 ****

**TECHNO INTERNATIONAL NEW TOWN**

**(Formerly known as Techno India College of Technology)**

**Block-DG, Action Area 1, New Town, Kolkata -700156, West Bengal, India**

**Department of Computer Science & Engineering**

**Eighth Semester Project Report (CS892)**

***Project Title***

***Prepared by***

***SUMANA KUNDU (Roll No: -18700116013)***

***SRIJA DAS GUPTA (Roll No: -18700116022)***

***SOHINI JANA (Roll No: -18700116028)***

***SHREYASHI GANAI (Roll No: -18700116035)***

***SHIVANSHI (Roll No: -18700116037)***

***Under the Guidance of***

***Prof. DEBRAJ CHATTERJEE***

***Batch: - 2016-2020******Semester :8 th(2020 – EVEN) Year: 2020(FEBRUARY-JULY)***

***Stream: - Computer Science & Engineering Year of Study: 4th***

***Affiliated to***

****

**MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, WESTBENGAL**

**(FORMERLY KNOWN AS WEST BENGAL UNIVERSITY OF TECHNOLOGY)**

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to Prof. DEBRAJ CHATTERJEE of the Department of Computer Science & Engineering, whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest he took in advising us, for the books and reference materials provided for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all non-teaching staff for the cordial support they offered.

Place: Techno International New Town

Date: \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

SUMANA KUNDU (18700116013)

SRIJA DAS GUPTA (18700116022)

SOHINI JANA (18700116028)

SHREYASHI GANAI (18700116035)

SHIVANSHI (18700116037)

Department of Computer Science & Engineering,

Techno International New Town

Kolkata – 700 156

West Bengal, India.

**Approval**

This is to certify that the project report entitled “ROBOTIC ARM USING ARDUINO” prepared under my supervision by SUMANA KUNDU (13) ,SRIJA DAS GUPTA(22),SOHINI JANA(28),SHREYASHI GANAI(35),SHIVANSHI(37) be accepted in partial fulfillment for the degree of Bachelor of Technology in Computer Science & Engineering which is affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal (Formerly known as West Bengal University of Technology).

It is to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

…………………………………………

Prof. DEBRAJ CHATTERJEE

………………………………………….

Prof. Bikash Sadhukhan,

HOD, Computer Science & Engineering,

Techno International New Town

### ABSTRACT

Today, technology is developing in the same direction in line with rapidly increasing human needs. The work done to meet these needs makes life easier every day, and these studies are concentrated in robotic arm studies. Robot arms work with an outside user or by performing predetermined commands. Nowadays, the most developed field of robot arms in every field is the industry and medicine sector. Designed and realized in the project, the robot arm has the ability to move in 4 axis directions with 5 servo motors. Thanks to the holder, you can take the desired material from one place and carry it to another place, and also mix it with the material it receives. While doing this, robot control is provided by connecting to the android application via Bluetooth module connected to Arduino Nano microcontroller.

### Table of Contents

ABSTRACT

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTERS

1. INTRODUCTION

2. THEORETICAL INFRASTRUCTURE

2.1. Servo Motors

2.2. Servo Controller

2.3. Arduino Nano Microcontroller

2.4. Stepper Motor

2.5. Stepper Motor Driver

2.6.H Bridge motor controller

2.7. HC-06 Bluetooth Module

3. ROBOT ARM MECHANICS

4. DESIGN BRIEF

4.1. Mechanical Design

4.2. Mechanical Part Mounting

4.3. Android Programming

5.Arduino Programming

6. EXPERIMENTAL STUDIES

6.1. Projected Method

6.2 Battery

6.3 Stepper Motor Controller

6.4 Servo Motor Controller

6.5. Robot Arm Control

7.Future Scope

8.Problems While Making Robotic Arm

9. Conclusion

10.Bibliography

**INTRODUCTION**

These days’ people always needed additional help systems. With the rapid increase in the flow of information, people are now guided to search for different markets and people have entered the competition to manufacture quality products cheaply. Automation systems are also needed to realize this. Because standardized automation systems are required to minimize errors as well as to have experienced and well-trained employees for quality products. Because of their physical characteristics, people needed to use auxiliary machines in places where their strength was not enough. These machines, which are operated with the need for human assistance in advance, have been made to operate spontaneously without the need of human power with the progress of technology. One of the most used components of automation systems is robots. Robotic systems; Mechatronics Engineering, Mechanical Engineering, Electrical Engineering and Computer Engineering have all come together to work together. In the project, researchers have been done and implemented in order to have knowledge about mechanics and software during the operations carried out by the robot arm which is designed to fulfil the tasks determined in accordance with predetermined commands. First, it was determined what function the robot arm would be and what movements it could make. Robotic arm made of Android phone or tablet control; it can carry the desired material, mix it up and perform the commands previously determined by a user. If this project is also a designated task; the robotic arm takes a piece of material and brings it to the desired position and then records its movements and lets it do the same action until we stop it. The servomotor is preferred in order to be able to perform these operations properly since the motor to be selected must operate precisely and must be at high torque. The robot arm is composed of 5 servo motors and can move in 4 axis directions with these motors. In the project, Arduino Nano microcontroller written in Java language is programmed and servo motor control is provided. Thus, it is possible to perform the desired operations by means of the elements located on the Arduino without any circuit construction other than the circuit where the servo motor inputs are located. For the mechanical part, the robot arm is drawn with the SolidWorks program and the dimensions of the robot arm are specified. A 5V power supply is also preferred for the robot to work.

### 2.THEORETICAL INFRASTRUCTURE

The theoretical background of the project is examined below as main headings and subheadings.

### 2.1. Servo Motors

Servo: Detects the operation error of a mechanism, provides feedback and corrects faults. The servo motor can have alternating current (AC), direct current (DC) or stepper motors. In addition to these, there are drive and control circuits. Servo motors are the kinds of motors that can fulfil the commands we want. They can operate steadily even at very small or very large speeds. In these motors, the large moment can be obtained from the small size. Servo motors are used in control systems such as fast operation, excessive axis movement, condition control and so on. Servo motors are the last control element of a mechanism. They are highly sensitive and servo motors are used in conjunction with electronic or programmable circuits. These engines are divided into AC and DC. When the AC servo motors are brushless type motors, the servo motors brush. Servo motors are mostly three cables. These are a red cable for power, black for grounding and yellow cables for control (data, data). One of the servomotors used in the production phase of the project is shown in Fig.1



Figure 1DC servo motor

Servo motors are controlled according to the signal condition. In doing so, the supplied pulse width modulated (PWM) signal is used with the data bus. Each servo motor is controlled by a PWM signal at 10-20 ms and at 0.5-1.5 ms. The position of the motor shaft is determined according to the duration (tk) of this signal at logic 1.

