commandWord == "z")
{
    delay = Math.Max(delay, Math.Abs(zOld - commandInt));
    zOld = commandInt;
}
else if (commandWord == "y")
{
    delay = Math.Max(delay, Math.Abs(yOld - commandInt));
    yOld = commandInt;
}
}
Console.WriteLine("Delay:" + delay * 90);
for (int l = 1; l <= 100; l++)
{
    progressBar1.Value = l;
    await Task.Delay(delay * 90 / 100); //Задержка перед отправкой
следующей строки с командами
    while (!enable) await Task.Delay(100);
}
}
EditButton.Enabled = true; //Разблокировать кнопку Изменить
StartButton.Enabled = true; //Разблокировать кнопку Изменить
panel1.Enabled = true; //Разблокировать панель
}

private void Homing_Click(object sender, EventArgs e) //Команда откалибровать
{
    SendToArduino(-500, "x");
    SendToArduino(-500, "z");
    SendToArduino(0, "y");
    SendToArduino(60, "g");
    SendToArduino(60, "t");
    ScrollBarX.Value = 0; //Сброс положений ползунков
    ScrollBarZ.Value = 0;
    ScrollBarY.Value = 0;
}
}
}
}

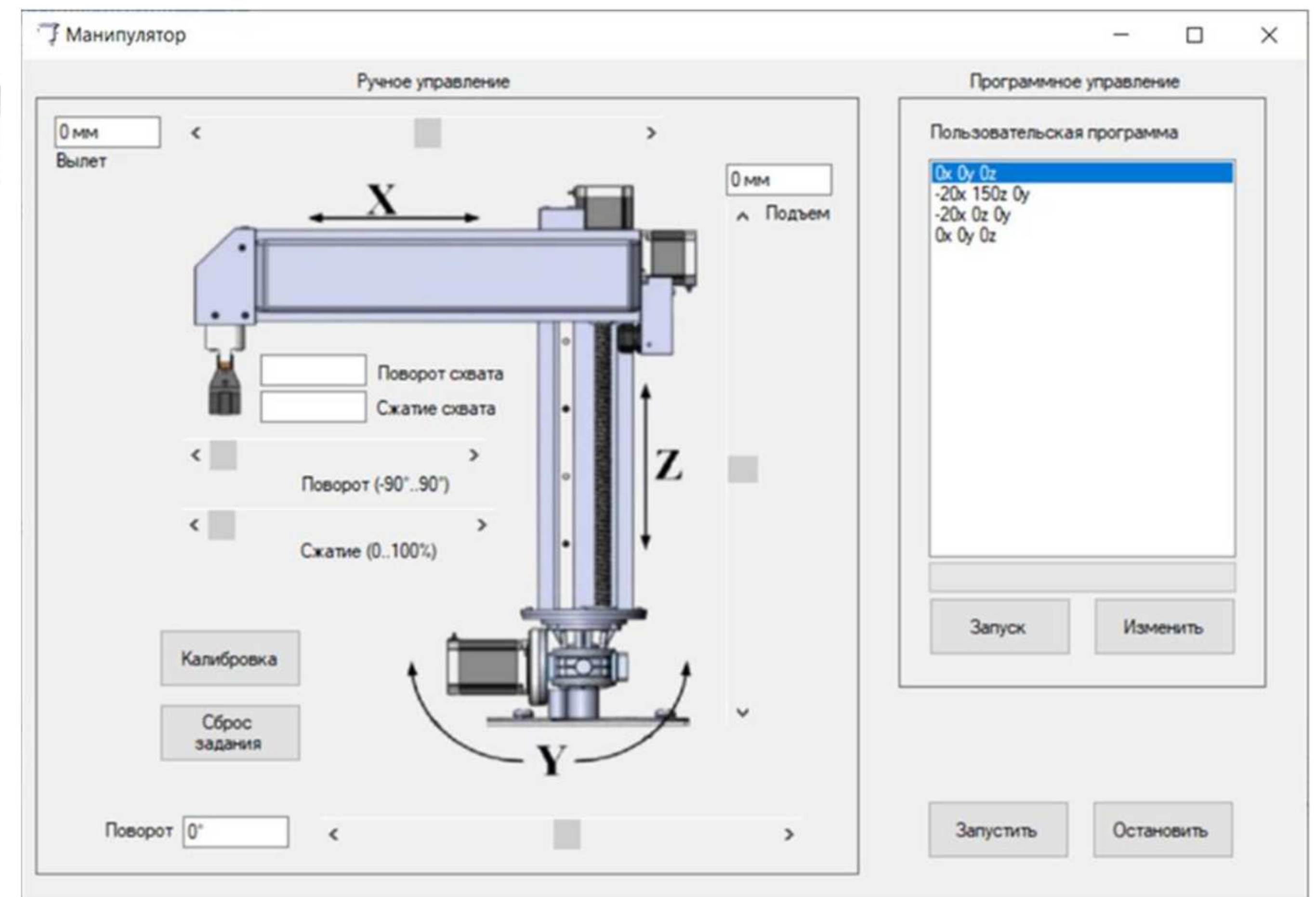
```

Мобильный робот-манипулятор

Микроконтроллер
ESP 32

Wi-Fi модуль

Окно программы для управления манипулятором



Wi-Fi 2.4 ГГц

Персональный компьютер с Wi-Fi



RC - приемник

Протокол - PPM



Пульт для управления мобильным роботом

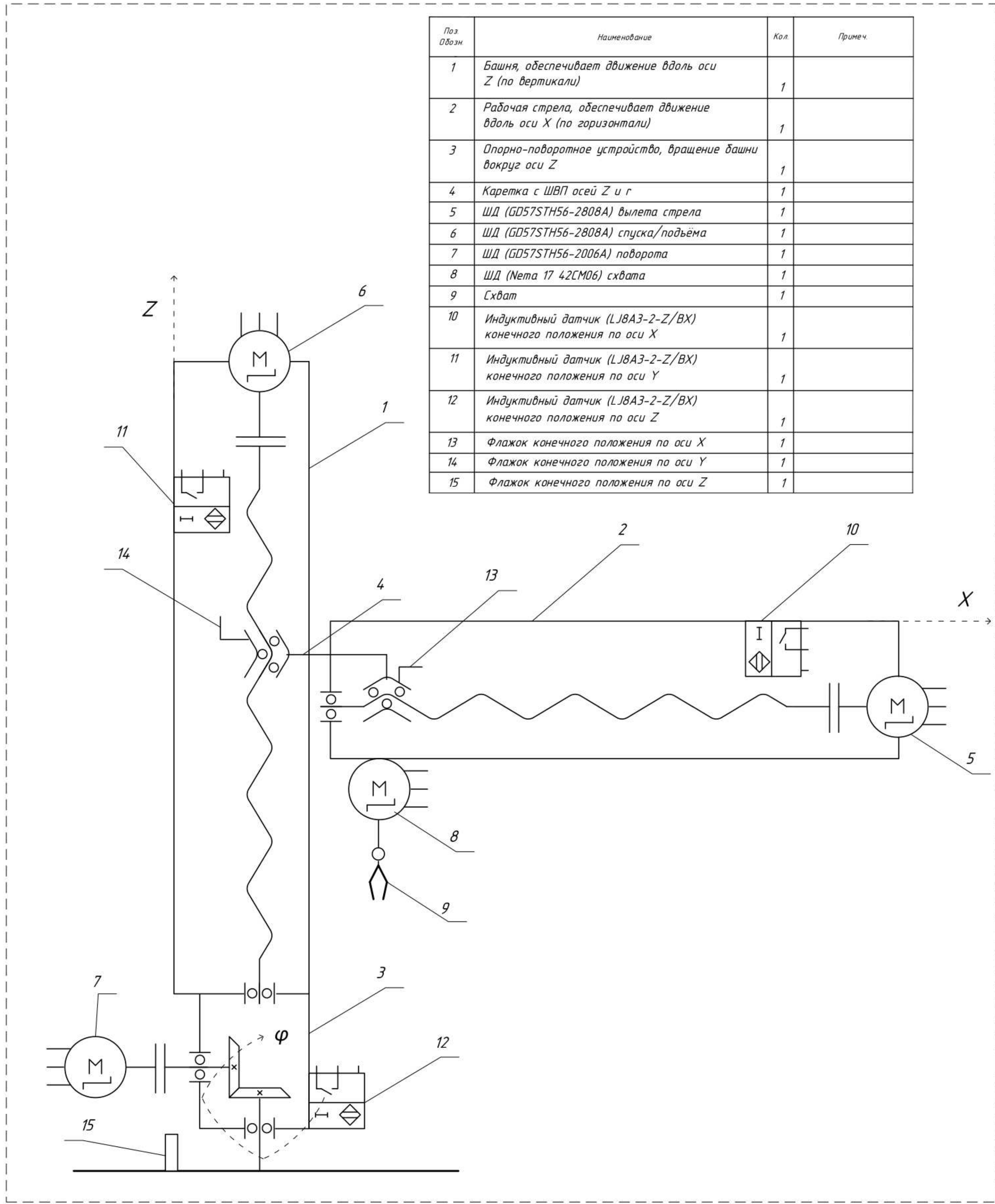
С помощью пульта можно управлять роботом в нескольких режимах:

1. Режим передвижения
 2. Режим управления манипулятором
 3. Режим программного управления.
