UNIVERSITY OF GÄVLE

FACULTY OF ENGINEERING AND SUSTAINABLE DEVELOPMENT
**Department of Electrical Engineering, Mathematics and Science**

# Evaluation, Design & Development of a Prototype 3-Link Mini Robot Manipulator

Rijon Ahmed

January, 2022

Student thesis, Advanced level (Master degree, two years), 30 HE
Electronics
Master Programme in Electronics/Automation

Supervisor: Dr Sajid Rafique
Assistant supervisor: Shaikh Masud Rana
Examiner: José Chilo

# Acknowledgements

The Master's thesis was written in the Electrical Engineering Mathematics, and science at the University of Gävle in Gävle city, Sweden.

At first, I would like to thank my supervisor, Dr. Sajid Rafique for providing the opportunity to do this thesis work. Special thanks to Shaikh Masud Rana, my assistant supervisor, for addressing all of my queries. Their knowledge has proven to be invaluable.

I would also like to express my gratitude to José Chilo, my thesis examiner.

Finally, I would like to express my gratitude to my family and friends. This would not be possible without your love and support.

Thank you

Gävle, January 2022

Rijon Ahmed

# Abstract

A 3-link robotic arm was designed, constructed, and tested as part of the thesis work. This prototype will assist students in applying their robotics and control system theory knowledge to real results. The kinematic equations are created to help with trajectory planning. Evaluation of different parameters was determined like (angles at which the servo motor operated, link length at which the servo can carry, spatial velocity, DH parameter, and Homogeneous transformation matrix). An Arduino-based closed-loop control system is built. Four Servo motors were used which are being controlled by Arduino UNO and Leonardo. The Arduino IDE is used to write proper codes. The main aim of the study was to apply the knowledge of robotics and control systems to develop a functional mini robot from scratch. Specifically, the thesis presents how to build a robotic arm that can move and lift objects. And, this task is done by using different controlling techniques where potentiometer, Bluetooth Module, and IR senor were used and compared which technique gives better results. The structural components had several issues. The project's linkages foundation and gripper are all 3-D printed pieces that are being designed using Autodesk Inventor Professional 2021 software. Although they did not cause any problems with strength, there were some difficulties with properly tightening the gears onto the shaft. To overcome these difficulties glue is being used so that all servo motors hold with the link properly. In a broad sense, the robot can position and orient the end-effector to pick and place the object accurately from a distance.

# Table of contents

## Contents

# List Of Figures

# 1 Introduction

A robotic arm can be described as a series of arm-like links that are connected and programmed to do tasks such as picking and placing, assembling, and so on. Robotic arms can be used in a variety of applications. For various purposes, many industries such as automobiles profit from them. They are more accurate than humans and do not perform poorly. These are just a few of the factors why industries and large corporations have begun to hire employees with industrial robots. The growth of robots is expected to become an important aspect of engineering. As a result, students will be able to learn about various aspects of engineering. The main aim of the study was to apply the knowledge of robotics and control system which are being covered in this master's degree program to develop the robotic arm.

This chapter presents a high-level overview of the design and development of a 3-Link Mini Robot manipulator and evaluation of the kinematic model, as well as the thesis's foundation. Through furthermore, the whole section will cover the project's objectives and deliverables. The thesis structure, objective of research, study methodology, and Sustainable Development Goals, which are all contained in this thesis, will be thoroughly studied. Furthermore, with dynamic modeling of robot manipulators with joints becoming commercially accessible at a reduced cost and with increased accuracy, a point-to-point interpolating trajectory is becoming a prominent topic of this study.

## 1.1 Background

The purpose of this research is to design and develop a 3-Link Mini Robot that will operate point to point and replace the working piece in an industry or institution environment, as well as to attract more people to learn more about this modeling process. The University of Gävle produced this project to aid industrial or educational areas when researchers or students modeling robots for seeing their frequent requiring jobs. This project focuses on design, development and structural analysis of a robotic arm. Prototypes helps in product design process to explore design possibilities, test theories, and confirm performance before initiating. This prototype will help the student to apply their theory knowledge of robotics and control system to compare with the practical results. This will also assist students in identifying issues and places for improvement so that essential changes may be made prior to development and students can also apply different control theories to operate the arm more effectively.

## 1.2 Goals

The main goal of the thesis is to have a comprehensive set of created resources and knowledge that can be used to duplicate the robot modeling technique and include it into design approaches for future projects for any institution.

Hence, the work is to structurally develop mini robot for a specific task like picking and placing and evaluation like (determining angles at which servo motor operates, maximum weight servo can carry, DH parameter, Homogeneous transformation matrix and spatial velocity) and also programmed various control techniques where Bluetooth module and IR sensor is used to analyze the performance.

## 1.3 Overview

In chapter 2 the overall theory will be discussed which will cover the diagram Kinematic model of the Robot arm, DH parameter, Homogenous transformation matrix equation, kinematics, Jacobian, Degree of Freedom (DOF) and Dynamics. In the chapter 3 the step-by-step process of the thesis will be discussed which includes design, implementation, list of hardware, software's used, Circuit diagram, Block diagram for different control techniques, flowchart of the system how it works, App development tools for the Bluetooth module. In chapter 4 all the results will be provided which includes the picture of the whole model, Homogeneous transformation matrix, Jacobian, spatial velocity of the joints, output graph of the servo motor and how to determine the operation angle of the servo motor and the results of a successful pick and place of an object by using the robot arm. In chapter 5 all the control techniques will be analyzed and discuss the best techniques that works well in the system. In chapter 6 where the thesis will be concluded and scope for the future improvement will be discussed.

## 1.4 Sustainable Development Goals

In terms of sustainable development objectives, this study aims to address one objective (9.1) and one major goal (9) from Agenda 2030. Besides, this initiative will aid industry, innovation, and infrastructure by delivering dependable precision approaches, which is a main aim in terms of improvements in the industrial and educational fields. Furthermore, this study should contain detailed information on how to operate the 3-link tiny robot with an IR sensor, manually utilizing potentiometers, Bluetooth module Hc-06, and Arduino code to evaluate performance at point-to-point movement with workspace. Therefore, this mini robot designing modules is available and economical to all sectors, eventually developing a high-quality infrastructure.

This initiative solely stimulates several additional long-term sustainable goals as subsidiary goals (7 and 11) from Agenda 2030. In terms of secondary and long-term consequences, it strongly encourages individuals to research or create industrial robots rather than relying on other ways of performance at point-to-point mobility with workspace. For these considerations, it offers the possibility of a broad range of applications in which additional characteristics such as pedestrians or gravity may be taken into account. Finally, all types of robots designing modules are helped industrial settings, which will help to reduce manual labor in industrial environments.

# 2 Theory

The design and development of a 3-link micro robot manipulator is the focus of this thesis. To accomplish this, several stages were taken, from design to installation, and a suitable link was constructed so that the servo motor could handle the maximum load [1]. Different techniques were done while pick and place the object from one place to another. For the robot arm movement Servo motors were used and for the controlling Arduino Leonardo and UNO were used. 3D software is used in designing all the arm and all the components are printed with 3D printer Different techniques are used in controlling the robot which includes manual controlling by using potentiometer, automatic movement of the arm, using IR sensor to control the servo motor of the end effector and lastly using the Bluetooth module control the end effector by smartphone.

## 2.1 Robot Modules and Parameters

Controlling a robot requires a vast number of mathematical equations. For dealing with the calculation, it is preferable to break it into segments, with each unit doing a specific duty. Besides, according to [2] the most significant modules are kinematics, inverse kinematics, spatial velocity, and feedback control. The selection of robot characteristics such as degrees of freedom, joint allocation, link lengths, link masses, and motor parameters influences the robot's behavior. As part of the prototype environment, an optimum design system will be built to pick the ideal values for those parameters based on the mentioned performance standards.

## 2.2 Kinematics

Kinematics is a fundamental and classic topic in robotics that studies the link between a robot's joint coordinates [3]. This can help in designing a mechanism to move an object from one point to another. Kinematics, in particular, is important in robotics, particularly when studying the behaviors of industrial manipulators [4]. As a result, analyzing and modeling the manipulator kinematics is a critical stage in any robotics system. As a result, analyzing and modeling the manipulator kinematics is a critical stage in any robotics system. Forward and inverse kinematics are two types of kinematics.

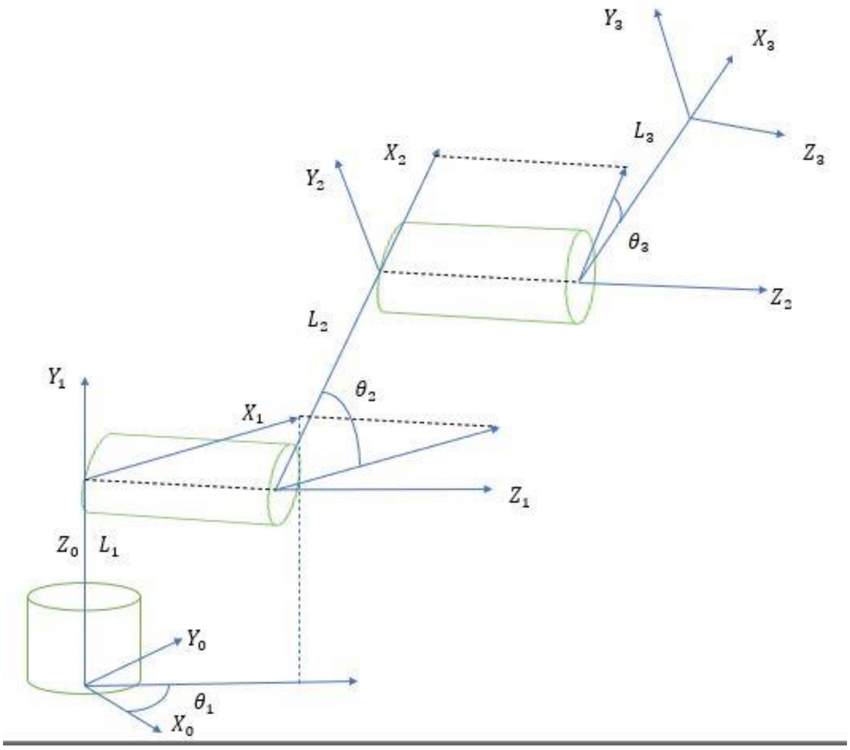


*Figure 2.1. Kinematic model of a 3-link mini robot manipulator*

### 2.2.1 DH Parameter

In the kinematics study of robotic manipulators, the Denavit-Hartenberg (DH) parameter is often utilized [4]. It works by attaching a coordinate frame to each joint and specifying three DH parameters for each connection, then constructing a DH table using these parameters. As in this thesis work the robot has 3-link.

| $\text{Frame}_i$ | $a_i$ | $\alpha_i$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| 1 | 0 | 90 | 0 | $\theta_1$ |
| 2 | $L_2$ | 0 | 0 | $\theta_2$ |
| 3 | $L_3$ | 0 | 0 | $\theta_3$ |