• When tk = 0.5 ms, the motor shaft rotates to the end,

• When tk= 0,5- 1 ms, the position of the motor shaft is in the middle,

• When tk= 1 - 1.5 ms, the motor shaft turns to the right,

• When tk= 10-20 ms (when the same signal is given again) it remains in its old position,

The position control of these motors is determined using the required pulses. The servo motors DC used in the project are kept at about 5V during operation.

### 2.2 SERVO CONTROLLER

More specific types of controllers, such as servo controllers, are used to control servomotors. When dealing with servo systems, it’s important to note that definitions for servomotor, servo system, and servo controllers can differ widely throughout industry. When selecting a servo system for an application, it’s best to ask suppliers what exactly their offerings entail.

A servo controller is the heart of a servo system. A typical servo system consists of a motor, feedback device, and the controller. The control circuitry typically involves a motion controller, which generates the motion profile for the motor, and a motor drive which supplies power to the motor based on the commands from the motion controller. Servo systems are closed-loop systems which have some benefits over open-loop systems including the fact that they improve transient response times, reduce steady state errors and reduce system sensitivity to load parameters.

Servo controllers perform two types of tasks; tracking some commanded input and improving a system’s disturbance rejection. One of the most powerful methods of control is PID control, which stands for proportional-integral-derivative control. PID control is a combination of proportional control, integral control and derivative control. A PID control method works on the error signal which is the difference between a commanded value and the actual value of an output variable, and driving the error to zero. The proportional value can be thought of as a simple gain value. The integral value integrates the error over a period of time and helps to drive the error to zero. The derivative value helps to stabilize a system that uses an integral and proportional term only.



### 2.3. Arduino UNO Microcontroller

Although microcontroller type PIC is usually used in programming and software field, Arduino has become very popular in the world in recent times. It is based on Arduino's past wiring and processing projects. Processing is written for nonprogramming users. Arduino wiring is produced on the basis of the programming language. The common feature of both is that it provides an environment where even the basic knowledge of electronics and programming can easily design. Arduino is now becoming more and more common nowadays. Even unmanned aerial vehicles made with Arduino, which is used almost every field, are visible. The causes of the spread of Arduino at such a rapid rate are;

• It can be used on all platforms due to the simplicity of the development environment with driver usage.

• With the help of the advanced library, even complex operations can be easily solved.

• Programs written in Arduino can run fast because they are not run on any other platform.

• There is a lot of hardware support that is compatible with Arduino and can work together.

• Communication with the environment is easy because it is open source.

• If there are any problems due to a large number of Arduino users, the solution can be easily reached.



Figure 2

**Arduino Uno** is a microcontroller board based on the ATmega328P ([datasheet](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)). 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. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards.

### General pin functions

* **LED**: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.
* **VIN**: The input voltage to the Arduino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
* **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
* **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* **GND**: Ground pins.
* **IOREF**: This pin on the Arduino/Genu board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source, or enable voltage translators on the outputs to work with the 5V or 3.3V.
* **Reset**: Typically used to add a reset button to shields that block the one on the board.[[7]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-website-7)

**Special pin functions**

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, under software control (using pinMode(), digitalWrite(), and digitalRead() functions). They operate at 5 volts. Each pin can provide or receive 20 mA as the recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50K ohm. A maximum of 40mA must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5; each provides 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though it is possible to change the upper end of the range using the AREF pin and the analogReference() function.[[7]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-website-7)

In addition, some pins have specialized functions:

* **Serial** / [UART](https://en.wikipedia.org/wiki/UART): pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
* **External interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* [**PWM**](https://en.wikipedia.org/wiki/Pulse-width_modulation) (pulse-width modulation): pins 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite() function.
* [**SPI**](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface) (Serial Peripheral Interface): pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.
* **TWI** (two-wire interface) / [I²C](https://en.wikipedia.org/wiki/I%C2%B2C): pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.
* **AREF** (analog reference): Reference voltage for the analog inputs.[[7]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-website-7)

### 2.4 STEPPER MOTOR

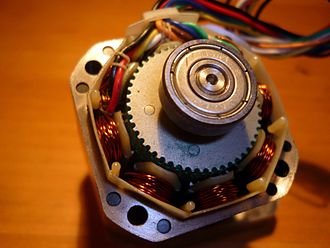
A **stepper motor**, also known as **step motor** or **stepping motor**, is a business DC electric motor that divides a full rotation into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without any position sensor for feedback (an open-loop controller), as long as the motor is carefully sized to the application in respect to torque and speed.

Switched reluctant motors are very large stepping motors with a reduced pole count, and generally are closed-loop commutated. DC motors  rotate continuously when DC voltage is applied to their terminals. The stepper motor is known by its property of converting a train of input pulses (typically square wave pulses) into a precisely defined increment in the shaft position. Each pulse moves the shaft through a fixed angle.

Stepper motors effectively have multiple "toothed" electromagnets arranged around a central gear-shaped piece of iron. The electromagnets are energized by an external driver circuit or a micro-controller. To make the motor shaft turn, first, one electromagnet is given power, which magnetically attracts the gear's teeth. When the gear's teeth are aligned to the first electromagnet, they are slightly offset from the next electromagnet. This means that when the next electromagnet is turned on and the first is turned off, the gear rotates slightly to align with the next one. From there the process is repeated. Each of those rotations is called a "step", with an integer numbers of steps making a full rotation. In that way, the motor can be turned by a precise angle.

The circular arrangement of electromagnets is divided into groups, each group called a phase, and there is an equal number of electromagnets per group. The number of groups is chosen by the designer of the stepper motor. The electromagnets of each group are interleaved with the electromagnets of other groups to form a uniform pattern of arrangement. For example, if the stepper motor has two groups identified as A or B, and ten electromagnets in total, then the grouping pattern would be ABABABABAB.