- Включать/Выключать робота,
и управлять схватом.

Задание для манипулятора в пользовательской программе формируется в мм (для вылета и подъема), градусах (для поворота) и процентах (для сжатия схвата). После числа обязательно должна следовать буква, обозначающая звено манипулятора:

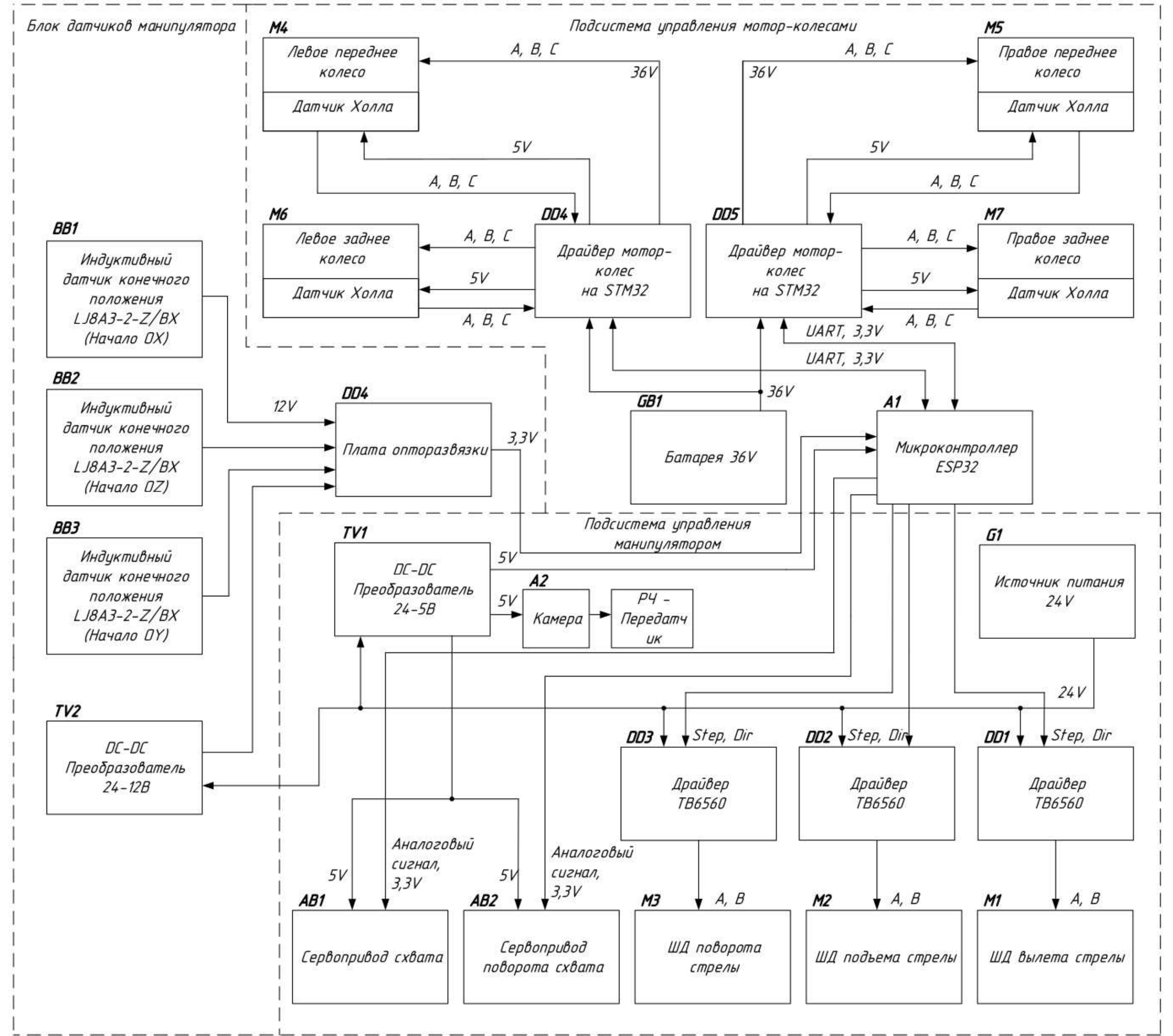
x — для вылета стрелы
y — для поворота стрелы
z — для подъема стрелы
d — для сжатия схвата
t — для поворота схвата
Буквы должны вводиться в нижнем регистре.