*Table 2.2. Denavit-Hartenberg Parameter*

$a_i$= Distance from $z_i$ to $z_{i+1}$ measured along $x_i$

$\alpha_i$= Angle of twist from $z_i$ to $z_{i+1}$ measured along $x_i$

$d_i$= Distance from $x_{i-1}$ to $x_i$ measured along $z_i$

$\theta_i$= Angle from $x_{i-1}$ to $x_i$ measured along $z_i$

### 2.2.2 Forward Kinematics

Forward kinematics is an application of kinematic equation it is used to calculate the position of the end effector from a given joints [5]. In the kinematics analysis of robot manipulators and other mechanical systems, the DH parameters and product of exponentials are frequently used. This is a generic matrix that can be used to represent a revolute or prismatic joint in a certain orientation. By multiplying the matrices of rotation and linear motion, the following equation [5] can be used to determine the transformation. When angles and link length are known, this computation can be performed.

$$T_i = Rot_{z,\theta i} * Trans_{z,di} * Trans_{x,\alpha i} * Rot_{x,\alpha i} \qquad (2.1)$$

Transformation Matrices are obtained from the DH parameter

$$T_1^0 = \begin{vmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad T_2^1 = \begin{vmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cdot\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\cdot\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_3^2 = \begin{vmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_2\cdot\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_3\cdot\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Homogeneous transformation matrices are the product of all the transformation matrices of each Link, therefore the homogeneous transformation matrices are

$$T_3^0 = T_1^0\cdot T_2^1\cdot T_3^2 \qquad (2.2)$$

$$T_3^0 = \begin{vmatrix} (\cos\theta_1\cdot\cos\theta_2\cdot\cos\theta_3) & (-\cos\theta_1\cdot\sin\theta_2\cdot\sin\theta_3) & \sin\theta_1 & \cos\theta_1\cdot(L_3\cdot\cos\theta_2\cdot\cos\theta_3 + L_2\cdot\cos\theta_2) \\ (\sin\theta_1\cdot\cos\theta_2\cdot\cos\theta_3) & (-\sin\theta_1\cdot\sin\theta_2\cdot\sin\theta_3) & -\cos\theta_1 & \sin\theta_1\cdot(L_3\cdot\cos\theta_2\cdot\cos\theta_3 + L_2\cdot\cos\theta_2) \\ (\sin\theta_2\cdot\sin\theta_3) & (\cos\theta_2\cdot\cos\theta_3) & 0 & (L_3\cdot\sin\theta_2\cdot\sin\theta_3 + L_2\cdot\sin\theta_2) \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (2.3)$$

### 2.2.3 Jacobian

In robotics, Jacobian is a Matrix that describes the relationship between joint and end-effector velocities of a robot manipulator [6]. If the robot's joints move at a specific speed, we might be interested in knowing how fast the end effector moves. The following is the relationship between joint velocities and end-effector velocities

$$\dot{X} = j\cdot\dot{q} \qquad (2.4)$$

As the robot has 3 links solving the Jacobian by using the following equation

$$j=\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}-a_2\sin(\theta_1+\theta_2)-a_1\sin\theta_1-a_3\sin(\theta_1+\theta_2+\theta_3) & -a_2\sin(\theta_1+\theta_2)-a_3\sin(\theta_1+\theta_2+\theta_3) & -a_3\sin(\theta_1+\theta_2+\theta_3)\\ a_2\cos(\theta_1+\theta_2)+a_1\cos\theta_1+a_3\cos(\theta_1+\theta_2+\theta_3) & a_2\cos(\theta_1+\theta_2)-a_3\cos(\theta_1+\theta_2+\theta_3) & a_3\cos(\theta_1+\theta_2+\theta_3)\\ 1 & 1 & 1\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{bmatrix} \quad (2.5)$$

### 2.2.4 Degree of Freedom (DOF)

The range of a system's freely and independently moving dimensions is defined by the degrees of freedom (DOF) [7]. The sum of all individual joints is called degree of freedom.

## 2.3 Dynamics

The link between the forces applied on a robot mechanism and the accelerations they produce is called robot dynamics [8]. Robot dynamics is the application of rigid-body dynamics to robotics when the robot mechanism is typically treated as a rigid-body system. The equations of motion for a manipulator are solved in Dynamics to compute the forces and torques that cause the manipulator to move. There are two primary methods of computing:

- Forward Dynamics: In forward dynamics joint motions are calculated based on joint forces and torques. This method is used in robot dynamic simulations.
- Inverse Dynamics: Inverse dynamics is a method for calculating joint forces and torques based on a body's kinematics. This is mostly applied in the control of robots.

For the dynamics, the Euler Lagrange method will be used. Develops an energy-based "Lagrangian Function" that connects the manipulator's Kinetic and Potential Energy as it moves, allowing the manipulator to be used in force/torque calculations.

## 2.4 Pulse Width Modulation

Pulse Width Modulation, a technique for translating digital outputs to control analog circuits, is used to operate servomotors [9]. The servomotor starts at 90 degrees and can move the same amount in both directions for a total of 180 degrees. This technique is used here while the robot arm movement was done manually by using potentiometer. The PWM will instruct the motor to move to the specified shaft position, and the duration of the pulse will cause the motor to rotate to that position.

# 3 Methodology

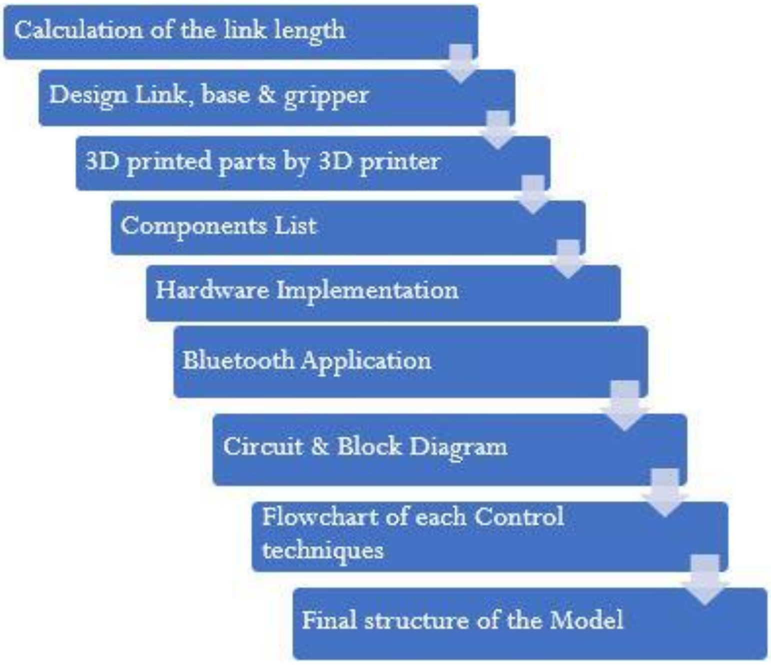

Calculation of the link length
Design Link, base & gripper
3D printed parts by 3D printer
Components List
Hardware Implementation
Bluetooth Application
Circuit & Block Diagram
Flowchart of each Control techniques
Final structure of the Model

*Figure 3.1. Flowchart construction of robotic arm*

This study goes through a number of phases, such as determining the length of the link that servo motors can carry, designing the base, link and gripper of the arm, 3D-printed parts by using 3D-printer, hardware's list, hardware implementation, developing application for Bluetooth module to control the robot arm with smartphone, designing circuit & block diagram, Flowchart for different control techniques, Arduino code is provided in Appendix, and lastly the final structural model of 3-link robot arm.

## 3.1 Calculation of the link length Servo Motor can carry

The Servo MG966r is employed in this thesis work, and it weighs 55 grams and the dimensions is (40.7*19.7*42.9) mm which is gathered from the data sheet of the servo [10]. The servo motor has a torque of 9400g/cm at 4.8v.

For an appropriate arrangement the length of the link is chosen to be 13.8cm. From here, we'll figure out how much weight the servo motor can handle. Bu using the following equation we will determine the maximum weight a servo motor can handle with a length of 13.8cm.

We know,

$$\tau = M * G * R \qquad (5)$$ [11]

Here,

$\tau$= Torque

M= Mass (grams)

R= Distance(cm)

G= gravitational acceleration

So, M = 9400/ (9.81*13.8)

M = 69.5grams

With a link length of 13.8cm, the servo will be able to transport a maximum mass of 69.5 grams, according to the calculations.

## 3.2 CAD Design

I used Autodesk Inventor Professional 2021 to construct the mechanical links of the robot [12]. The student version of this software is being used as per instructions of the supervisor to design all the links. Professional-grade 3D mechanical design, research, and product simulation tools are available in Autodesk Inventor software. It uses a powerful combination of parametric, direct, freeform, and rules-based design features to get the job done quickly.

Following the installation of the software, various video lessons were used to master some basic design concepts. The drawing will be created in 2D first, then converted to a 3D shape from there the thickness of the link will be given.

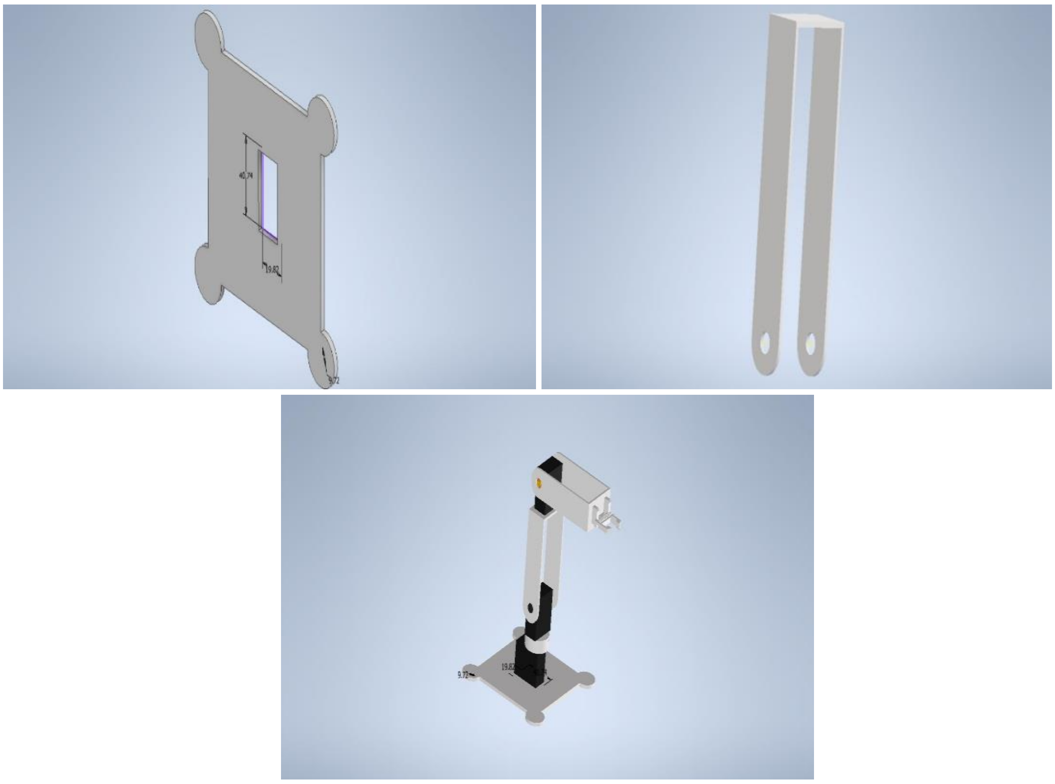


*Figure 3.2. 3D printed parts of Robot Arm*

In the first picture a square-shaped base with a length of 90mm and a hole is created for the servo motor with a dimension of (40.74*19.82) mm so that the servo motor can be installed. In the second picture the arm link is being designed the arm link is 138mm with a thickness of 2.5mm.

