Universiti Malaysia PAHANG

Engineering • Technology • Creativity

FACULTY OF MANUFACTURING AND MECHATRONIC ENGINEERING TECHNOLOGY

BFM4503 - ROBOTICS FOR ENGINEERS

SECTION 1

5 DOF ROBOT ARM WITH SMARTPHONE CONTROL

PROJECT REPORT

GROUP NO:

LECTURER'S NAME: DR. SMAYUZRI BIN ISHAK

GROUP MEMBERS NAMES & ID:

1- **MARWAN ABDULLAH SALEH AL-ABAB FB15108**
2- **YAHYA MOHAMMED AL-NAGGAR FB16046**

INTRODUCTION

In this project we learnt how to make an Arduino Robot Arm which is wirelessly controlled and programmed using a custom-build Android application. This project shows the entire process of building it, starting from designing and 3D printing the robot parts, connecting the electronic components and programming the Arduino, to developing the Android application for controlling the Robot Arm.

OBJECTIVES

- To apply the outcomes of what we have learnt in our class of **Robotics for engineers** during this semester.
- To design and develop a robot arm with a smartphone controller consist of five degrees of freedom.
- To build the developed robot arm using 3D printer.

METHODOLOGY

Using the sliders in the app we can manually control the movement of each servo or axis of the robot arm. Also using the “Save” button we can record each position or step and then the robot arm can automatically run and repeat these steps. With the same button we can pause the automatic operation as well as reset or delete all steps so that we can record new ones.

Robot Arm 3D Model

To begin with, we designed the Robot Arm using **Catia 05** 3D modeling software. The arm has 5 degrees of freedom as it shown in figure 1.

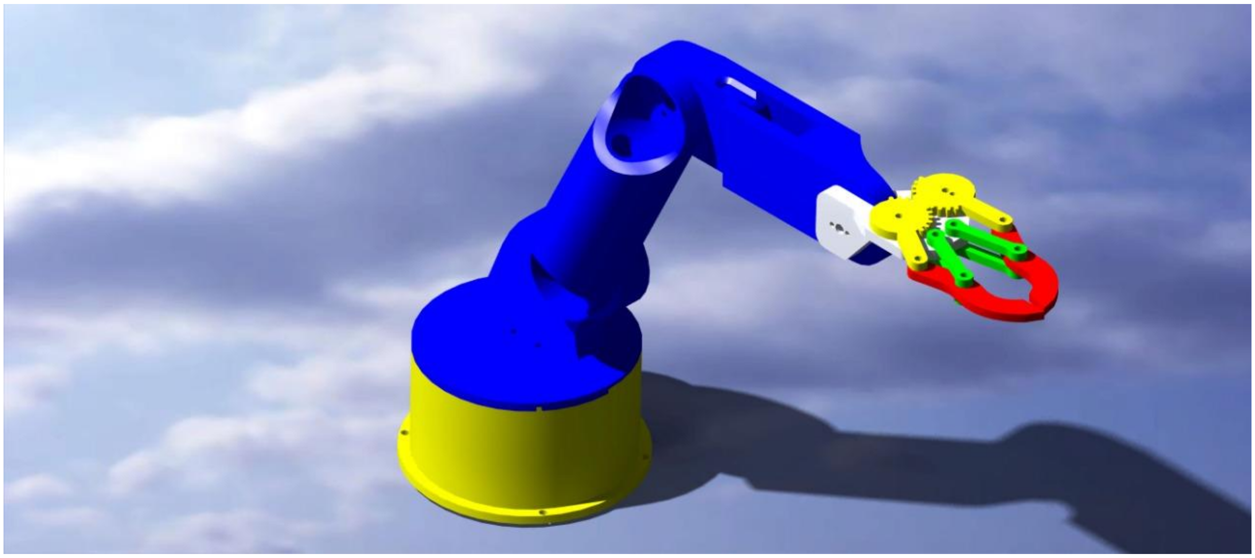


Figure 1 Design of Arm Robot

For the first 3 axis, the waist, the shoulder and the elbow, we have used the MG996R servos, and for the other 2 axis, the wrist roll and wrist pitch, as well as the gripper we used the smaller SG90 micro servos.

3D Printing the Robot Arm

Using a 3D Printer, we have printed all of the parts for the Arduino robot arm. The 3D printer printing quality is amazing for its price point and what's also great about it is that it comes almost 90% pre-assembled. In order to complete the assembly we just have to connect the upper and lower parts frames using some bolts and brackets, and then connect the electronic components with the control box using the provided cables. Before trying it, we were recommended to check whether the roller wheels are tight enough, and if they are not, we just simply use the eccentric nuts to tight them up. And that's it, after leveling our 3D printing bed, we are ready to transform our 3D creations into reality.