				ВКР.214.010.150304.08		
Изм.	Лист	Н. док.	Подп.	Дата	Схема системы управления манипулятора	Лист
Разраб.	Курочкин В.А.	Скрипко О.В.	Скрипко О.В.	20.05.14		Масса
Проб.	Скрипко О.В.	Скрипко О.В.	Скрипко О.В.	20.05.14		Масштаб
Т. контр.	Скрипко О.В.	Скрипко О.В.	Скрипко О.В.	20.05.14		Лист 1
Н. контр.	Скрипко О.В.	Скрипко О.В.	Скрипко О.В.	20.05.14	Разработка системы управления мобильным роботом-манипулятором	Листов 6
Утв.	Скрипко О.В.	Скрипко О.В.	Скрипко О.В.	20.05.14		АМГУ 141-08



Кинематическая схема манипулятора

Поз. Обозн.	Наименование	Кол.	Примеч.
1	Башня, обеспечивает движение вдоль оси Z (по вертикали)	1	
2	Рабочая стрела, обеспечивает движение вдоль оси X (по горизонтали)	1	
3	Опорно-поворотное устройство, вращение башни вокруг оси Z	1	
4	Каретка с ШВП осей Z и X	1	
5	ШД (GD57STH56-2808A) вылета стрела	1	
6	ШД (GD57STH56-2808A) спуска/подъема	1	
7	ШД (GD57STH56-2006A) поворота	1	
8	ШД (Nema 17 42CM06) схвата	1	
9	Схват	1	
10	Индуктивный датчик (LJ8A3-2-Z/BX) конечного положения по оси X	1	
11	Индуктивный датчик (LJ8A3-2-Z/BX) конечного положения по оси Y	1	
12	Индуктивный датчик (LJ8A3-2-Z/BX) конечного положения по оси Z	1	
13	Флажок конечного положения по оси X	1	
14	Флажок конечного положения по оси Y	1	
15	Флажок конечного положения по оси Z	1	



Структурная схема всей системы

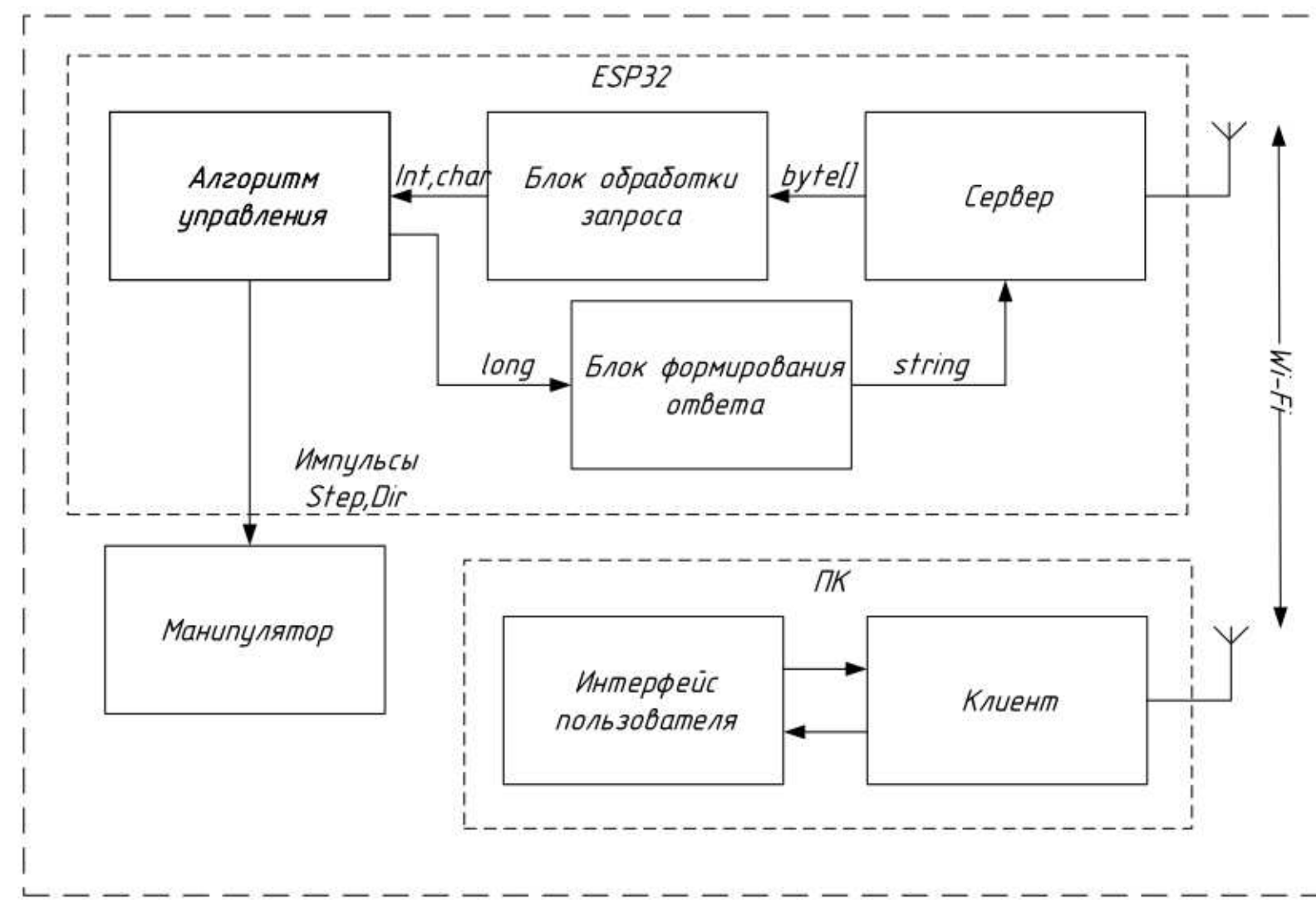
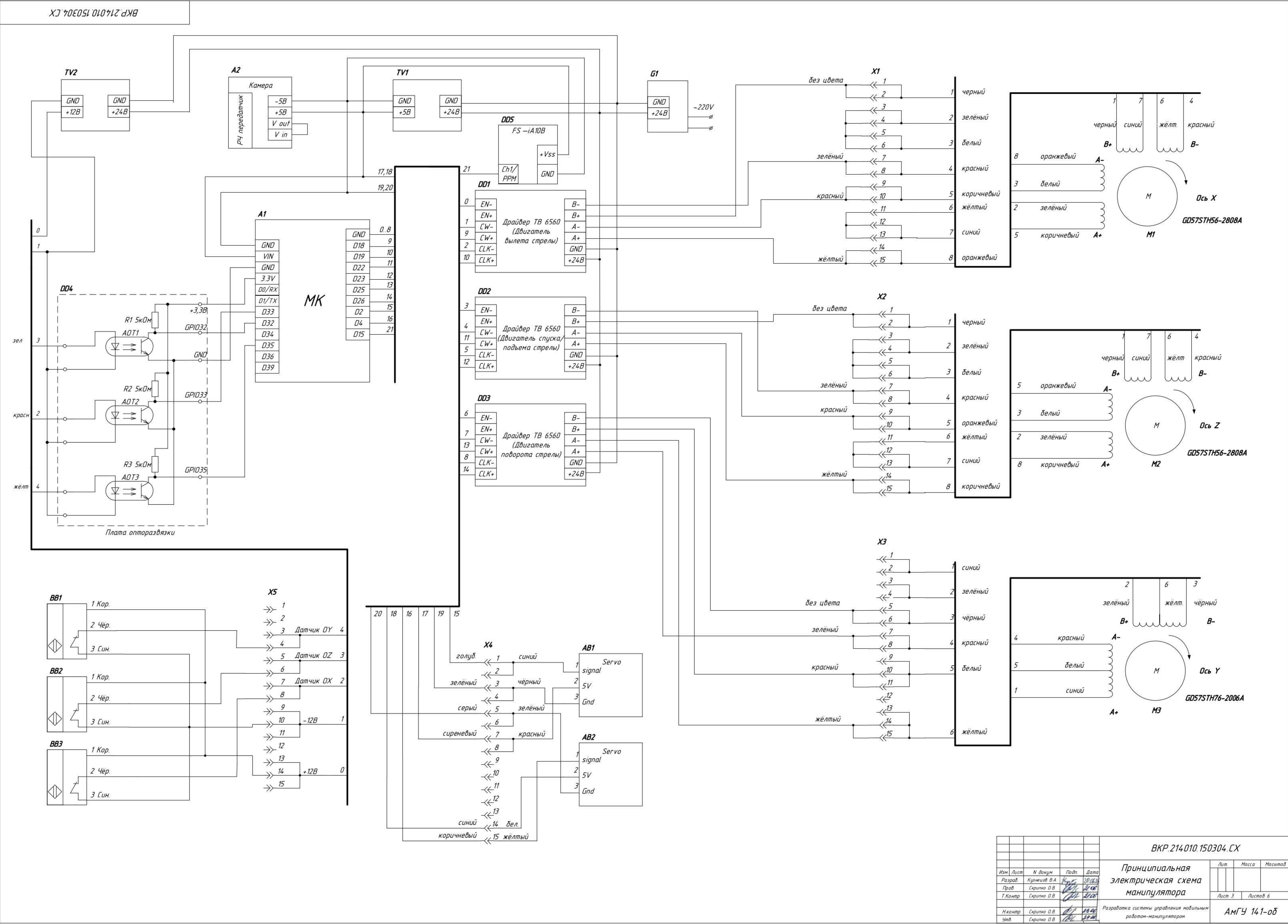
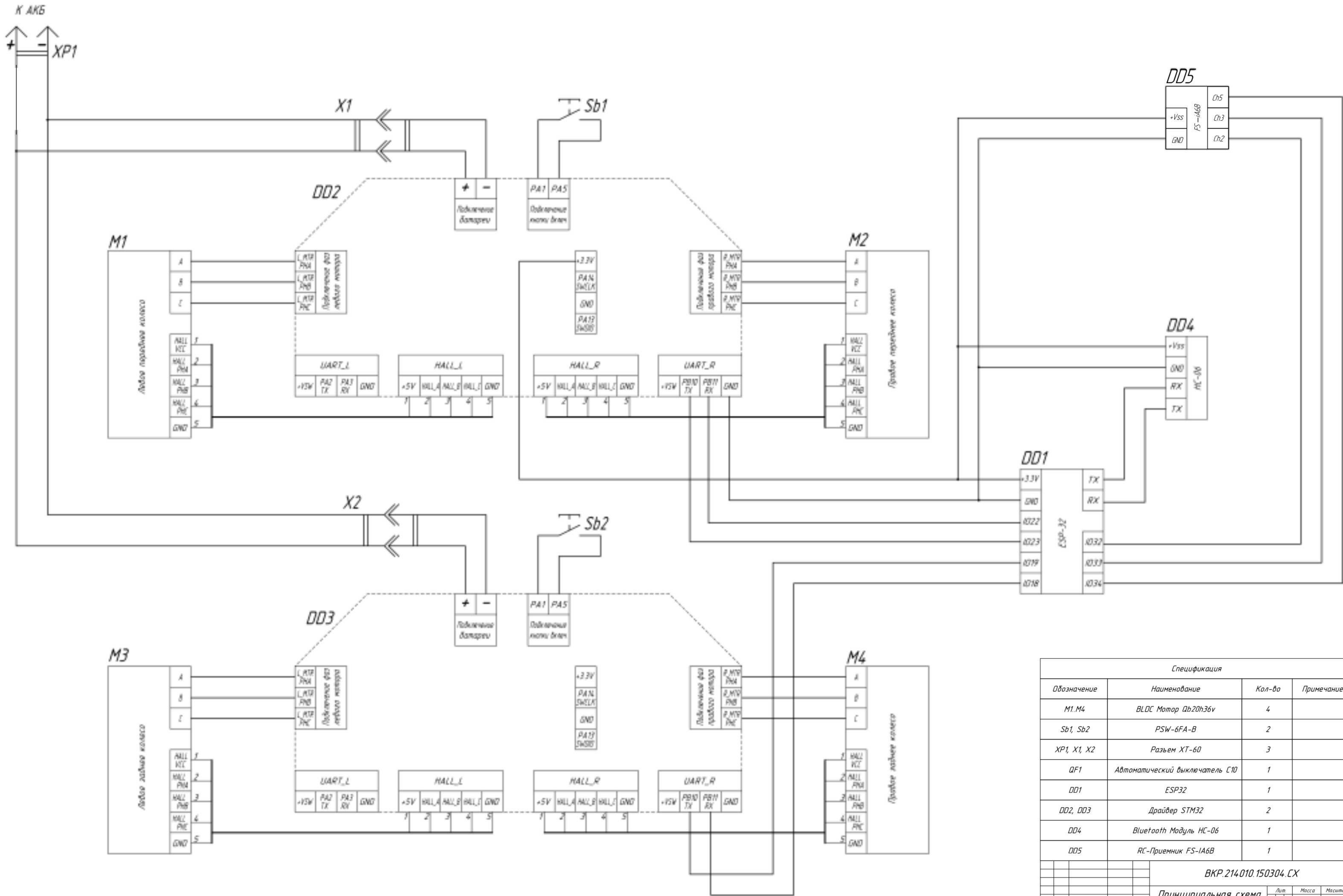


Схема управления манипулятором по сети Wi-Fi

