EFFICENT ROBOTIC ARM WITH SMARTPHONE CONTROL

A PROJECT REPORT

Submitted by

SANYATJEET PAWDE [RA1711004010048]
ROHAN TEHALYANI [RA1711004010050]
MITRANSH CHOUBISA [RA1711004010058]
MAYANK KUMAR [RA1711004010064]

Under the guidance of
Dr. DAMODAR PANIGRAHY,Ph.D
(Assitant Professsor, Department of Electronics and Communication Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

OCT 2019

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**EFFICENT ROBOTIC ARM WITH SMARTPHONE CONTROL**" is the bonafide work of "**SANYATJEET PAWDE [RA1711004010048], ROHAN TEHALYANI [RA1711004010050],MITRANSH CHOUBISA[RA1711004010058] ,MAYANK KUMAR [RA1711004010064],** ", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. DAMODAR PANIGRAHY,Ph.D
GUIDE
Assitant Professsor
Dept. of Electronics and Communication Engineering

SIGNATURE

Dr ESWARAN P,Ph.D
CO-ORDINATOR
Associate Professor
Dept. of Electronics and Communication Engineering

ABSTRACT

Robotic arm is a analogical device used to reduce the human effort . Robot arm is based on the principle of movement of human arm . In this work, we have designed a robot arm to shift one device from one place to other place. Our designed robot arm has a compact console such as a mobile phone which controls the arm using bluetooth technology. The main advantage of our proposed robot arm is controlled by bluetooth so the size of console is feasible compared to existing technologies .

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to our guide, Dr. Damodar Panigrahy and to our co-ordinator Dr.Eswaran P .for his valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to me for completing this research work.

Project Group

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

DOF	Degrees Of Freedom
GUI	Graphical User Interface

LIST OF SYMBOLS

F	Force, N
T	Torque ,kgcm
L	length, mm
m	mass, gm
a	acceleration, m/(s)^2

CHAPTER 1

INTRODUCTION

1.1 Robotic Arm

Industry-specific robots perform several tasks such as picking and placing objects, and movement adapted from observing how similar manual tasks are handled by a fully-functioning human arm. Such robotic arms are also known as robotic manipulators. These manipulators were originally used for applications concerning bio-hazardous or radioactive materials or use in inaccessible places.

A series of sliding or jointed segments are put together to form an arm-like manipulator that is capable of automatically moving objects within a given number of degrees of freedom. Every commercial robot manipulator includes a controller and a manipulator arm. The performance of the manipulator depends on its speed, payload weight and precision. However, the reach of its end-effectors, the overall working space and the orientation of the work is determined by the structure of the manipulator.

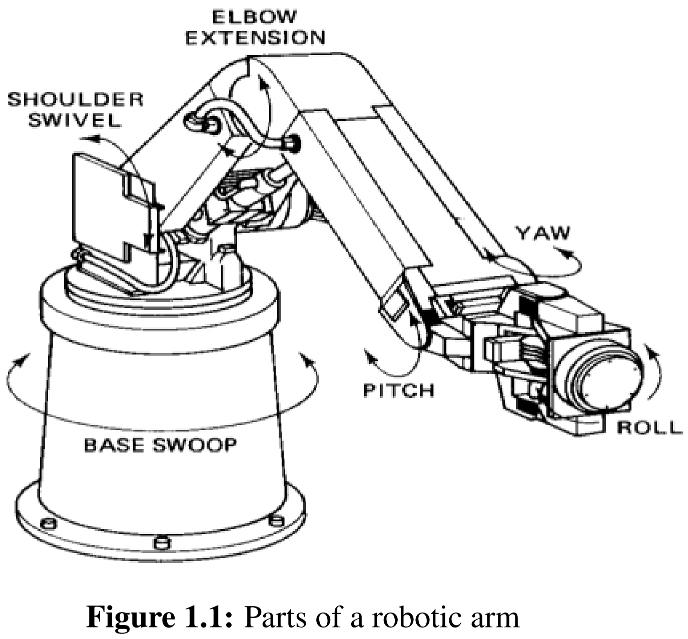


Figure 1.1: Parts of a robotic arm

Robotic manipulators are not a new idea and have already been in operation in many areas of manufacturing for several years. As AI advances, increasing the accuracy and function of robotics will allow a greater range of tasks to be achieved by such manipulators, thus providing more function and provision of labor than the robotic manipulators of the past decades.

1.1.1 Kinematics of a Robotic Arm

A robot manipulator is constructed using rigid links connected by joints with one fixed end and one free end to perform a given task, such as moving a box from one location to the next. The joints to this robotic manipulator are the movable components, which enables relative motion between the adjoining links. There are also two linear joints to this robotic manipulator that ensure non-rotational motion between the links, and three rotary type joints that ensure relative rotational motion between the adjacent links.[6][4]

The manipulator can be divided into two parts, each having different functions:

Arm and Body

The arm and body of the robot consist of three joints connected by large links. They can be used to move and place objects or tools within the workspace.

Wrist

The function of the wrist is to arrange the objects or tools at the workspace. The structural characteristics of a robotic wrist consist of two or three compact joints.

1.1.2 Robotic Manipulator Arm Configuration

Manipulators are grouped into several types based on the combination of joints, which are as follows:[9]

Cartesian geometry arm

This arm employs prismatic joints to reach any position within its rectangular workspace by using Cartesian motions of the links.

Cylindrical geometry arm

This arm is formed by the replacement of the waist joint of the Cartesian arm with a revolute joint. It can be extended to any point within its cylindrical workspace by using a combination of translation and rotation.

Polar/spherical geometry arm

When a shoulder joint of the Cartesian arm is replaced by a revolute joint, a polar geometry arm is formed. The positions of end-effectors of this arm are described using polar coordinates.

Articulated/revolute geometry arm

Replacing the elbow joint of the Cartesian arm with the revolute joint forms an articulated arm that works in a complex thick-walled spherical shell.

Selective compliance automatic robot arm (SCARA)

his arm has two revolute joints in a horizontal plane, which allow the arm to extend within a horizontal planar workspace.

1.1.3 Wrist Configuration

The two main types of wrist design include:

Roll-pitch-roll or spherical wrist

Pitch-yaw-roll

The spherical wrist is more common because of its mechanically simpler design. It has 6 degrees of freedom and consists of a Hooke shoulder joint followed by a rotary elbow joint.[1]

1.2 Objectives

To make an Arduino Robot Arm

Control it using a custom build Android application.

Program it using arduino ide to control the movement of the arm.

Automate it for the repetitive process.

To take inspiration form human hand and develop a model with similar degree of freedom.

Thereby making the controlling console , compact in size , user friendly and more feasible to use than the already existing technology.

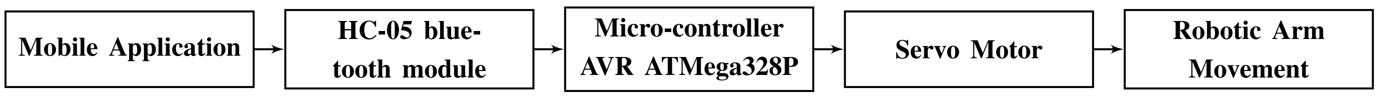


Figure 1.2: Overall Framework

1.3 Applications

Some of the major applications of robotic manipulators are presented below:Pachaiyappan et al. [8]

Motion planning

Remote handling

Teleoperation

Micro-robots

Humanoid robots

Machine tools

Space shuttle operations

Military EOD(Explosive Ordinance Disposal)

Medical applications, such as surgery

CHAPTER 2

LITERATURE SURVEY

2.1 Design of the arm

After going through all the papers mention in the reference the design of a typical manipulator should be similar to that of a **human hand**. The design must have the complete degree of freedom of both rotational and lateral. The design of robot should fulfil 5 Degrees Of Freedom (DOF) for pick and place application. It should consist of a base, shoulder, elbow, wrist joint and gripper. 'The of the structures should be manipulated according to the application of the robotic arm. The prime demand of the arm is to pick and place things but the requirement varies on the basis of the industry where it is applied. The selection of motors is based on the torque, which operates on 5V regulated supply. DC servo motor has faster cycle rate than stepper motor and lighter in weight compared to AC motor so we are using DC servo motors **MG996R and SG90**. [11]

2.2 Graphical User Interface

The basic aim is to design a Graphical User Interface (GUI) which can eliminate the big control station of the already available pick and place technology in the market and replace it with a simple user-friendly application. This can be done with the help of a normal GUI or a mobile application. The main aim of the GUI is to allow the user to give simple mechanism for controlling the robotic arm. This method is time saving and can be used for picking and placing the material over conveyor assembly. The motion of the arm is not completely closed loop system. User must have limited limb functionality to move their wrist or gripper. GUI takes the user input and move the joints accordingly. GUI can have slider option for giving inputs to the system. According to slider position angles value will be recorded and stored in the backend and transmitted to the micro-controller. After getting the angle values, the controller actuates the servo motors to desired angle and arm movement takes place.[10]

