

Некоммерческое Акционерное Общество «Казахский Национальный  
Исследовательский Технический Университет имени К.И.Сатпаева»

Институт Автоматики и информационных технологий

Кафедра «Робототехники и технических средств автоматики»



Кокшаров Даниил Олегович

Разработка системы управления робота манипулятора

## **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

Специальность 6В07113 – Робототехника и мехатроника

Алматы 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ  
КАЗАХСТАН

Некоммерческое акционерное общество «Казахский национальный  
исследовательский технический университет имени К.И.Сатпаева»



Институт Автоматики и информационных технологий

Кафедра Робототехники и технических средств автоматки



**ДОПУЩЕН К ЗАЩИТЕ**  
Заведующий кафедрой РТиТСА  
кандидат технических наук,  
ассоциированный профессор  
Ожикенов К. А.  
«31» мая 2024 г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
к дипломному проекту

На тему: «Разработка системы управления робота манипулятора»

6B07111 – Робототехника и мехатроника

Выполнил

Кокшаров Даниил Олегович

Рецензент

Научный руководитель

Кандидат технических наук, доцент

Ассоциированный профессор

Жеменкеев Е.К.

Тулешов Е. А.

«30» май 2024 г.

«31» май 2024 г.

Алматы 2024

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ  
КАЗАХСТАН

Некоммерческое акционерное общество «Казахский национальный  
исследовательский технический университет имени К.И.Сатпаева»

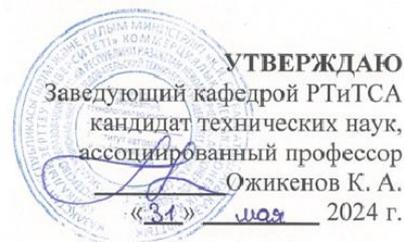


SATBAYEV  
UNIVERSITY

Институт Автоматики и Информационных Технологий

Кафедра Робототехники и технических средств автоматики

6B07111 – Робототехника и мехатроника



**ЗАДАНИЕ**

**на выполнение дипломного проекта**

Студенту Кокшарову Даниилу Олеговичу

Тема: Разработка системы управления робота манипулятора

Утверждена приказом ректора университета № 548-Т/19 от «4» декабря 2023 г.

Срок сдачи законченной работы «4» июня 202 г.

Исходные данные к дипломному проекту:

Теоретические материалы по Конструирование узлов и механизмов

Теоретические материалы по Arduino UNO

Теоретические материалы по ESP8266

Перечень подлежащих разработке в дипломном проекте вопросов:

- Исследовать актуальность систем управления для робота манипулятора
- Исследовать проблему использования методов систем управления.
- 

Перечень графического материала (с точным указанием обязательных чертежей): представлены слайдов презентации работы

Рекомендуемая основная литература: из \_\_\_\_\_ наименований

**ГРАФИК**  
подготовки дипломной работы (проекта)

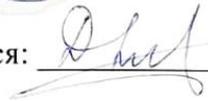
Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Исследовательская часть	02.03.2024	Выполнено
Теоретическая часть	18.03.2024	Выполнено
Практическая часть	15.04.2024	Выполнено
Специальная часть	17.05.2024	Выполнено

**Подписи**

консультантов и норм контролера на законченную дипломную работу  
(проект) с указанием относящихся к ним разделов работы (проекта)

Наименование разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Норм контролер	Игембай Ерболат Айдынулы, магистр технических наук, преподаватель	30.05.2024	

Научный руководитель:  Тулешов Е.А.

Задание принял к исполнению обучающийся:  Кокшаров Д. О.

Дата : «30» мая 2024

## **АННОТАЦИЯ**

Дипломная работа посвящена разработке системы управления робота манипулятора. Основная мысль проекта заключается в том, чтоб воспроизводить работу манипулятора дистанционно без дополнительного блока управления, для облегчения работы пользователя. Система управления манипулятором имеет решающее значение для обеспечения точного, эффективного и безопасного функционирования робота. Она является центральной частью робота, ответственной за получение входных данных, обработку и генерацию выходных сигналов, управляющих приводами робота. Эффективная система управления роботом-манипулятором имеет решающее значение для расширения возможностей робототехники.

## **АҢДАТПА**

Дипломдық жұмыс Робот-манипуляторды басқару жүйесін дамытуға арналған. Жобаның негізгі мақсаты-пайдаланушының жұмысын жеңілдету үшін қосымша басқару блогынсыз манипулятордың жұмысын қашықтан ойнату. Роботтың дәл, тиімді және қауіпсіз жұмысын қамтамасыз ету үшін манипуляторды басқару жүйесі өте маңызды. Бұл робот жетектерін басқаратын кіріс, өңдеу және шығыс сигналдарын жасауға жауап беретін роботтың орталық бөлігі. Робототехниканы басқарудың тиімді жүйесі робототехниканың мүмкіндіктерін кеңейту үшін өте маңызды.

## **ANNOTATION**

The thesis is devoted to the development of a Robot manipulator control system. The main goal of the project is to remotely reproduce the operation of the manipulator without an additional control unit to facilitate the user's work. The manipulator control system is crucial to ensure accurate, efficient and safe operation of the robot. It is the central part of the robot, which is responsible for generating input, processing and output signals that control the robot drives. An effective robotics control system is crucial to enhance the capabilities of robotics.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Исследовательская часть	5
1.1 Актуальность	5
1.2 Виды СУ для роботов манипуляторов	6
1.3 Постановка цели и задачи	7
2 Теоретическая часть	8
2.2 Расчет модели манипулятора	13
2.3 Разработка алгоритма решения задачи системы управления	20
3 Практическая часть	25
3.1 Проектирование электросхемы	25
3.2 Проектирование 3D – модели	25
3.3 Расчет себестоимости	27
4 Тестирование разработанного робота манипулятора	28
5 Заключение	30
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	31
Приложение А	31

## ВВЕДЕНИЕ

С начала появления первых манипуляторов использовалось ручное управление, но с развитием электроники и ЭВМ системы управления начали переходить на новые уровни от исполнительного, в котором происходит выполнение отдельных команд до высшего уровня с корректировкой действий под контролем искусственного интеллекта. Основная проблема работы на предприятиях - не полное понимание работы оборудования, у работников уходят годы, чтоб освоить все аспекты системы. Создание системы управления с понятным и легко обучаемым интерфейсом повысит КПД производства и облегчит работу пользователей.

В наше время, на предприятиях уже не только используют роботов манипуляторов как отдельное оборудования, но и как полноценные отлаженные системы.

Например, в компании Amazon существуют полностью автономные склады с роботами погрузчиками и сортировщиками товаров, где люди работают как операторы и технический персонал.

На территории нашей страны тоже имеются автономные предприятия, такие как ТОО «Astana Ceramic». Завод представляет собой современное автоматизированное производство, производительная мощность которого после модернизации составляет 30 млн. условного кирпича в год. На заводе полностью автоматизирован процесс от загрузки глины до обжига и сортировки готового кирпича.

Для эффективной работы робота-манипулятора необходима надежная и гибкая система управления, которая позволит программировать его действия, контролировать движение и взаимодействие с окружающей средой.

## **1 Исследовательская часть**

В разделе представлены результаты исследований в актуальности данной работы и проекта, а также Виды СУ для роботов манипуляторов и вывод, отражающий необходимый состав характеристик и функционал систем управления робота манипулятора.

### **1.1 Актуальность**

Для повышения качества и объема в производственном процессе применяют роботов, которые могут не только заменять человека на рабочем месте, но также дополнять и улучшать его работу, тем самым увеличивая КПД производства и не увольняя сотрудников.

Зачастую, пользователям необходимо владеть определенными навыками и знаниями для управления промышленным роботом. Основная задача разработчиков является улучшение характеристик робота, и создание более понятных систем для пользователя.

Робота манипулятор - автоматическое устройство, предназначенное для выполнения различных задач, таких как поднятие, перемещение, сборка и размещение объектов. Разработка систем управления для робота манипулятора включает в себя создание алгоритмов и программ, которые позволяют роботу эффективно выполнять поставленные перед ним задачи. Большинство роботов, которые используются в промышленности, являются манипуляторами, управляемые микроконтроллерами.

К этому классу относятся промышленные роботы, а также манипуляционные роботы, предназначенные для замены человека в тех случаях, когда он не может присутствовать на месте выполнения операции или выполнять ее самостоятельно – под водой, в космическом пространстве, в условиях повышенной радиации и т.п. Существует так же понятие коллаборативный робот (кобот) – устройство, предназначенное для совместной работы вместе с человеком для производства или создания различных продуктов.

В связи с наличием указанных особенностей, для внедрения манипуляционного робота в производственный процесс требуется специально разрабатываемые системы управления. Они служат для организации взаимодействия между человеком-оператором и роботом, и обеспечивают

выполнение процессов, необходимых для автоматизации технологической операции.

## **1.2 Виды СУ для роботов манипуляторов**

В зависимости от типа робота и сложности выполняемого технологического процесса структура системы управления может иметь различное количество уровней.

Чаще всего выделяют четыре уровня управления:

1. Высший (четвертый) – уровень искусственного интеллекта, на котором происходит распознавания обстановки, решение задачи принятия решения, накопление опыта работы и самообучение;

2. Стратегический (третий) – уровень организации адаптивного управления с помощью ЭВМ, которая поставленное задание расчленяет на элементарные операции (подвести схват к детали, сориентировать схват относительно детали, захватить ее и т.п.);

3. Tактический (второй) – уровень, где элементарные операции распределяются на движения отдельных СП;

4. Исполнительный (первый) – уровень, на котором осуществляются заданные движения отдельных СП.