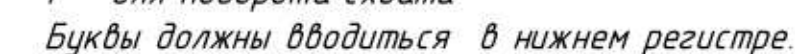
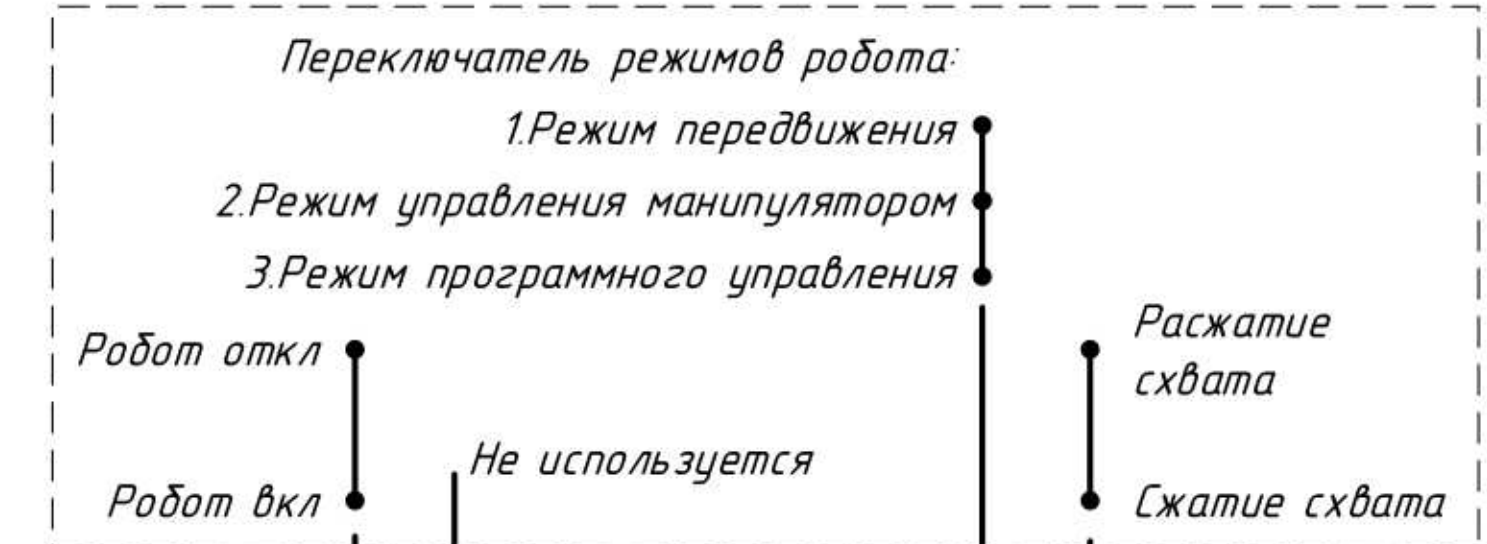
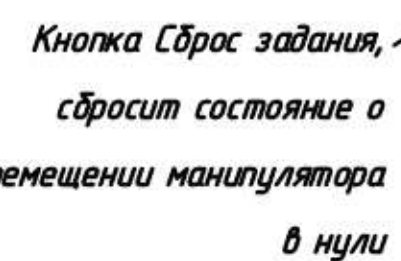
Спецификация к структурной схеме			
Обозначение	Наименование	Кол-во	Примечание
A1	Плата Esp32-WROOM DevKit-V1	1	
A2	Камера FPV LST S2 OSD с передатчиком	1	
G1	Блок питания KJ S-24-24 AC-DC	1	
GB1	Аккумулятор 10S2P 36V 4.4Ah Li-ion	1	
TV1	Понижающий преобразователь HW-319	1	
TV2	DC-DC преобразователь LM2596S	1	
DD1, DD3	Драйвер ШД Toshiba TB6560AHQ	3	
D4	Плата опторазвязки	1	
BB1, BB3	Индуктивный датчик LJ8A3-2-Z/AX	3	
M1, M2	Шаговый двигатель GD57STH56-2808A	2	
M3	Шаговый двигатель GD57STH76-2006A	1	
AB1, AB2	Сервопривод MG996R	2	
DD4, DD5	Плата драйвера мотор-колес STM32	2	
M4, M7	Мотор-колесо Qb20h36v	4	
ВКР.214.010.150304.CX			
