

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

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



**SATBAYEV  
UNIVERSITY**

Институт автоматки и информационных технологий

Кафедра «Робототехники и технических средств автоматки»

Специальность 6В07111– Робототехника и мехатроника

Тулешов Елнур Ержанұлы

«Разработка исполнительного механизма для портативного устройства с обширной системой  
анализа данных»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

Специальность 6В07111– Робототехника и мехатроника

Алматы 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

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



SATBAYEV  
UNIVERSITY

Институт автоматки и информационных технологий

Кафедра «Робототехники и технических средств автоматки»



**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

«Разработка исполнительного механизма для портативного устройства с обширной  
системой анализа данных»

по специальности 6В07111– Робототехника и мехатроника

Выполнил

Тулешов Елнур Ержанұлы

Рецензент

д.т.н., асоц. профессор ИМиМ

Научный руководитель

канд. техн. наук, асоц. профессор

Балбаев Г.К.

Тулешов Е.А.

«30» май 2023 г.

«30» май 2023 г.

Алматы 2023

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



SATBAYEV  
UNIVERSITY

Институт автоматизации и информационных технологий

Кафедра «Робототехники и технических средств автоматизации»

6B07111 – Робототехника и мехатроника



УТВЕРЖДАЮ

Заведующий кафедрой РТ и ТСА

канд. техн. наук

К. А. Ожикенов

«30» май 2023 г.

### ЗАДАНИЕ

на выполнение дипломной работы

Обучающемуся Тулешов Елнұр Ержанұлы

Тема: Разработка исполнительного механизма для портативного устройства с обширной системой анализа данных

Утверждена приказом Ректора Университета № 408/П/р от «23» 11 2023г.

Срок сдачи законченной работы «30» май 2023 г.

Исходные данные к дипломному проекту: Arduino Uno

Перечень подлежащих разработке вопросов в дипломном проекте:

- а) изучение аналогов манипуляторов
- б) выявить плюсы и минусы
- в) разработка оптимального манипулятора
- г) разработать макет робота-манипулятора

Перечень графического материала (с точным указанием обязательных чертежей):

*Представлены 11 слайдов презентации работы*

Рекомендуемая основная литература: из 20 наименований 9

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

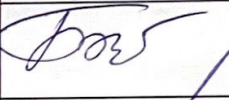

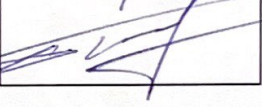
**ГРАФИК**

подготовки дипломной работы (проекта)

Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
Технологическая часть	13.03 – 03.04.2023 г.	Выполнено
Практическая часть	20.03-30.04.2023 г.	Выполнено

**Подписи**

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

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Нормоконтролер	Ж.С. Бигалиева, магистр технических наук, старший преподаватель	30.05.23	
Основная часть	Е.А. Тулешов, кандидат технических наук, ассоциированный профессор	30.05.23	
Расчетная часть	Е.А. Тулешов, кандидат технических наук, ассоциированный профессор	30.05.23	

Научный руководитель

 Тулешов Е.А.

Задание принял к исполнению обучающийся

 Тулешов Е.Е.

Дата

" 30 " май 2023г.

## **АҢДАТПА**

Дипломдық жобаның мақсаты – түрлі командалар орындайтын манипулятор жасау. Дипломдық жобада үлгі ретінде, манипуляторлардың өзге түрлері келтірілген. Жақсы және жаман жақтары қарастырылған. Сонымен қатар, принциптары мен болашақтағы мүмкіндіктері.

Бірінші бөлімде, роботтардың қандай түрлері болатындығы, не үшін, қалай және қандай мақсаттармен қолданылатындығы жазылған.

Екінші бөлімде, робот-манипуляторлар тереңірек қарастырылған, кинематикалық анализ бен математикалық есептеулер жазылған.

Практикалық бөлімде жалпы жұмыстың орындалуында қандай құрылғылардың қолданылғаны жазылған.

## **АННОТАЦИЯ**

Целью дипломного проекта является создание манипулятора для выполнения различных команд. В дипломном проекте, в качестве примера, были приведены другие виды роботов-манипуляторов. Рассматривались преимущества и слабые стороны устройства. Также принципы и дальнейшие возможности.

В первой части идет пояснение, какие роботы бывают, зачем они нужны, как использовать и для какой цели.

Во второй части, более углубленно рассматриваем роботов-манипуляторов, кинематический анализ и математические расчеты.

В практической части указано какие приложения были использованы при выполнении общей работы.

## **ANNOTATION**

The goal of my diploma project is to create a manipulator for execution of various commands. As an example, other types of robotic manipulators were given. Also, considered advantages and disadvantages of the device. Principles and further possibilities too.

The first part is about an explanation of kinds of robots, why they are needed, how to use them and for what purposes.

In the practical part, it is indicated which applications were used when performing general work.

## СОДЕРЖАНИЕ

Введение	7
1 Общие понятия о роботах	8
1.1 История роботов	8
1.2 Типы роботов	8
1.3 Роботы манипуляторы	11
2 Разработка манипулятора	15
2.1 Создание манипулятора	15
2.2 Архитектура тела	15
2.3 Кинематика	15
2.4 Архитектура управления	16
2.5 Автономия	16
2.6 Создание корпуса	16
2.7 Особенности при работе	17
2.8 Стоимость конструкции	17
3 Описание используемых технологии	18
3.1 Платы Arduino UNO	18
3.1.1 Среда разработки Arduino IDE	18
3.2 Сервоприводы MG996R и SG90	18
3.3 Аккумулятор MJXRC X600 2S	19
4 Практическая часть	20
4.1 Разработка приложения	20
4.2 Конструкторская часть	20
Заключение	27
Список использованной литературы	28
Приложение А	
Приложение Б	

## ВВЕДЕНИЕ

На сегодняшний день, роботы все больше и больше начинают пользоваться спросом не только в производстве, но даже и в быту. С развитием технологии, развиваются и роботы, они становятся более комплексные и выполняют больше команд. Роботы также используются в военных и медицинских целях. Когда рушится здание, происходит пожар или заброшенный дом, который стоит несколько лет, то для человека, заходить в такие места может быть опасным. Для таких случаев можно создать портативного робота с манипулятором, которого можно управлять и совершать разного рода операции, проверять на исправность, опасность ту или иную вещь.

Манипуляторы – устройства, которые предназначены для того чтобы управлять механическими системами или роботами. На сегодняшнее время, манипуляторы используются в самых разных областях, где требуется точность и скорость перемещения тех или иных объектов и где требуется выполнение трудоемких, физических операций.

Производство. В этой отрасли манипуляторы применяются на производственных линиях для таких операций, как перемещение и обработки товаров, манипулирования инструментами, частями оборудования, сборки и упаковки.