Если на первом и втором уровне осуществляется управление состоянием в виде управления движением распознавания обстановки, то на третьем – управление сменой состояний. Кроме иерархичности структуры системы управления, одним из основополагающих принципов управления МР является комбинированный характер управления, в основе которого лежит сочетание, комбинация нескольких способов управления. Так, например, с точки зрения участия человека в управлении возможны как ручное, дистанционное или автоматическое управление, так и их сочетание – интерактивное (автоматизированное, дистанционно-автоматическое (ДАУ)) управление. Наилучшее качество управления на тактическом и исполнительном уровне достигается применением комбинированных систем, сочетающих в себе принципы программного управления и управления в функции текущего состояния внешней среды.

В зависимости от того, для каких целей будет использовано ПО, оно подразделяется на средства:

Технического обслуживания, позволяющие поддерживать роботов в наилучшем рабочем состоянии с помощью инструментов устранения неполадок, технического обслуживания и калибровки.

Визуализация и мониторинг автоматизированных промышленных роботов, отображающие состояние устройств и уровни их работоспособности и производительности.

Моделирования и симуляции работа технических комплексов. Эти инструменты и средства автономного программирования помогают разрабатывать и отлаживать решения роботизации до их внедрения.

Разработки дополнительного ПО. С помощью этих средств создаются и настраиваются индивидуальные приложения для управления роботами.

Программного управления контроллерами, предназначенные для программирования роботов непосредственно через логические контроллеры с использованием существующих периферийных средств.

Прикладного программного обеспечения - в виде интерфейсов к внешним компонентам или устройствам, предназначенных для включения конкретных действий роботов.

Также программное обеспечение для роботов можно разделить на две категории: контролирующее и управляющее ПО, включающее в себя графические интерфейсы для работы с телеуправляемыми системами, ПО для взаимодействия с автономно действующими роботами и для формирования схем эксплуатации мобильных промышленных роботов;

ПО для задач, которое включает в себя простые интерфейсы с возможностью перетаскивания и настройки маршрутов перемещения и специализированные программы, созданные для развертывания конкретных приложений.

### **1.3 Постановка цели и задачи**

Цель:

1. Разработка СУ для робота манипулятора на основе Arduino
2. Настроить управления через Web-сервер.

Задачи:

1. Исследовать виды СУ для роботов манипуляторов.
2. Спроектировать робота манипулятора для СУ.
3. Провести тестирование программы.

## 2 Теоретическая часть

### 2.1 Расчет кинематики манипулятора

Для расчётов кинематики робота, необходимо построить кинематическую схему манипулятора. Кинематическая схема – изображение кинематической связи между отдельными звеньями механизма или изделия, также показана последовательность передачи движения от двигателя через передаточный механизм к рабочему органу.

Разработанная кинематическая схема манипулятора

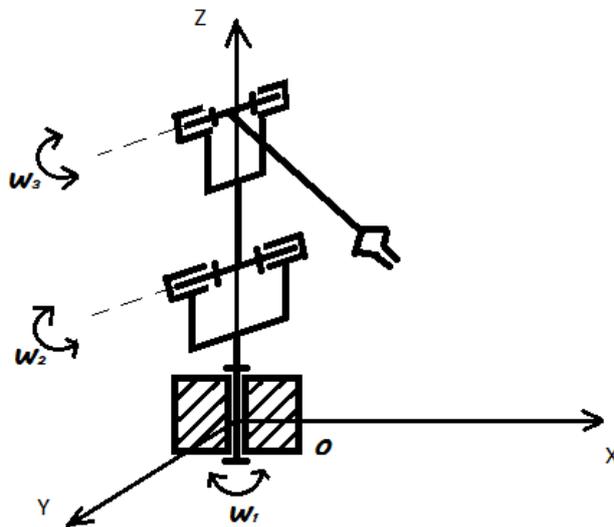


Рисунок 2.1 - Разработанная мной кинематическая схема моего манипулятора

На схеме изображен манипулятор, представляющий собой систему из трех подвижных частей. Подвижной оси  $w_1$ ,  $w_2$ ,  $w_3$ .

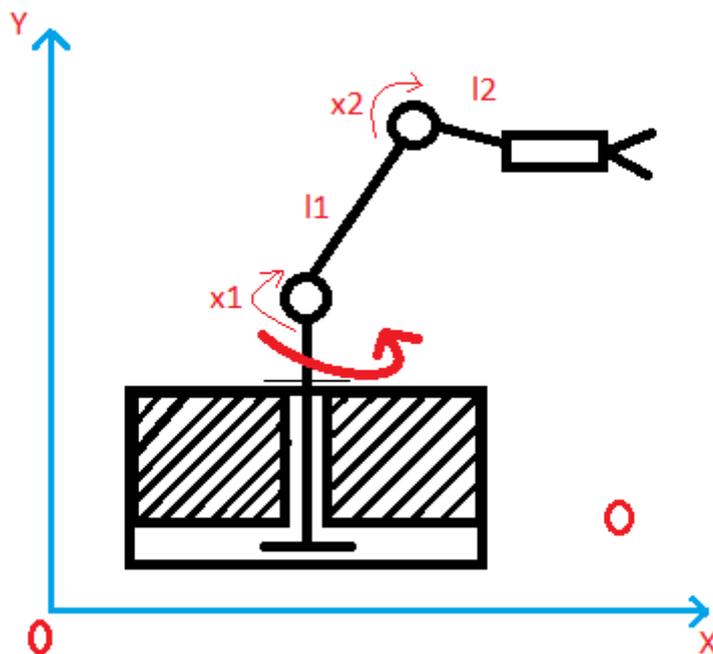


Рисунок 2.2 - Кинематическая схема в плоскости XY для расчета прямой и обратной задач кинематики.

Сперва, рассчитаем степень свободы данного манипулятора.

Степень свободы – это количество способов, при которых система движется, при этом не нарушая наложенных ограничений, и имеет минимум координат, которые могут полностью определять положение системы.

Каждая кинематическая пара имеет свой класс, который отображает сколько степеней свободы она связывает в перемещении.

1. 3 кинематических вращательные пары, низшего вида.
2. Все 3 пары имеют 5 класс.
3. Каждая пара имеет 1 степень свободы.

Для расчета степени свободы всей кинематической цепи, используют формулу Сомова-Малышева:

$$W=6n-5P_5-4P_4-3P_3-2P_2-P_1 \quad (1)$$

Где,  $n$  – количество звеньев в цепи,  $P_5, P_4, P_3, P_2, P_1$  – количество пар пятого, четвертого, третьего, второго, первого классов.

В данном случае:

$$n = 3, P_5 = 3$$

Тогда:

$$W = 6 * 3 - 5 * 3 - 4 * 0 - 3 * 0 - 2 * 0 - 0 = 3 \quad (3)$$

Следует, что данная система манипулятора имеет 3 степени свободы.

В ходе выполнения работы, манипулятор будет менять положение рабочего органа. Для этого нужно определять местоположение и траекторию движения. Определять будем эти значения с помощью прямой и обратной задачи кинематики.

Прямая задача — это вычисление положения  $(X, Y, Z)$  рабочего органа манипулятора по его *кинематической схеме* и заданной ориентации  $(A_1, A_2 \dots A_n)$  его звеньев ( $n$  — число степеней свободы манипулятора,  $A$  — углы поворота).

Обратная задача — это вычисление углов  $(A_1, A_2 \dots A_n)$  по заданному положению  $(X, Y, Z)$  рабочего органа и опять же известной схеме его кинематики.

Решение прямой задачи — где будет находиться рабочий орган манипулятора, при заданных углах его суставов, а обратная задача, наоборот: как нужно сделать манипулятору, чтобы его рабочий орган оказался в заданном положении.

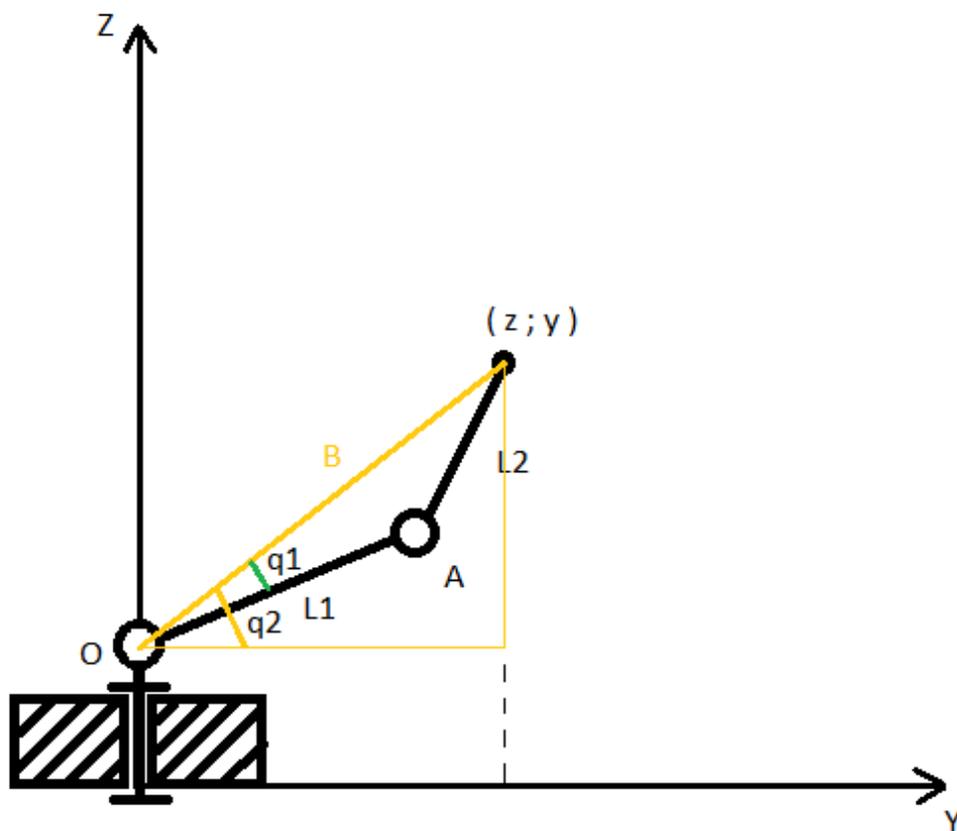


Рисунок 2.3 - Кинематическая схема в плоскости ZY для расчета прямой и обратной задач кинематики.