Electromagnets within the same group are all energized together. Because of this, stepper motors with more phases typically have more wires (or leads) to control the motor.



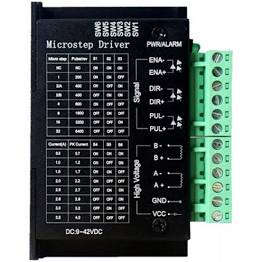
**2.5 STEPPER MOTOR DRIVER**

A **stepper motor driver** (or stepper motor drive) is a circuit which is used to drive or run a stepper motor. It is often called a stepper motor driver. A stepper motor driver usually consists of a controller, a driver and the connections to the motor.

A lot of drive circuits are available in the market today. Many circuits are so easy to interface to a motor that you can almost instantly connect the stepper motor to it and you are ready to run the motor. These circuits come in a variety of ratings for current and voltage and one should select them according to the needs of the motor which will be used.

**ESSENTIAL COMPONENTS FOR STEPPER MOTOR DRIVER:**

* Controller (Essentially a micontroller or a microprocessor)
* A driver IC to handle the motor current
* A power supply unit



### 2.6. H-BRIDGE MOTOR CONTROLLER

### The most commonly used actuator in any electronic device/machine will be motors next to solenoids, pneumatics and hydraulics. From a simple vibration motor inside a mobile phone to complex [stepper motors](https://components101.com/tags/stepper-motor) in CNC machines, these DC machines can be found everywhere. To control a motor using a [Microcontroller](https://components101.com/microcontrollers) or processors we need something called a **Motor Driver** or **Motor Controller**. Depending upon the type of motor and type of control required the type of Motor Drivers will also change. In this article we will focus only of [DC motors](https://components101.com/tags/dc-motor) and how to control a DC motor using a Motor Driver with the most popular H-bridge Topology. This technique will help us drive small or large DC Motors and also control its direction.

### ****Why do we need motor driver IC****

As the world is witnessing new technology every day, autonomous robots are one of them. We use these ICs in autonomous robots mainly to control them. Microprocessors operate on low-level voltage/current, unlike motors. For example the popular Arduino microcontrollers or PIC microcontroller has an operating voltage of 5V or 3.3V, but a decent DC motor requires 5V or 12V to operate.

### ****How the motor driver works****

Motor driver receives signals from the microprocessor and eventually, it transmits the converted signal to the motors. It has two voltage pins (VCC1 and VCC2), and one of them is used to turn on the motor driver, and another pin is used to apply the voltage to the motor through this motor IC. This motor IC will continuously toggle the output signal according to the input wave it is receiving from the microprocessor.

The small IC transmits the signal it receives, but it will not change the value of the signal. For example, if the microprocessor sends a high input (1) to the Driver Ic then, driver Ic will pass the same High (1) though it's an output pin. The **H-bridge circuit** will look like this in the picture below. Four switches will form an **''H'' shape**, and these four switches are used to enable/disable the supply.

### 

**Rotate in clockwise direction**

Now, in the first condition, when **S1** and **S4** switches are closed, and **S3** and **S2** are open, the voltage will pass from the**S1** switch to the Motor and then to the **S4**. Hence we have a complete circuit that will allows current to flow from V to M through S1 and S4. This state will be a short circuit in S1 and S4 switch condition. In this case, the Motor will be in ON, and the direction of the Motor will be in a **clockwise rotation.**

**Rotate in counter-clockwise direction**

Coming to the Next state, when we enable **S3**and **S2**by giving voltage input, then the **S1** and**S4**switches will close, and the voltage travels from S3 and S2. This is also a positive connection by enabling two parallel-connected switches, but the rotation of the Motor will be in a **counter-clockwise direction**.

**How to turn off the Motor (Braking)**

To turn off the Motor, either we can turn off the voltage supply, or we can open all the switches in Motor. And note that you can only turn on two parallel connected switches at a time.  And when we close S1 and S3 switches, Motor will receive only the positive signal from two sides, and it has no direction to send the ground signal. As a result, Motor will get into the ‘***STALL***’ condition. The direction of the motor rotation will be based on the inputs we are giving and also the switches we opened in the circuit.

**Advantages of Motor Driver IC**

* High level functionality and better performance
* The circuit is easy to operate. So, we can easily control the robot using inputs.
* It can be used in autonomous and commercial robots, as well.
* The motor deals with heavy current. Due to this current flow, IC gets heated, and we need a heatsink to reduce the heating.
* We use 4 capacitors in this circuit to avoid the fluctuations of voltage while using the motor in one direction and suddenly when we take the opposite direction. This works as a direction shifter without loss.

### 2.7 HC -06 Bluetooth Module

The HC-06 BT module is a slave operating only and uses a serial communication protocol. In Bluetooth communication, master and slave are determined according to the state of connection start. A master module can initiate the connection, but the slave module cannot initiate the connection. In our project, we will provide an external device to connect to a slave PC or an android device. Bi-directional data can be sent and received in a healthy way. Page | 6 Figure 3HC-06 Bluetooth module After adding the device to the Bluetooth devices, a virtual com port associated with the module is created on our computer. It is now possible to communicate via Bluetooth via the module. The codes that we send via UART with the microcontroller (PIC18F46K22) can be received via Bluetooth with PC, which is connected to the BT module. Data sent from the PC and COM overreaches the microcontroller.



### 3. ROBOT ARM MECHANICS

3.1. General Characteristics of Robot Arm Mechanics

Kinematics in robotics is the science of motion investigation. Robot arm links can be rotated or offset according to the reference coordinate frame. A systematic and general approach developed by Denavit and Hartenberg establishes the relationship between the robot endpoint and the total displacement of robot arm links (1). Angular and linear displacements between limbs are called joint coordinates and are defined by limb variables. In order to determine the amount of rotation and displacement according to the reference coordinate system of the endpoint, the matrices A which represent the amounts of each limb rotation and displacement are multiplied in turn. If the coordinates of the end point are given, limb variables can be obtained by going backward. These operations are called forward and inverse kinematics. The next section will explain how to determine forward and reverse kinematics. The general transformation matrix can be quite complex even for simple robots. It can be found in standard textbooks such as the Jacobian matrix for standard robots (2) and (3).