2.3 Design Analysis of a Remote Controlled Pick and Place Robotic Vehicle by B.O. Omijeh

In this paper The design of a Remote Controlled Robotic Vehicle has been completed. A prototype was built and confirmed functional. This system would make it easier for man to unrivalled the risk of handling suspicious objects which could be hazardous in its present environment and workplace. Complex and complicated duties would be achieved faster and more accurately with this design.[7]

2.4 ROBOTICS ARM CONTROL USING HAPTIC TECHNOLOGY by Vipul J. Gohil

In this paper the system robotic arm based on real-world haptics. The primary goals of haptic guidance is to facilitate the learning of complex human motion skills by providing haptic cues that are helpful to induce desired movements. The proposed system is utilized to recognize the human motion..Large potential for applications in critical fields as well as for leisurely pleasures. Haptic devices must be smaller so that they are lighter, simpler and easier to use.Haptic technology allows interactivity in real-time with virtual objects. [2]

2.5 DESIGN AND ANALYSIS OF AN ARTICULATED ROBOT ARM FOR VARIOUS INDUSTRIAL APPLICATIONS by S.Pachaiyappan

Using basic formulae from strength of materials. Two possible hollow cross sections, considering the electrical, control and feedback wiring to pass through, is modelled using commercially available 3D modelling tool, SolidWorks, for further study and comparison. The Model is used for analysis using a commercially available analysis tool, ANSYS, taking into account the various critical loads acting on the base arm alone. Since the base arm is the major component in which maximum magnitude of the critical loads considered occur, it is enough to analyze the base arm alone. Considering the shapes, sizes, deflections during working and stresses occurring, both the AIAs are workable comparatively. Considering the manufacturability, ease of transport, assembly, and weight, the circular section AIAs are preferred over the rectangular section AIAs. Pachaiyappan et al. [8]

2.6 Survey of Robotic Arm and Parameters by Ritu Tiwari

Based on this research worksDiscussed robotic arm and there different parameters. Understand which factor affect the performance of a robotic arm and how it change a robotic arm in work efficient arm. Know how multiple axis uses to change the mass of an arm, DOF increased by simply by adding joints, working envelope and space should decide according to the situation, kinematics improved movement of the robot, speed and acceleration vary in different works, accuracy and repeatability is the important factor for any robotic arm. Also, use diagrams for making proper understanding of robotic arm. Then discussed gaps in research and issues, its use as a guideline for future research work, at last give suggestions how we try to improve a robotic arm by working on effective algorithms and simulations.[9]

CHAPTER 3

SYSTEM ANALYSIS

Torque-Force Calculation of Model[5]

Torque is the twisting force which requires for the joint to move along the axis.

For the selection of motor, the Torque-force calculation is considered. T-F calculations are different at each joint. Torque is given by,

$$Torque = Force * Length \tag{3.1}$$

Acting force is,

$$Force = Mass * Acceleration \tag{3.2}$$

Acceleration due to gravity 9.80665 m/s2

Payload considered =250 gram (Max)

T-F calculation of Gripper Motor-D:

Total weight for gripper =125 gram

F1 = m1* a1 N

= 0.145 * 9.8

= 1.42 N.......... (1)

T1 = F1 * L1

= 1.42 N * 0.01m

= 0.14 kgcm........ (2)

T-F calculation of Motor A, B, C:

Total weight for Motor A, B, C = 250 gram

F2 = 0.25 * 9.8

= 2.45 N............ (3)

T2 = 2.45 N * 70m

= 1.75 kgcm.......... (4)