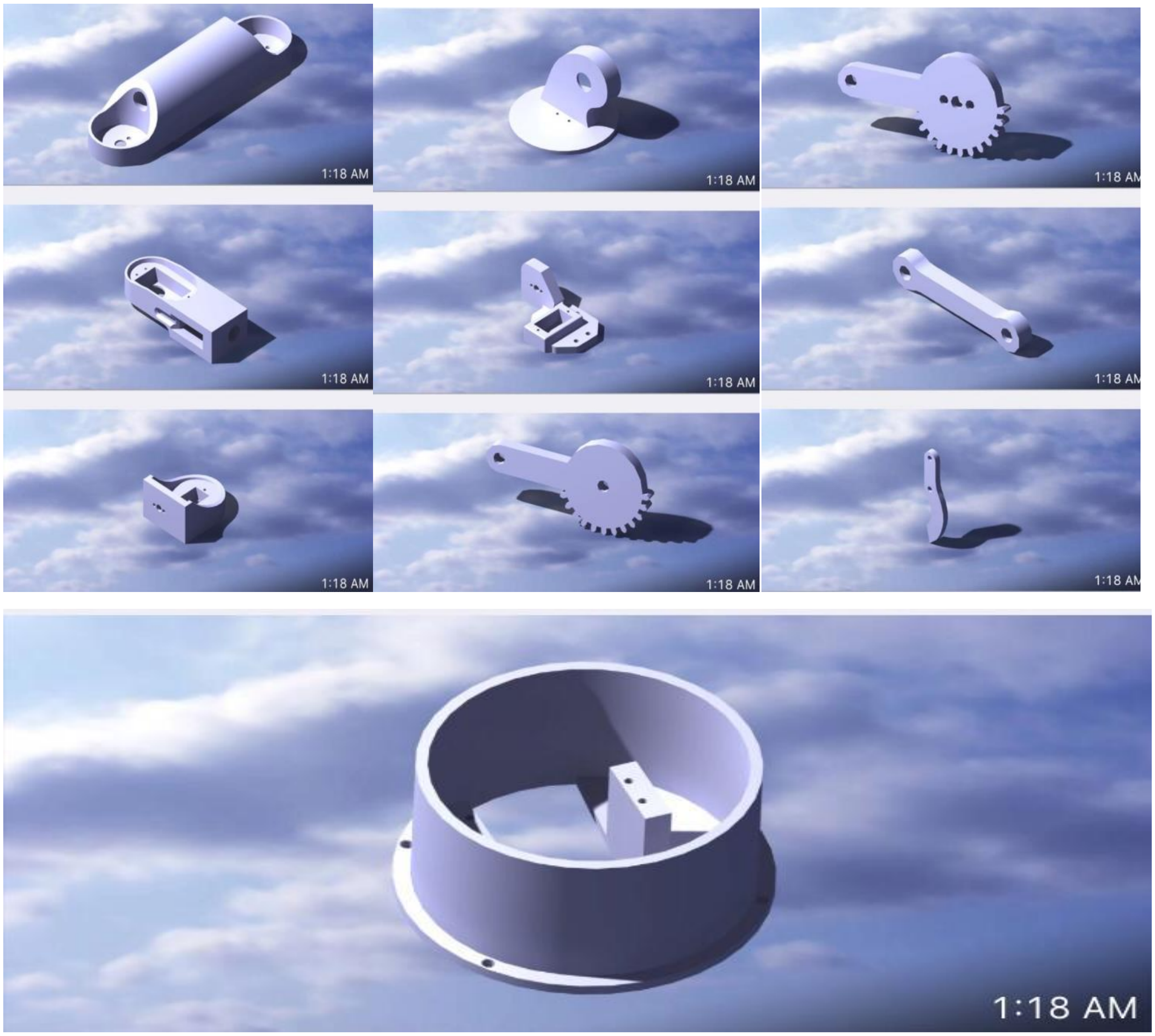


Figure 2 Robot Arm Parts

Assembling the Robot Arm

At this point we were ready to assemble the robot arm. We started with the base on which we attached the first servo motor using the screws included in its package. Then on the output shaft of the servo we secured a round horn a bolt.