### 4. DESIGN BRIEF

The design part is divided into two parts, the mechanical part design, and the mechanical part installation. In the design of the mechanical part, the millimetric drawings of the parts to be used in the robot arm construction were made through the help program. In the installation of the mechanical part, the naming of the servomotors used in the robot arm and the tasks during the operation of the robot are explained. The construction of the project consists of several steps. These steps are;

• Determination of the mechanical materials required for the production of the project,

• Determination of microcontroller and software to be used in the project,

• Search and selection of servo motors that will run the robot arm in a proper way,

• Proper selection of mechanical parts,

• Implementation of robot arm assembly,

• Testing the system to see if it works properly with the microcontroller we choose,

• Possible faults have been given in the form of restructuring the system by passing through the eye.

These steps have been completed and the design of the robot has been completed.

### 4.1. Mechanical Design

Materials needed for the mechanical part of the robot arm were supplied, and then the materials were drawn on SolidWorks in millimetric form. The mechanical part was assembled with these materials. Drawings of the mechanical part of the project are given in Figures 5, 6, 7, 8.

### 4.2. Mechanical Part Mounting

The cut parts are assembled together with the servo motors and assembly of the robot arm is completed. The robot arm moves by 4 axes and performs this movement with 5 Mini Servo Motors. After drawing the parts on SolidWorks and making the necessary parts, the design of the mechanical part is started and the robot arm is made of a stationary lower body and movable upper body. There is one servo motor in the stationary lower part of the robot arm and this robot is called axis 1. The 1st axis provides rotation of the robot arm to the right or left. There are 2 servo motors in the moving upper body part. Since these servo motors must operate parallel to each other, both start and end positions are set simultaneously. These two servo motors are named as 2nd axis and 3rd axis respectively. The tasks of 2nd and 3rd axes are to move the robot arm up and down. Mounting of servo motors in 1st and 2nd and 3rd axes is given in Fig.11.

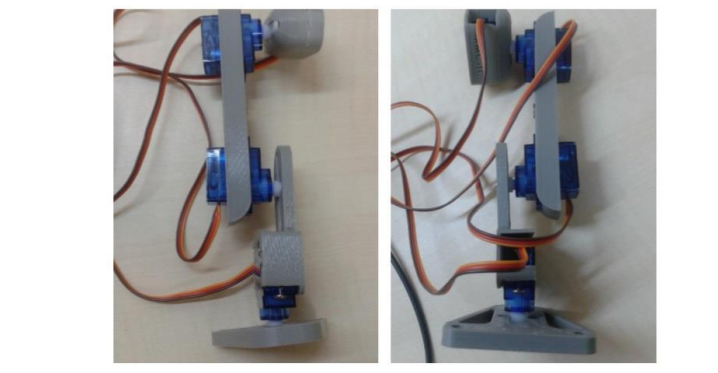


Figure 11

Mounting of servo motors in 1st, 2nd, and 3rd axes.

### 4.3. Android Programming

The Android application we used in this project was done with App Inventor. What is App Inventor? App Inventor is a free web application developed by Google and later developed by the Massachusetts Institute of Technology (MIT). Other MIT projects use a graphical interface, such as Scratch and StarLogo TNG, to drag and drop visual objects instead of printing lines of code in mobile applications running on the Android operating system. App Inventor Design is the part of the application that we will design for visual and active components of the application, from Buttons, Pictures, Text Fields; it can be added to the Media Components such as Audio, Video, etc., besides the GPS, Acceleration and Bluetooth sensor can be added to the device. In this section, you have a design criterion where many variables like size, colour, the position can be controlled. The design of the android application we use in the project is shown in Figure 12