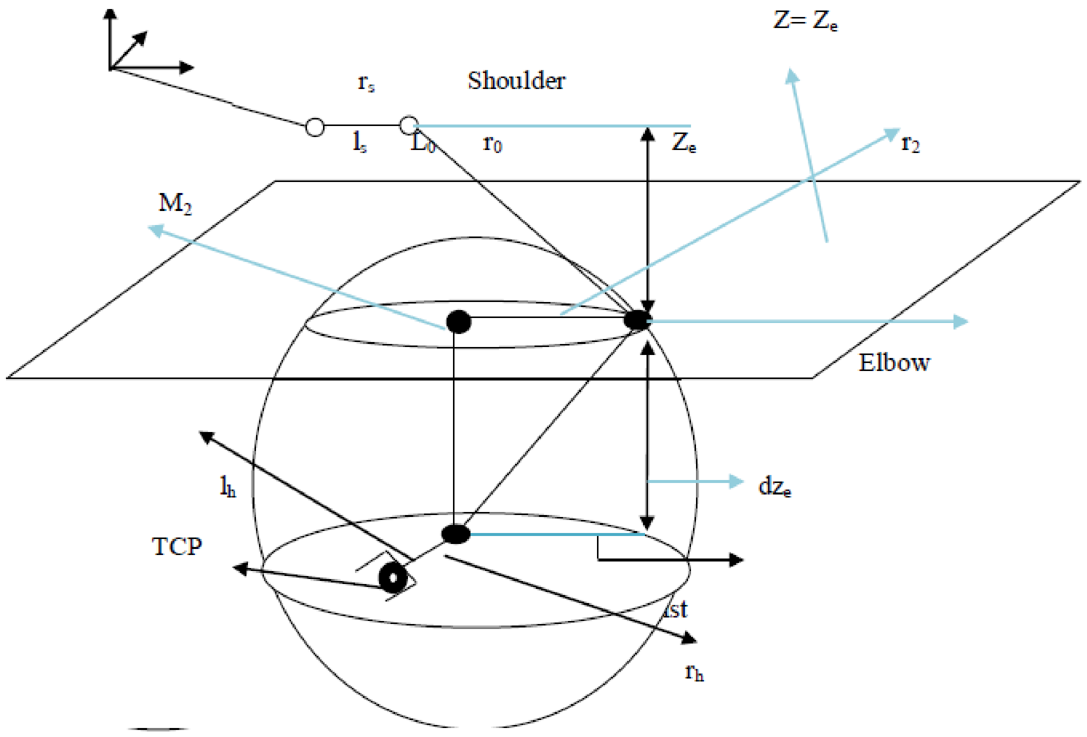


Figure 3.1: Plane Z = Ze and ellipsoid

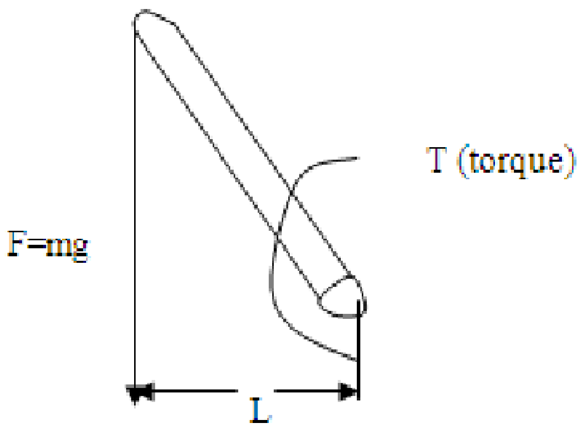


Figure 3.2: Torque Balance

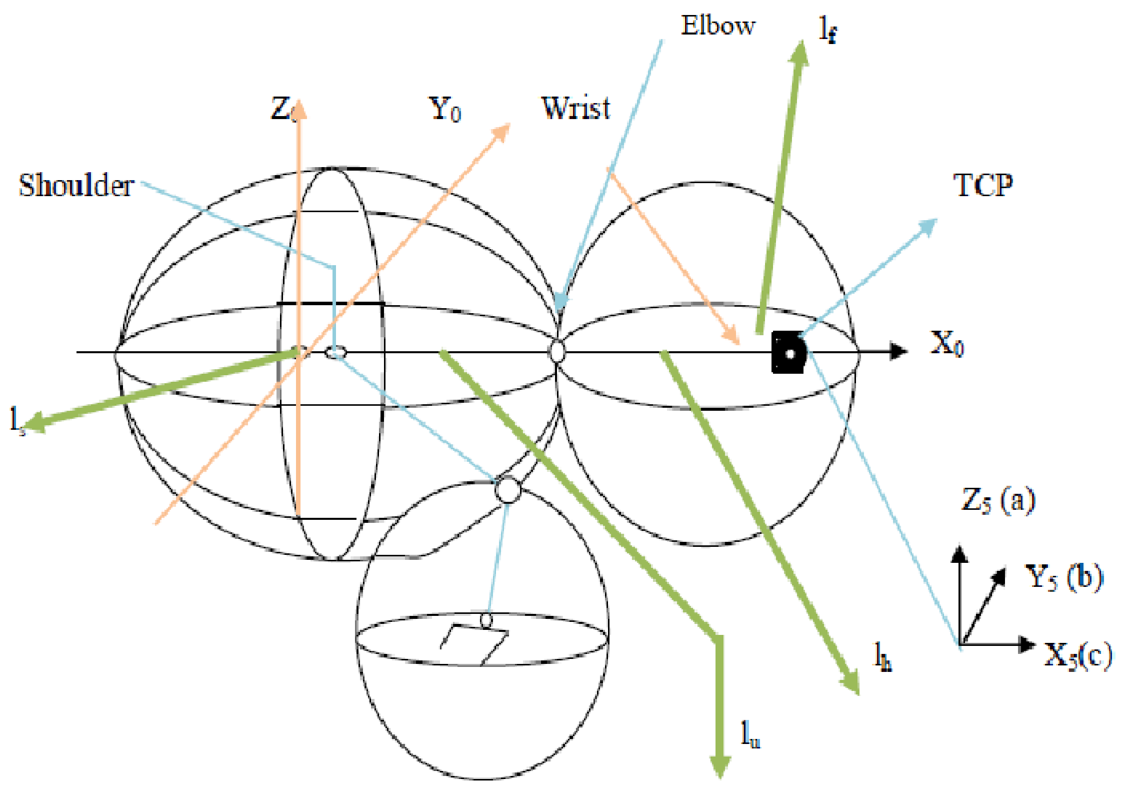


Figure 3.3: work space of the upper arm and forearm of this arm

CHAPTER 4

SYSTEM DESIGN

4.1 System Integrated Design

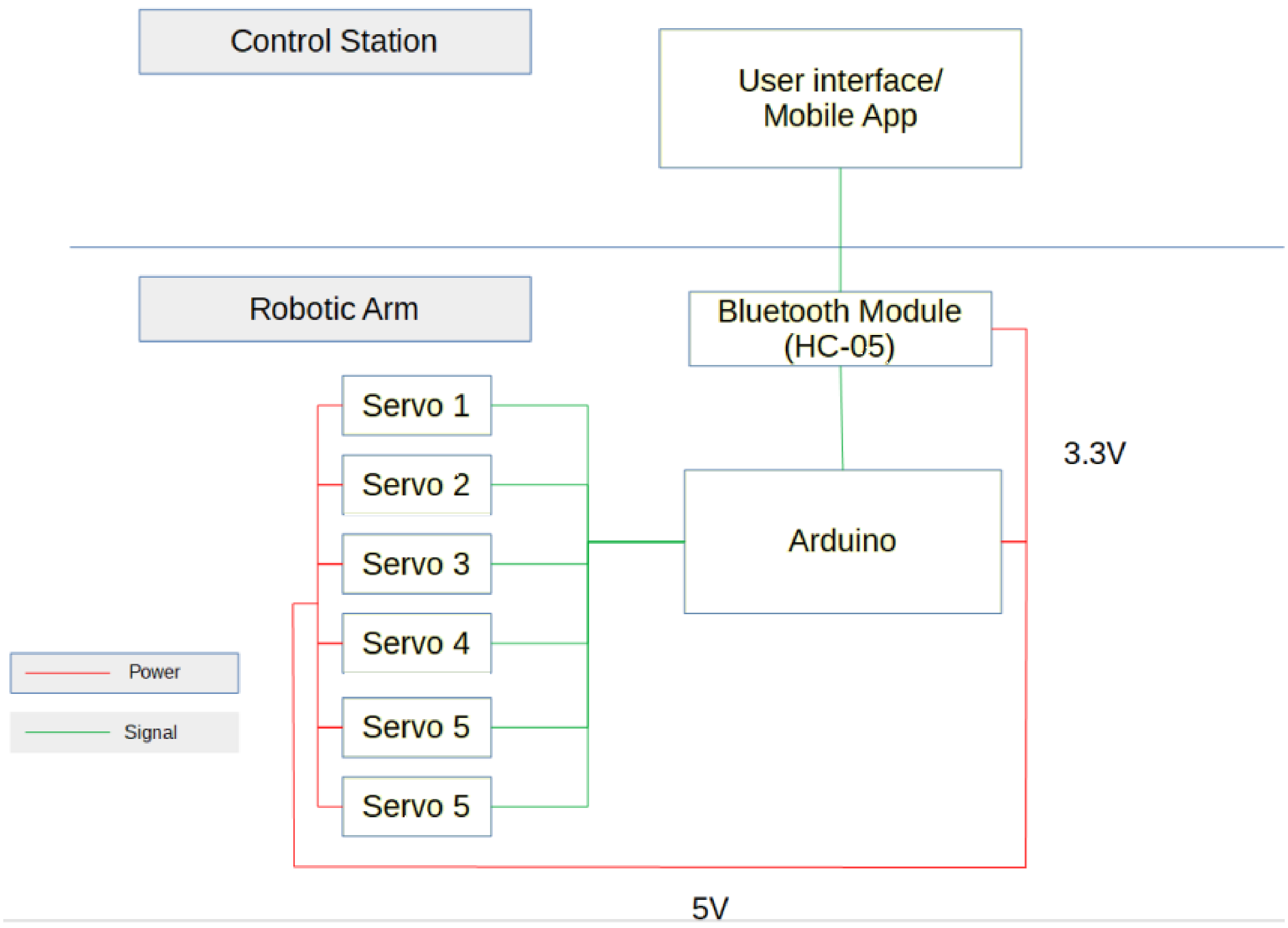


Figure 4.1: System Integrated Design

By considering the above approaches, to avoid sensor cost, huge training time, and programming skills we have developed 3D printed prototype having an easy operating methodology to train the robotic arm. So basically, the designed system is easy to handle, requires less time user-friendly that uses GUI for training purpose. Also system has the good adjustable response ability to pick the object of any shapes and size without touching the arm. The user who doesn't have programming skills and technical knowledge will also be able to train the robot easily. Graphical user interface (GUI) helps the operator to train robot efficiently within a short period of time. Also, an additional feature is that once you train the robot, in future you dont need to train it again and again for the same task. While training it will automatically save the trained angle of each DOF at backend. So,as per our need, we can directly use it without any repeated training. So we called it as a **Trainable Multipurpose Robotic Arm**. [3]

Table 4.1: 3D model Specifications

Name of the part	Motion	Dimensions	Infill	Weight	Torque
Base	rotation(swoop)	121*121*51 mm	60%	100gm	0.36kgcm
Shoulder	swivel(up and down)	97*97*67.8 mm	60%	50 gm	1.75kgcm
Elbow to Shoulder joint		46*166*21 mm	20%	40 gm	
Elbow	extension	38*115*27 mm	20%	33 gm	1.62kgcm
Wrist	pitch and yaw	33*46*280 mm	60 %	17 gm	0.6kgcm
Gripper joints	linear	8.5*20.4*65.3mm	60%	5 gm	0.14kgcm

.

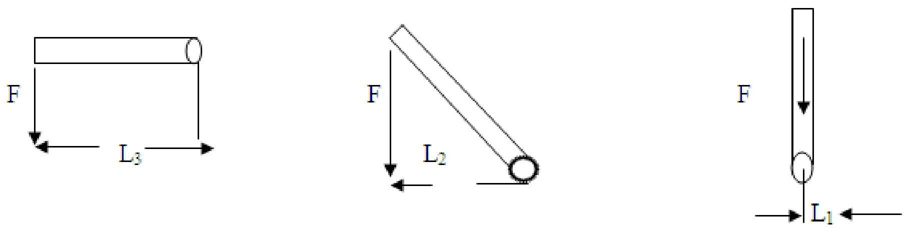


Figure 4.2: Required torque at each joint

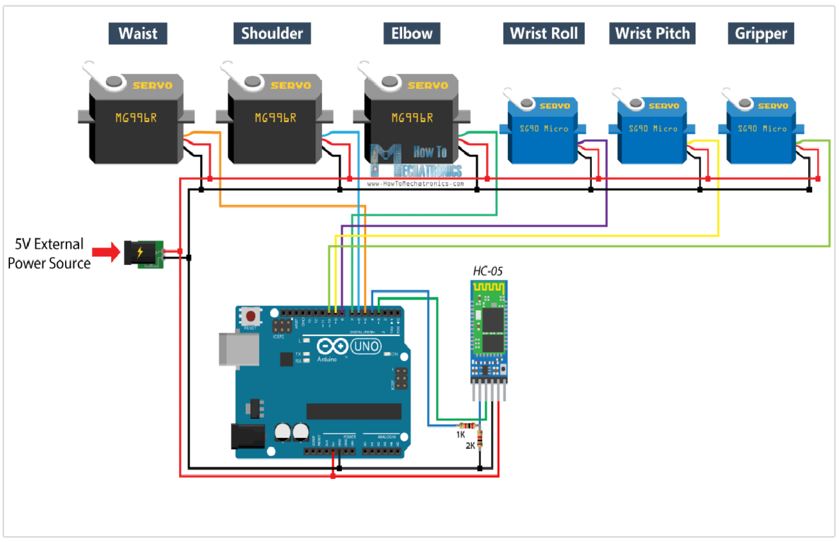


Figure 4.3: Prototype Model

4.1.1 Design Specifications (Servo Motors)

Main Body Servo(MG996R)