Структурная и кинематическая схемы манипулятора			
Разработка системы управления мобильным роботом-манипулятором			
Изм.	Лист	И. докум.	Подп.
Разраб.	Курт	18.06.21	
Проб.	Курт	18.06.21	
Т. контр.	Курт	18.06.21	
И. контр.	Курт	18.06.21	
Утв.	Курт	18.06.21	
Лист 2		Листов 6	
АмГУ 141-08		Формат А1	



ВКР.214.010.150304.СХ					Лит			Масса	Масштаб
Изм	Лист	И докум	Подп	Дата	Принципиальная электрическая схема манипулятора				
Разраб	Кузнецов В.А.	Кур	10.06.18						
Проб	Скрипка О.В.	Скр	10.08.18						
Т.Контр	Скрипка О.В.	Скр	10.08.18		Разработка системы управления мобильным роботом-манипулятором				
Н.контр	Скрипка О.В.	Скр	10.08.18						
Утв	Скрипка О.В.	Скр	10.08.18						
Лист 3					Листов 6				
АМГУ 141-08					Формат А1М				



Спецификация			
Обозначение	Наименование	Кол-во	Примечание
M1..M4	BLDC Мотор Qb20h36v	4	
Sb1, Sb2	PSW-6FA-B	2	
XP1, X1, X2	Разъем XT-60	3	
QF1	Автоматический выключатель C10	1	
DD1	ESP32	1	
DD2, DD3	Драйвер STM32	2	
DD4	Bluetooth Модуль HC-06	1	
DD5	RC-Приемник FS-1A6B	1	
ВКР.214.010.150304.CX			
Принципиальная схема управления мотор-колесами			
Изм	Лист	№ докум	Подп
Разраб	Кузнецов В.А.	Дата	20.06.2018
Проб	Скрипка О.В.	Скрипка О.В.	21.06.2018
Т.контр	Скрипка О.В.	Скрипка О.В.	21.06.2018
Н.контр	Скрипка О.В.	Скрипка О.В.	21.06.2018
Утв	Скрипка О.В.	Скрипка О.В.	21.06.2018
Разработка системы управления мобильным роботом-манипулятором			
Лит			
Масса			
Масштаб			
Лист 4			
Листов 6			
АМГУ 141-08			



*Схема управления роботом с помощью РС -
пультa*

PROGRAM AIM