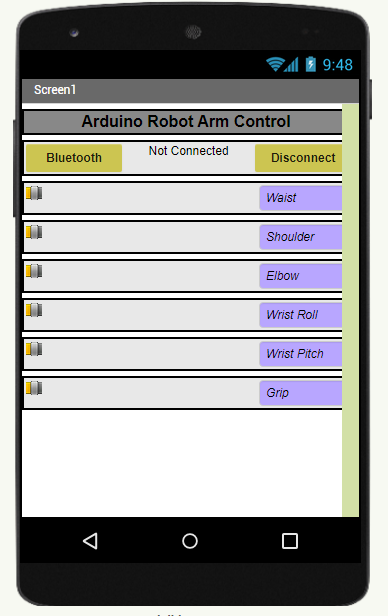
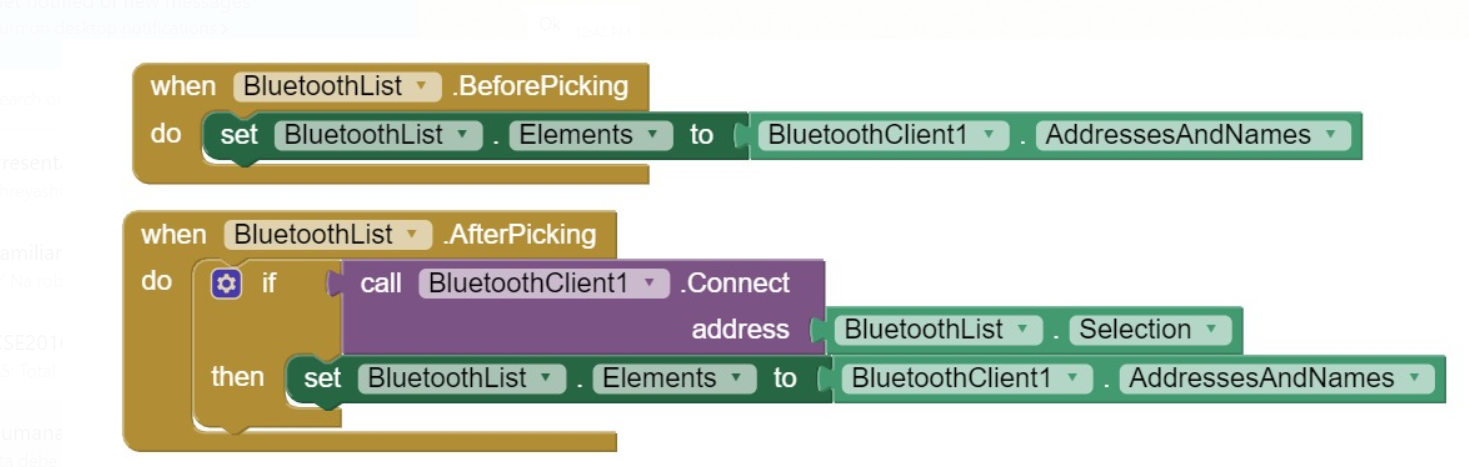
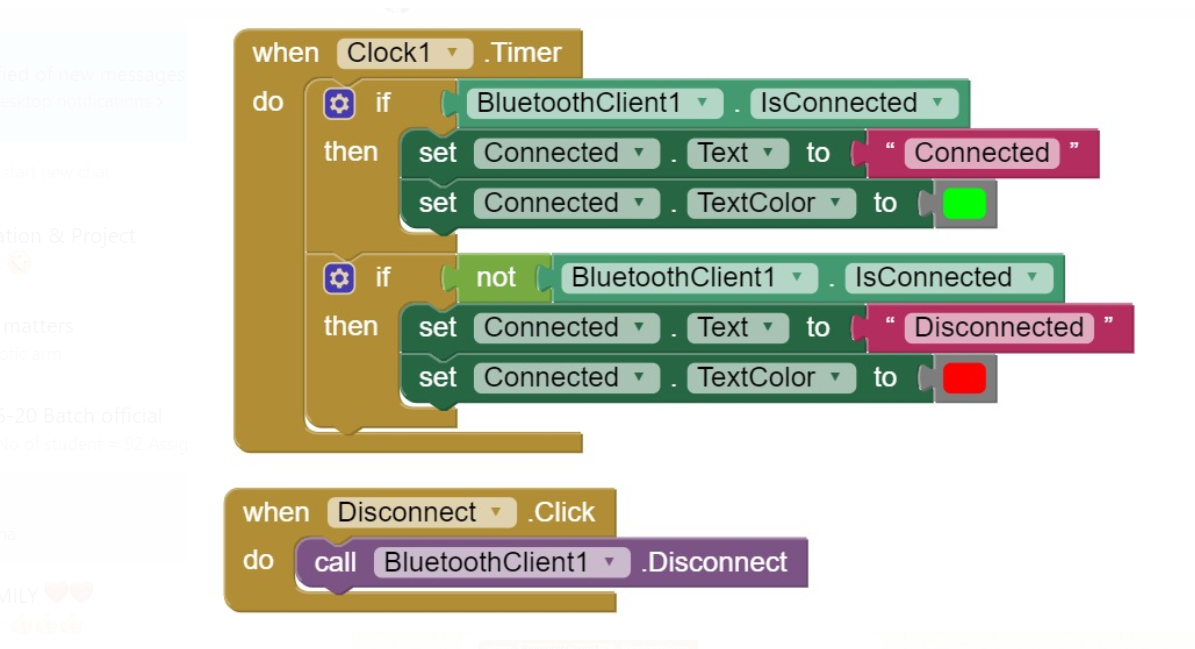
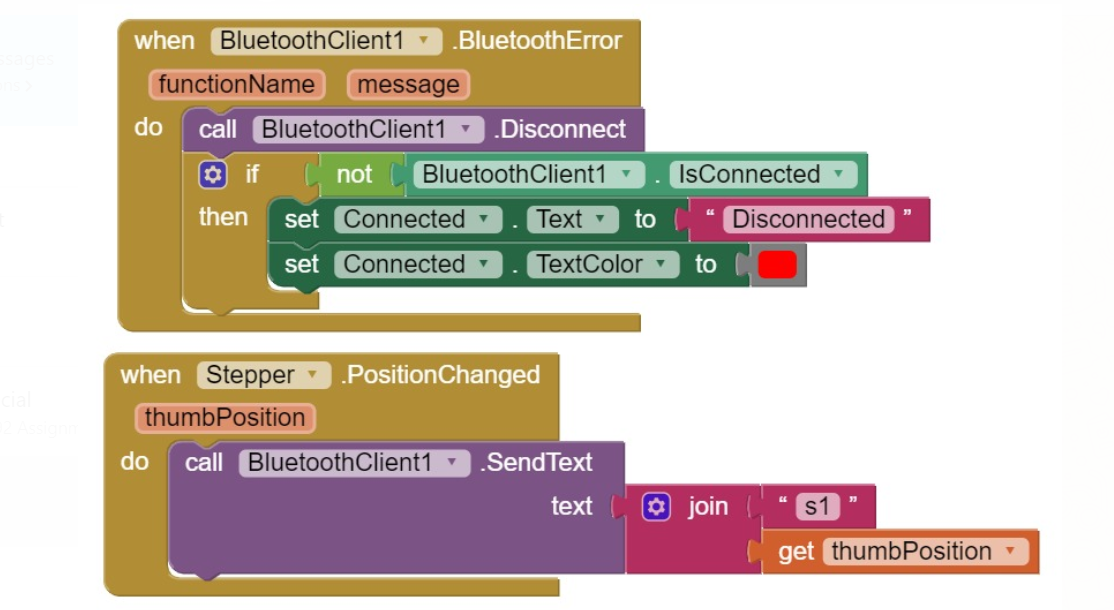