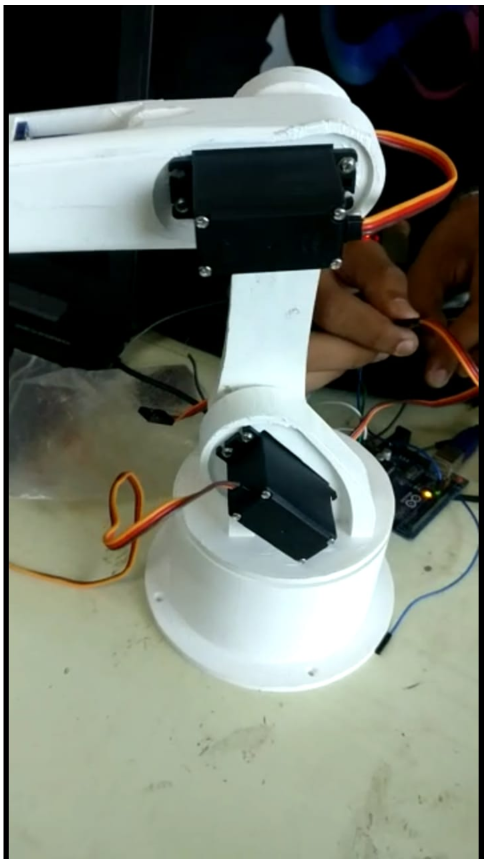


Figure 4.4: MG996R attached to shoulder and elbow

1. **Size** 38 x 11.5 x 24mm (Include tabs) 28 x 12.7 x 27mm (Not include tabs)
2. **Weight** 17g (Not include a cable and a connector) 18g (Include a cable and a connector)
3. **Speed** 0.14sec/60degrees (4.8V) 0.12sec/60degrees (6.0V)
4. **Torque** 2.5kgf-cm (4.8V) 3.0kgf-cm (6.0V)
5. **Voltage range** 4.8V-6.0V
6. **Connector type** JR type '(Yellow: Signal, Red: VCC, Brown:GND)'

Wrist and Gripper Servo(SG90)

1. **Size** 32 x 11.5 x 24mm (Include tabs) 23.5 x 11.5 x 24mm (Not include tabs)
2. **Weight** 8.5g (Not include a cable and a connector) 9.3g (Include a cable and a connector)
3. **Speed** 0.12sec/60degrees (4.8V) 0.10sec/60degrees (6.0V)

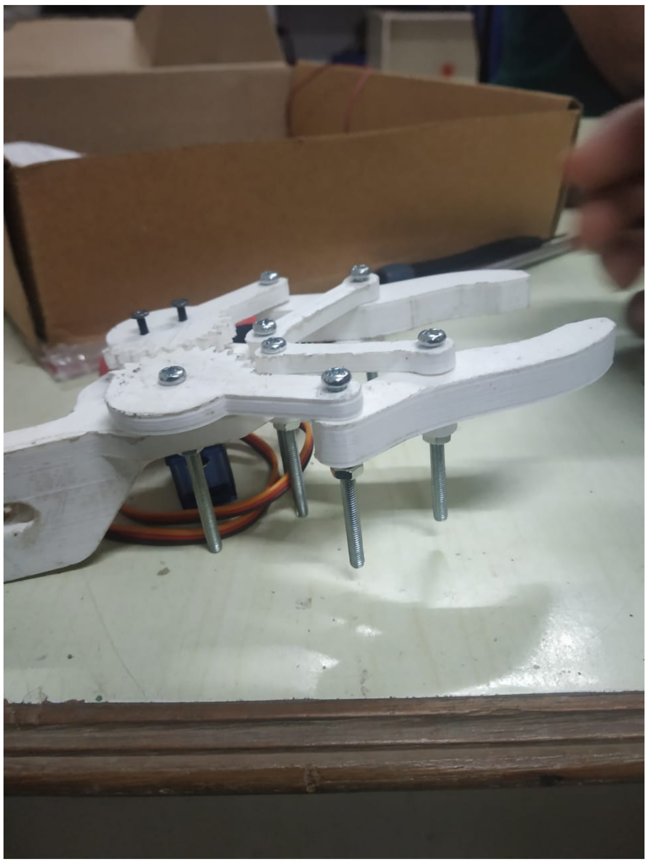


Figure 4.5: SG90 attached to gripper

4. **Torque** 1.5kgf-cm (4.8V) 2.0kgf-cm (6.0V)
5. **Voltage range** 4.8V-6.0V
6. **Connector type** JR type '(Yellow: Signal, Red: VCC, Brown:GND)'

4.1.2 MICROCONTROLLER Specifications

MICROCONTROLLER- Arduino UNO board with 8 bit ATmega328p AVR

1. **Power** Vin,3.3V,5V,GND pins for power I/O. operating voltage of 5V and recommended input of 7-12 volts. 50mA current is drawn.
2. **I/O pins** digital pins 0-13, analog pins(A0-A5 0-5V),PWM pins(3,5,6,9,11) for 8 bits pwm I/O.
3. **Serial pins** Rx(0);Tx(1); for transmission and reception of TTL serial data
4. **Microcontroller** ATmega 328p AVR 8 bit microcontroller high performance ,low power controller from Microchip previously manufactured by atmel.
5. **CPU** AVR CPU - Advanced Virtual RISC CPU RISC - reduced instruction set computer Alf-Egil Bogen Vegard Wollan RISC .

Figure 4.6: Arduino Uno

6. **Crystal Oscillator** used for square pulse generation as clock with 16MHz frequency. as it takes (1/16micro seconds) for 1 state
7. **EEPROM** 1Kb
8. **SRAM** 2Kb
9. **Flash Memory** 32Kb for storage of input code

4.1.3 Bluetooth Module Specifications

Bluetooth Module HC-05

1. **Power** (Vcc) Powers the module. Connect to +5V Supply voltage
2. **TX** Transmitter Transmits Serial Data.
3. **RX** Receiver Receive Serial Data.
4. **Operating Voltage**: 3.3V
5. **Speed** :1-2MBPS

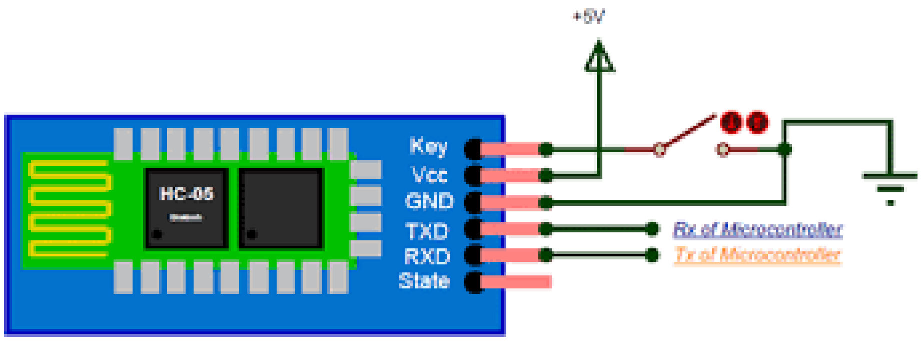


Figure 4.7: HC-05 bluetooth module

6. **Frequency**:2.4GHz ISM band
7. **Operating Current**: 50mA

8. **Range**: <100m
9. Works with Serial communication (USART) and TTL compatible
10. Can operate in Master, Slave or Master/Slave mode
11. Supported **baud rate**: 9600,19200,38400,57600,115200,230400,460800.