Прямая задача:

Имеем две системы отсчёта — первая, связанная с точкой крепления плеча  $L1$  —  $O$ , а вторая — с началом координат в точке крепления локтя —  $A$ . Найдём смещение второй системы относительно первой (координаты точки  $A$  в системе отсчёта  $O$ ):

$$Y_A = L1 * \cos(Q1) \quad (2)$$

$$Z_A = L1 * \sin(Q1) \quad (3)$$

Координаты  $(x, y)$  в системе отсчёта локтя:

$$Y'' = L2 * \cos(Q2) \quad (6)$$

$$Z'' = L2 * \sin(Q2) \quad (7)$$

По рисунку видно, что в системе O, локоть L2 повёрнут относительно плеча на Q1+Q2:

$$Y' = L2 * \cos(Q1+Q2) \quad (8)$$

$$Z' = L2 * \sin(Q1+Q2) \quad (9)$$

Значит:

$$y = XA + x' = L1 * \cos(Q1) + L2 * \cos(Q1+Q2) \quad (10)$$

$$z = YA + y' = L1 * \sin(Q1) + L2 * \sin(Q1+Q2) \quad (11)$$

Обратная задача:

Рисунок 2, теперь нужно найти такие углы Q1 и Q2, которые позволят манипулятору с плечом L1 и локтем L2 поместить рабочий орган в заданную точку (x, y)

Проведём прямую B, соединяющую начало координат O с заданной точкой (y, z).

$$B^2 = y^2 + z^2 \quad (12)$$

$$y = B * \cos(q1) \quad (13)$$

$$z = B * \sin(q1) \quad (14)$$

q1 — угол между осью OX и прямой B  
q2 — угол между прямой B и плечом L1

отсюда:

$$Q1 = q1 - q2 \quad (15)$$

$$q1 = \arccos( y/B ) \text{ или } q1 = \arctg( z/y ) \quad (16)$$

$q_2$  находим при помощи **теоремы косинусов**, которая говорит: Для плоского треугольника со сторонами  $a, b, c$  и углом  $\alpha$ , противолежащим стороне  $a$ , справедливо соотношение:

$$a^2 = b^2 + c^2 - 2*b*c*\cos(\alpha) \quad (17)$$

в нашем случае, по теореме косинусов:

$$L_2^2 = B^2 + L_1^2 - 2*B*L_1*\cos(q_2) \quad (18)$$

$$\Rightarrow q_2 = \arccos( (L_1^2 - L_2^2 + B^2) / (2*B*L_1) ) \quad (19)$$

$$Q_1 = q_1 - q_2 = \arccos( y/B ) - \arccos( (L_1^2 - L_2^2 + B^2) / (2*B*L_1) ) \quad (20)$$

По той же теореме косинусов найдём угол  $Q_2$ : как видно по рисунку, угол  $Q_2 = 180$  — угол  $OAx$

$$B^2 = L_1^2 + L_2^2 - 2*L_1*L_2*\cos(\pi - Q_2) \quad (21)$$

$$Q_2 = \pi - \arccos( (L_1^2 + L_2^2 - B^2) / (2*L_1*L_2) ) \quad (22)$$

Вывод данных уравнений обеспечит качественное движения для автономного передвижения манипулятора из относительной точки  $A$  в точку  $B$ .

## 2.2 Расчет модели манипулятора

Для этого необходимо построить кинематическую схему узлов, для определения комплектующих (привод, материал передач).

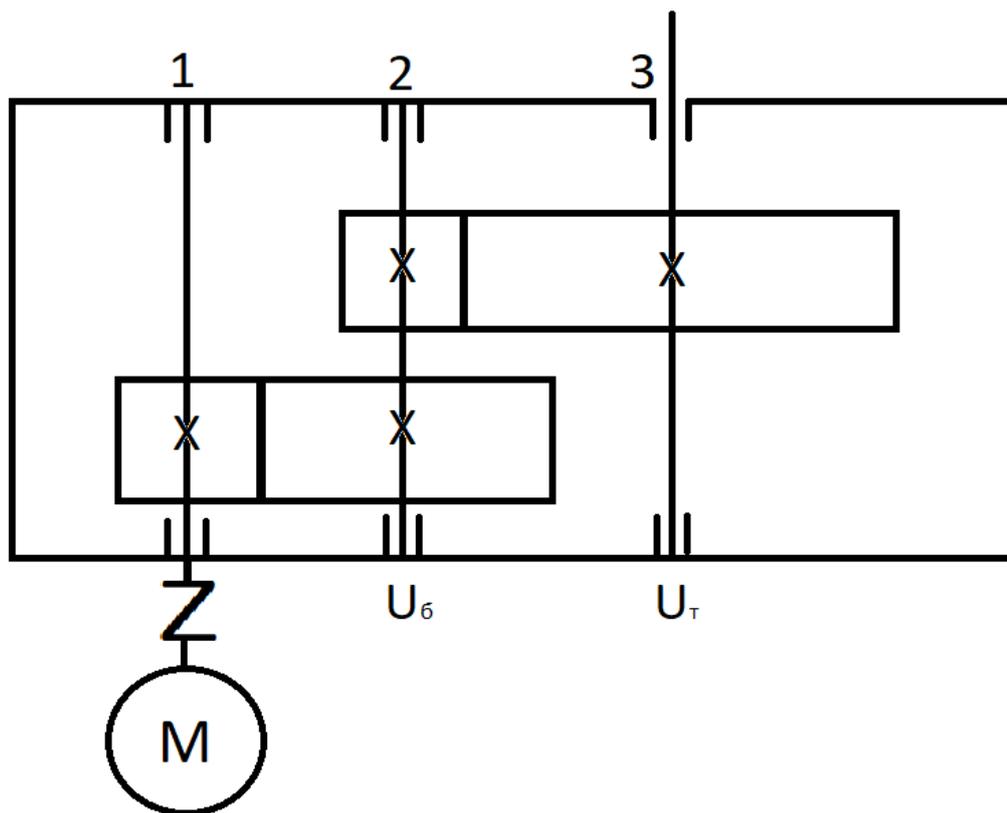


Рисунок 2.4 - На рисунке изображена схема редуктора, на ось Z (Основание манипулятора).

1, 2, 3 – валы редуктора

Uб – быстроходная зубчатая передача

Uт – тихоходная зубчатая передача.

Входные характеристики:

$F = 15 \text{ Н}$  (Сила, которая должна быть на выходном валу )

$V = 0,2 \text{ м/с}$  (скорость перемещения по оси Z)

$Dб = 0.08 \text{ м}$  (Диаметр вращающегося барабана)

Для начала необходима выбрать электродвигатель.

Нужно выбрать потребляемую мощность и частоту вращения двигателя.

Потребляемая мощность определяется по формуле:

$$P_v = F * V \text{ (Вт)} \quad (23)$$

$$P_v = 15 * 0.2 = 3 \text{ Вт} \quad (24)$$

Тогда требуемая мощность:

$$P_{э.тр} = P_v / \eta \text{ (Вт)} \quad (25)$$

Где  $n = n_1 * n_2 * n_3 * \dots$

Здесь  $n_1, n_2, n_3$  – КПД отдельных звеньев кинематической цепи  
Значение которых можно принимать из таблицы 2.1

Тип передачи	$\eta$
Зубчатая (с опорами, закрытая):	
цилиндрическая	0,96...0,98
коническая	0,95...0,97
Планетарная (закрытая):	
одноступенчатая	0,9...0,95
двухступенчатая	0,85...0,9
Червячная (закрытая) при передаточном числе:	
св. 30	0,7...0,8
св. 14 до 30	0,75...0,85
св. 8 до 14	0,8...0,9
Ременная (все типы)	0,94...0,96
Цепная	0,92...0,95
Муфта соединительная	0,98
Подшипники качения (одна пара)	0,99