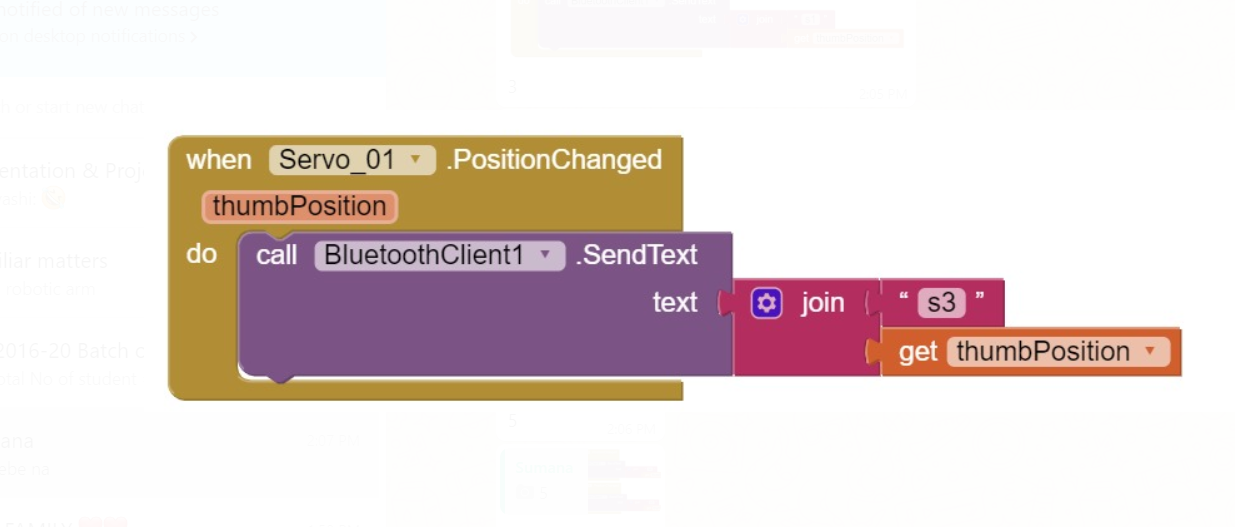
Figure 12

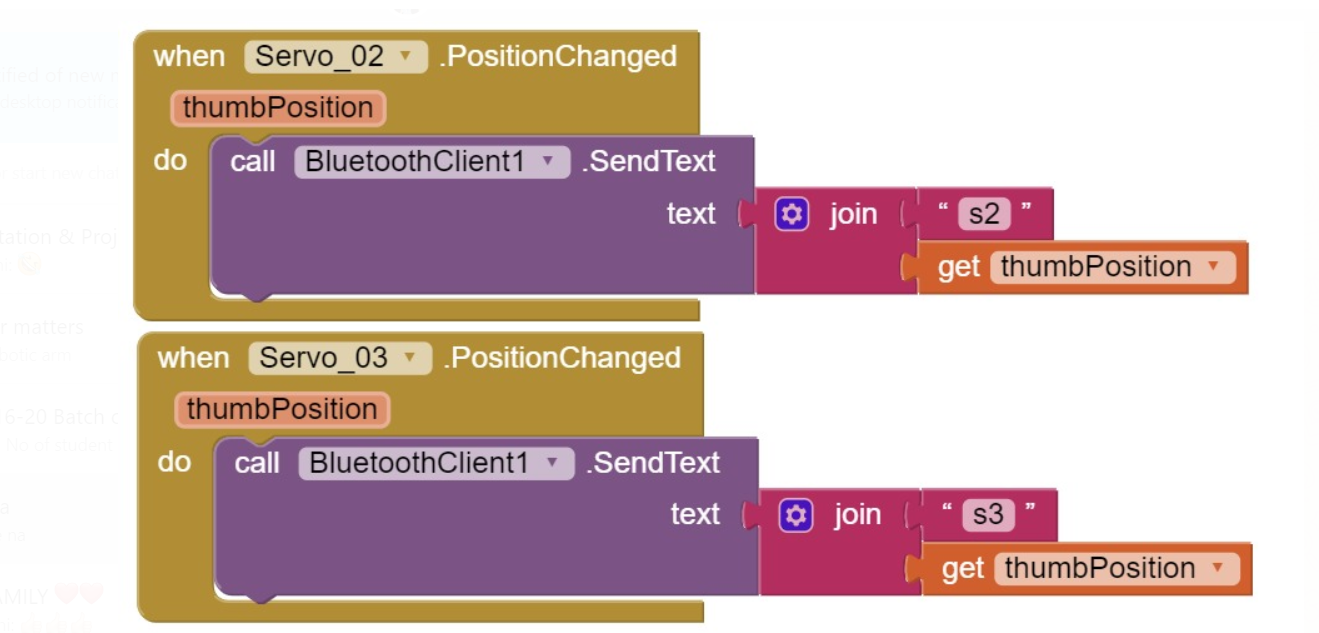
the design of the android application In the App Inventor Block section, we create the code that will work in the background of the application we are doing with the interlocking blocks with code fragments, just like a Lego. We make sure that all the components that I add to my application have all the events/properties that can be used and combine them into the corresponding blokes. In the project we have done, android blocks are shown in Figures 13, 14, 15, 16.

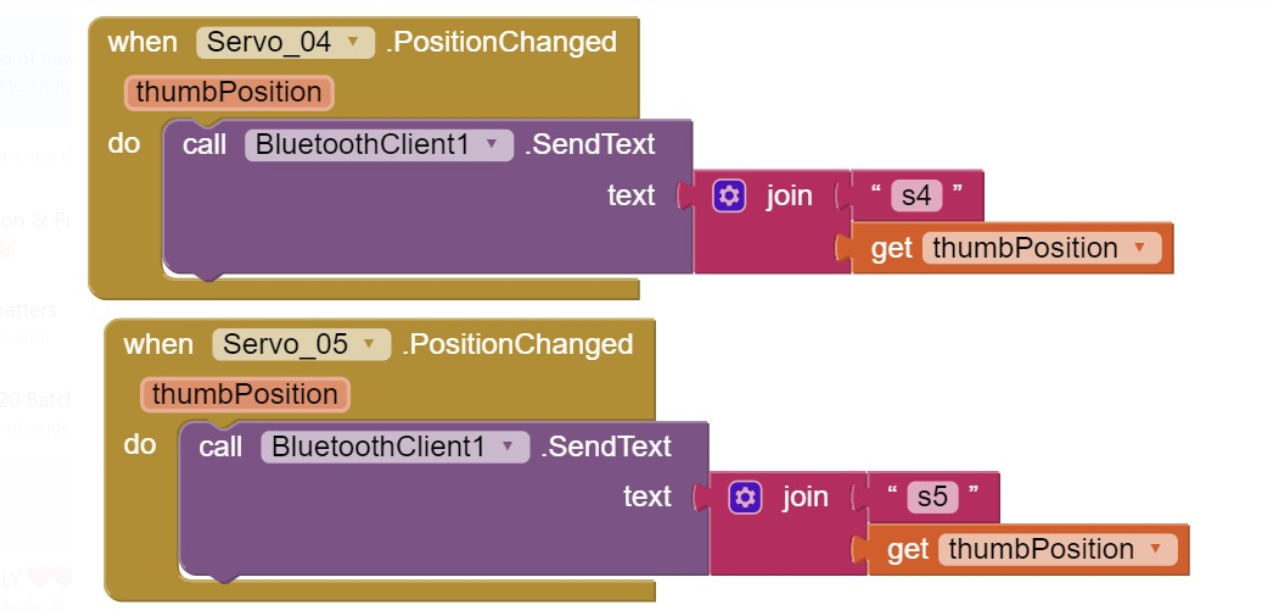












### 5.ARDUINO PROGRAMMING

#include <SoftwareSerial.h>

#include <Servo.h>

#include <Stepper.h>

Servo servo01;