4.1.4 Conceptual Sketches using CAD

These sketches portray the **5 Degrees Of Freedom**

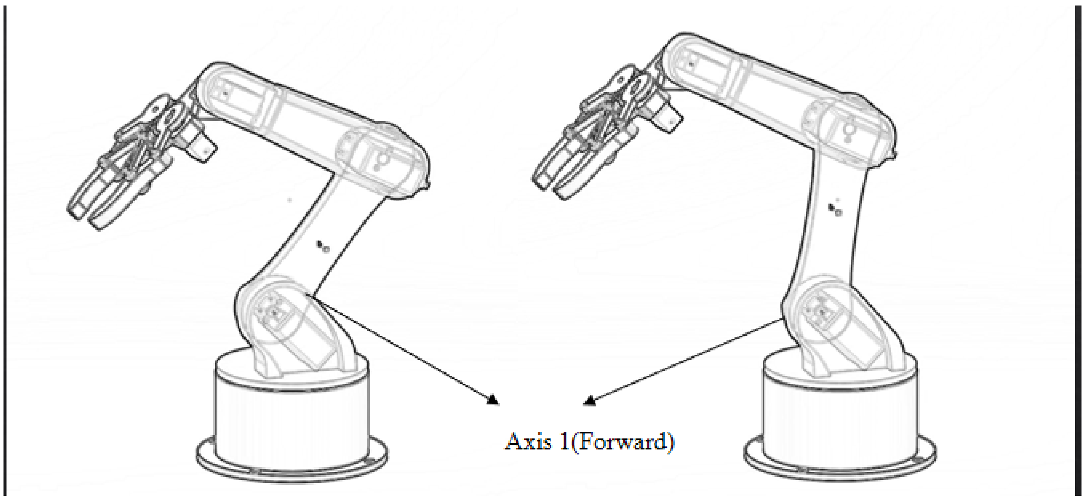


Figure 4.8: 1st Degree

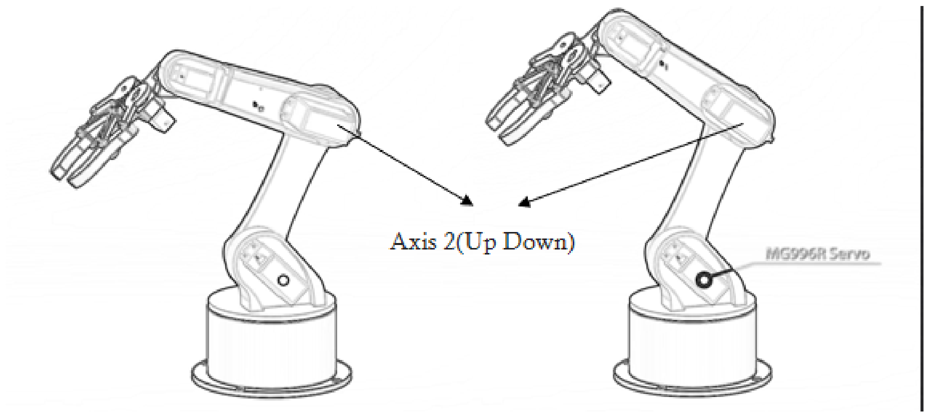


Figure 4.9: 2nd degree

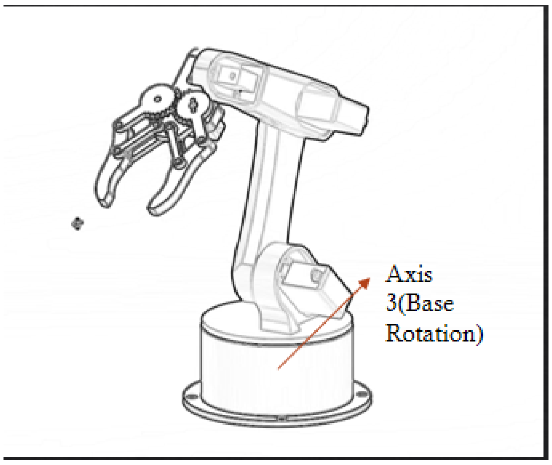


Figure 4.10: 3rd Degree

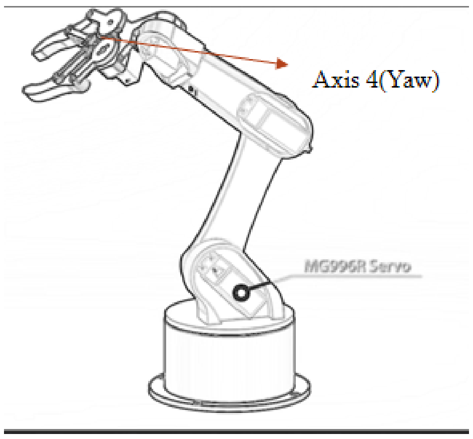


Figure 4.11: 4th Degree

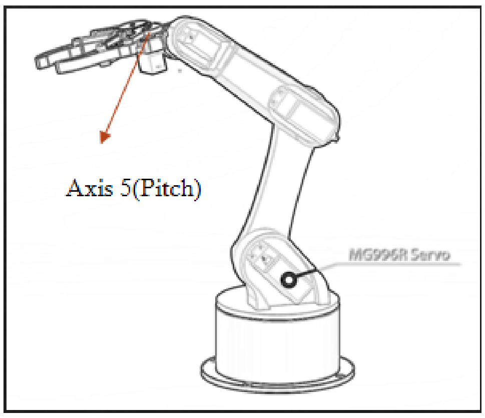


Figure 4.12: 5th Degree

CHAPTER 5

CODING, TESTING

5.1 CODE

First we need to include the **SoftwareSerial library** for the serial communication of the Bluetooth module as well as the servo library. Both of these libraries are included with the **Arduino IDE** so you dont have to install them externally. Then we need to define the six servos, the **HC-05 Bluetooth module** and some variables for storing the current and previous position of the servos, as well as arrays for storing the positions or the steps for the automatic mode.

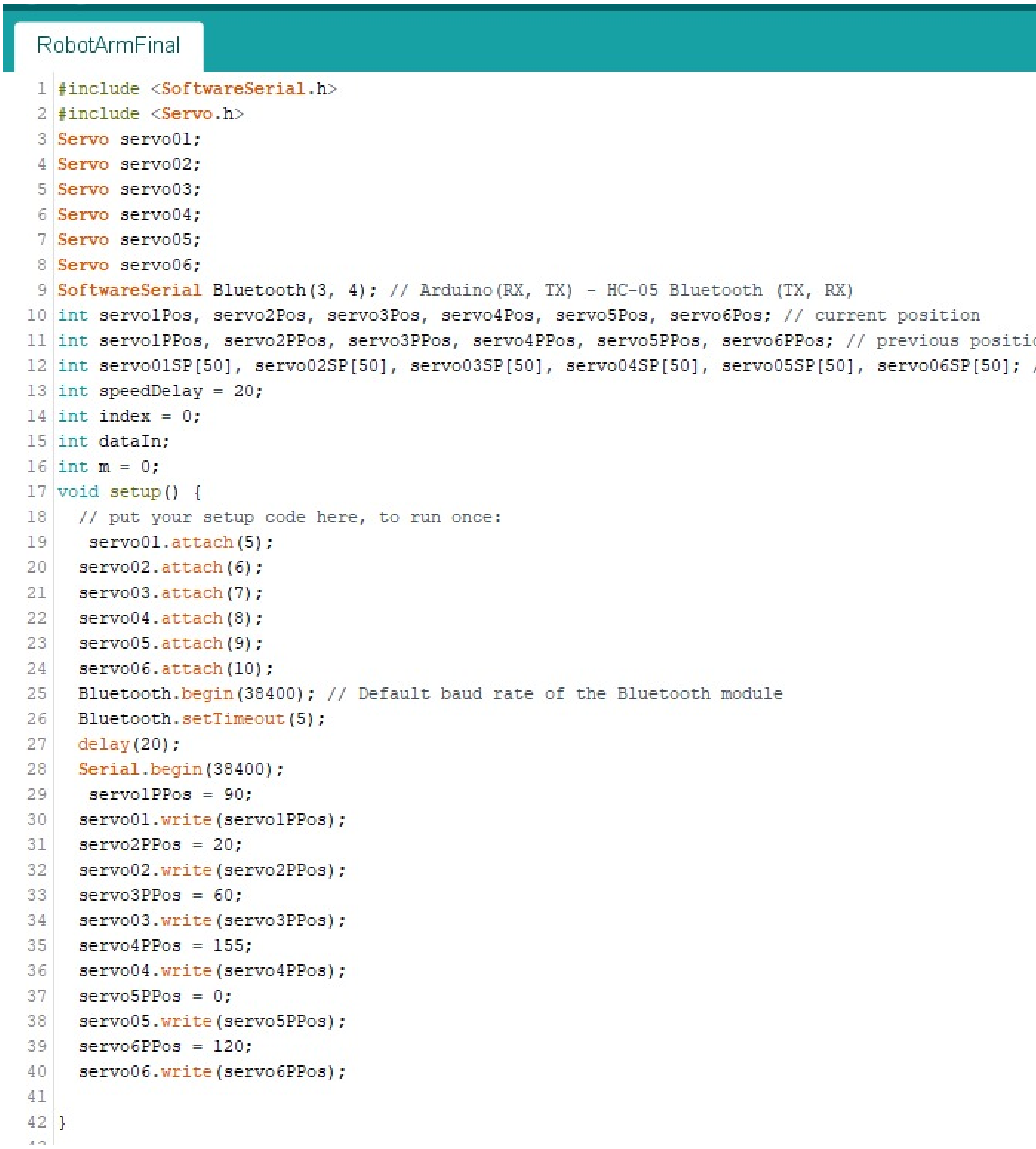
1. In the **setup section** we need to initialize the servos and the Bluetooth module and move the robot arm to its initial position. We do that using the write() function which simply moves the servo to any position from 0 to 180 degrees.