Медицина. В медицине, где требуется максимальная точность и где ошибка обходится очень дорогой ценой, манипуляторы нашли свое применение и комфортно себя чувствуют. Выполнение хирургических операций, диагностика и лечение пациентов, работа с радиоактивными и другими опасными для здоровья веществами, где контакт человека может привести к плохим последствиям.

Пространство. Здесь манипуляторы применяются для того чтобы управлять космические аппараты, разные роботы, выполнять нужные фиксированные задачи на орбите и на поверхности других планет, как Марс

Логистика. Отрасль, где много физической работы по типу загрузки и разгрузки грузовых машин, перемещения и складывания грузов в складах, помещениях и на производственных площадках, также для автономного управления транспортными средствами.

Обслуживание. Манипуляторы здесь используются для управления разными устройствами, например, для обслуживания технического оборудования, ремонта и обслуживания инфраструктуры.

Манипуляторы сильно упрощают выполнение сложных задач, экономят время выполнения и минимизируют вероятность ошибок, что позволяет работать более эффективно и качественно.

Целью моей дипломной работы является разработка манипулятора для комплексного робота, создание 3D модели, распечатка, разработка программного обеспечения. Задача – изучить аналоги манипуляторов, более тонко понять плюсы и минусы того или иного вида, подвести итоги и взять самое лучшее и оптимальное и создать свой манипулятор.

# 1 ОБЩИЕ ПОНЯТИЯ О РОБОТАХ

## 1.1 История роботов

История робототехники сильно связана с огромным количеством изобретений, сделанным за все время. Его невозможно отделить от общей истории развития технологий.

Обыватели знают, что роботы, используемые в качестве протезов, уже помогает людям с ограниченными возможностями, роботы машины проникают в труднодоступные или жизнеопасные места чтобы человеческие жизни получали наименьший вред, а в лучшем случае без вреда.

Робот-манипулятор играет основную роль в командном проекте. Поскольку, может выполнять разного рода действия.

Целью данной дипломной работы является разработка исполнительного механизма для портативного устройства с обширной системой анализа данных.

## 1.2 Типы роботов

За последние десятилетия, технология сильно перешагнула вперед, это касается роботов тоже. На сегодняшний день существует огромное количество роботов для разных функций, со своими предназначениями на любой вкус.

Несмотря на такой выбор, всех роботов объединяет одно. То – что каждый робот выполняет то что прописано в программной части.

В введении мы рассмотрели сферы где применяются роботы, а сейчас более углубленно рассмотрим механическую часть роботов, как они двигаются и их типы.

**Колесные.** Скопированный от машин метод передвижения. Простой в создании и эффективный в применении. Характеристика каждого робота зависит от количества установленных колёс. Меньшее количество колёс означает большую маневренность. А большее количество колёс означает устойчивость робота, улучшать проходимость.



Рисунок 1.2.1 – колесный робот



**Гусеничные роботы.** Популярный метод среди военной техники. Популярность именно этого типа легко объясняется тем, что ножки помогают машине без особой сложности передвигаться по пересеченной местности.



Рисунок 1.2.2 – Гусеничный инспекционный робот

**Шагающие.** Имитирующие людей роботы. Данный вид является одним из сложных в создании. Поскольку ту самую золотую середину в устойчивости очень сложно найти, зачастую невозможно.



Рисунок 1.2.3 – Шагающий робот для спасательных операций

**Летающие.** Наверняка, вы несколько раз видели летающих дронов во время разных мероприятий, концертов. Управляются дистанционно, требуют

нужной мощности, без которой манипулировать таким роботом будем попросту невозможно.



Рисунок 1.2.4 - Дрон

**Плавающие.** Роботы, которые пользуются силами извне. Передвигаются с помощью силы ветра или других факторов. Могут быть использованы как на суше, так и под водой.



Рисунок 1.2.5 – Плавающий робот

### 1.3 Роботы манипуляторы

Поскольку тема моей дипломной работы напрямую относится к манипуляторам, более углубленно рассмотрим эти роботы и их аналоги.

Манипуляторы позволяют выполнять разного рода действий, если большинство обычных роботов ограничены в своем функционале, то роботы манипуляторы могут похвастаться приличным арсеналом.

Манипуляторы бывают разных типов и назначений.

Промышленные. Перемещение грузов на производстве.



Рисунок 1.3.1 – Промышленные роботы манипуляторы

Хирургические манипуляторы для проведения операций на расстоянии. В медицине точность манипуляторов высоко ценится, поскольку не бывает человеческого фактора и оплошностей, соответственно и процесс операций происходит эффективнее и безопаснее.



Рисунок 1.3.2 – Операция робота-хирурга

Космические манипуляторы для монтажа и обслуживания космических объектов.