Servo servo02;

Servo servo03;

Servo servo04;

Servo servo05;

SoftwareSerial Bluetooth(0, 1); // Arduino(RX, TX) - HC-06 Bluetooth (TX, RX)

int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos; // current position

int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos; // previous position

int index = 0;

String dataIn = "";

void setup() {

servo01.attach(8);

servo02.attach(9);

servo03.attach(10);

servo04.attach(11);

servo05.attach(12);

Bluetooth.begin(38400); // Default baud rate of the Bluetooth module

Bluetooth.setTimeout(1);

delay(20);

// Robot arm initial position

servo1PPos = 90;

servo01.write(servo1PPos);

servo2PPos = 150;

servo02.write(servo2PPos);

servo3PPos = 35;

servo03.write(servo3PPos);

servo4PPos = 140;

servo04.write(servo4PPos);

servo5PPos = 85;

servo05.write(servo5PPos);

}

void loop() {

// Check for incoming data

if (Bluetooth.available() > 0) {

dataIn = Bluetooth.readString(); // Read the data as string

// If "Waist" slider has changed value - Move Servo 1 to position

if (dataIn.startsWith("s1")) {

String dataInS = dataIn.substring(2, dataIn.length()); // Extract only the number. E.g. from "s1120" to "120"

servo1Pos = dataInS.toInt(); // Convert the string into integer

// We use for loops so we can control the speed of the servo

// If previous position is bigger then current position

if (servo1PPos > servo1Pos) {

for ( int j = servo1PPos; j >= servo1Pos; j--) { // Run servo down

servo01.write(j);

delay(20); // defines the speed at which the servo rotates

}

}

// If previous position is smaller then current position

if (servo1PPos < servo1Pos) {

for ( int j = servo1PPos; j <= servo1Pos; j++) { // Run servo up

servo01.write(j);

delay(20);

}

}

servo1PPos = servo1Pos; // set current position as previous position

}

// Move Servo 2

if (dataIn.startsWith("s2")) {

String dataInS = dataIn.substring(2, dataIn.length());

servo2Pos = dataInS.toInt();

if (servo2PPos > servo2Pos) {

for ( int j = servo2PPos; j >= servo2Pos; j--) {

servo02.write(j);

delay(50);

}

}

if (servo2PPos < servo2Pos) {

for ( int j = servo2PPos; j <= servo2Pos; j++) {

servo02.write(j);

delay(50);

}

}

servo2PPos = servo2Pos;

}

// Move Servo 3

if (dataIn.startsWith("s3")) {

String dataInS = dataIn.substring(2, dataIn.length());

servo3Pos = dataInS.toInt();

if (servo3PPos > servo3Pos) {

for ( int j = servo3PPos; j >= servo3Pos; j--) {

servo03.write(j);

delay(30);

}

}

if (servo3PPos < servo3Pos) {

for ( int j = servo3PPos; j <= servo3Pos; j++) {

servo03.write(j);

delay(30);

}

}

servo3PPos = servo3Pos;

}

// Move Servo 4

if (dataIn.startsWith("s4")) {

String dataInS = dataIn.substring(2, dataIn.length());

servo4Pos = dataInS.toInt();

if (servo4PPos > servo4Pos) {

for ( int j = servo4PPos; j >= servo4Pos; j--) {

servo04.write(j);

delay(30);

}

}

if (servo4PPos < servo4Pos) {

for ( int j = servo4PPos; j <= servo4Pos; j++) {

servo04.write(j);

delay(30);

}

}

servo4PPos = servo4Pos;

}

// Move Servo 5

if (dataIn.startsWith("s5")) {

String dataInS = dataIn.substring(2, dataIn.length());

servo5Pos = dataInS.toInt();

if (servo5PPos > servo5Pos) {

for ( int j = servo5PPos; j >= servo5Pos; j--) {

servo05.write(j);

delay(30);

}

}

if (servo5PPos < servo5Pos) {

for ( int j = servo5PPos; j <= servo5Pos; j++) {

servo05.write(j);

delay(30);

}

}

servo5PPos = servo5Pos;

}

}

//Stepper Motor

int step\_pin\_1 = 8;

int step\_pin\_2 = 9;

int step\_pin\_3 = 10;

int step\_pin\_4 = 11;

float delay\_time;

int value\_bluetooth = 0;

String readString;

void setup()

{

Serial.begin(9600);

}

void loop()

{

while (Serial.available()){

char c= Serial.read();

readString+=c;

}

if(readString.length() >0) {

value\_bluetooth =readString.toInt();

if (value\_bluetooth > 0 ){

delay\_time = 15;

}

else

{

delay\_time=0;

}

Serial.println(value\_bluetooth);

digitalWrite(8, HIGH); digitalWrite(step\_pin\_2, HIGH); digitalWrite(step\_pin\_3, LOW); digitalWrite(step\_pin\_4, LOW);

delay(delay\_time);

digitalWrite(step\_pin\_1, LOW); digitalWrite(step\_pin\_2, HIGH); digitalWrite(step\_pin\_3, HIGH); digitalWrite(step\_pin\_4, LOW);

delay(delay\_time);

digitalWrite(step\_pin\_1, LOW); digitalWrite(step\_pin\_2, LOW); digitalWrite(step\_pin\_3, HIGH); digitalWrite(step\_pin\_4, HIGH);

delay(delay\_time);

digitalWrite(step\_pin\_1, HIGH); digitalWrite(step\_pin\_2, LOW); digitalWrite(step\_pin\_3, LOW); digitalWrite(step\_pin\_4, HIGH);

delay(delay\_time);

readString="";

}

}

}

### 6. EXPERIMENTAL STUDIES

Experimental studies are within themselves; the method followed in the project, robot arm control system, and programming. In the method and robot arm control system part of the project, the basic commands are explained after the basic information is given and the robot can perform the required tasks. In the program section, there is the software information required for the movement of the robot. The programming part of the pro- gram is given in Appendix to the project to make the project more regular.