The third picture shows the whole 3D demonstration of the Robot where base, links and the gripper are connected with the servo motor.

### 3.2.1 3D printing parts

Apart from servo motors all the links base and gripper are printed by using 3D printer from University of Gävle workshop building 45. F123 series 3D printer is used. The F123 Series printers combine manufacturing performance with ease of use [13]. ABS (Acrylonitrile Butadiene Styrene) is a dense thermoplastic and amorphous polymer that is employed in this design [14]. ABS is known for its strength and resilience while remaining lightweight and robust, making it ideal for most 3D printing applications. Fig 3.8. shows the final version of the Mini Robot manipulator.

## 3.3 Components List

The University of Gävle has provided funding for this initiative, which has examined and assisted in the development of an industry 4.0 research field. Besides, rotational cubes are employed in the suggested idea to rotate the three joints with tiny increments, and the difference between the actual state as well as the intended configuration is evaluated. Researchers can use the toolbox below to build any mini robot

- Arduino Leonardo, UNO
- Servo motor MG966r
- Micro servo
- Bluetooth module Hc-06
- IR sensor
- Breadboard
- Connecting cables
- Potentiometer (10K-ohm)

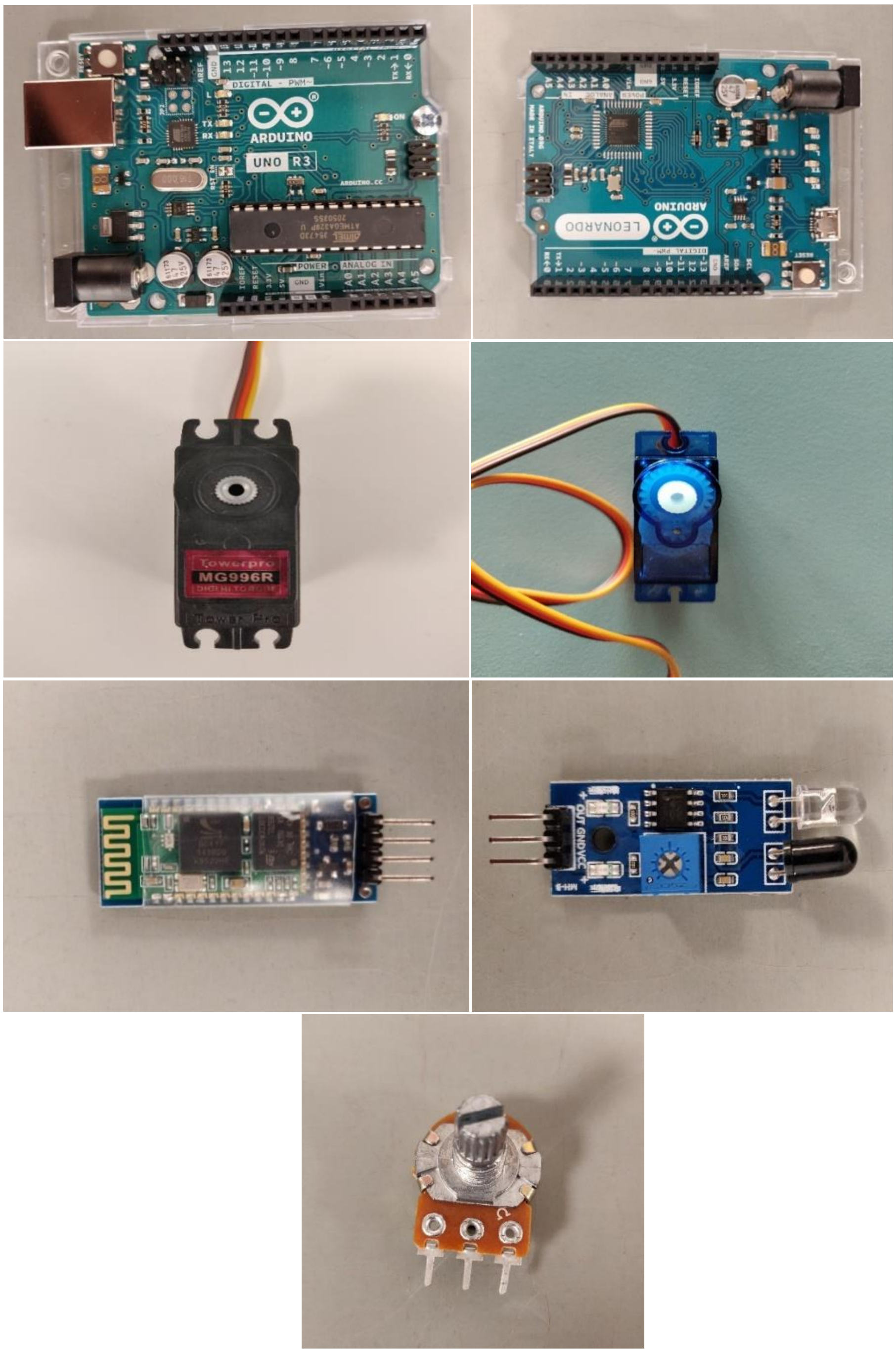


*Figure 3.3. Hardware Components*

Arduino Leonardo is a microcontroller with 14 digital pins and 6 analog pins. A USB port, a power jack, an ICSP header, and a reset button are included. It comes with everything you'll need to get started with the microcontroller [15]. Because it has more ports and allows for the expansion of other sensors, the Arduino Leonardo single-chip microcontroller was chosen as the electronic core control system in this design. This single-chip microcomputer has the advantages of being compact and efficient, as well as being more expandable, which satisfies the small, precise, and accurate criteria of a multi-point monitoring system. Furthermore, Arduino is user-friendly, allowing for the integration of a variety of sensors.

Arduino UNO is a microcontroller it is also similar to Arduino Leonardo based on the ATmega328 microcontroller [16]. This particular Arduino is used to connect with the IR sensor which will control the gripper of the robot. Purchasing or assembling Arduino hardware is inexpensive. The application is also free to download, allowing the user to study and update it as needed. The IDE can run on a variety of platforms, including Microsoft, Linux, and Mac OS, allowing it to gain popularity.

The servo motor MG996R has a maximum stall torque of 11 kg/cm. The motor rotates from 0 to 180 degrees based on the duty cycle of the PWM wave given to its signal pin, just as other servos [17]. This is the main motors which is used to operate the three joints of the robot arm. It has 3 wires brown is the ground wire that will be connected to the ground of the system. Red wire is the Vcc to power the servo motor normally, +5V is used and to drive the motor, a PWM signal is sent through the orange wire.

TS90A is a micro servo small and light with a high output power. The servo can rotate about 180 degrees. This particular servo is used in the gripper end of the robot arm to pick and place any light weight object. It functions similarly to the standard types, but on a smaller scale. They can run at any speed, whether it's very slow or very fast. The high force can be acquired from these servo motor [18]. Servo motors are utilized in a variety of control systems, including quick operation, axis movement, and many others. It can also operate in both AC and DC current.

The HC-06 Bluetooth module is a wireless serial communication slave module. Master and slave are determined in Bluetooth communication based on the status of the connection when it is established [19]. Bluetooth module is used in this design as one of the control techniques where the robot arm will be controlled by using smartphone by connecting with the Bluetooth module. When the module is not paired with the app on the phone, the LED blinks frequently, however when the module is paired with the app on the phone, the LED stops blinking and remains static. The Bluetooth module has 4 pin that is Vcc, Gnd, Tx, & Rx and the pin configuration is easy to define as each pin name is printed on the back side module.

An infrared sensor is a type of electronic device. IR sensor emits light in order to detect a nearby item. The infrared sensor can both detect and measure the heat of an object. Almost all items emit some type of thermal radiation in the infrared range. These sorts of radiations are invisible to our sight, but they can be detected by an IR sensor [20]. This IR sensor is used at the end effector/gripper, and it is controlled by an Arduino UNO. Therefore, the gripper closes automatically when an IR sensor detects any objects. IR sensor has 3 pin that is Vcc, GND, & output pin it also has an adjustable knob at the top this is used to increase or decrease the intensity of the detection and the name of the pins are printed on the body, making them easier to spot.

When it comes to adjusting a system's electrical settings, potentiometers are quite beneficial [21]. With a turning knob, it's a single-turn 10k potentiometer. In this work Potentiometers are used as control technique where the robot arm can be controlled manually by rotating the knob in both clockwise and anticlockwise direction of the Potentiometers. By moving the knob on these three-terminal devices, it can change the resistance from 0 to 10k ohms. It has 3 pin that is Vcc, GND, and signal output pin. The angle is mapped and supplied to the servo motors via the Arduino through coding. In this thesis work 4 analog potentiometers are used for servo motors to operate each servo independently.

## 3.4 Hardware Installation

In the first step the servo motor is connected with the base followed by circular link on top of that servo motor. In the second step the second servo was attached on top of the circular link and then the second link is attached with the second servo. In the third step same method was used to attached the third servo with the third link. And lastly a gripper is attached on top of the third link.

## 3.5 Software

The Arduino environment is free and simple to use. It runs on a different operating system, including Windows, Mac OS X, and Linux. It's designed to teach programming to people who aren't familiar with the field, making it ideal for students [22]. With a single click, it is capable of compiling and uploading codes to the board. “Sketch” is a place where code is written for the Arduino. C or C++ language is used to write Arduino code.

The Arduino board must be connected to the system via USB connection after the software has been installed. The correct Arduino board and port should be chosen before writing the code.

### 3.5.1 App for Bluetooth Module

To control the gripper movements wirelessly using a smartphone, an application must be created. To do that MIT app inventor is used, MIT App Inventor is a user-friendly, visual programming environment that allows anyone to create fully working Android and iOS apps [23]. As it is a block-based tool, it does not require typical programming environments. To begin working with MIT inventor, you must first create a user account. By, using the Gmail account this could be done and anyone having a valid Gmail ID can create the account. After creating the account it's ready to develop the application and by clicking the create app a new window will come fig 3.4. is illustrated below.

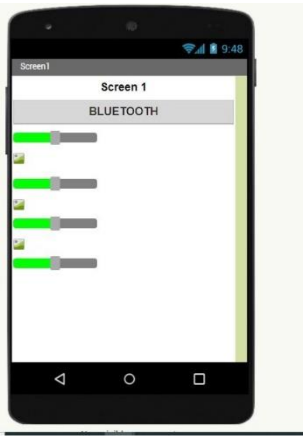


*Figure 3.4. MIT App inventor for making Bluetooth Interface*

After creating the application's interface by putting and labeling the buttons, a Bluetooth client should be added to the interface, but it will be hidden. When this procedure is completed, the next part is to build the blocks for the application so that the Bluetooth module can connect with the smart phone and can be operated by the smart phone to do that click the **"Blocks"** button to start see fig 3.5. The blocks will be provided in the appendix.