2. Next, in the loop section, using the Bluetooth.available() function , we constantly check whether there is any incoming data from the Smartphone. If true, using the readString() function we read the data as string an store it into the dataIn variable. Depending on the arrived data we will tell robot arm what to do.

3. At the Arduino, using the startsWith() function we check the prefix of each incoming data and so we know what do to next. For example, if the prefix is s1 we know that we need to move the servo number one. Using the substring() function we get the remaining text, or thats the position value, we convert it into integer and use the value to move the servo to that position.

4. At the Arduino, using the startsWith() function we check the prefix of each incoming data and so we know what do to next. For example, if the prefix is s1 we know that we need to move the servo number one. Using the substring() function we get the remaining text, or thats the position value, we convert it into integer and use the value to move the servo to that position.

5. Below them is the SAVE button. If we press the SAVE button, the position of each servo motor is stored in an array. With each pressing the index increases so the array is filled step by step.



```
RobotArmFinal
#include <SoftwareSerial.h>
#include <Servo.h>
Servo servo01;
Servo servo02;
Servo servo03;
Servo servo04;
Servo servo05;
Servo servo06;
SoftwareSerial Bluetooth(3, 4); // Arduino(RX, TX) - HC-05 Bluetooth (TX, RX)
int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos; // current position
int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos; // previous positi
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50]; /
int speedDelay = 20;
int index = 0;
int dataIn;
int m = 0;
void setup() {
  // put your setup code here, to run once:
   servo01.attach(5);
  servo02.attach(6);
  servo03.attach(7);
  servo04.attach(8);
  servo05.attach(9);
  servo06.attach(10);
  Bluetooth.begin(38400); // Default baud rate of the Bluetooth module
  Bluetooth.setTimeout(5);
  delay(20);
  Serial.begin(38400);
   servo1PPos = 90;
  servo01.write(servo1PPos);
  servo2PPos = 20;
  servo02.write(servo2PPos);
  servo3PPos = 60;
  servo03.write(servo3PPos);
  servo4PPos = 155;
  servo04.write(servo4PPos);
  servo5PPos = 0;
  servo05.write(servo5PPos);
  servo6PPos = 120;
  servo06.write(servo6PPos);

}
```

Figure 5.1: Final Code using IDE

6. Then if we press the RUN button we call the runservo() custom function which runs stored steps. Lets take a look at this function. So here we run the stored steps over and over again until we press the RESET button. Using the FOR loop we run through all positions stored in the arrays and at the same time we check whether we have any incoming data from the smartphone. This data can be the RUN/PAUSE button, which pauses the robot, and if clicked again continues with the automatic movements. Also if we change the speed slider position, we will use that value to change the delay time between each iteration in the FOR loops below, which controls the speed of the servo motors.

7. In similar way as explained earlier with these IF statements and FOR loops we move the servos to their next position. Finally if we press the RESET button we will clear all the data from the arrays to zero and also reset the index to zero so we can reprogram the robot arm with new movements.

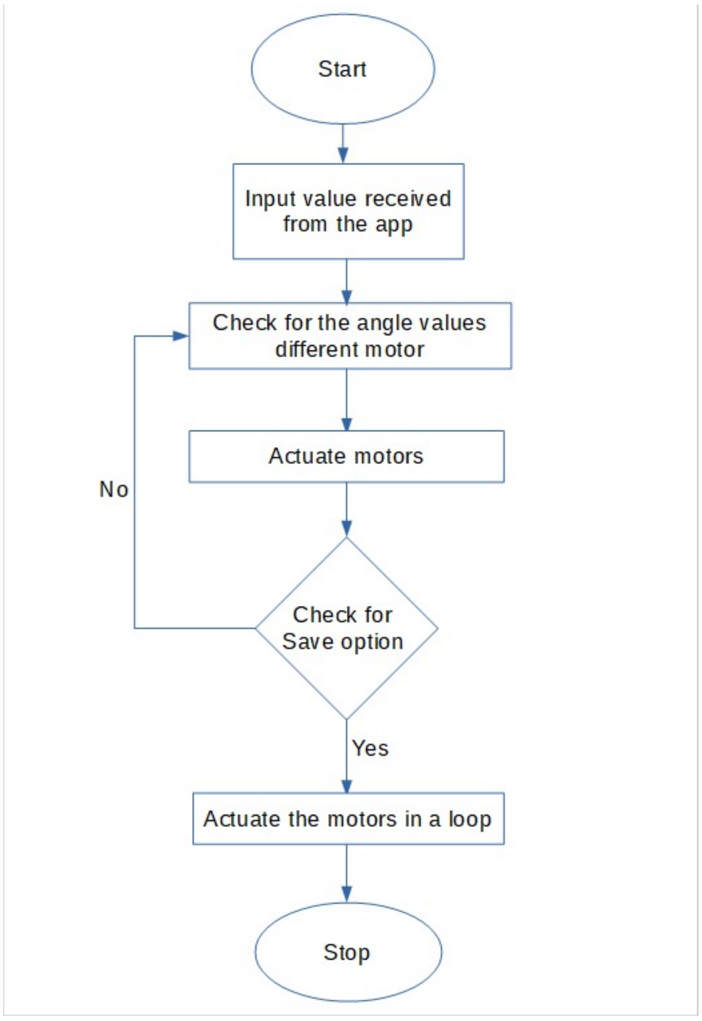

Start
Input value received from the app
Check for the angle values different motor
Actuate motors
No
Check for Save option
Yes
Actuate the motors in a loop
Stop

Figure 5.2: Code Flowchart and Loop

5.1.1 Arduino Robot Arm Control Android App

MIT App Inventor -Software

OLD APP

At the top we have two buttons for connecting the smartphone to the HC-05 Bluetooth module. Then on the left side we have an image of the robot arm, and on the right side we have the six slider for controlling the servos and one slider for the speed control.

Each slider have different initial, minimum and maximum value that suits the robot arm joints. At the bottom of the app, we have three button, SAVE, RUN and RESET through which we can program the robot arm to run automatically. There is also a label below which shows the number of steps we have saved.

First, on the left side we have the blocks for connecting the smartphone to the Bluetooth module.Then we have the sliders blocks for the servo position control and the buttons blocks for programing the robot arm. So if we change the position of the slider, using the Bluetooth function .SendText, we send a text to the Arduino. This text consist of a prefix which indicates which slider has been changed as well as the current value of the slider.

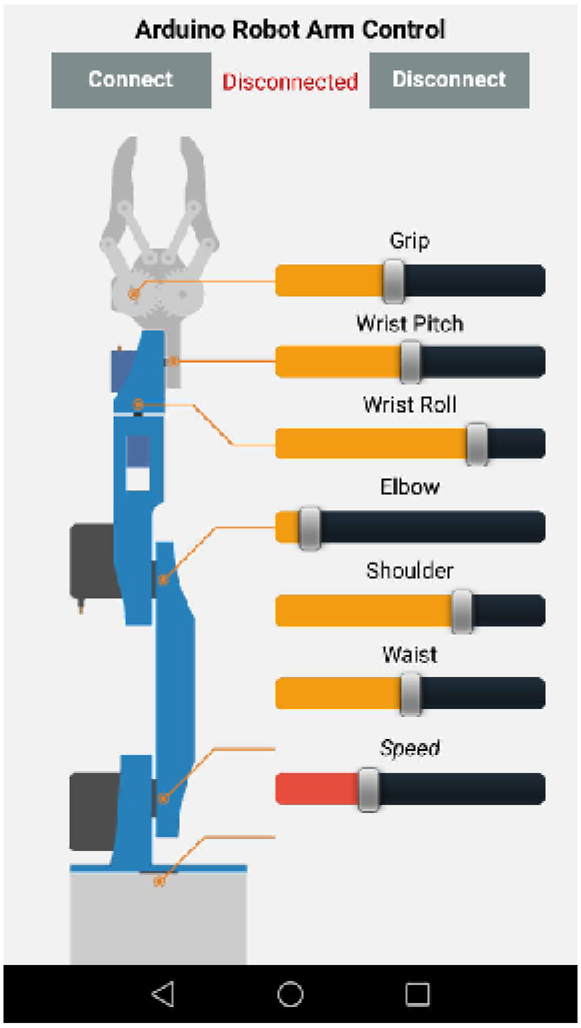
DRAWBACKS OF OLD APP

The previous robot arm control app actually had sliders for controlling the robot joints, but that was causing some problems with the arm stability and it was not easy for the user to control the arm. So we made a new app to gradually control the servo motors using discrete values and more user friendly using forward and backward arrows.