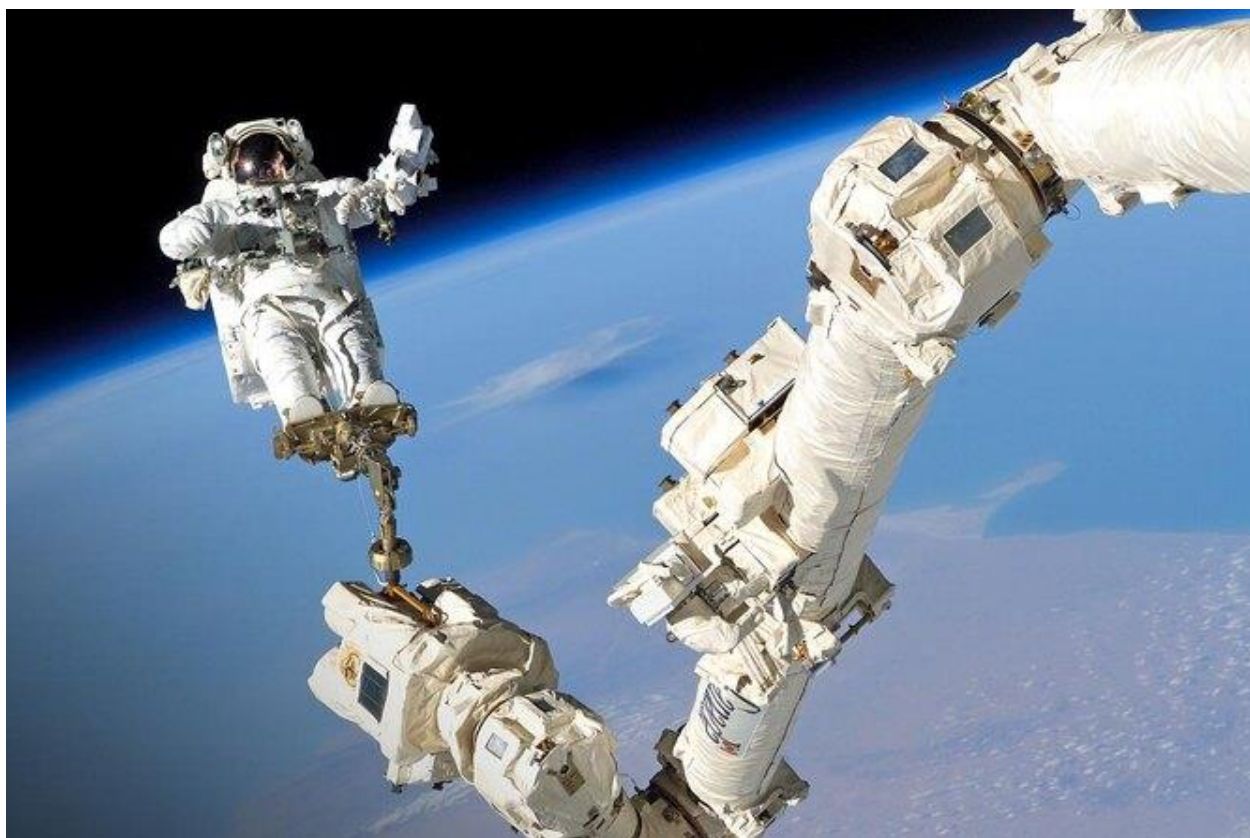


Рисунок 1.3.3 – Космический манипулятор

Манипуляторы для управления судном или самолетом.



Рисунок 1.3.4 – Кран-манипулятор от австрийской компании Palfinger

Гидравлические, механические или электрические манипуляторы для уборки улиц, ухода за растениями и т. д.



Рисунок 1.3.5 – Подметальная машина

Выше были приведены общие виды манипуляторов. Теперь более углубленно рассмотрим конкретные аналоги нашего манипулятора.

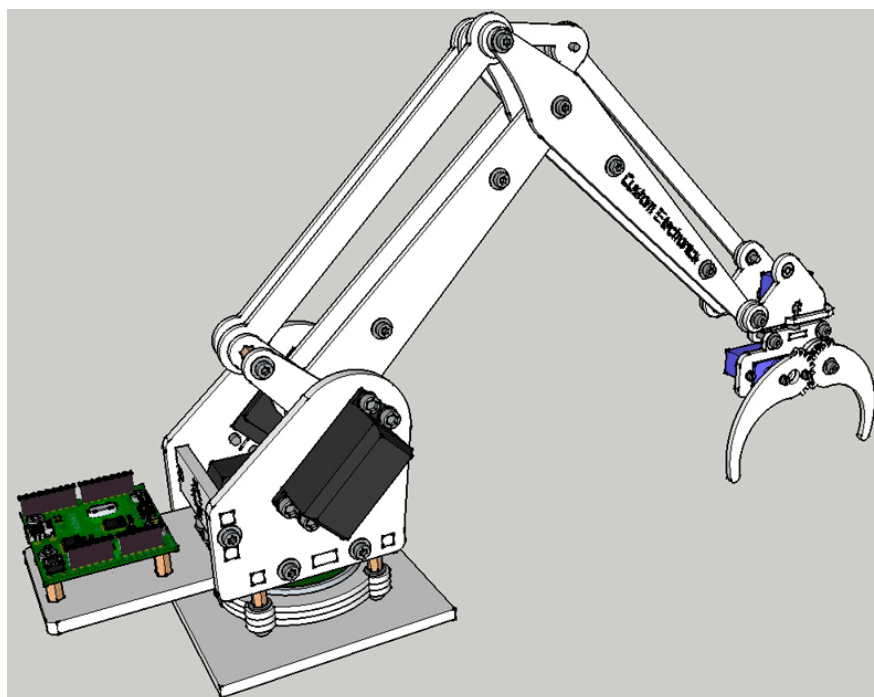


Рисунок 1.3.6 - Настольная робо-рука манипулятор из оргстекла на сервоприводах

Этот аналог имеет ряд преимуществ. Он является крепким манипулятором за счет вспомогательной помощи ребру манипулятора. А минусом является расположение ардуино. Он не удобен и занимает больше пространства.

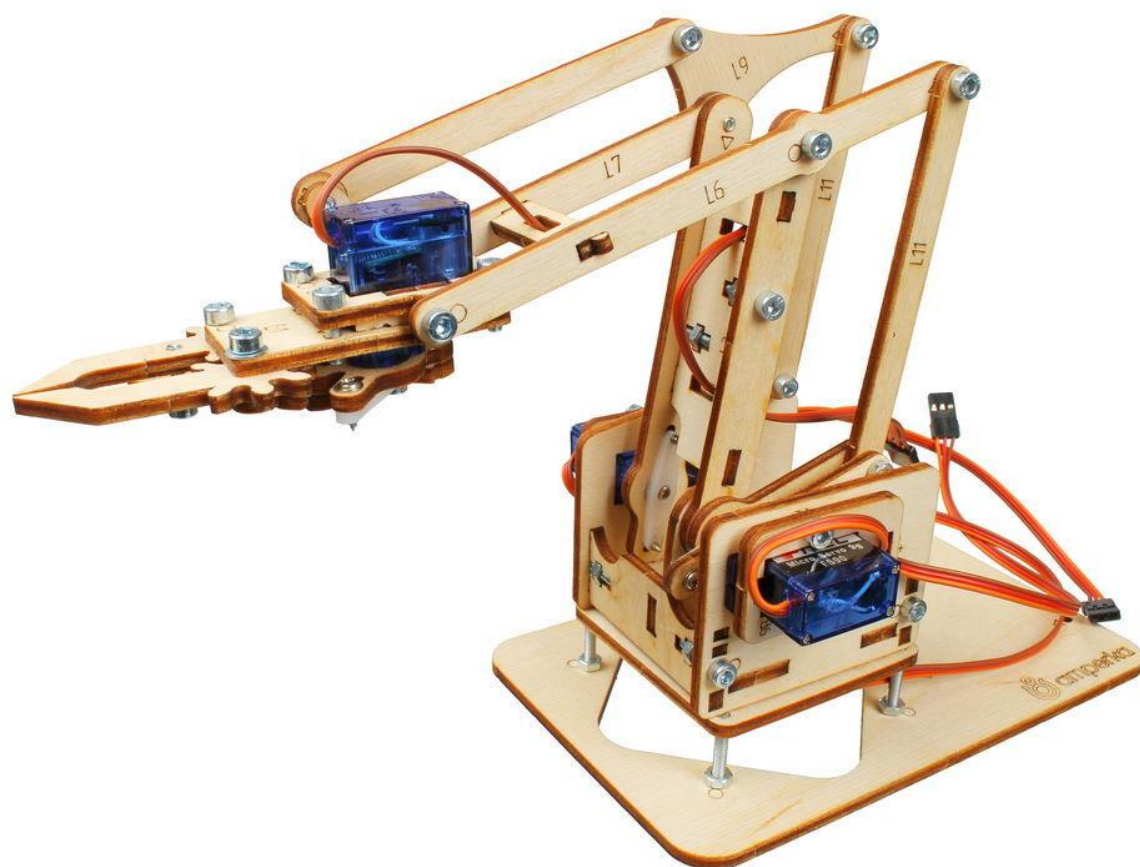


Рисунок 1.3.7 – Рука-манипулятор XIII

## **2 РАЗРАБОТКА МАНИПУЛЯТОРА**

### **2.1 Создание манипулятора**

Роботы-манипуляторы в наше время играют весомую роль благодаря их простоте и эффективности. Есть ряд характеристик, требованиям которых должен соответствовать мой манипулятор, чтобы он был максимально оптимален и эффективен.