Таблица 2.1 – КПД механических передач

Рассчитаем:

$$n = n_m * n_z^2 * n_{оп}^3 = 0.98 * 0.97^2 * 0.99^3 = 0.913 \quad (26)$$

$$n_m = 0.98$$

$$n_z = 0.97$$

$$n_{оп} = 0.99$$

Определим мощность требуемого электродвигателя

$$P_{э.тр} = P_v / n \text{ (Вт)} \quad (27)$$

$$P_{э.тр} = 3 / 0.913 = 3.3 \text{ Вт} \quad (28)$$

Частота вращения приводного вала:

$$n_v = 60 * V / \pi * D_b \quad (29)$$

$$n_v = 60 * 0.2 / 3.14 * 0.08 = 47.77 \text{ об/мин} \quad (30)$$

Предварительно назначаем передаточное число из редуцируемого диапазона из таблицы 2.2

Таблица 1.2

Вид передачи	Твердость зубьев	Передаточное число	
		$u_{\text{эк}}$	$u_{\text{ред}}$
Зубчатая цилиндрическая: тихоходная ступень во всех редукторах ( $u_T$ )	$\leq 350 \text{ HB}$	2,5...5,6	6,3
	40...56 HRC <sub>3</sub>	2,5...5,6	6,3
	56...63 HRC <sub>3</sub>	2...4	5,6
быстроходная ступень в редукторах по развернутой схеме ( $u_6$ )	$\leq 350 \text{ HB}$	3,15...5,6	8
	40...56 HRC <sub>3</sub>	3,15...5	7,1
	56...63 HRC <sub>3</sub>	2,5...4	6,3
быстроходная ступень в соосном редукторе ( $u_6$ )	$\leq 350 \text{ HB}$	4...6,3	8
	40...56 HRC <sub>3</sub>	4...6,3	7,1
	56...63 HRC <sub>3</sub>	3,15...5	6,3
Коробка передач	Любая	1...2,5	3,15
Коническая зубчатая	$\leq 350 \text{ HB}$	1...4	6,3
	$\geq 40 \text{ HRC}_3$	1...4	5
Червячная	—	16...50	80
Цепная	—	1,5...3	4
Ременная	—	2...3	5

Таблица 2.2 – КПД механических передач

Также можно определить из какого материала будут шестерни. По шкале твердости HB у пластика из которого можно изготовить эти детали от 190 до 260 HB, тогда:

$U_6 = 3.15$  (быстроходная зубчатая передача)

$U_T = 2.5$  (тихоходная зубчатая передача)

Значит, требуемая частота вращения электродвигателя:

$$n_{\text{э. тр}} = n_{\text{в}} * U_6 * U_T \quad (31)$$

$$n_{\text{э. тр}} = 47.77 * 3.15 * 2.5 = 376.2 \text{ об/мин} \quad (32)$$

Следует, что для работы данного редуктора необходима как минимум:  
Требуемая мощность электродвигателя:

$$P_{\text{э. тр}} = 3.3 \text{ Вт}$$

Требуемая частота вращения электродвигателя:

$$n_{\text{э. тр}} = 376.2 \text{ об/мин}$$

Самый близкий по характеристикам электродвигатель это Nema 17 HS44401:

$P = 5 \text{ Вт}$   
 $n = 500 \text{ об/мин}$

Находим уточненное значение передаточного отношения:

$$U = n / n_b = 500 / 47.77 = 8.37 \quad (35)$$

$$U_T = 0.88 * \sqrt{8.37} = 2.54 \quad (34)$$

$$U_b = U / U_T = 8.37 / 2.54 = 3.3 \quad (33)$$

Так же рассчитываем все значения на 2 подвижных звеньях.

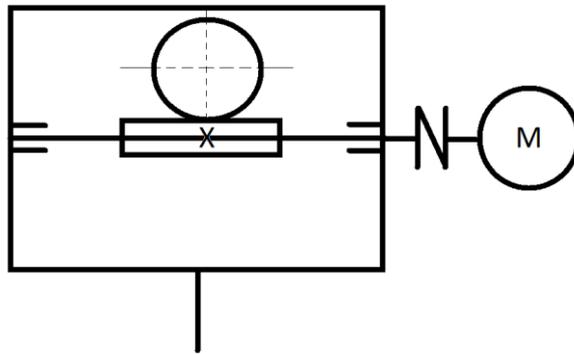


Рисунок 2.5 - Кинетическая схема червячной передачи с приводом.

$F = 15 \text{ Н}$  (Сила, которая должна быть на выходном валу )

$V = 0,1 \text{ м/с}$  (скорость перемещения по оси Z)

$D_b = 0.08 \text{ м}$  (Диаметр вращающегося барабана)

Потребляемая мощность определяется по формуле (23), подставим значения:

$$P_b = 15 * 0.1 = 1.5 \text{ Вт} \quad (35)$$

Тогда требуемая мощность:

$$P_{э.тр} = P_b / n \text{ (Вт)} \quad (25)$$

$$\text{Где } n = n_1 * n_2 * n_3 * \dots$$

Здесь  $n_1, n_2, n_3$  – КПД отдельных звеньев кинематической цепи

Значение которых можно принимать из таблицы 2.1

Рассчитаем:

$$n = n_m * n_z * n_{оп} = 0.98 * 0.97 * 0.99 = 0.94 \quad (26)$$

$$n_m = 0.98$$

$$n_z = 0.97$$

$$n_{оп} = 0.99$$

Определим мощность требуемого электродвигателя

$$P_{э.тр} = P_B / n \text{ (Вт)} \quad (27)$$

$$P_{э.тр} = 3 / 0.94 = 1.6 \text{ Вт}$$

Частота вращения приводного вала:

$$n_B = 60 * V / P_i * D_b \quad (29)$$

$$n_B = 60 * 0.1 / 3.14 * 0.08 = 23.9 \text{ об/мин}$$

Предварительно назначаем передаточное число из редуцируемого диапазона из таблицы 2.2

$$U_ч = 16 \text{ (червячная передача)}$$

Значит, требуемая частота вращения электродвигателя:

$$n_{э.тр} = n_B * U_б * U_T \quad (31)$$

$$n_{э.тр} = 23.9 * 16 = 382.4 \text{ об/мин}$$

Следует, что для работы данного редуктора необходима как минимум:

Требуемая мощность электродвигателя:

$$P_{э.тр} = 1.6 \text{ Вт}$$

Требуемая частота вращения электродвигателя:

$$n_{э.тр} = 382.4 \text{ об/мин}$$

Самый близкий по характеристикам электродвигатель это Nema 17 HS44401:

$$P = 5 \text{ Вт}$$

$$n = 500 \text{ об/мин}$$

Находим уточненное значение передаточного отношения:

$$U = n / n_B = 500 / 16 = 31.25 \quad (36)$$

Для третьего звена используется нагрузка только на мотор. Движение осуществляется при вращении электродвигателя без дополнительных механических передач.

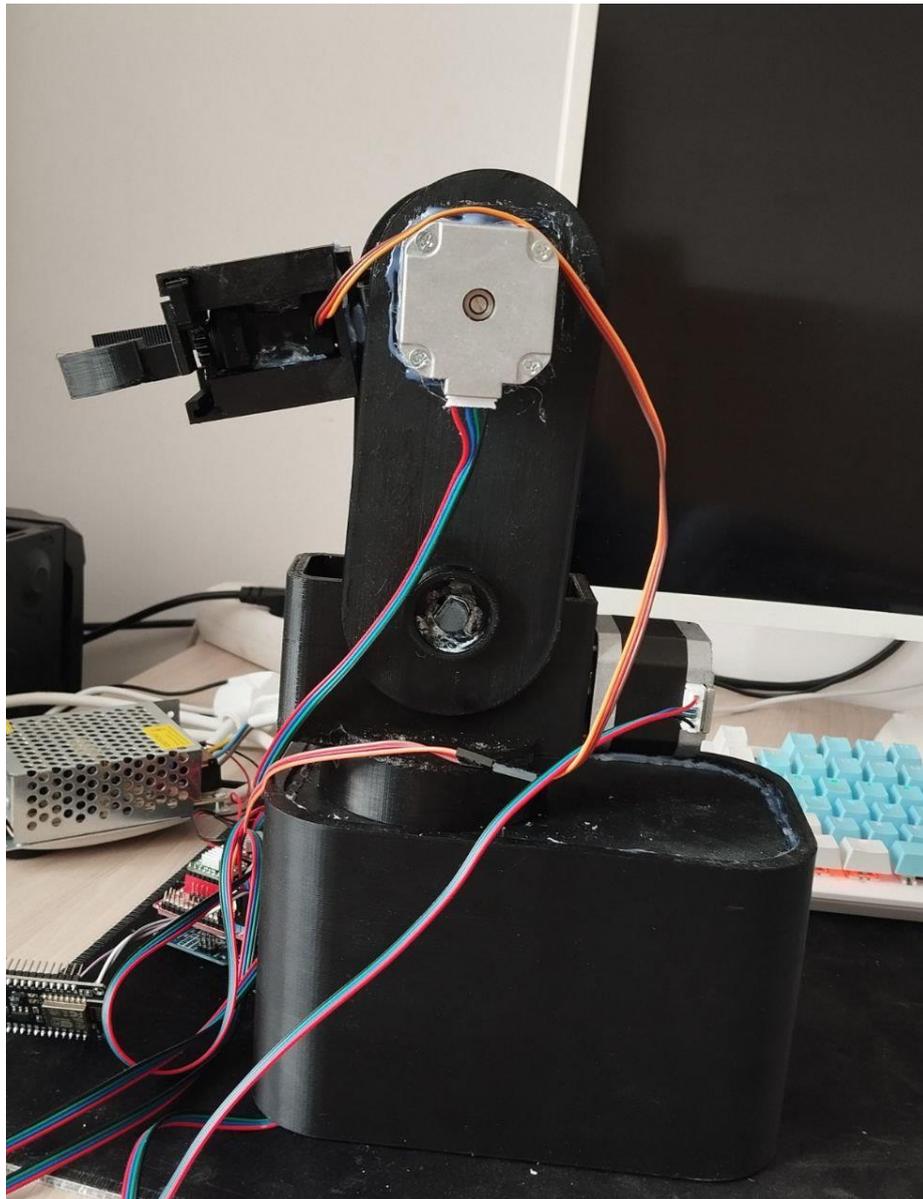


Рисунок 2.6 – Вид манипулятора в сборке

Определяем вращающий момент на валах приводов

Момент силы на приводном валу оси Z:

$$T1 = (F * D) * 0.5 \quad (37)$$

$$T1 = 15 * 0.08 * 0,5 = 0.6 \text{ Н * м} \quad (38)$$

Момент силы на приводном валу оси Y1 используя формулу (37):

$$T1 = 15 * 0.13 * 0,5 = 0.975 \text{ Н * м}$$

Момент силы на приводном валу оси Y2 используя формулу (37):

$$T1 = (F * L) * 0.5$$

$$T1 = 15 * 0.09 * 0,5 = 0.675 \text{ Н * м}$$

### 2.3 Разработка алгоритма решения задачи системы управления

В данном алгоритме процесс начинается с подачи (обмена) команд с веб-сервера на плату ESP8266, далее ESP8266 посылает команды на Arduino Uno, который принимает их в виде чисел от 0 до 10. Каждая команда соответствует своему числу. Далее команды и импульсы передаются на драйвера, затем драйвер подает импульсы на привод. Всей этой системой можно управлять дистанционно через устройство которое имеет доступ в интернет.

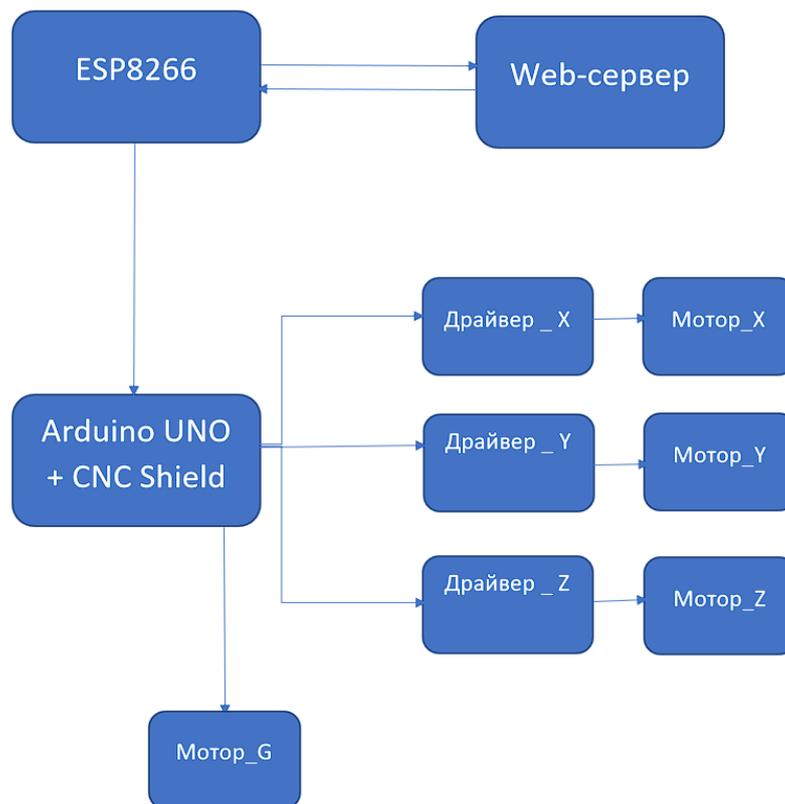


Рисунок 2.7 – Схема работы системы управления

Известна схема работы манипулятора, теперь строим алгоритм программы. Состоять будет из 2 частей:

1. Выполнение приводных функций на Arduino UNO
2. Обмен сервера и ESP8266, и подача сигналов управления на Arduino UNO

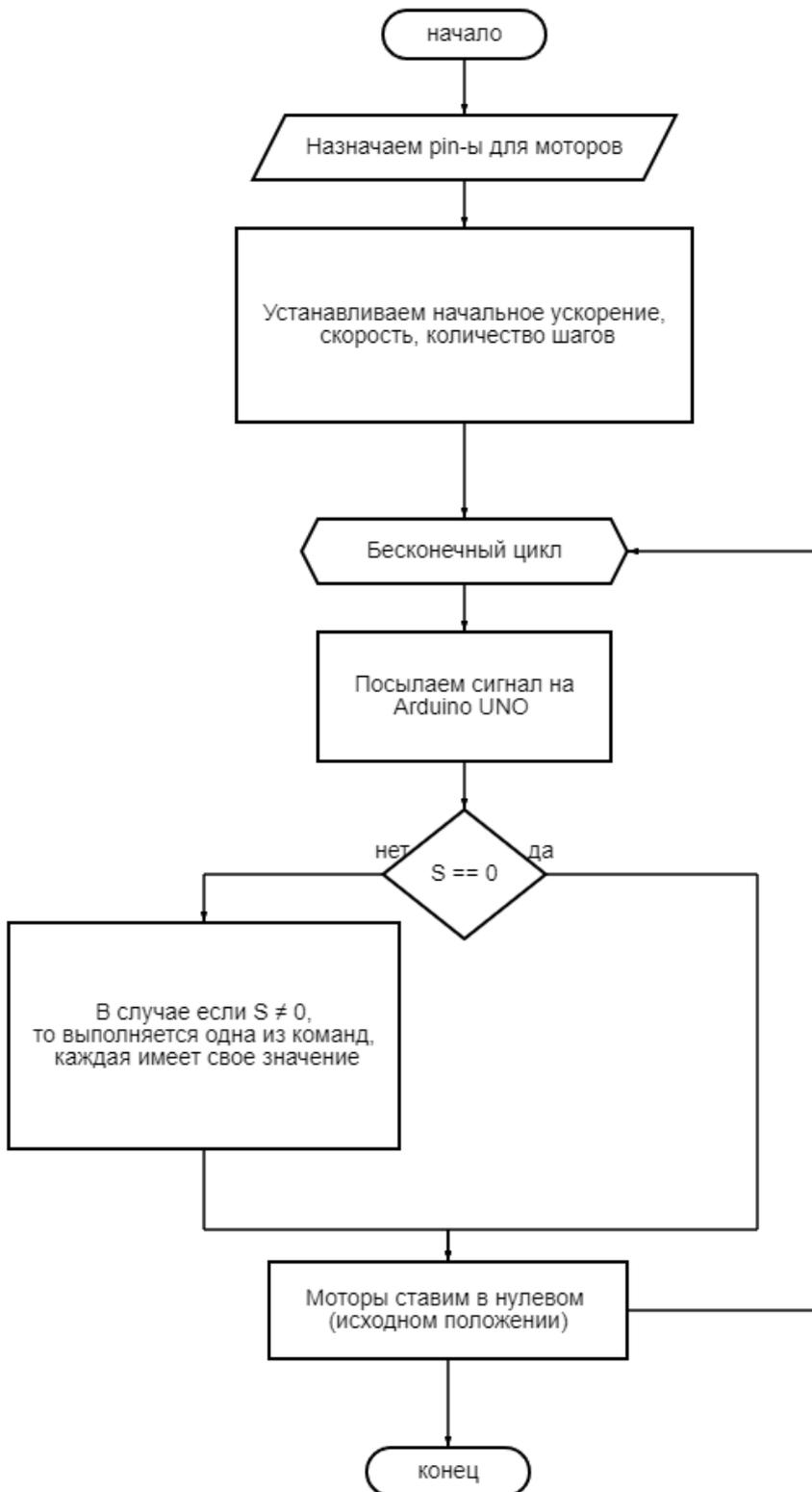


Рисунок 2.7 – Алгоритм приводных функций на Arduino UNO

В алгоритме начинаем с назначения выводов для моторов. Назначаем начальные значения для моторов: максимальную скорость, ускорение, скорость. В данном случае:

```
myStepperX.setMaxSpeed(1000);  
myStepperX.setAcceleration(50);  
myStepperX.setSpeed(200);
```

Эти операции нужно сделать с каждым шаговым мотором. Далее задаем адрес I2C равный 8, получаем отправленные сигналы на Arduino UNO.

```
Wire.begin(8); /* задаем на шине i2c 8 адрес */  
Wire.onReceive(receiveEvent1); /* регистрируем полученное событие */
```

При получении искомого сигнала который имеется в списке команд выполняется определенное действие. После выполнения команды, моторы записывают конечное положение как начальное. Для того, чтобы возвращаться в исходное положение используется счетчик, который записывает количество шагов для каждого привода, при нажатии команды Home моторы делают такое же количество шагов как на счетчике и робот возвращается в начальное положение.