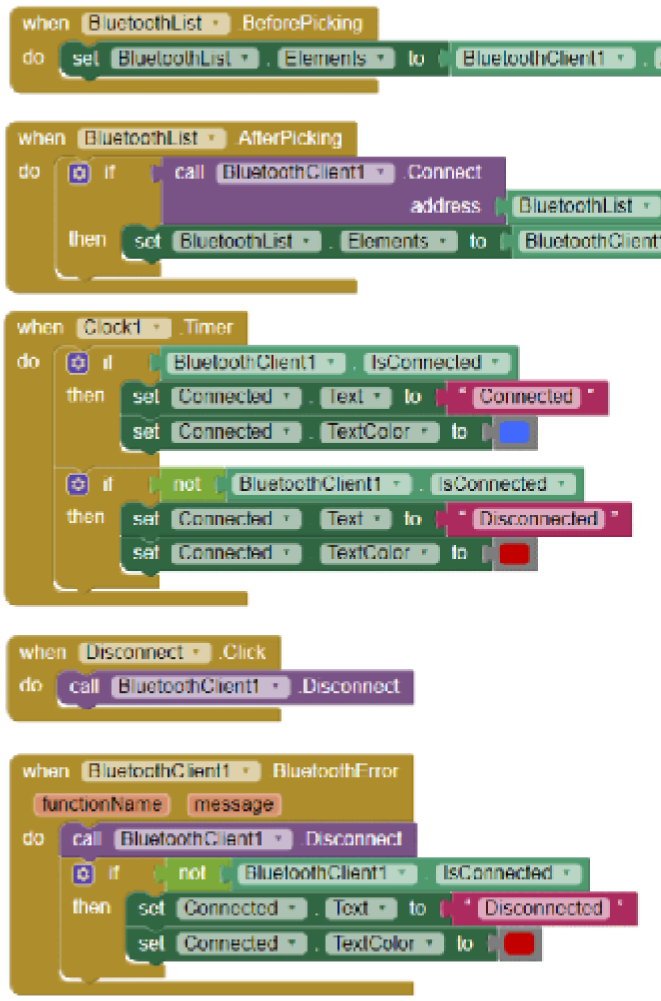
NEW APP

It sends one-byte numbers rather than string when the buttons are clicked. So, depending on clicked button, we tell the Arduino what to do. For controlling the robot arm,when holding the buttons, the robot arm joints move in the particular direction.

In this way we simple send a single 1-byte number when a particular button is touched down. The Arduino code enters the while loop of that number, and stays there until we touch up the button, because at that moment we send the number 0 which means the robot should do nothing.



FRONTEND



```
when BluetoothList .BeforePicking
do  set BluetoothList . Elements to BluetoothClient1 .

when BluetoothList .AfterPicking
do  if   call BluetoothClient1 .Connect
                            address BluetoothList .
    then set BluetoothList . Elements to BluetoothClient

when Clock1 .Timer
do  if   BluetoothClient1 . IsConnected
    then set Connected . Text to " Connected "
         set Connected . TextColor to 🟦
    if   not BluetoothClient1 . IsConnected
    then set Connected . Text to " Disconnected "
         set Connected . TextColor to 🟥

when Disconnect .Click
do  call BluetoothClient1 .Disconnect

when BluetoothClient1 .BluetoothError
  functionName  message
do  call BluetoothClient1 .Disconnect
    if   not BluetoothClient1 . IsConnected
    then set Connected . Text to " Disconnected "
         set Connected . TextColor to 🟥
```

BACKEND

Figure 5.3: Previous App design

So, depending on the touched buttons the servos move either in positive or negative direction. The same working principle applies for all servo motors. For changing the speed of movement, we use the values coming from the slider which range from 100 to 250. By dividing them by 10 we get values from 10 to 25, which are used as delay in microseconds in the whiles loops for driving the servos.

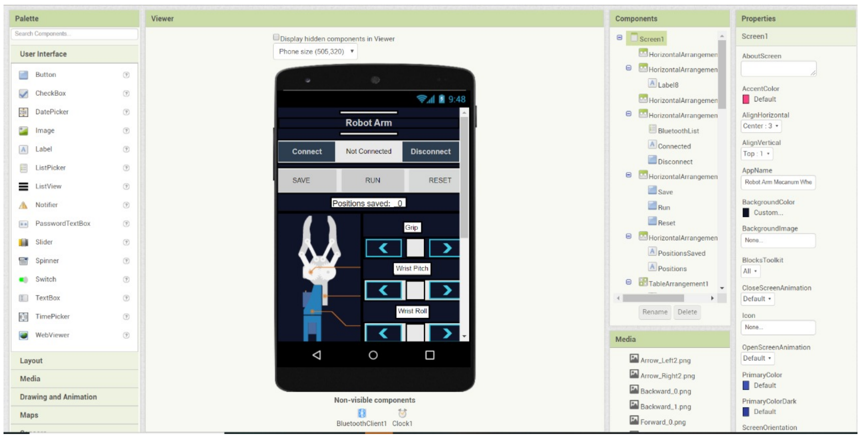


Figure 5.4: Rectified App

For storing the robot movements, we simply save the current positions of the servos and the steppers into arrays, each time the Save button is clicked.

Then when we press the Run button we call the runSteps() custom function. This custom function runs through all stored steps using some for and while loops.

The coolest feature of this robot is the ability to store the movements and then automatically repeat them. Using the Save button, we can simply store the positions of the motors for each step. Then we just need to click the Run button and the robot will automatically repeat the stored movements over and over again.

5.2 Tests and Results

5.2.1 Tests

1. **Test 1**: To compensate the stall torque of the base motor , the shoulder joint was adjusted with an elastic band to provide tension .
2. **Test 2**: after assembly of the arm, speed of the waist motor was tested.

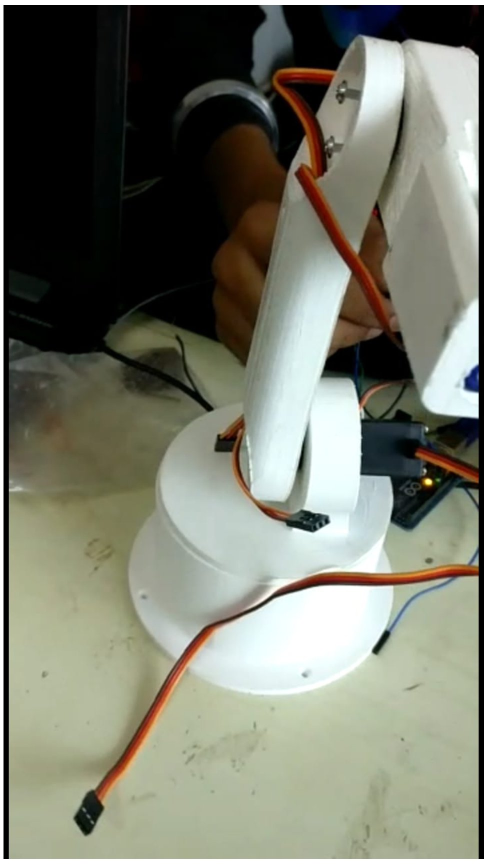


Figure 5.5: Elastic Band at waist shoulder joint

3. **Test 3**: The manipulator gears slipped over each other at extension of the claw , as one of the claw was provided with a servo and the other was not.
4. **Test 4**: For constant current source we used Li-Po battery and 7805 voltage regulator as all 6 motors were not able to get proper amount of current from the microcontroller.
5. **Test 5**: Test with different weights varying from 50gm-200gm.

5.2.2 Results

1. **Result 1**: The degree of rotation of motors. .