- Архитектура тела
- Кинематика
- Архитектура управления
- Автономия
- Создание корпуса
- Особенности при работе
- Стоимость конструкции

### **2.2 Архитектура тела**

Манипулятор состоит из нескольких составляющих компонентов:

- Стойка или рама: это основание манипулятора, которое держит все остальные компоненты и определяет его геометрическую форму.
- Соединительные элементы: это механизмы, которые связывают основание манипулятора с плечами, локтевым и запястным соединениями.
- Плечи: эти элементы соединяются с основанием манипулятора и позволяют манипулятору двигаться в горизонтальной плоскости вверх и вниз.
- Локтевые соединения: эти соединения позволяют манипулятору сгибаться и разгибаться вверх и вниз.
- Запястные соединения: это соединения, которые позволяют манипулятору вращаться вокруг своей оси и управлять ориентацией захвата.
- Захват или эффектор: это элемент, который в конечном итоге выполняет задачу манипулятора, например, подъем или перемещение объекта.

Помимо этих основных составляющих, к моему манипулятору также прикреплены датчики, задача которых является работой другого студента из моей группы в выполнении нашей командной работы.

### **2.3 Кинематика**

Кинематика манипулятора руки с 6 сервоприводами включает в себя определение положения и ориентации наконечника манипулятора относительно базового звена, используя значения углов поворота каждого из шести сервоприводов. Для решения этой задачи используется обратная кинематическая задача (*inverse kinematics problem*).

В общем случае, положение и ориентация наконечника могут быть определены с помощью трех координат и трех углов Эйлера. Для решения

обратной кинематической задачи в таком случае используются методы численного решения систем уравнений. Однако, для частных случаев (например, для манипуляторов со сферической кинематикой), существуют аналитические решения данной задачи.

При проектировании манипулятора руки с 6 сервоприводами, необходимо учитывать механические ограничения поворотов каждого из звеньев. Также важно обеспечить возможность регулировки углов поворота сервоприводов для достижения нужного положения конечника.

Кинематическое уравнение манипулятора руки с 6 сервоприводами может быть записано в форме матрицы преобразования, которая описывает положение и ориентацию манипулятора в трехмерном пространстве.

Для шестизвенной манипулятора руки с 6 сервоприводами кинематическое уравнение может быть представлено следующим образом:

$$T = T1 * T2 * T3 * T4 * T5 * T6$$

Где T1, T2, T3, T4, T5 и T6 являются матрицами преобразования, которые моделируют изменение положения и ориентации каждого звена манипулятора. Каждая матрица состоит из элементов, описывающих перемещение и поворот в каждой из трех осей координат.

Кинематическое уравнение позволяет вычислять координаты конечной точки (End Effector) манипулятора в зависимости от положения и ориентации его звеньев. Это позволяет контролировать движения манипулятора и его точность при выполнении задач.

Для управления манипулятором руки с 6 сервоприводами обычно используются алгоритмы обратной кинематики, которые позволяют определить значения углов поворота каждого сервопривода, необходимые для достижения заданного положения и ориентации конечной точки манипулятора.

## **2.4 Архитектура управления**

К манипулятору дистанционно подключается телефон на операционной системе Android. Управление происходит на базе приложения под названием “MIT App Inventor”.

## **2.5 Автономия**

Манипулятор будет подключен к аккумулятору и будем питаться от него. На каждом сервоприводе соединены два провода. Они подключаются соответственно, к минусу и плюсу.

## **2.6 Создание корпуса**

Как указано в приложении А, макет робота был разработан на Solid Works, после, были трансформированы в STL файлы и отправлены на 3D печать.



## **2.7 Особенности при работе**

Были использованы три разных сервопривода в виду их расположения. Были выбраны разные размеры и разной мощности сервоприводы.

## **2.8 Стоимость конструкции**

Общая стоимость конструкции составила около 50 000 тг.

### 3 Описание используемых технологии

Были использованы сервоприводы, аккумулятор, ардуино для реализации проекта, сейчас подробнее рассмотрим все элементы.

#### 3.1 Платы Arduino Uno

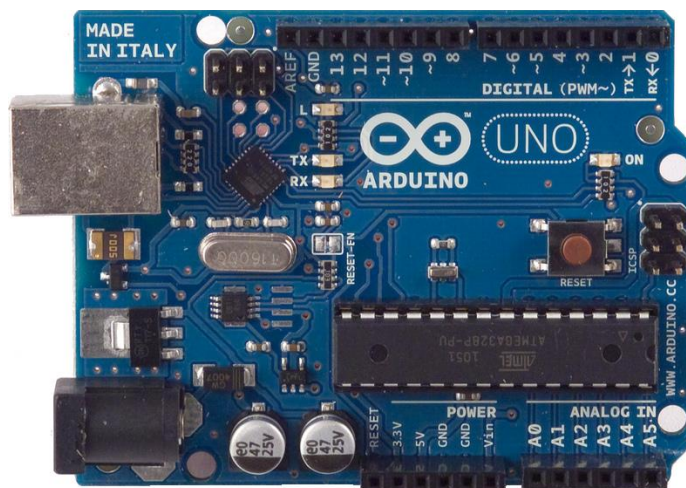


Рисунок 3.1.1 – Arduino UNO

##### 3.1.1 Среда разработки Arduino IDE

Это интегрированная среда разработки для разных операционных систем как Windows, MacOS, Linux, разработанная на языке C++ для создания программ на Arduino

#### 3.2 Сервоприводы MG996R и SG90



Рисунок 3.2.1 – Сервопривод MG996R

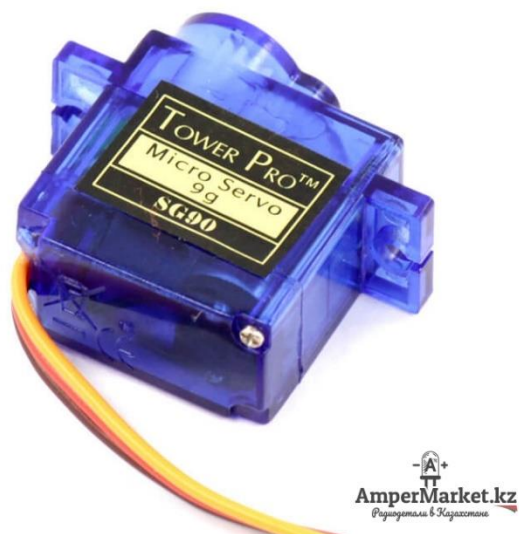


Рисунок 3.2.2 – Сервопривод SG90

### 3.3 Аккумулятор MJXRC X600 2S



Рисунок 3.3.1 – Аккумулятор MJXRC X600 2S

## 4 ПРАКТИЧЕСКАЯ ЧАСТЬ

### 4.1 Разработка приложения

Робот управляется с помощью андроид приложения на мобильном телефоне, интерфейс которого выглядит следующим образом. Приложение было разработано с помощью онлайн сайта “Mit App Inventor”