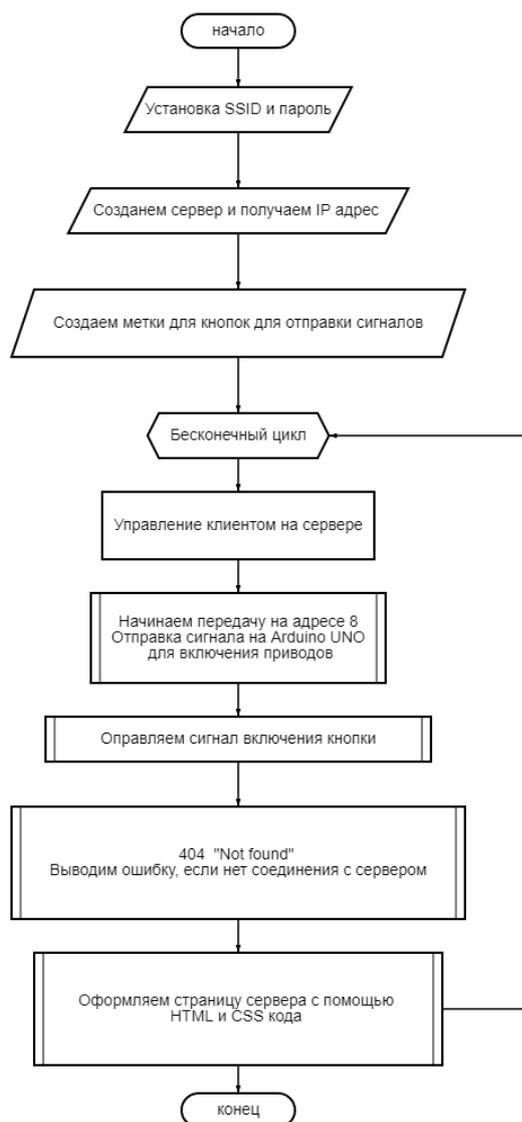


Рисунок 2.8 – Алгоритм принятие сигнала на движение моторов

С начала создается сервер и подключение к точке доступа. Задается пароль и название сети для подключения к ней платы ESP8622. Задаем все необходимые переменные и ссылки для передвижения по сайту.

В выполнении цикла подается не прерывный сигнал, в котором отправляются команды в виде чисел для робота.

В цикле работают несколько подпрограмм, которые отвечают за каждую команду отдельно, где есть и передача сигнала на Arduino Uno, и вывод ошибок при неправильной работе программы, и оформление сайта с встроенным HTML и CSS кода.

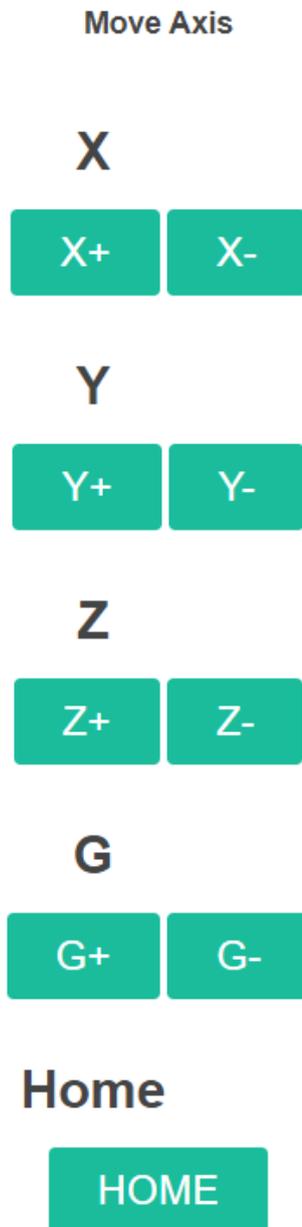


Рисунок 2.9 -интерфейс системы управления с сервера

Программа состоит из двух частей. Первая программа из управления приводами с контроллером Arduino UNO, вторая часть необходима для передачи сигналов и отправки команд с сервера на Arduino UNO, посредником является плата ESP8266.

## 3 Практическая часть

### 3.1 Проектирование электросхемы

Для сборки электросхемы используются:

1. 3 Шаговых мотора Nemo 17
2. 3 драйвера
3. Плата Arduino Uno
4. Блок питания 12 В 5 А
5. Плата ESP8266
6. Понижатель напряжения 12 – 5 В

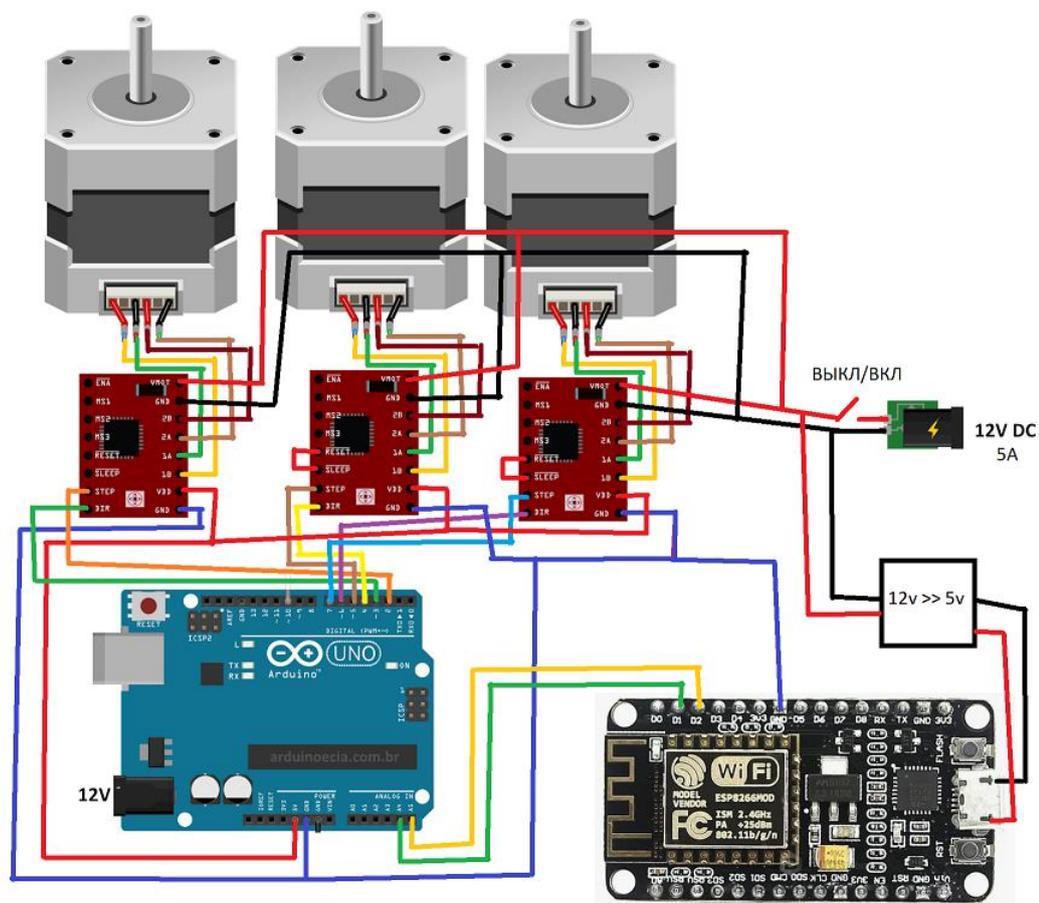


Рисунок 3.1 – Электросхема собранная мной для данного манипулятора

### 3. 2 Проектирование 3D – модели

Для моделирования и сборки манипулятора используем программу КОМПАС 3D для проектирования механических передач.

Для моделирования и сборки манипулятора используем программу Sharp 3D для проектирования корпуса.