Table 5.1: The degree of rotation of motors

Motor	Operating range of motor(angle in degree)
Waist motor	0 to 180
Shoulder motor	0 to 120
Elbow motor	60 to 140
Wrist roll motor	50 to 155
Wrist pitch motor	0 to 90
Gripper motor	80 to 120

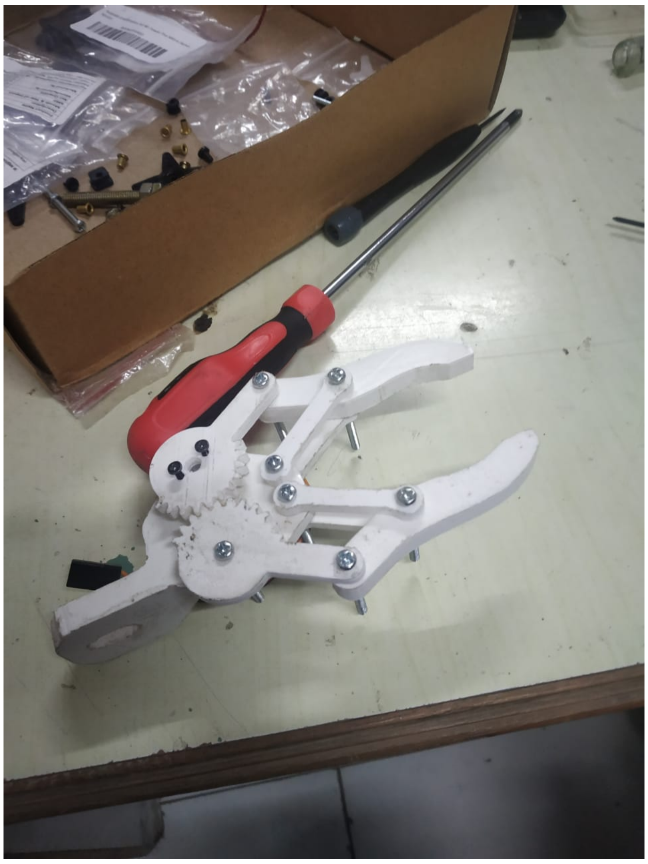


Figure 5.6: Manipulator Gears

2. **Result 2**:The maximum weight manipulator can lift is turned out to be 200gm

CHAPTER 6

CONCLUSION

Our robotic arm could be efficiently controlled by a more compact and feasible GUI rather than the much bulkier console in already existing technology , which to operate is a tedious task. This can reduce the human effort by very closely imitating to action of a human hand.

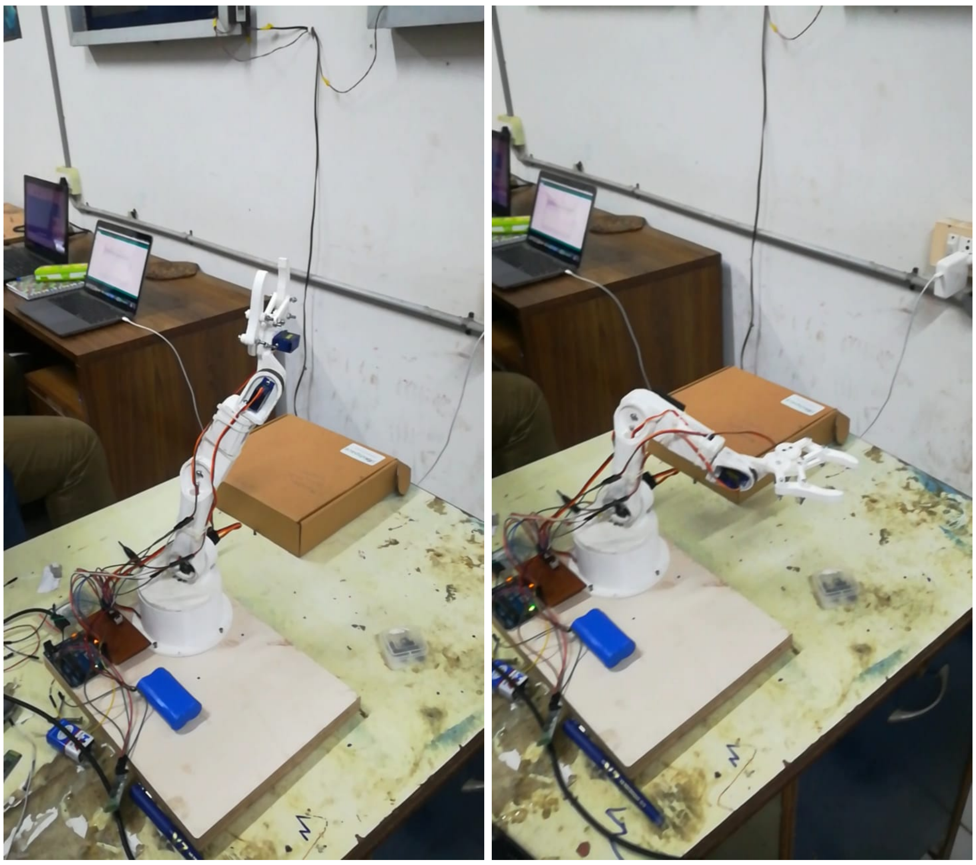


Figure 6.1: Finally Assembled Arm in different positions while picking an object

CHAPTER 7

FUTURE ENHANCEMENT

This prototype can be implemented commercially by replacing bluetooth by **Radio Frequency Technology** and also replacing servo motors by more efficient **stepper motors**

It can be wirelessly controlled

It can be made autnomous by using caterpillar tracks and Artificial Intelligence.

In medical fields the manipulator can be modified to perform delicate surgeries and diagnosis.

APPENDIX A

MATERIALS

A.1 Hardware

1. Arduino Microcontroller:Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started

2. Carbon Fibre Filament

3. HC-05 bluetooth module:The HC-05 is a very cool module which can add two-way (full-duplex) wireless functionality to your projects. You can use this module to communicate between two microcontrollers like Arduino or communicate with any device with Bluetooth functionality like a Phone or Laptop. There are many android applications that are already available which makes this process a lot easier. The module communicates with the help of USART at 9600 baud rate hence it is easy to interface with any microcontroller that supports USART.

4. Servomotors:The term servomotor does not refer to one single kind of motor. Instead it refers any type of motor that receives a command signal from a controller. In this same respect, any closed loop system can be referred to as a servo system.This flexibility allows for several suitable types of electric motors to be used in servo systems. These electric motors include:

   Permanent Magnet DC Motor

   Brushless DC Motor

   Induction AC Motor

   Electromagnetic motors operate based on the principle that the magnetic force on an electrical conductor in a magnetic field is perpendicular to that field. This force is defined by,

   F= qv*B

   Where:

   F is the vector describing the magnetic force

   q is the magnitude of the electrical charge

   v is the vector magnitude of the charged particles velocity

   B is the vector describing the magnetic field

However, in the case of an electric motor the force can be quantified as a scalar

F=I*L*B

Where: F is the magnetic force

I is the electric current in the coil

L is the length of the coil contained within the magnetic field

B is the strength of the magnetic field

A.2 Software

1. Arduino IDE:Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

2. Utilmaker Cura:Cura is an open source 3D printer slicing application. ... Ultimaker Cura is used by over one million users worldwide, handles 1.4 million print jobs per week, and is the preferred 3D printing software for Ultimaker 3D printers, but it can be used with other printers as well.

3. Solidworks:SolidWorks is a solid modeling computer-aided design (CAD) and computer-aided engineering (CAE) computer program that runs on Microsoft Windows. SolidWorks is published by Dassault SystÃÍmes. For those of us having served any amount of time as a 3D CAD designer, the name SolidWorks is a name well known.

REFERENCES

[1] Arefin, S. E., Ashrafi Heya, T., and Uddin, J. (2018). "Real-life implementation of internet of robotic things using 5 dof heterogeneous robotic arm." *2018 Joint 7th International Conference on Informatics, Electronics Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, 486–491 (June).

[2] Gohil, V. J., Bhagwat, D. S. D., Raut, A. P., and Nirmal, P. R. (2013). "Robotics arm control using haptic technology." *International Journal of Latest Research in Science and Technology*, 2(2), 98–102.

[3] Jahnavi, K. and Sivraj, P. (2017). "Teaching and learning robotic arm model." *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 1570–1575 (July).

[4] Jimin Liang, Gong Zhang, Xianshuai Chen, and Wang, D. (2015). "Development of two degrees of freedom deterministic parallel robotic arm unit." *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, 348–351 (Feb).

[5] Khan, A., Ahmed, T., and Sajjad, M. (2013). "Design, analysis and implementation of a robotic arm-the animator." *International Journal of Engineering Research*, 02, 298.

[6] Kruthika, K., Kumar, B. M. K., and Lakshminarayanan, S. (2016). "Design and development of a robotic arm." *2016 International Conference on Circuits, Controls, Communications and Computing (I4C)*, 1–4 (Oct).

[7] Omijeh, B., Uhunmwangho, R., and Ehikhamenle, M. (2014). "Design analysis of a remote controlled pick and place robotic vehicle." *International Journal of Engineering Research and Development*, 10(5), 57–68.

[8] Pachaiyappan, S., Balraj, M. M., and Sridhar, T. (2014). "Design and analysis of an articulated robot arm for various industrial applications." *J. Mech. Civil Eng*, 42–53.

[9] Patidar, V. and Tiwari, R. (2016). "Survey of robotic arm and parameters (01).

[10] Songmin Jia, Tomoyuki Murakami, Daisuke Chugo, and Kunikatsu Takase (2008). "Human-assistance robotic system using interactive gui." *2008 SICE Annual Conference*, 2291–2294 (Aug).

[11] Yenorkar, R. and Chaskar, U. M. (2018). "Gui based pick and place robotic arm for multipurpose industrial applications." *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 200–203 (June).