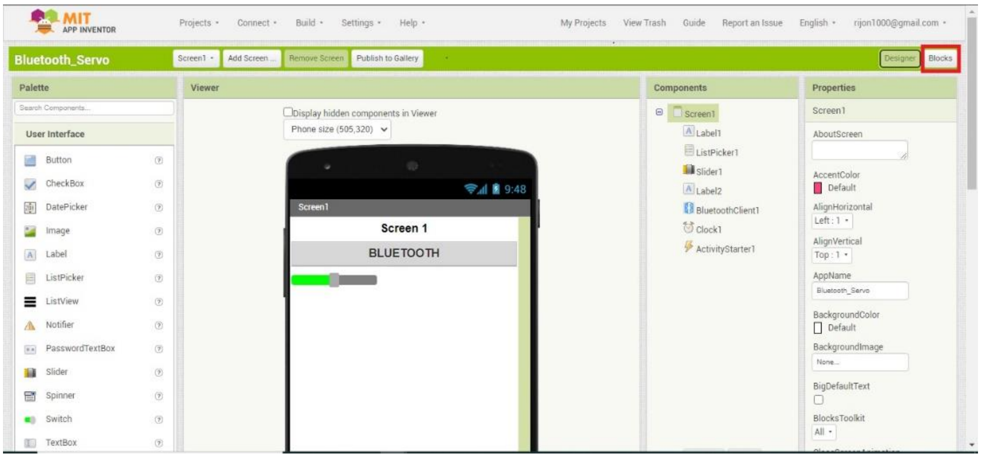
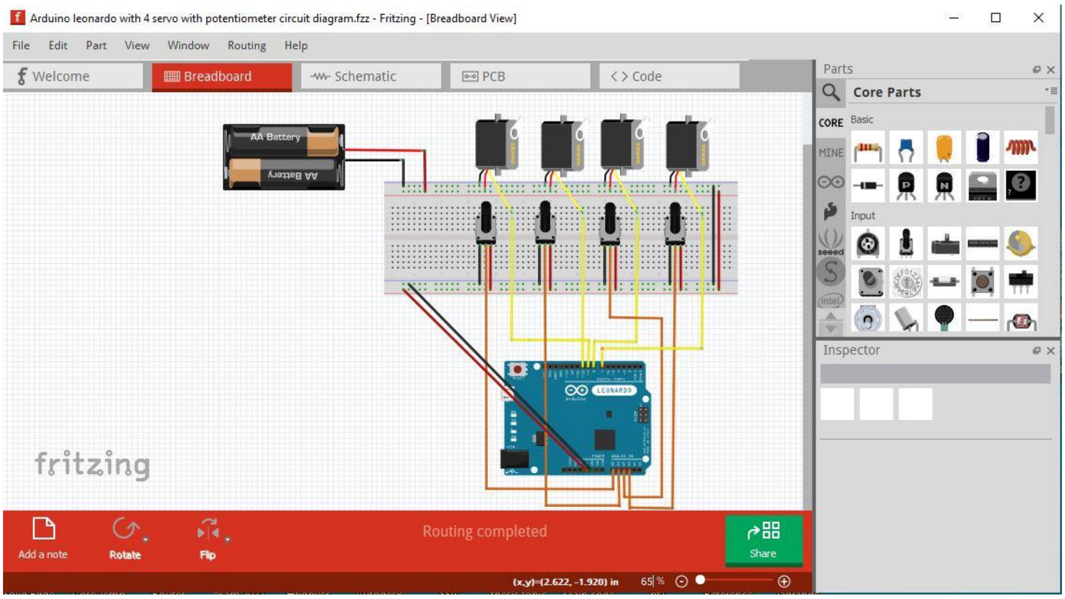


*Figure 3.5. start working with the app by clicking "**Blocks**" button*

## 3.6 Circuit & Block Diagram

### 3.6.1 Circuit Diagram for controlling the servo using potentiometer



Routing completed

*Figure 3.6. Circuit diagram of 3-link manipulator controlled by potentiometers*

Fig 3.6. shows this. Each potentiometer and servo motor has three pin connections (Vcc, out, and Gnd). The output pin of the potentiometer is connected to the Arduino's analog pin (A0, A1, A2, A3). And the output pin of the Servo motor is connected with the Arduino's PWM/Digital pin to control the motion of the servo motor. In this thesis work the Arduino is powered by connecting the Arduino with the laptop via USB cable.

### 3.6.2 Block Diagram for controlling the Robot Arm

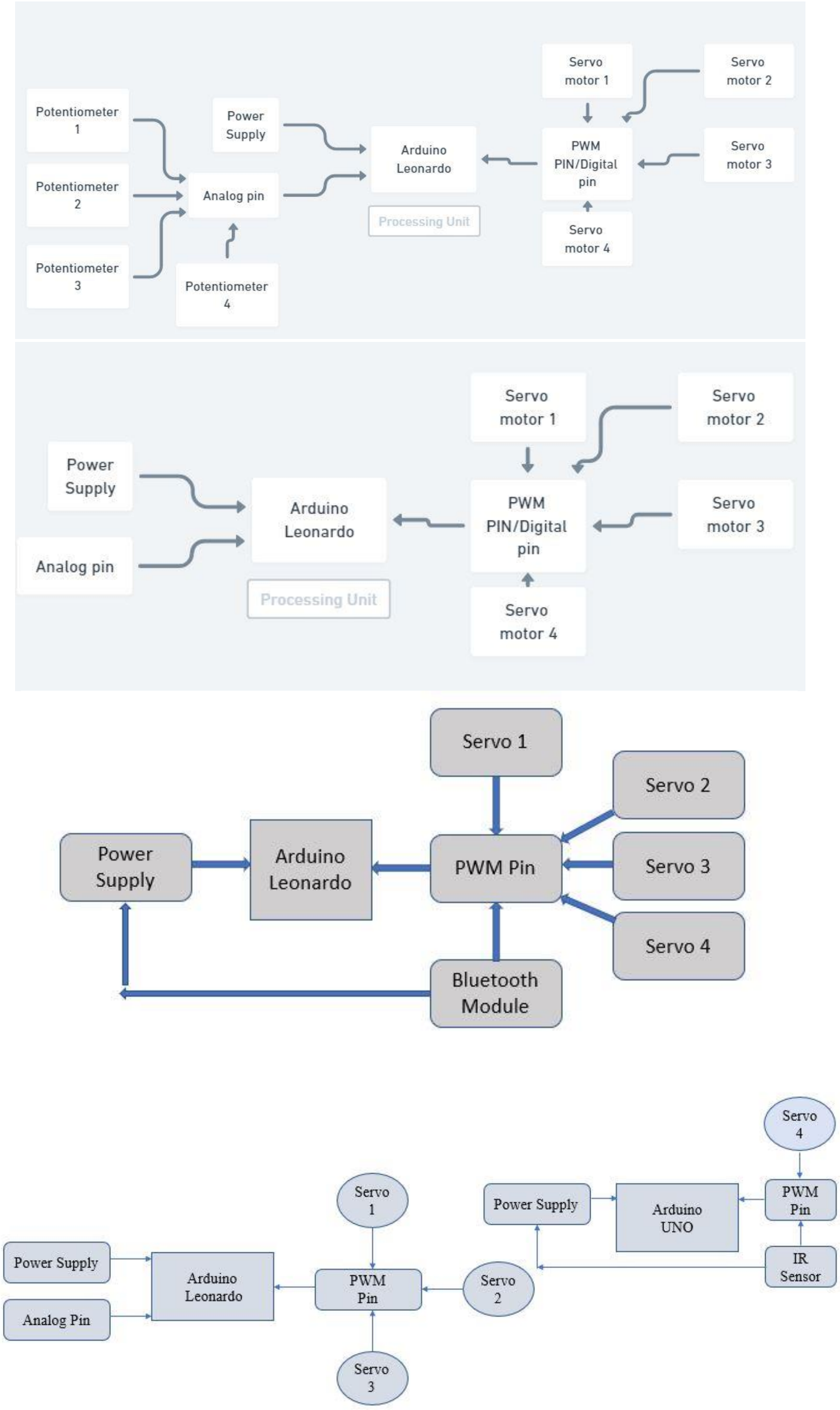


*Figure 3.7. Block diagram of different control system for 3-link manipulator*

First figure shows the block diagram where the robot arm is controlled manually by using potentiometers. Four Potentiometers were used for each servo motor and all the potentiometers is connected with the analog port of the Arduino and all the servo motors are connected with the PWM pin of the Arduino. Arduino Leonardo microcontroller is used for the control system and an external power source is used to power up the system.

The second figure shows the block diagram of the servo motor which is connected with Arduino Leonardo. This is almost same as the first figure but no potentiometers were used as the robot arm will operate automatically. All the servo motors are connected with PWM pin of the Arduino and an external power supply is used power up the whole system.

In the third figure the block diagram is shown for the robot arm controlled by using Bluetooth module. Arduino Leonardo is used for the control system. All servo motor and the Bluetooth module is connected with the Arduino's PWM pin. External power source is used to power up the system.

In fourth figure the block diagram shows two Arduino board is used for the operation three servo motor is connected with the PWM pin of the Arduino Leonardo and the 4th servo which is the gripper is connected with the PWM pin of the Arduino UNO and with that an IR sensor is used so that when an object will be found Infront of the IR sensor the gripper will automatically close.

## 3.7 Program flowchart for the Mini Robot controlled by Bluetooth Module

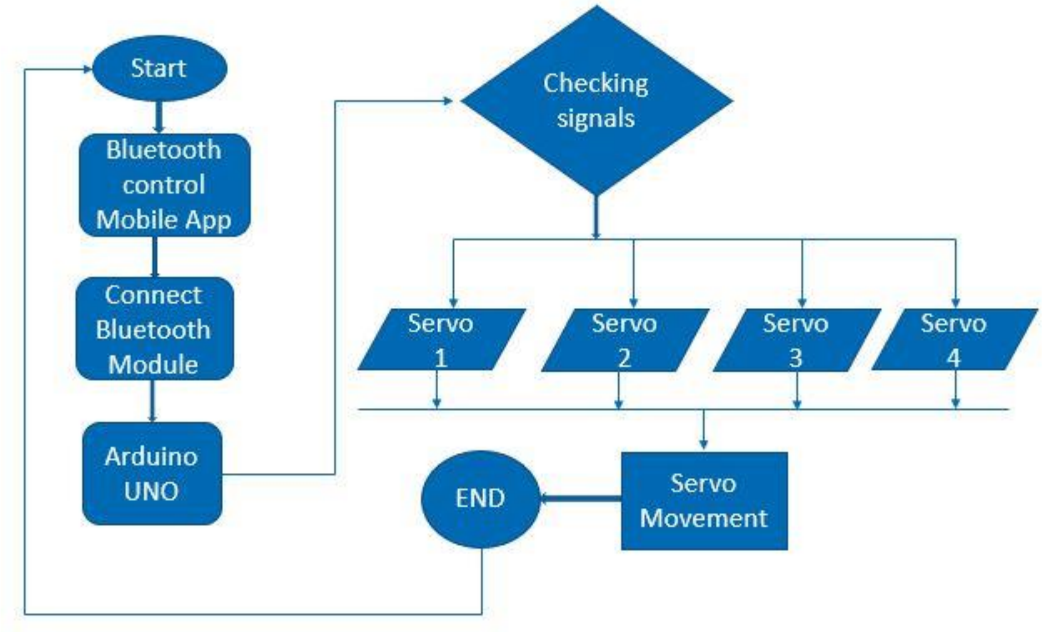


*Figure 3.8. Flowchart for 3 Link Manipulator controlled by Bluetooth Module*

The above figure shows the flowchart of a 3-Link manipulator controlled by Bluetooth module and step by step working principals. For this a smart phone is needed with the Bluetooth power on and also an application to run the whole system. After opening the application in the mobile, it needs to be paired with the HC-06 Bluetooth module which is connected with the Arduino to operate the robot arm. When the Bluetooth connection is established, it is ready to transmit and receive signals. When a command is sent via smartphone to the Bluetooth module of the system the transmitting signal will be checked and when it receives the signal the servo motors move according to command that is send from the smartphone.