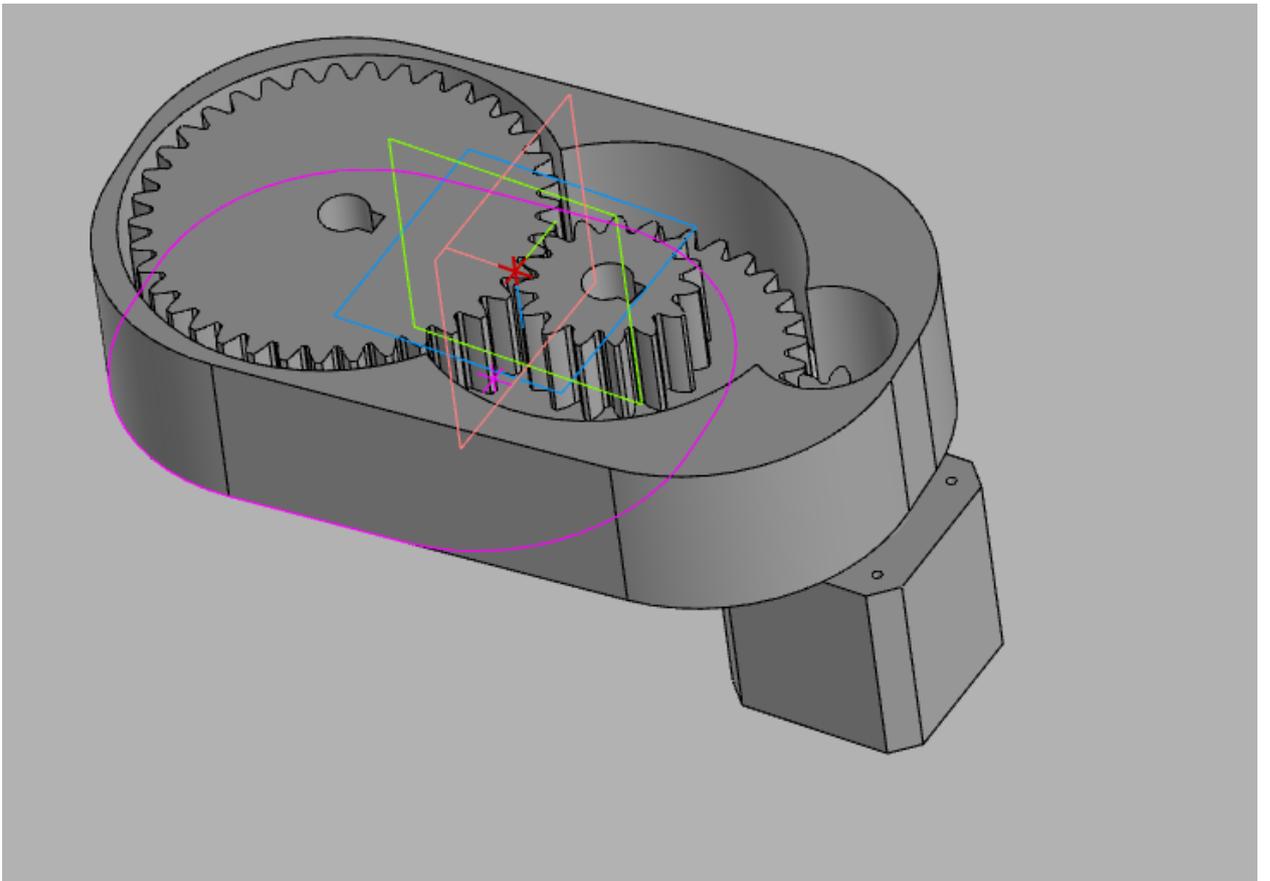


Рисунок 3.2 - Сборка редуктора в программа КОМПАС – 3D

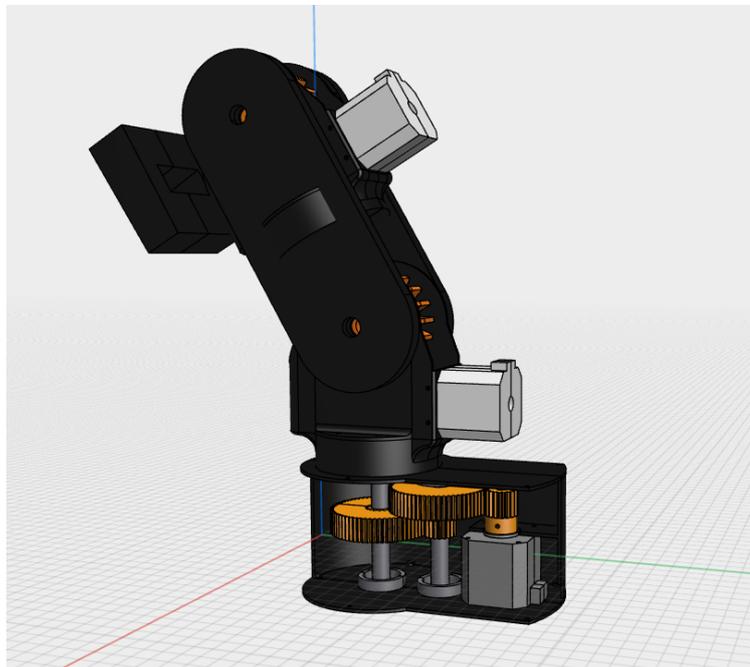


Рисунок 3.3 – полная сборка манипулятора в программа Sharp 3D

### 3.3 Расчет себестоимости

Расчет себестоимости представлен в виде таблицы, в которой имеются данные наименования, количество, цена и итоговая сумма.

№	Наименование	Количество (шт)	Цена (тг)
1	Arduino Uno	1	1 500
2	CNC Shield	1	600
3	Esp8266	1	760
4	Nemo 17	3	10 500
5	Servo	1	2 000
6	Провода		1 000
7	Вентилятор	1	2 500
8	Блок питания	1	2 500
9	Драйвера	3	2 400
10	Пластик	700 грамм	3 750
11	Подшипники	12	2 600
Итого			30 110

Таблица 3.1 – Расчет себестоимости

Процесс изготовления корпуса манипулятора. Манипулятор был спроектирован в системах САПР Sharp3d и КОМПАС 3D. Модели были подготовленные для 3D печати в программе PrusaSlicer 2.7.4. Детали были распечатаны на 3D принтере Ender 3 v2.

#### 4 Тестирование разработанного робота манипулятора

Для испытаний создадим задачу. Переместить из точки А в точку В груз массой 100 грамм с использованием системы управления разработанная для данного манипулятора.

В качестве груза используется подшипник массой 20 грамм, в количестве 5 штук.



Рисунок 4.1 – Подшипник весом примерно 20 грамм.

Выполнение операции было 3 раза без каких-либо существенных отклонений. Результаты эксперимента записаны на видео.

Вывод, робот манипулятор сможет выполнять операции перемещения груза около 100 грамм.



Рисунок 4.2 – Кадры перемещения груза из видео

Построим циклограмму по результатам экспериментов. В итоге манипулятор может стабильно работать с грузами массой 100 грамм.

	Время (сек)	5	10	15	20	25	30	35	40	45	
Перемещение к грузу	7	■									
Захват	1		■								
Подъем груза	5		■								
Перемещение из точки А в точку Б	10			■							
Опустить груз	8					■					
Разжать	1							■			
Начальное положение	10							■			
Общее время	42	■									

Таблица 4.1 – Циклограмма для задачи с перемещением груза из условной точки А в точку Б.

## 5 Заключение

В данной дипломной работе была разработана система управления робота манипулятором, включающая в себя механическую часть, программное обеспечение и алгоритмы управления.

Результаты работы:

Разработана кинематическая модель робота-манипулятора, позволяющая определить положение и ориентацию его рабочего органа в пространстве.

Разработана система управления движением робота, основанная на методах обратной кинематики, позволяющая задавать траекторию движения робота с использованием различных алгоритмов планирования.

Реализована система управления роботом, включающая в себя интерфейс пользователя, программное обеспечение для управления двигателями и исполнительными механизмами.

Разработанная система управления роботом-манипулятором позволяет автоматизировать выполнение различных задач, требующих точных движений и манипуляций.

Программное обеспечение системы управления разработано с использованием современных технологий, что обеспечивает ее высокую надежность и производительность.

Перспективы дальнейших исследований:

Разработка алгоритмов адаптивного управления, позволяющих роботу-манипулятору адаптироваться к изменениям в окружающей среде.

Внедрение систем машинного обучения для повышения точности и эффективности работы робота манипулятора.

В заключение, разработанная система управления роботом-манипулятором является ценным инструментом для автоматизации различных задач и обладает высоким потенциалом для дальнейшего развития

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. П. Ф. Дунаев, О. П. Леликов, Конструирование узлов и механизмов
2. И. Н. ЕГОРОВ В. П. УМНОВ СИСТЕМЫ УПРАВЛЕНИЯ ЭЛЕКТРОПРИВОДОВ ТЕХНОЛОГИЧЕСКИХ РОБОТОВ И МАНИПУЛЯТОРОВ
3. Торхова Е.К. Вахрушев А.С. Кинематические схемы
4. Волжанова О.А. В673 Схемы кинематические принципиальные: учебно-метод. пособие / Ижевск: Изд-во «Удмуртский университет», 2012. 34 с.
5. Б. А. БЕЛЯЕВ А. П. ШЕВЧЕНКО ТЕОРИЯ МЕХАНИЗМОВ И МАШИН
6. Прямая и обратная задача кинематики <https://robocraft.ru/mechanics/756>
7. <https://astanaceramic.com/about>

### Приложение А

```
#include <Wire.h>
#include <AccelStepper.h> // Подключаем библиотеку AccelStepper

int signal1;
bool flag1 = false;

#define dirPinX 5
#define stepPinX 2
#define dirPinY 6
#define stepPinY 3
#define dirPinZ 7
#define stepPinZ 4
#define motorInterfaceType1 1 // Определение тип интерфейса двигателя

// Создаем экземпляр
AccelStepper myStepperX(motorInterfaceType1, stepPinX, dirPinX);
AccelStepper myStepperY(motorInterfaceType1, stepPinY, dirPinY);
AccelStepper myStepperZ(motorInterfaceType1, stepPinZ, dirPinZ);

void receiveEvent1()
{
  while (Wire.available())
  {
    signal1 = Wire.read();
    Serial.println(signal1);
  }
}
```

```

}

// Функция для извлечения любых отправляемых данных от мастера на шину
// void requestEvent()
// {
//   if (flag1 == true)
//   {
//     flag1 = false;
//     Wire.write(1);
//     Serial.println("STOP1");
//   }
// }

void setup()
{
  Wire.begin(8);          /* задаем на шине i2c 8 адрес */
  Wire.onReceive(receiveEvent1); /* регистрируем полученное событие */
  // Wire.onRequest(requestEvent); /* регистрируем запрошенное событие */

  Serial.begin(9600);    /* открываем серийный порт для дебаггинга */

  // Параметры для X
  myStepperX.setMaxSpeed(1000);
  myStepperX.setAcceleration(50);
  myStepperX.setSpeed(200);
  myStepperX.moveTo(100);

  // Параметры для Y
  myStepperY.setMaxSpeed(1000);
  myStepperY.setAcceleration(50);
  myStepperY.setSpeed(200);
  myStepperY.moveTo(100);

  // Параметры для Z
  myStepperZ.setMaxSpeed(1000);
  myStepperZ.setAcceleration(50);
  myStepperZ.setSpeed(200);
  myStepperZ.moveTo(100);
}

void loop()
{
  // вращение по часовой стрелке X