### 4.2 Конструкторская часть

После разработки макета на SolidWorks, я преобразовал данные в STL-файлы, чтобы отправить на 3D печать. По итогу я получил мой макет манипулятора по частям и уже с нужными отверстиями, где буду устанавливать сервоприводы, вставлять винтики с гайками и прочее.

Все детали манипуляторы приведены на рисунке:



Рисунок 4.2.1

После, я нарезал грани и все лишнее с деталей, чтобы в дальнейшем задействовать к самой конструкции манипулятора.



Рисунок 4.2.2

Приступил к установке первого сервопривода к базе манипулятора.



Рисунок 4.2.3

После, установил станцию, где в последствии закрепил ребро манипулятора. Также, небольшой прямоугольник слева служит для крепления резинок. С помощью резинок ребро будет тянуться в изначальную форму, что даёт больше эффективности и с меньшей силой сделать больше работы, чтобы падало меньше нагрузки относительно начальной. База крутится горизонтально.

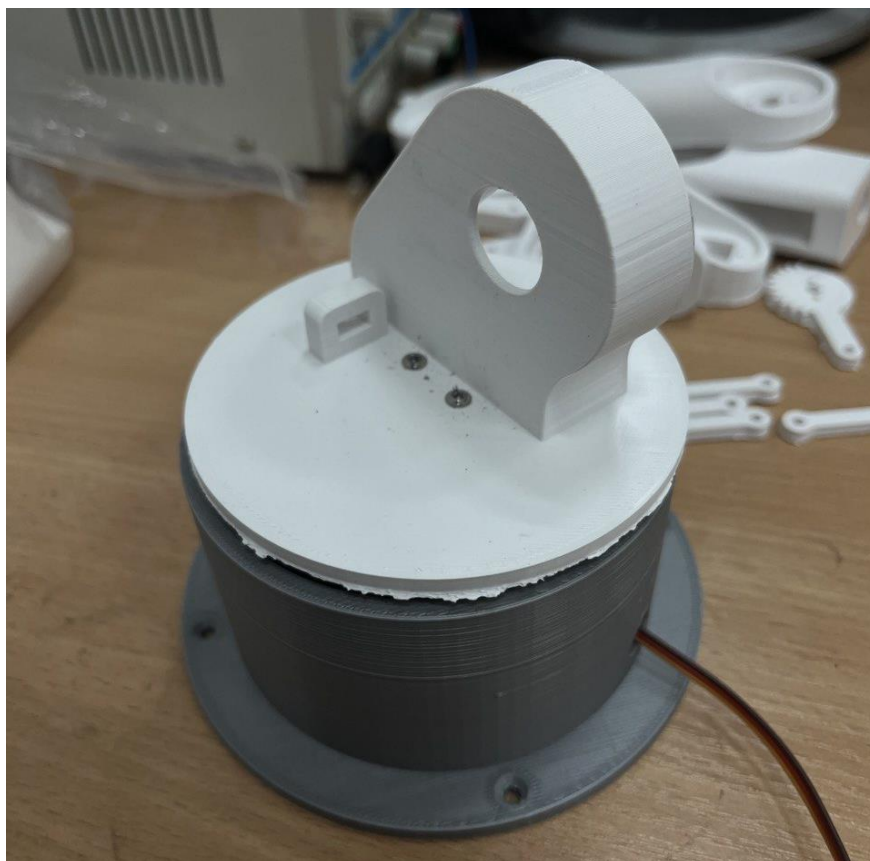


Рисунок 4.2.4

Установил то самое ребро манипулятора. Если база крутится горизонтально, то ребро крутится вертикально, также в круговое движение.



Рисунок 4.2.5

После, к основному ребру устанавливаем вспомогательное второе, на котором будем крепиться основа для клещи, то есть для рук.



Рисунок 4.2.6



На второе ребро устанавливаем основу и поверхность для клещи.

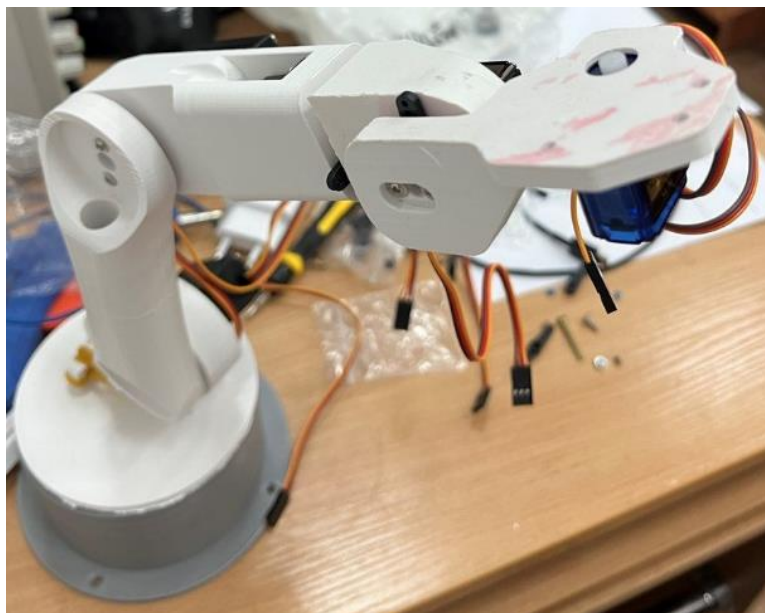


Рисунок 4.2.7

Над плоскостью для клещи устанавливаем сервопривод.