### 3.7.1 Program Flowchart for the Mini Robot Manipulator controlled Manually using Potentiometer

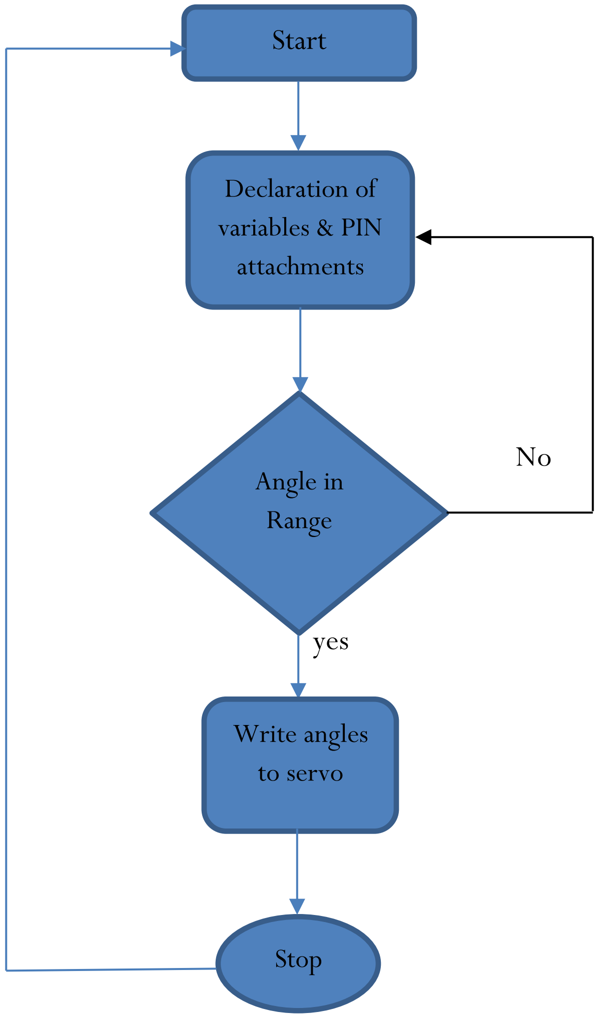


*Figure 3.9. Flowchart for 3-Link robot manipulator controlled by potentiometer*

The above figure shows the flowchart for the 3-Link robot manipulator controlled by potentiometer. When the system is turned on the servo is connected with the PWM pin to receive any signal from the potentiometer which is connected with the analog port of the Arduino. When potentiometer knob is rotate at any angle it will check whether the angle is in range hence, the PWM will instruct the motor to move to the specified shaft position, and the duration of the pulse will cause the motor to rotate to that position.

# 4 Results

## 4.1 Homogeneous transformation matrix

To find the homogeneous transformation matrix Link length and angle (θ) is needed the length of all 3- Links are given below, and the angles are given that is used to pick the object.

$L_1 = 1.2\text{cm} \qquad \theta_1 = 90°$

$L_2 = 13.8\text{cm} \qquad \theta_2 = 75°$

$L_3 = 13.8\text{cm} \qquad \theta_3 = 0°$

Using equation (2.3) the following homogeneous transformation matrix are determined

$$T_3^0 = \begin{vmatrix} \left(\cos\theta_1 \cdot \cos\theta_2 \cdot \cos\theta_3\right) & \left(-\cos\theta_1 \cdot \sin\theta_2 \cdot \sin\theta_3\right) & \sin\theta_1 & \cos\theta_1 \cdot \left(L_3 \cdot \cos\theta_2 \cdot \cos\theta_3 + L_2 \cdot \cos\theta_2\right) \\ \left(\sin\theta_1 \cdot \cos\theta_2 \cdot \cos\theta_3\right) & \left(-\sin\theta_1 \cdot \sin\theta_2 \cdot \sin\theta_3\right) & -\cos\theta_1 & \sin\theta_1 \cdot \left(L_3 \cdot \cos\theta_2 \cdot \cos\theta_3 + L_2 \cdot \cos\theta_2\right) \\ \left(\sin\theta_2 \cdot \sin\theta_3\right) & \left(\cos\theta_2 \cdot \cos\theta_3\right) & 0 & \left(L_3 \cdot \sin\theta_2 \cdot \sin\theta_3 + L_2 \cdot \sin\theta_2\right) \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_1^0 = \begin{vmatrix} \cos 90° & 0 & \sin 90° & 0 \\ \sin 90° & 0 & \cos 90° & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_2^1 = \begin{vmatrix} \cos 75° & -\sin 75° & 0 & (0.138 \cdot \cos 75°) \\ \sin 75° & \cos 75° & 0 & (0.138 \cdot \sin 75°) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 0.25 & -0.966 & 0 & 0.0345 \\ 0.966 & 0.25 & 0 & 0.133 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_3^2 = \begin{vmatrix} \cos 0° & -\sin 0° & 0 & 0.138 \cdot \cos 0° \\ \sin 0° & \cos 0° & 0 & 0.138 \cdot \sin 0° \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0.138 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_3^0 = \begin{vmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 0.25 & -0.966 & 0 & 0.0345 \\ 0.966 & 0.25 & 0 & 0.133 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0.0138 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_3^0 = \begin{vmatrix} 0 & 0 & 1 & 0 \\ 0.25 & -0.966 & 0 & 0.07 \\ 0.966 & 0.25 & 0 & 0.266 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

## 4.2 Jacobian

By using the equation (2.5) spatial velocity will be determined. The joint velocity is

$$j=\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}-a_2\sin(\theta_1+\theta_2)-a_1\sin\theta_1-a_3\sin(\theta_1+\theta_2+\theta_3) & -a_2\sin(\theta_1+\theta_2)-a_3\sin(\theta_1+\theta_2+\theta_3) & -a_3\sin(\theta_1+\theta_2+\theta_3)\\ a_2\cos(\theta_1+\theta_2)+a_1\cos\theta_1+a_3\cos(\theta_1+\theta_2+\theta_3) & a_2\cos(\theta_1+\theta_2)-a_3\cos(\theta_1+\theta_2+\theta_3) & a_3\cos(\theta_1+\theta_2+\theta_3)\\ 1 & 1 & 1\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{bmatrix}$$

$$\dot{\theta} = \left|\, 6.16 \;\; 6.16 \;\; 6.16 \,\right|^{T}$$

Which is found from the data sheet [41] of the servo motor angle used for the picking the object is

$$\dot{\theta}_1 = 90°$$

$$\dot{\theta}_2 = 75°$$

$$\dot{\theta}_3 = 0°$$

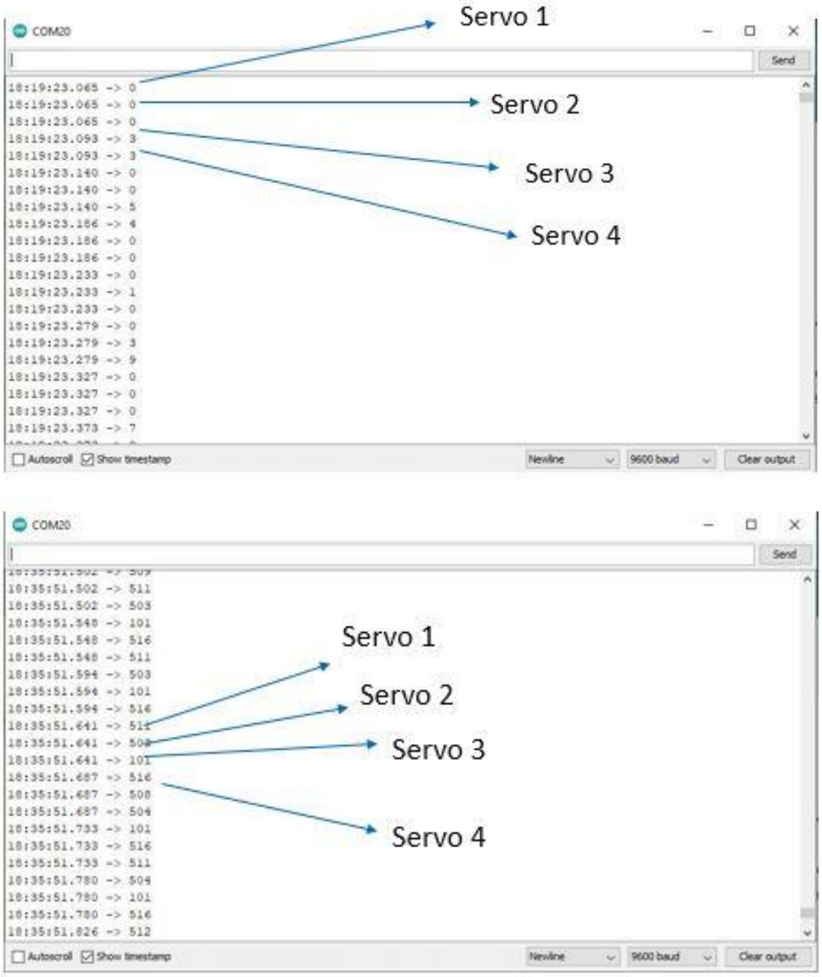
$$\nu = \mathrm{J}\cdot\dot{\theta}$$

$$\mathrm{J} = \begin{bmatrix}-0.83 & -0.071 & -0.035\\ -0.266 & -0.266 & -0.133\\ 1 & 1 & 1\end{bmatrix}\cdot\begin{bmatrix}6.16\\ 6.16\\ 6.16\end{bmatrix}$$

$$\nu = \begin{bmatrix}-5.75\\ -4.0964\\ 18.48\end{bmatrix}\mathrm{rad/sec}$$

One of the outcomes of robotics techniques is the robotic arm, which is made up of a variety of complicated and structured designs. The goal of this thesis is to build and develop a mini robot manipulator that can pick and place objects from one location to another using various strategies. The Arduino Leonardo is used to operate the Robot arm manually using a potentiometer as well as for automatic movement, there are two more control mode is performed to see which one performs best in this situation. where the gripper is attached with Arduino Uno an IR sensor and a Bluetooth module is added to control the whole robot arm by using smart phone. While the gripper is connected with IR sensor it will automatically open and close when any object is detected.