### 6.1. Projected Method

First, a historical research on robot arms was carried out and the basic information needed to establish the system was obtained. The robot used in the project types with arm joint and can move in 4 axis directions (left and right, up and down) and also can hold and swing motion thanks to the holder on it. The microcontroller Arduino Nano is used to provide optimal control of the robot arm. The reason for preferring this microcontroller is that it is more accessible to be able to get a solution to a possible error because the open source code is easier to use than the other microcontrollers and the number of users is higher. After these studies, detailed information has been obtained about the servo motors to be used. The servomotor is preferred because it can be carried out smoothly in the robot project, the motor can be operated precisely and it must be at high torque. The robot arm, 5 servo motors are formed. Servo motors are numbered from top to bottom in order to explain their tasks because of the excess.

### 6.2. Battery

### Three lines from battery are going to buck converters.The buck converters are converting the voltages into the following voltages respectively-

### 1. 5Volt:

### It goes directly to the microcontroller from the buck converter.

### 2. 6.6Volt:

### It splits into two lines-First one goes to the Servo motors.It can rotate upto 16 servos but here we are using only five servo.The second one goes to the Stepper controller.

### 3. 12Volt: It goes to the Gear motor for driving the bot around.

### 6.3 Stepper Motor Controller

### We are using only one Stepper controller.Stepper controller can control only one stepper at a time.It has two connections connected to the microcontroller.

### 6.4 Servo Motor Controller

### We are using four motors and two motor controllers. We are using L298n motor controller.It has six communication lines, three for each motor.One line is used for speed and the rest two for direction.

### 6.5Robot Arm Control

The connection box is made to distribute the supply source to the servo motors. In doing so, servo motor inputs, Arduino pin inputs, and communication circuit elements are used. The mechanical part of the robot arm is designed by combining the pre-selected parts appropriately. In order to move the robotic arm properly, software with the selected Arduino microcontroller has been implemented and then the experiment with the Bluetooth module and servo motors has been done to learn about the system operation. The software has been implemented with the appropriate Arduino microcontroller selected so that the robot arm can be moved appropriately for the intended purpose. Control of the robot arm is achieved by moving the axes of the android application in the '-' and '+' directions.

We can connect the microcontroller with routers,bluetooth,wifi.Here we are using bluetooth to connect with mobile and controlling the arm with the app.We can also use webpage to connect the mictrocontroller but in that case it will put a lot of pressure to the microcontroller.Since the microcontroller already has to control the motor,servo and stepper so it will not be a good idea to put more pressure into it.When we are using the app it makes the mobile as server and microcontroller as client. So in this way we can control multiple robots like this using the app.

### 7.FUTURE SCOPE

**Medical Science** - Brain Computer Interface (BCI) is an immerging field of

research, can be used to acquire signals from the human brain and control the

arm. They may perform complex operations.

**Clothing Retail Industry** - Help the users to feel the texture of the clothes on

the internet.

**Mechanical Design** - More efficient, reliable, improved power. Shoulder,

elbow & wrist movement allowing circular & angular rotations.

**Universal Gripper** - Capable of doing multiple tasks.

**Intelligence** - Capable of making decisions about the task it performs.

### 8. PROBLEMS WHILE MAKING ROBOTIC ARM

### As the global pandemic CORONA VIRUS(COVID-19) disease has affected everything around us,we too faced some problems while making the arm.

### We have used L298n motor driver which is 70% efficient and gets heated some times because due to lockdown we were not able to arrange Mosfet servo arduino which is 90% efficient and can work for long time.

### Due to lockdown we were not being able to modify the arm anymore.

### 9. CONCLUSION

Robotic arms, many areas are developable. Thanks to the robotic arms, many tasks are made easier and the resulting error level has been reduced to a minimum. For example; some pharmacy-based drug-giving robots and a projected robot arm have been developed. In addition to this, the ability to move the robot arm is further increased, and when the camera is placed in the finger area and the sensitivity is increased, it can be used in a wide range of applications from the medical sector to the automation systems. With the robotic arms developed in this way, the risk of 1 Axis + 1 Axis - 2 Axis + 2 Axis - 3 Axis + 3 Axis - 4 Axis + 4 Axis - Gripper + Gripper - Save Play BT Names Page | 24 infecting the patient in the medical sector is minimized, while the human errors are minimized during the surgical intervention. Despite the fact that the robotic arm made by this project is of prototype quality, it has a quality that can be improved for more robotic systems. Besides these, robotic arm sector, which is open to development, will keep its importance in the future. The purpose of the project is to provide control of 4 axes moving robot arm design and this robot arm with a suitable microcontroller and Bluetooth module with android application. The necessary theoretical and practical information for this purpose has been obtained and the necessary infrastructure has been established for the project. During the process of making and developing the project, a lot of theoretical knowledge has been transferred to the practice and it has been ensured that it is suitable for the purpose of the project.

### 10.BIBLIOGRAPHY

1. WMHW Kadir, RE Samin, BSK Ibrahim. Internet controlled a robotic arm. Procedia Engineering. 2012.

2. MAK Yusoff, RE Samin, mobile robotic arm. Procedia Engineering. 2012.

3. AM Al-Busaidi, Development of an educational environment for online control of a biped robot using MATLAB and Arduino, (MECHATRONICS), 9th France-Japan.2012.

4. HS Juang, KY Lurrr. Design and control of a two-wheel self-balancing robot using the Arduino microcontroller board. Control and Automation (ICCA), 2013.

5. R Krishna, GS Bala, SS ASC, BBP Sarma. Design and implementation of a robotic arm based on haptic technology. Int. J. of Eng. Research. 2012.

6. Electric Electronic Technology-Step and Servo Motors, SVET, 2007.