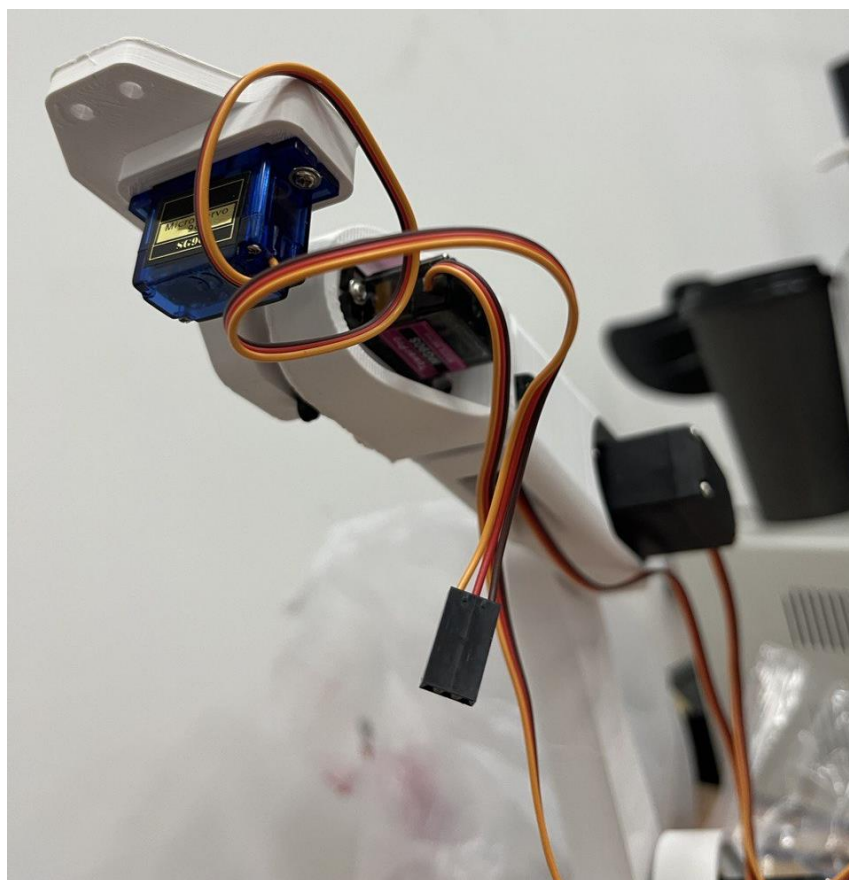


Рисунок 4.2.8

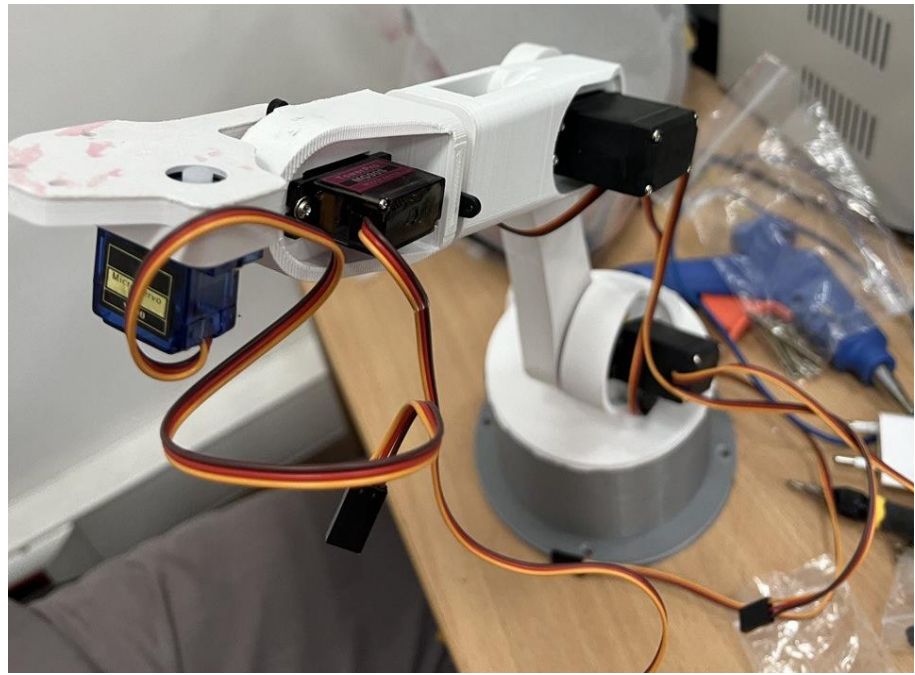


Рисунок 4.2.9

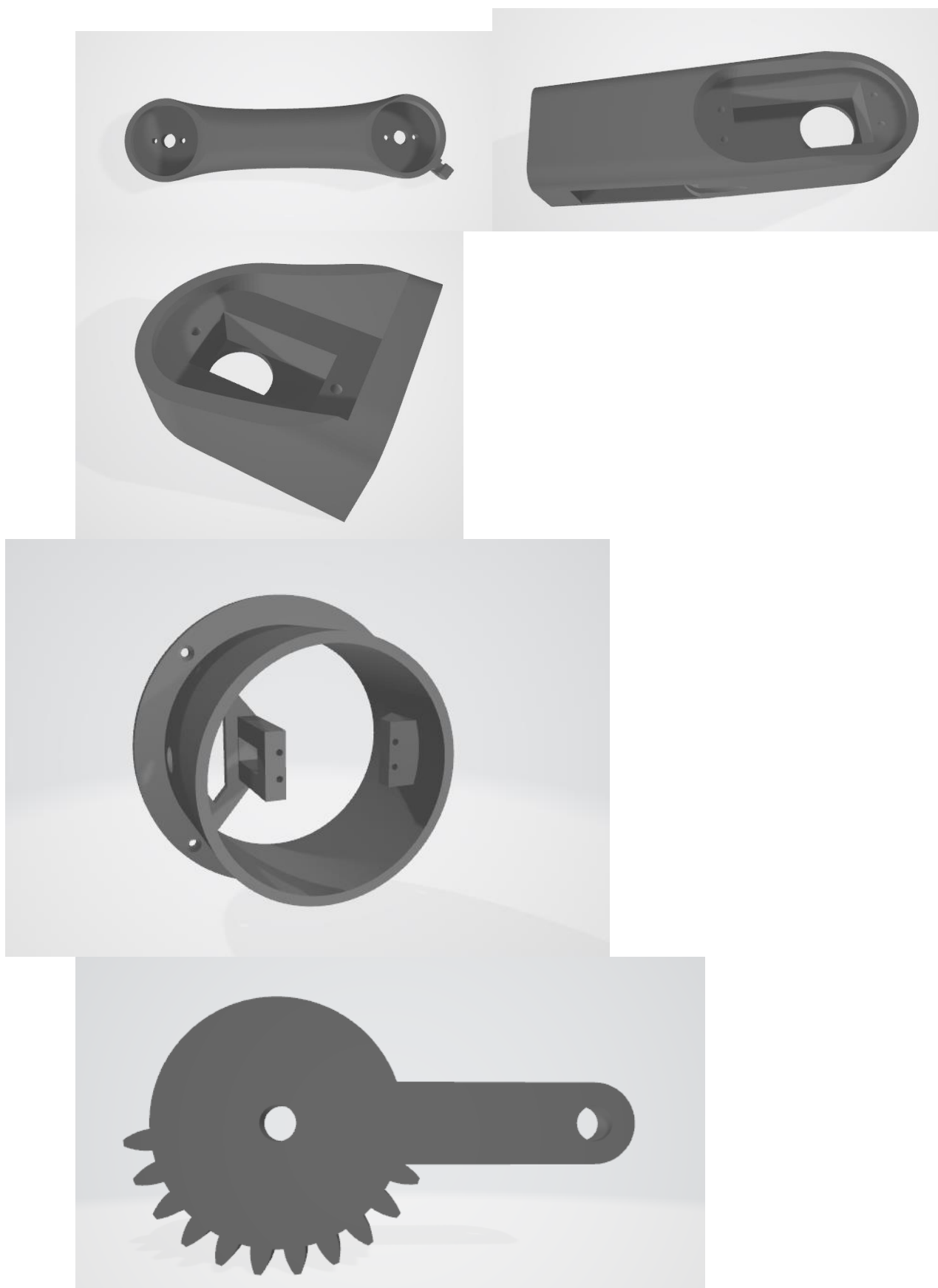
## **ЗАКЛЮЧЕНИЕ**

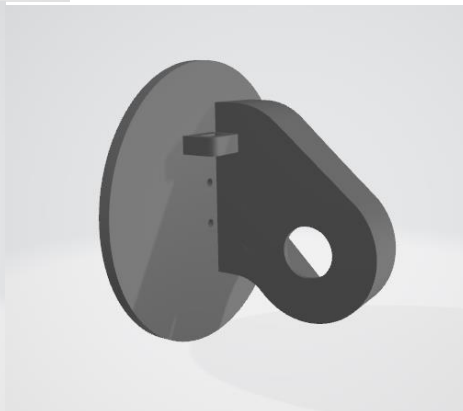
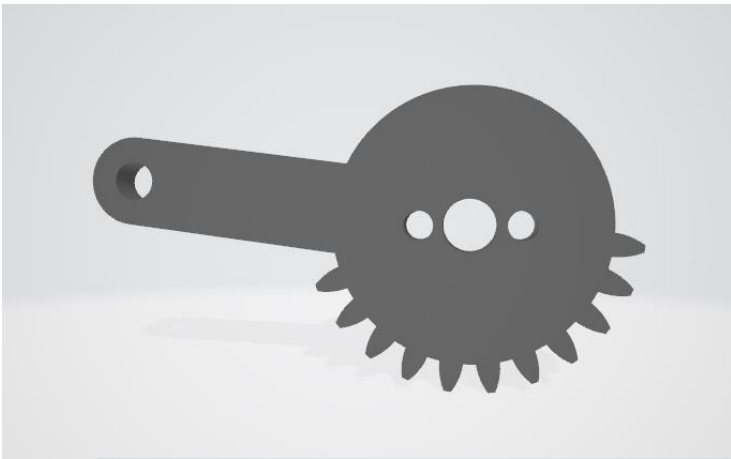
Данный робот манипулятор является самым оптимальным. Он может использоваться как отдельно вручную, так и часть комплексного робота. Такие роботы-манипуляторы в значительной степени могут спасти жизни людей в чрезвычайных ситуациях.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

[https://www.keldysh.ru/papers/2006/prep12/prep2006\\_12.files/image006.jpg](https://www.keldysh.ru/papers/2006/prep12/prep2006_12.files/image006.jpg)  
[https://img.directindustry.com.ru/images\\_di/photo-mg/7945-16382731.jpg](https://img.directindustry.com.ru/images_di/photo-mg/7945-16382731.jpg)  
[https://e-notabene.ru/img/articles/35862\\_28.jpg](https://e-notabene.ru/img/articles/35862_28.jpg)  
[https://cdn21.img.ria.ru/images/100747/73/1007477354\\_0:121:2448:1510\\_600x0\\_80\\_0\\_0\\_d0533bb9b16fd2c514913e386284878e.jpg](https://cdn21.img.ria.ru/images/100747/73/1007477354_0:121:2448:1510_600x0_80_0_0_d0533bb9b16fd2c514913e386284878e.jpg)  
[https://mirrobotics.ru/wp-content/uploads/2018/11/robotsYaskawa\\_960-370.jpg](https://mirrobotics.ru/wp-content/uploads/2018/11/robotsYaskawa_960-370.jpg)  
<https://static.mk.ru/upload/entities/2022/12/20/10/articles/facebookPicture/8d/1c/6b/93/d288f061d18b4b44a70d47dd0ae17ae9.jpg>  
<https://habrastorage.org/files/e98/53d/f17/e9853df17214430283ac112e4839a877.gif>  
[https://doc.arduino.ua/img/hardware/ArduinoUno\\_r2\\_front.jpg](https://doc.arduino.ua/img/hardware/ArduinoUno_r2_front.jpg)  
<https://appinventor.mit.edu/>

## ПРИЛОЖЕНИЕ А





## ПРИЛОЖЕНИЕ Б

```
/*
    DIY Arduino Robot Arm Smartphone Control
    by Dejan, www.HowToMechatronics.com
*/
#include <SoftwareSerial.h>
#include <Servo.h>
Servo servo01;
Servo servo02;
Servo servo03;
Servo servo04;
Servo servo05;
Servo servo06;
SoftwareSerial Bluetooth(3, 4);
int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos;