## 4.3 Controlling the Robot arm manually by using potentiometers



```
18:19:23.065 -> 0
18:19:23.065 -> 0
18:19:23.065 -> 0
18:19:23.093 -> 3
18:19:23.093 -> 3
18:19:23.140 -> 0
18:19:23.140 -> 0
18:19:23.140 -> 5
18:19:23.186 -> 4
18:19:23.186 -> 0
18:19:23.186 -> 0
18:19:23.233 -> 0
18:19:23.233 -> 1
18:19:23.233 -> 0
18:19:23.279 -> 0
18:19:23.279 -> 3
18:19:23.279 -> 9
18:19:23.327 -> 0
18:19:23.327 -> 0
18:19:23.327 -> 0
18:19:23.373 -> 7
```

□ Autoscroll ☑ Show timestamp

```
18:35:51.502 -> 509
18:35:51.502 -> 511
18:35:51.502 -> 503
18:35:51.548 -> 101
18:35:51.548 -> 516
18:35:51.548 -> 511
18:35:51.594 -> 503
18:35:51.594 -> 101
18:35:51.594 -> 516
18:35:51.641 -> 511
18:35:51.641 -> 503
18:35:51.641 -> 101
18:35:51.687 -> 516
18:35:51.687 -> 508
18:35:51.687 -> 504
18:35:51.733 -> 101
18:35:51.733 -> 516
18:35:51.733 -> 511
18:35:51.780 -> 504
18:35:51.780 -> 101
18:35:51.780 -> 516
18:35:51.826 -> 512
```

□ Autoscroll ☑ Show timestamp

*Figure 4.1. Servo motors initial position and picking position using potentiometers*

From the first figure initially the all the servo position/angle is 0 which can be found from the Arduino analog value.

In the second figure each servo motor angle changes when the motor turns. When the Robot picks up the object from spot 1, and the analog value can be seen using the Arduino IDE program.

From analog values the angle of the servo can be determined. The maximum analog value for the potentiometer is 1023. For the analog value of 1023 the rotation of the servo motor will be 180 degrees. By doing Unitary method the angle of each servo motor can be determined

$$\begin{aligned} &1023 \rightarrow 180^\circ \\ &1 \quad\;\; \rightarrow \frac{180^\circ}{1023} \\ &511 \;\; \rightarrow \frac{180^\circ}{1023} \cdot 511 = 89.9^\circ \approx 90^\circ \end{aligned}$$

By performing this calculation, the angle of servo motor 1 can be obtained, and by repeating the process for the remaining servo motors, the angles of all the remaining servo motors can be determined.

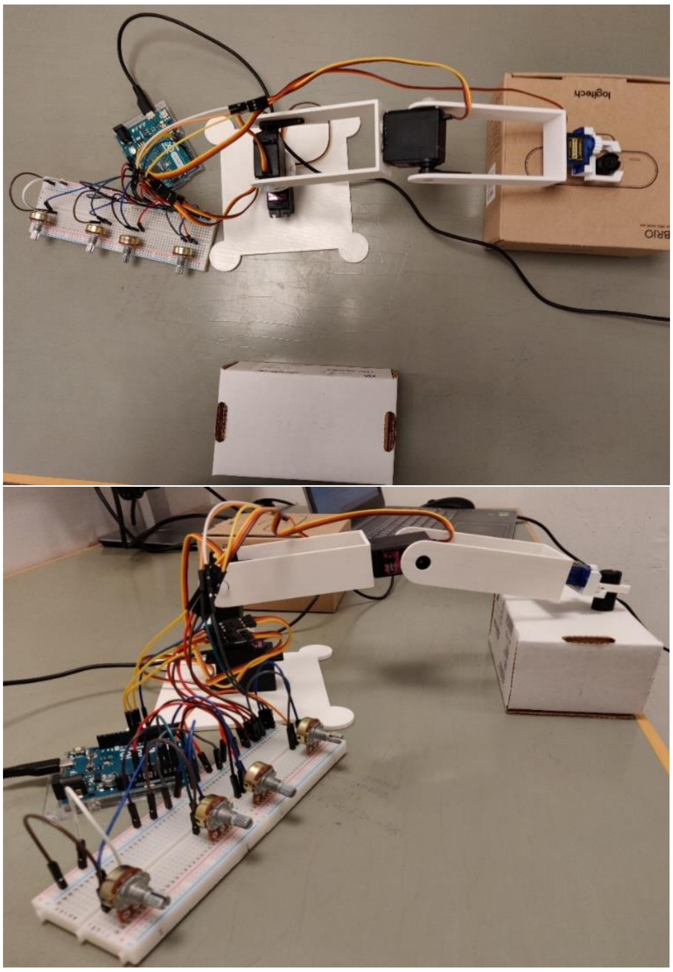


*Figure 4.2. Picking object from location 1 to location 2 using potentiometers*

The first figure shows all the servo motor position is control manually. depicts the first way, which involves the use of a potentiometer and manual control of the servo motor to pick the object from location 1.

The second figure shows the picked object is placed to location 2 by using potentiometer which is being operated manually.

After picking the object from location 1 the object is placed in location 2 and the resulting graph is shown by using the serial plotter of Arduino IDE software.

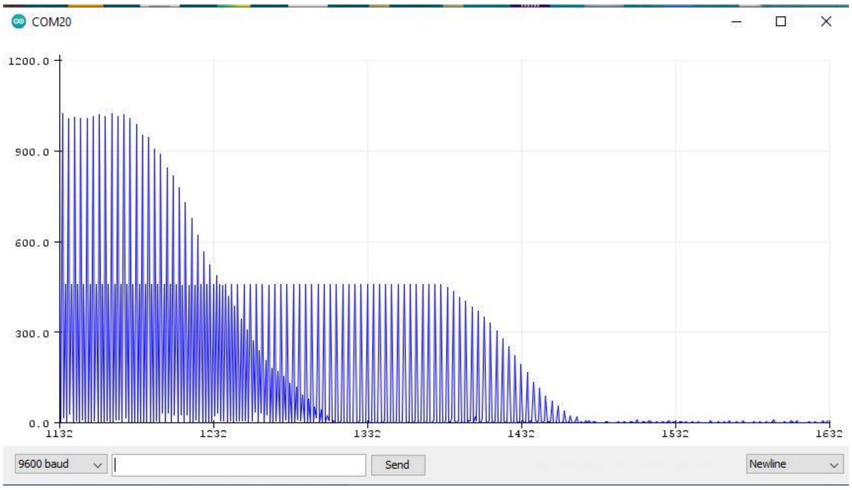


*Figure 4.3. serial plotter graph of placing the object to location 2 by using potentiometers*

## 4.4 Automatic operation of the robot arm

All of the servo motors will work automatically in this operation, with no human assistance. Here the servo is connected with the PWM pin of Arduino Leonardo and the Arduino is powered by external power source.

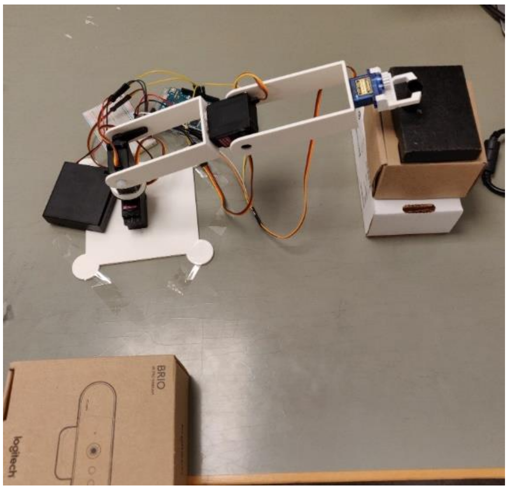


*Figure 4.4. picking the object from location 1 by automatic operation*

## 4.5 Controlling the gripper of the robot by using IR sensor

In this section of the work, all three servos are connected to an Arduino Leonardo, which will run automatically, and the micro servo is connected to an IR sensor, which is connected to Arduino UNO. The IR sensor will control the gripper here, and if the IR sensor detects any object close by, the gripper will automatically close to hold that object.

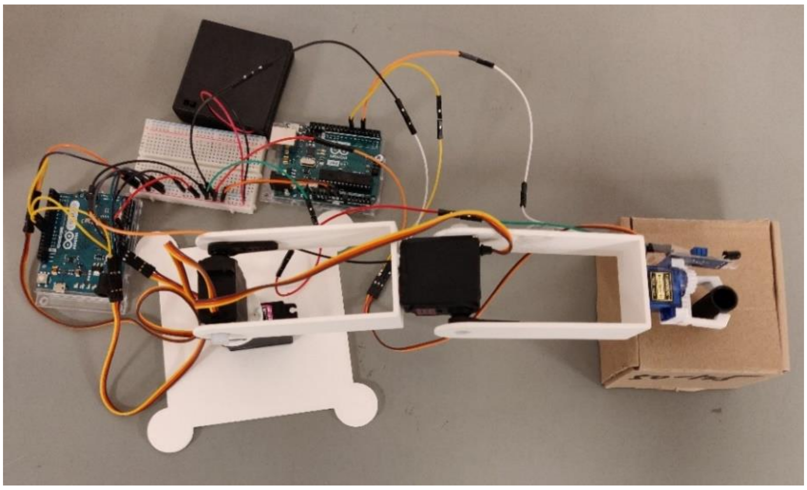
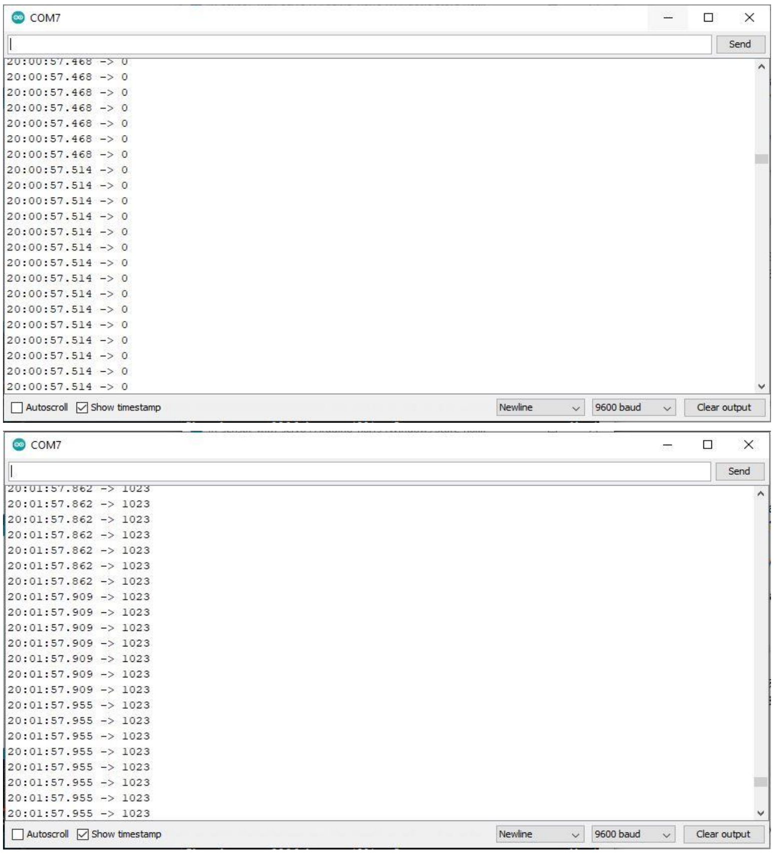


*Figure 4.5. Controlling the Gripper by using IR sensor*



```
20:00:57.468 -> 0
20:00:57.468 -> 0
20:00:57.468 -> 0
20:00:57.468 -> 0
20:00:57.468 -> 0
20:00:57.468 -> 0
20:00:57.468 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
20:00:57.514 -> 0
```

```
20:01:57.862 -> 1023
20:01:57.862 -> 1023
20:01:57.862 -> 1023
20:01:57.862 -> 1023
20:01:57.862 -> 1023
20:01:57.862 -> 1023
20:01:57.862 -> 1023
20:01:57.909 -> 1023
20:01:57.909 -> 1023
20:01:57.909 -> 1023
20:01:57.909 -> 1023
20:01:57.909 -> 1023
20:01:57.909 -> 1023
20:01:57.909 -> 1023
20:01:57.955 -> 1023
20:01:57.955 -> 1023
20:01:57.955 -> 1023
20:01:57.955 -> 1023
20:01:57.955 -> 1023
20:01:57.955 -> 1023
20:01:57.955 -> 1023
20:01:57.955 -> 1023
```

*Figure 4.6. IR sensor value while detecting the object*

In the first figure when the infrared sensor detects no item, it sends a signal to the Arduino UNO microcontroller, which shows 0 on the serial monitor.