```

```
if (signal1==1)
{
  // Serial.println(signal1);
  if (myStepperX.distanceToGo() == 0)
  {
    myStepperX.move(400);
    flag1 = true;
  }
  myStepperX.run(); // Передвинуть на 1 шаг
}
// вращение по против часовой стрелке X
if (signal1==2)
{
  // Serial.println(signal1);
  if (myStepperX.distanceToGo() == 0)
  {
    myStepperX.move(-400);
    flag1 = true;
  }
  myStepperX.run(); // Передвинуть на 1 шаг
}

// вращение по часовой стрелке Y
if (signal1==3)
{
  // Serial.println(signal1);
  if (myStepperY.distanceToGo() == 0)
  {
    myStepperY.move(400);
    flag1 = true;
  }
  myStepperY.run(); // Передвинуть на 1 шаг
}

// вращение по против часовой стрелке Y
if (signal1==4)
{
  // Serial.println(signal1);
  if (myStepperY.distanceToGo() == 0)
  {
    myStepperY.move(-400);
    flag1 = true;
  }
}
```

```
myStepperY.run(); // Передвинуть на 1 шаг
}

// вращение по часовой стрелке Z
if (signal1==5)
{
  // Serial.println(signal1);
  if (myStepperZ.distanceToGo() == 0)
  {
    myStepperZ.move(400);
    flag1 = true;
  }
  myStepperZ.run(); // Передвинуть на 1 шаг
}

// вращение по против часовой стрелке Z
if (signal1==6)
{
  // Serial.println(signal1);
  if (myStepperZ.distanceToGo() == 0)
  {
    myStepperZ.move(-400);
    flag1 = true;
  }
  myStepperZ.run(); // Передвинуть на 1 шаг
}

//Занулить все сигналы на моторы по X, Y, Z
if (signal1==0)
{
  // Serial.println(signal1);
  if (myStepperX.distanceToGo() == 0)
  {
    myStepperX.move(0);
    signal1 = 0;
    flag1 = true;
  }
  myStepperX.run(); // Передвинуть на 1 шаг
}
if (signal1==0)
{
  // Serial.println(signal1);
```

```

if (myStepperY.distanceToGo() == 0)
{
  myStepperY.move(0);
  signal1 = 0;
  flag1 = true;
}
myStepperY.run(); // Передвинуть на 1 шаг
}
if (signal1==0)
{
  // Serial.println(signal1);
  if (myStepperZ.distanceToGo() == 0)
  {
    myStepperZ.move(0);
    signal1 = 0;
    flag1 = true;
  }
  myStepperZ.run(); // Передвинуть на 1 шаг
}
}
}

```

```

#include <Wire.h>
#include <ESP8266WebServer.h>

int button;
/* Установите здесь свои SSID и пароль */
const char* ssid = "Redmi Note 12"; // SSID
const char* password = "jk4yenfr8d59agd"; // пароль

ESP8266WebServer server(80);

bool statusButton1 = false;
bool statusButton2 = false;

String command = "";

void setup()
{
  Serial.begin(9600);
  delay(100);

  Wire.begin(4, 5);

```

```

Serial.println("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
{
    delay(1000);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected..!");
Serial.print("Got IP: ");
Serial.println(WiFi.localIP());

server.on("/", handle_OnConnect);
server.on("/buttonX1", handle_buttonX1);
server.on("/buttonX2", handle_buttonX2);
server.on("/buttonY1", handle_buttonY1);
server.on("/buttonY2", handle_buttonY2);
server.on("/buttonZ1", handle_buttonZ1);
server.on("/buttonZ2", handle_buttonZ2);
server.onNotFound(handle_NotFound);
server.begin();
Serial.println("HTTP server started");
}

void loop()
{
    server.handleClient();
}

void signal_to_Arduino_Button1()
{
    Wire.beginTransmission(8); /* Начинаем передачу на адресе 8 */
    Wire.write(1); /* Отправляем "On" */
    Wire.endTransmission(); /* прекращаем передачу */
}

void signal_to_Arduino_Button2()
{
    Wire.beginTransmission(8); /* Начинаем передачу на адресе 8 */
    Wire.write(2); /* Отправляем "On" */
    Wire.endTransmission(); /* прекращаем передачу */
}

void signal_to_Arduino_Button3()

```

```

{
  Wire.beginTransmission(8); /* Начинаем передачу на адресе 8 */
  Wire.write(3); /* Отправляем "On" */
  Wire.endTransmission(); /* прекращаем передачу */
}
void signal_to_Arduino_Button4()
{
  Wire.beginTransmission(8); /* Начинаем передачу на адресе 8 */
  Wire.write(4); /* Отправляем "On" */
  Wire.endTransmission(); /* прекращаем передачу */
}
void signal_to_Arduino_Button0()
{
  Wire.beginTransmission(8); /* Начинаем передачу на адресе 8 */
  Wire.write(0); /* Отправляем "On" */
  Wire.endTransmission(); /* прекращаем передачу */
}
void signal_to_Arduino_Button5()
{
  Wire.beginTransmission(8); /* Начинаем передачу на адресе 8 */
  Wire.write(5); /* Отправляем "On" */
  Wire.endTransmission(); /* прекращаем передачу */
}
void signal_to_Arduino_Button6()
{
  Wire.beginTransmission(8); /* Начинаем передачу на адресе 8 */
  Wire.write(6); /* Отправляем "On" */
  Wire.endTransmission(); /* прекращаем передачу */
}

void handle_OnConnect()
{
  server.send(200, "text/html", SendHTML(true));
}

// Отправляем сигнал включения кнопки1 по X
void handle_buttonX1()
{
  signal_to_Arduino_Button1();
  Serial.println("ON_1");
  server.send(200, "text/html", SendHTML(true));
  signal_to_Arduino_Button0();
}

```

```
//Отправляем сигнал выключения кнопки2 по X
void handle_buttonX2()
{
    signal_to_Arduino_Button2();
    Serial.println("ON_2");
    server.send(200, "text/html", SendHTML(true));
    signal_to_Arduino_Button0();
}

//Отправляем сигнал выключения кнопки3 по Y
void handle_buttonY1()
{
    signal_to_Arduino_Button3();
    Serial.println("ON_1");
    server.send(200, "text/html", SendHTML(true));
    signal_to_Arduino_Button0();
}

//Отправляем сигнал выключения кнопки4 по Y
void handle_buttonY2()
{
    signal_to_Arduino_Button4();
    Serial.println("ON_1");
    server.send(200, "text/html", SendHTML(true));
    signal_to_Arduino_Button0();
}

//Отправляем сигнал выключения кнопки5 по Z
void handle_buttonZ1()
{
    signal_to_Arduino_Button5();
    Serial.println("ON_1");
    server.send(200, "text/html", SendHTML(true));
    signal_to_Arduino_Button0();
}

//Отправляем сигнал выключения кнопки6 по Z
void handle_buttonZ2()
{
    signal_to_Arduino_Button6();
    Serial.println("ON_1");
    server.send(200, "text/html", SendHTML(true));
    signal_to_Arduino_Button0();
}
```

```

}

void handle_NotFound()
{
    server.send(404, "text/plain", "Not found");
}

String SendHTML(bool status1)
{
    String ptr = "<!DOCTYPE html> <html>\n";
    ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";

    ptr += "<title>Move Axis</title>\n";

    ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center; }\n";
    ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px; padding-right:80px;} h3 {color: #444444;margin: 50px auto 30px;}\n";

    ptr += ".button {width: 80px; background-color: #1abc9c; color: white;padding: 13px 30px; text-decoration: none; font-size: 25px; cursor: pointer;border-radius: 4px;}\n";
    // ptr += ".button {display: block;width: 80px;background-color: #1abc9c;border: none;color: white;padding: 13px 30px;text-decoration: none;font-size: 25px;margin: 0px auto 35px;cursor: pointer;border-radius: 4px;}\n";

    ptr += ".button-on {background-color: #1abc9c;}\n";
    ptr += ".button-on:active {background-color: #16a085;}\n";

    ptr += ".button-off {background-color: #34495e;}\n";
    ptr += ".button-off:active {background-color: #2c3e50;}\n";

    ptr += "p {font-size: 14px;color: #888;margin-bottom: 10px;}\n";
    ptr += "</style>\n";
    ptr += "</head>\n";

    ptr += "<body>\n";

    ptr += "<h3>Move Axis</h3>\n";

    if(status1)
    {
        ptr += "<h1>X</h1>\n";
    }
}

```

```
ptr += "<a class=\"button button-on\" href=\"/buttonX1\">X+</a>\n";
ptr += "<a class=\"button button-on\" href=\"/buttonX2\">X-</a>\n";
ptr += "<h1>Y</h1>\n";
ptr += "<a class=\"button button-on\" href=\"/buttonY1\">Y+</a>\n";
ptr += "<a class=\"button button-on\" href=\"/buttonY2\">Y-</a>\n";
ptr += "<h1>Z</h1>\n";
ptr += "<a class=\"button button-on\" href=\"/buttonZ1\">Z+</a>\n";
ptr += "<a class=\"button button-on\" href=\"/buttonZ2\">Z-</a>\n";
}
ptr += "</body>\n";
ptr += "</html>\n";
return ptr;
}
```

Код программы