int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos;
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50],
servo05SP[50], servo06SP[50];
int speedDelay = 20;
int index = 0;
String dataIn = "";
void setup() {
    servo01.attach(5);
    servo02.attach(6);
    servo03.attach(7);
    servo04.attach(8);
    servo05.attach(9);
    servo06.attach(10);
    Bluetooth.begin(38400);
    Bluetooth.setTimeout(1);
    delay(20);
    servo1PPos = 90;
    servo01.write(servo1PPos);
    servo2PPos = 150;
    servo02.write(servo2PPos);
    servo3PPos = 35;
    servo03.write(servo3PPos);
    servo4PPos = 140;
    servo04.write(servo4PPos);
    servo5PPos = 85;
    servo05.write(servo5PPos);
    servo6PPos = 80;
    servo06.write(servo6PPos);
}
```

```

}
void loop() {
  if (Bluetooth.available() > 0) {
    dataIn = Bluetooth.readString();
    if (dataIn.startsWith("s1")) {
      String dataInS = dataIn.substring(2, dataIn.length());
      servo1Pos = dataInS.toInt(); // Convert the string into integer
      if (servo1PPos > servo1Pos) {
        for ( int j = servo1PPos; j >= servo1Pos; j--) {
          servo01.write(j);
          delay(20);
        }
      }
      if (servo1PPos < servo1Pos) {
        for ( int j = servo1PPos; j <= servo1Pos; j++) {
          servo01.write(j);
          delay(20);
        }
      }
      servo1PPos = servo1Pos;
    }
    if (dataIn.startsWith("s2")) {
      String dataInS = dataIn.substring(2, dataIn.length());
      servo2Pos = dataInS.toInt();
      if (servo2PPos > servo2Pos) {
        for ( int j = servo2PPos; j >= servo2Pos; j--) {
          servo02.write(j);
          delay(50);
        }
      }
      if (servo2PPos < servo2Pos) {
        for ( int j = servo2PPos; j <= servo2Pos; j++) {
          servo02.write(j);
          delay(50);
        }
      }
      servo2PPos = servo2Pos;
    }
    if (dataIn.startsWith("s3")) {
      String dataInS = dataIn.substring(2, dataIn.length());
      servo3Pos = dataInS.toInt();
      if (servo3PPos > servo3Pos) {
        for ( int j = servo3PPos; j >= servo3Pos; j--) {
          servo03.write(j);