Here it can be seen that when the IR sensor found any object close to it a signal value will be shown on serial monitor of Arduino IDE software when the IR sensor detects the object, it will show the value 1023.

## 4.6 Controlling the Robot arm by using smartphone connected with the Bluetooth module

In this stage all the 3 servo motors are connected with the Arduino Leonardo, and it will be operating automatically. The entire system is controlled by a smartphone through Bluetooth, and the system is powered by an external power supply and a USB cable connection to a computer.

# 5 Discussion

This thesis work is a real case of study, from here a different design and construction techniques were learned. The first task was to calculate the maximum link length the servo can carry followed by design the robot's by using Autodesk Inventor 2021 to construct the primary mechanical elements, which included the robot's linkages, base, and gripper according to particular standards. The third purpose was to combine mechanism, electronic, and computer control in order to build a working robotic prototype. Components such as mechanisms, structures, actuators, drivers, power sources, wiring, microprocessors, peripherals, and real-time programming were all included in the scope of topics required for constructing the actual robotic device. The fourth goal was to create an application for the Bluetooth module so that it can be controlled by using smartphone. The implementation of kinematics was the fifth target. Concepts were introduced here by starting with mathematical fundamentals like matrix algebra and trigonometry. The principles of joint rotations, translations, and trajectory computation were introduced. The next goal was to put together the entire experimental robotic system and get it moving. A connection between the PC and the Arduino microcontroller was formed. To test robot performance, multiple sets of controlling modes were applied for picking and placing the object from one place to another.

By observing the outcome of the results, it can be said that the automatic operations of the 3-link mini robot manipulator were better than the other three techniques. Because all the work were done automatically without human interaction.

Controlling the 3-Link mini robot manipulator was also satisfying as the direction of the motor can be controlled by using the potentiometer and can pick the object from various places and can place the object in different place.

The other two techniques were not so satisfying when it comes with the IR sensor and Bluetooth module. When using the IR sensor across the gripper the gripper will close when sensor detects any object. When the IR sensor detects the object, the gripper shuts and grabs the object however, when the object is placed, the gripper does not open to release the object since the IR sensor is still detecting the object. This is because due to inappropriate placing of the IR sensor.

Bluetooth module was used to control the whole Robot arm by smartphone but all of the servo was not controllable by smartphone as the MG966r servo is a bit heavy to handle. As a result, an additional power source, as well as the USB cable, are attached to Arduino. As a result, using an additional power source allowed the servo motors to rotate while being controlled over Bluetooth. Employing micro servos to improve the model will yield better results because they are lighter and consume less power than MG966r servo motors.

# 6 Conclusions

Through this thesis work the goal was accomplished from here the connections between theoretical and practical features of real-time robot manipulator motions. Different C program were carried out by using Arduino to evaluate the performance of the prototype. Pick, lift and position the object from one spot to another using a 3-link mini robot manipulator that was designed and developed. Pick and place procedures of various types were employed to compare which technique produced the best results. One of most important challenges faced while doing this work is to maintain a stable connection with the system as lots of connecting wires were used. Loose connections caused numerous issues, particularly with potentiometers. While carrying out this work a good knowledge has been gathered about building a whole structure and fabricating all of the parts from scratch. As it is a prototype and the proposed model is demonstrated through the use of a real-world scenario. And this can be employed in different industrial and educational purposes.

## 6.1 Future work

As it is a prototype in future it could be in future a fully developed robot model which may be used with stronger servo motors so that it can carry heavy object. Finally, more integrated circuits will be developed to eliminate the problem of loose connections.

# References

[1] Yılmaz, Kaan & Berkmen, Ekrem & Yurdakul, Boray. (2017). Design and Manufacturing of a 4 Degrees of Freedom Robotic Arm. 10.13140/RG.2.2.28339.32808.

[2] Craig, J. Introductzon to Robotzcs. Addison Wesley, 1989.

[3] "Kinematics," Illinois.edu. [Online]. Available: http://motion.cs.illinois.edu/RoboticSystems/Kinematics.html

[4] A. A. Mohammed and M. Sunar, "Kinematics modeling of a 4-DOF robotic arm," 2015 International Conference on Control, Automation and Robotics, 2015, pp. 87-91, doi: 10.1109/ICCAR.2015.7166008.

[5] A. Fredmer, "Application and Control of Robotic Manipulator through PLC," Dissertation, 2017.

[6] "Jacobian," Home. [Online]. Available: https://www.rosroboticslearning.com/jacobian

[7] "Kinematics," Illinois.edu. [Online]. Available: http://motion.cs.illinois.edu/RoboticSystems/Kinematics.html

[8] R. Featherstone, "Robot dynamics," Scholarpedia J., vol. 2, no. 10, p. 3829, 2007.

[9] C. Westerberg and M. Nore, "Robotic Arm controlled by Arm Movements MIKO NORE," Diva-portal.org. [Online]. Available: http://kth.diva-portal.org/smash/get/diva2:1373883/FULLTEXT01.pdf

[10] "MG996R servo motor," Components101. [Online]. Available: https://components101.com/motors/mg996r-servo-motor-datasheet

[11] "Torque," Khan Academy. [Online]. Available: https://www.khanacademy.org/science/physics/torque-angular-momentum/torque-tutorial/a/torque

[12] "Inventor," Autodesk.eu, 16-Jul-2021. [Online]. Available: https://www.autodesk.eu/products/inventor/overview?term=1-YEAR&tab=subscription

[13] "Stratasys F123 Series 3D printers," Stratasys. [Online]. Available: https://www.stratasys.com/3d-printers/f123-series

[14] T. Rogers, "Everything you need to know about ABS plastic," Creativemechanisms.com. [Online]. Available: https://www.creativemechanisms.com/blog/everything-you-need-to-know-about-abs-plastic

[15] "Arduino - ArduinoBoardLeonardo," Arduino.cc. [Online]. Available: https://www.arduino.cc/en/Main/Arduino_BoardLeonardo

[16] A. A. Galadima, "Arduino as a learning tool," 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), 2014, pp. 1-4, doi: 10.1109/ICECCO.2014.6997577.

[17] “MG996R servo motor,” Components101. [Online]. Available: https://components101.com/motors/mg996r-servo-motor-datasheet

[18] Ic.ac.uk. [Online]. Available: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

[19] “Bluetooth module (HC-06) + arduino,” Wustl.edu. [Online]. Available: https://classes.engineering.wustl.edu/ese205/core/index.php?title=Bluetooth_Module_(HC-06)_%2B_Arduino

[20] MACFOS, “IR sensor Working Principle and Applications,” Robu.in | Indian Online Store | RC Hobby | Robotics, 19-May-2020. [Online]. Available: https://robu.in/ir-sensor-working/

[21] “10K ohm potentiometer,” QuartzComponents. [Online]. Available: https://quartzcomponents.com/products/10k-potentiometer

[22] “Software,” Arduino.cc. [Online]. Available: https://www.arduino.cc/en/software

[23] “About us,” Mit.edu. [Online]. Available: https://appinventor.mit.edu/about-us

[24] “Servo Motor Basics with Arduino,” Arduino.cc. [Online]. Available: https://docs.arduino.cc/learn/electronics/servo-motors

[25] “Servo with Ir sensor,” ARDUINO PROJECTS, 29-Mar-2020. [Online]. Available: https://arduinoproyek.wordpress.com/servo-with-ir-sensor/

[26] “VarSpeedServo - a modified Servo library with speed control,” Arduino Forum, 18-May-2011. [Online]. Available: https://forum.arduino.cc/t/varspeedservo-a-modified-servo-library-with-speed-control/61404

# Appendix A

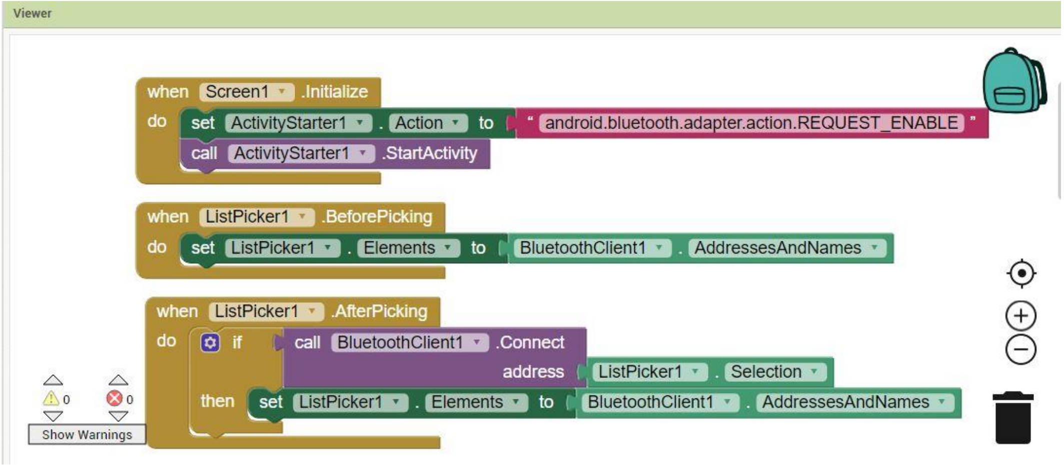
```
/*

 Controlling a servo motors position using a potentiometer [24].

*/

#include <Servo.h>

Servo servo1; // create servo object to control a servo1

Servo servo2; // create servo object to control a servo2

Servo servo3; // create servo object to control a servo3

Servo servo4; // create servo object to control a servo4

int potpin1 = A3; // analog pin used to connect the potentiometer1

int potpin2 = A2; // analog pin used to connect the potentiometer2

int potpin3 = A1; // analog pin used to connect the potentiometer3

int potpin4 = A0; // analog pin used to connect the potentiometer4

int val1; // variable to read the value from the analog pin A3

int val2; // variable to read the value from the analog pin A2

int val3; // variable to read the value from the analog pin A1

int val4; // variable to read the value from the analog pin A0

void setup () {

  Serial. Begin (9600);

  servo1.attach(9); // attaches the servo1 on pin 9 to the servo object

  servo2.attach(10); // attaches the servo2 on pin 10 to the servo object

  servo3.attach(8); // attaches the servo3 on pin 8 to the servo object

  servo4.attach(7); // attaches the servo4 on pin 7 to the servo object

}

void loop () {

  val1 = analogRead(potpin1); // reads the value of the potentiometer1 (value between 0 and 1023)

  Serial.println(val1);

  val1 = map (val1, 0, 1023, 0, 180); // scale it for use with the servo1 (value between 0 and 180)

  servo1.write(val1); // sets the servo1 position according to the scaled value

 delay (15); // waits for the servo1 to get there
val2 = analogRead(potpin2);
  Serial.println(val2);
  val2 = map(val2, 0, 1023, 0, 180);
  servo2.write(val2);
  delay(15);
  val3 = analogRead(potpin3);
  Serial.println(val3);
  val3 = map(val3, 0, 1023, 0, 180);
  servo3.write(val3);
  delay(15);
  val4 = analogRead(potpin4);
  Serial.println(val4);
  val4 = map(val4, 0, 1023, 0, 180);
  servo4.write(val4);
  delay (15);
}
```