```



```

        delay(30);
    }
}
if (servo3PPos < servo3Pos) {
    for ( int j = servo3PPos; j <= servo3Pos; j++) {
        servo03.write(j);
        delay(30);
    }
}
servo3PPos = servo3Pos;
}
if (dataIn.startsWith("s4")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo4Pos = dataInS.toInt();
    if (servo4PPos > servo4Pos) {
        for ( int j = servo4PPos; j >= servo4Pos; j--) {
            servo04.write(j);
            delay(30);
        }
    }
    if (servo4PPos < servo4Pos) {
        for ( int j = servo4PPos; j <= servo4Pos; j++) {
            servo04.write(j);
            delay(30);
        }
    }
    servo4PPos = servo4Pos;
}
if (dataIn.startsWith("s5")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo5Pos = dataInS.toInt();
    if (servo5PPos > servo5Pos) {
        for ( int j = servo5PPos; j >= servo5Pos; j--) {
            servo05.write(j);
            delay(30);
        }
    }
    if (servo5PPos < servo5Pos) {
        for ( int j = servo5PPos; j <= servo5Pos; j++) {
            servo05.write(j);
            delay(30);
        }
    }
}
servo5PPos = servo5Pos;

```

```

}
if (dataIn.startsWith("s6")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo6Pos = dataInS.toInt();
    if (servo6PPos > servo6Pos) {
        for ( int j = servo6PPos; j >= servo6Pos; j--) {
            servo06.write(j);
            delay(30);
        }
    }
    if (servo6PPos < servo6Pos) {
        for ( int j = servo6PPos; j <= servo6Pos; j++) {
            servo06.write(j);
            delay(30);
        }
    }
    servo6PPos = servo6Pos;
}
if (dataIn.startsWith("SAVE")) {
    servo01SP[index] = servo1PPos;
    servo02SP[index] = servo2PPos;
    servo03SP[index] = servo3PPos;
    servo04SP[index] = servo4PPos;
    servo05SP[index] = servo5PPos;
    servo06SP[index] = servo6PPos;
    index++;
}
if (dataIn.startsWith("RUN")) {
    runservo();
}
if ( dataIn == "RESET") {
    memset(servo01SP, 0, sizeof(servo01SP));
    memset(servo02SP, 0, sizeof(servo02SP));
    memset(servo03SP, 0, sizeof(servo03SP));
    memset(servo04SP, 0, sizeof(servo04SP));
    memset(servo05SP, 0, sizeof(servo05SP));
    memset(servo06SP, 0, sizeof(servo06SP));
    index = 0; // Index to 0
}
}
}
void runservo() {
    while (dataIn != "RESET") { "RESET" button is pressed
        for (int i = 0; i <= index - 2; i++) { // Run through all steps(index)

```

```

if (Bluetooth.available() > 0) {
  dataIn = Bluetooth.readString();
  if ( dataIn == "PAUSE") {
    while (dataIn != "RUN") {
      if (Bluetooth.available() > 0) {
        dataIn = Bluetooth.readString();
        if ( dataIn == "RESET") {
          break;
        }
      }
    }
  }
  if (dataIn.startsWith("ss")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    speedDelay = dataInS.toInt();
  }
}
if (servo01SP[i] == servo01SP[i + 1]) {
}
if (servo01SP[i] > servo01SP[i + 1]) {
  for ( int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {
    servo01.write(j);
    delay(speedDelay);
  }
}
if (servo01SP[i] < servo01SP[i + 1]) {
  for ( int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {
    servo01.write(j);
    delay(speedDelay);
  }
}
if (servo02SP[i] == servo02SP[i + 1]) {
}
if (servo02SP[i] > servo02SP[i + 1]) {
  for ( int j = servo02SP[i]; j >= servo02SP[i + 1]; j--) {
    servo02.write(j);
    delay(speedDelay);
  }
}
if (servo02SP[i] < servo02SP[i + 1]) {
  for ( int j = servo02SP[i]; j <= servo02SP[i + 1]; j++) {
    servo02.write(j);
    delay(speedDelay);
  }
}

```

```

}
if (servo03SP[i] == servo03SP[i + 1]) {
}
if (servo03SP[i] > servo03SP[i + 1]) {
    for ( int j = servo03SP[i]; j >= servo03SP[i + 1]; j--) {
        servo03.write(j);
        delay(speedDelay);
    }
}
if (servo03SP[i] < servo03SP[i + 1]) {
    for ( int j = servo03SP[i]; j <= servo03SP[i + 1]; j++) {
        servo03.write(j);
        delay(speedDelay);
    }
}
if (servo04SP[i] == servo04SP[i + 1]) {
}
if (servo04SP[i] > servo04SP[i + 1]) {
    for ( int j = servo04SP[i]; j >= servo04SP[i + 1]; j--) {
        servo04.write(j);
        delay(speedDelay);
    }
}
if (servo04SP[i] < servo04SP[i + 1]) {
    for ( int j = servo04SP[i]; j <= servo04SP[i + 1]; j++) {
        servo04.write(j);
        delay(speedDelay);
    }
}
if (servo05SP[i] == servo05SP[i + 1]) {
}
if (servo05SP[i] > servo05SP[i + 1]) {
    for ( int j = servo05SP[i]; j >= servo05SP[i + 1]; j--) {
        servo05.write(j);
        delay(speedDelay);
    }
}
if (servo05SP[i] < servo05SP[i + 1]) {
    for ( int j = servo05SP[i]; j <= servo05SP[i + 1]; j++) {
        servo05.write(j);
        delay(speedDelay);
    }
}
if (servo06SP[i] == servo06SP[i + 1]) {

```

```
    }
    if (servo06SP[i] > servo06SP[i + 1]) {
        for ( int j = servo06SP[i]; j >= servo06SP[i + 1]; j--) {
            servo06.write(j);
            delay(speedDelay);
        }
    }
    if (servo06SP[i] < servo06SP[i + 1]) {
        for ( int j = servo06SP[i]; j <= servo06SP[i + 1]; j++) {
            servo06.write(j);
            delay(speedDelay);
        }
    }
}
}
```