# Appendix B

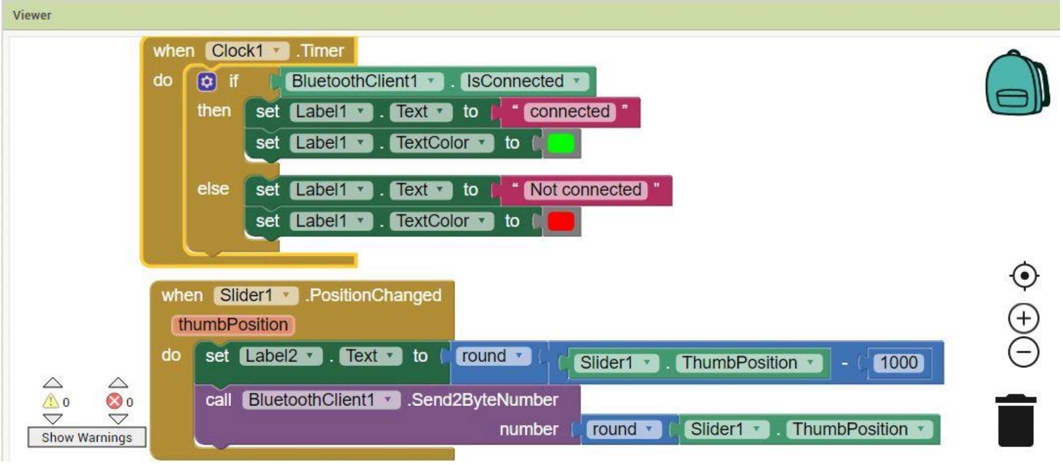
```
/*
 Controlling gripper with IR sensor [25]
*/
#include<Servo.h> // servo motor library
Servo motor; // Name servo motor

void setup () {
pinMode (7, INPUT); // Control pin of servo motor attached with PWM pin 7
pinMode (4, INPUT); // control pin of IR sensor attached with PWM pin 4
Serial.begin(9600);
motor.attach(7);
motor.write(130);
}
void loop() {
 if (digitalRead(4) ==LOW){
  Serial.println(1023);
motor.write(90);
}
if (digitalRead(4)==HIGH){
  Serial.println(0);
motor.write(0);

}

}
```

# Appendix C

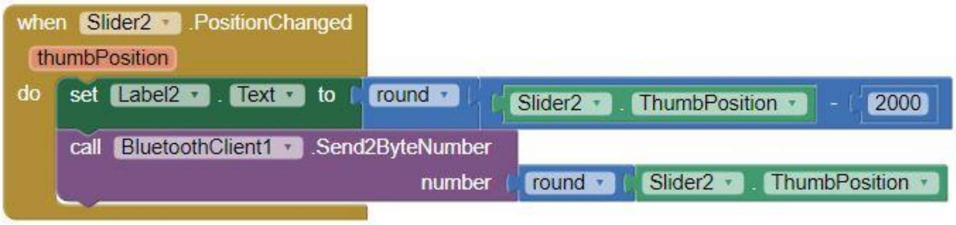
MIT app inventor blocks for making Bluetooth application for controlling the servo motor



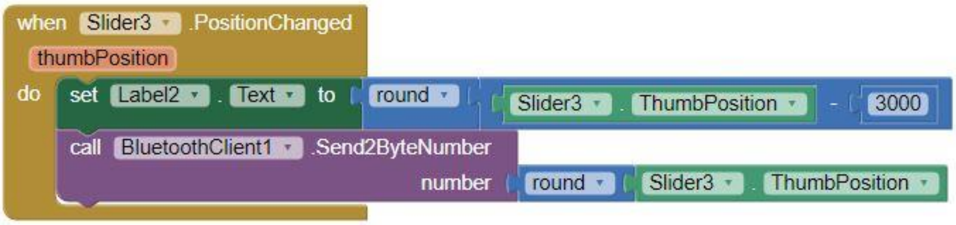
```
when Screen1 .Initialize
do  set ActivityStarter1 . Action to " android.bluetooth.adapter.action.REQUEST_ENABLE "
    call ActivityStarter1 .StartActivity

when ListPicker1 .BeforePicking
do  set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do  if   call BluetoothClient1 .Connect
                             address ListPicker1 . Selection
    then set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
```



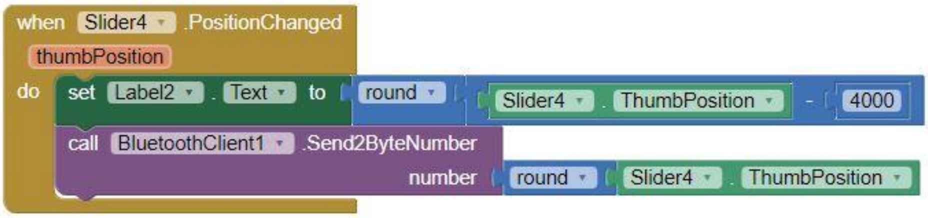
```
when Clock1 .Timer
do  if   BluetoothClient1 . IsConnected
    then set Label1 . Text to " connected "
         set Label1 . TextColor to 🟩
    else set Label1 . Text to " Not connected "
         set Label1 . TextColor to 🟥

when Slider1 .PositionChanged
 thumbPosition
do  set Label2 . Text to round Slider1 . ThumbPosition - 1000
    call BluetoothClient1 .Send2ByteNumber
                          number round Slider1 . ThumbPosition
```



```
when Slider2 .PositionChanged
  thumbPosition
do  set Label2 . Text to  round ( Slider2 . ThumbPosition - 2000 )
    call BluetoothClient1 .Send2ByteNumber
                            number  round ( Slider2 . ThumbPosition )
```



```
when Slider3 .PositionChanged
  thumbPosition
do  set Label2 . Text to  round ( Slider3 . ThumbPosition - 3000 )
    call BluetoothClient1 .Send2ByteNumber
                            number  round ( Slider3 . ThumbPosition )
```



```
when Slider4 .PositionChanged
  thumbPosition
do  set Label2 . Text to  round ( Slider4 . ThumbPosition - 4000 )
    call BluetoothClient1 .Send2ByteNumber
                            number  round ( Slider4 . ThumbPosition )
```

# Appendix D

```
// Servo Motor automatic control [26]

#include <VarSpeedServo.h>


VarSpeedServo servo1;  // create servo object to control a servo
VarSpeedServo servo2;
VarSpeedServo servo3;
VarSpeedServo servo4;



void setup() {
  servo1.attach(9);  // attaches the servo with pin 9 to the servo object
  servo1.write(0,10,true); // set the intial position of the servo1, wait until done
  servo2.attach(10);  // attaches the servo with pin 10 to the servo object
  servo2.write(90,10,true); // set the intial position of the servo2, wait until done
  servo3.attach(8); // attaches the servo with pin 8 to the servo object
  servo3.write(45,10,true); // set the initial position of the servo3, wait until done
  servo4.attach(7); // attaches the servo with pin 7 to the servo object
  servo4.write(0,10,true); // set the initial position of the servo4, wait until done
}
void loop() {
  servo1.write(90,10,true);  //Servo1 move 90 degree with speed 10
  servo2.write(75,10,true);  //Servo2 move 75 degree with speed 10
  servo3.write(0,10,true);   //Servo3 move 0 degree with speed 10
  servo4.write(160,10,true); //Servo4 move 160 degree with speed 10 to grab the object from location 1
  servo1.write(0,10,true);   //Servo1 move 0 degree with speed 10 from location 1 to location 2
  servo2.write(90,10,true);  //Servo2 move 90 degree with speed 10
  servo4.write(0,10,true);   //Servo4 move 0 degree with speed 10 to drop the object to location 2
  servo3.write(45,10,true);  //Servo3 coming back to the initial position
  servo2.write(0,10,true);   //Servo2 move 0 degree to come back to its initial position
}
```

```
//* Servo Motor controlled by smartphone via Bluetooth connection

Software serial multiple serial test

Receives from the hardware serial, sends to software serial.

Receives from software serial, sends to hardware serial [58]. //*

#include <SoftwareSerial.h> // TX RX software library for bluetooth

#include <wiring_private.h>

#include <Servo.h> // servo library

Servo myservo1, myservo2, myservo3, myservo4; // servo name


int bluetoothTx = 0; // bluetooth tx to 0 pin

int bluetoothRx = 1; // bluetooth rx to 1 pin

SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

void setup()

{

  myservo1.attach(9); // attach servo signal wire to pin 9

  myservo2.attach(10); // attach servo signal wire to pin 10

  myservo3.attach(8); // attach servo signal wire to pin 8

  myservo4.attach(7); // attach servo signal wire to pin 7

  Serial.begin(9600); //Setup usb serial connection to computer

  bluetooth.begin(9600); //Setup Bluetooth serial connection to android

}

void loop ()

{

  if(bluetooth.available()>= 2 ) //Read from Bluetooth and write to usb serial

  {

    unsigned int servopos = bluetooth.read();

    unsigned int servopos1 = bluetooth.read();

    unsigned int realservo = (servopos1 *256) + servopos;

    Serial.println(realservo);
if (realservo >= 1000 && realservo <1180) {
   int servo1 = realservo;
   servo1 = map (servo1, 1000, 1180, 0, 180);
   myservo1.write(servo1);
   Serial.println("Servo 1 ON");
   delay (10);
  }
  if (realservo >= 2000 && realservo <2180) {
   int servo2 = realservo;
   servo2 = map (servo2, 2000, 2180, 0, 180);
   myservo2.write(servo2);
   Serial.println("Servo 2 ON");
   delay (10);
  }
  if (realservo >= 3000 && realservo <3180) {
   int servo3 = realservo;
   servo3 = map (servo3, 3000, 3180, 0, 180);
   myservo3.write(servo3);
   Serial.println("Servo 3 ON");
   delay(10);
  }
  if (realservo >= 4000 && realservo <4180) {
   int servo4 = realservo;
   servo4 = map (servo4, 4000, 4180, 0, 180);
   myservo4.write(servo4);
   Serial.println("Servo 4 ON");
   delay (10);
  }}
}
```