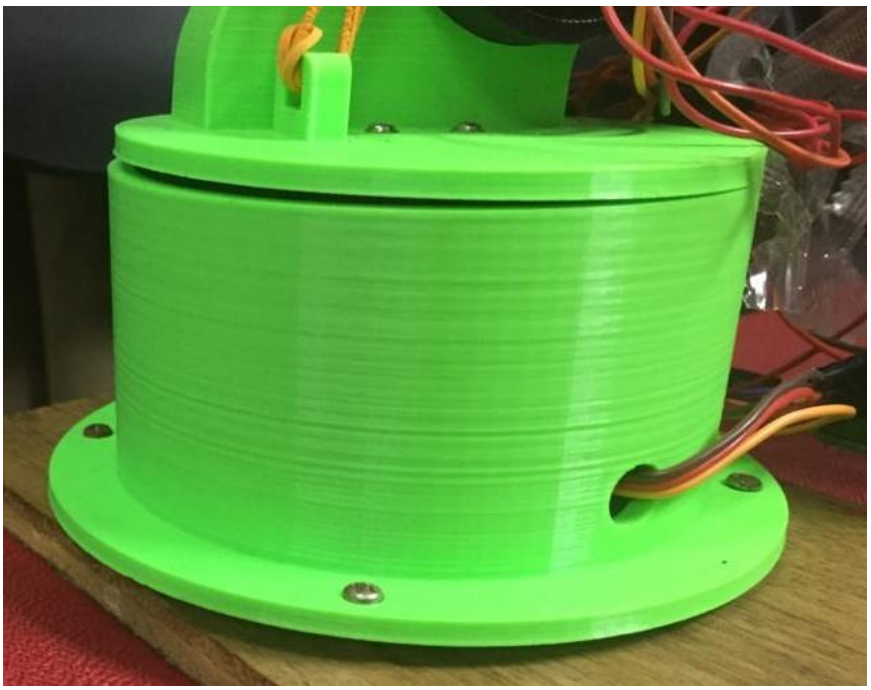


Figure 3 Base

And on top of the base we placed the upper part and secured it using two screws.

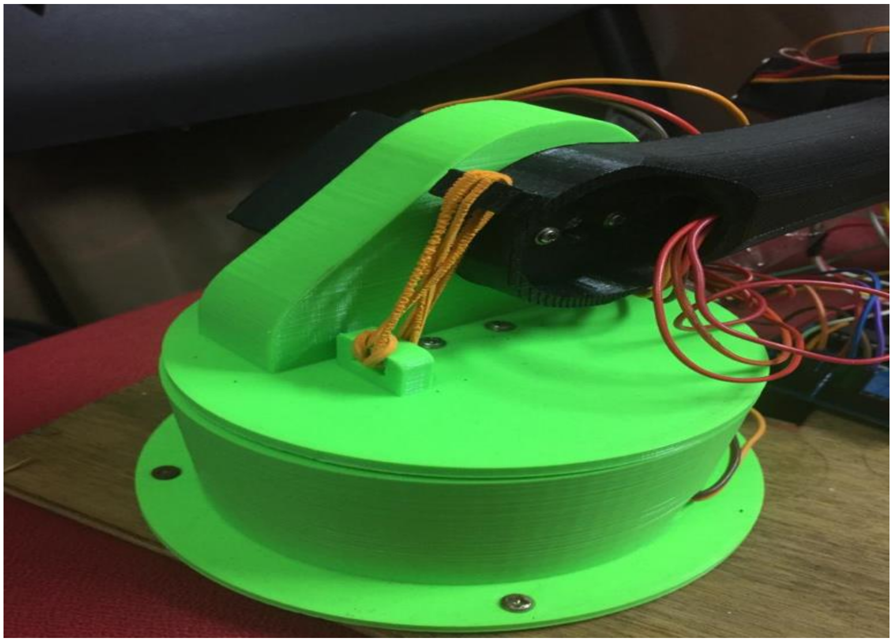


Figure 4 Upper part

Here again first goes servo, then the round horn onto the next part, and then they are secured to each other using the bolt on the output shaft.

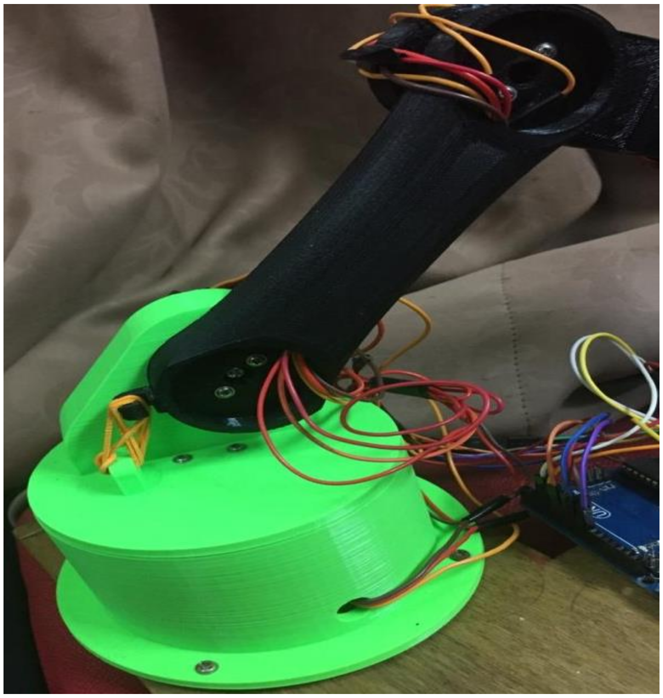


Figure 5 Round horn next part

It can be noticed here that at the shoulder axis it is good idea to include some kind of spring or in our case we used a rubber band to give some help to the servo because this servo carries the whole weight of the rest of the arm as well as the payload.

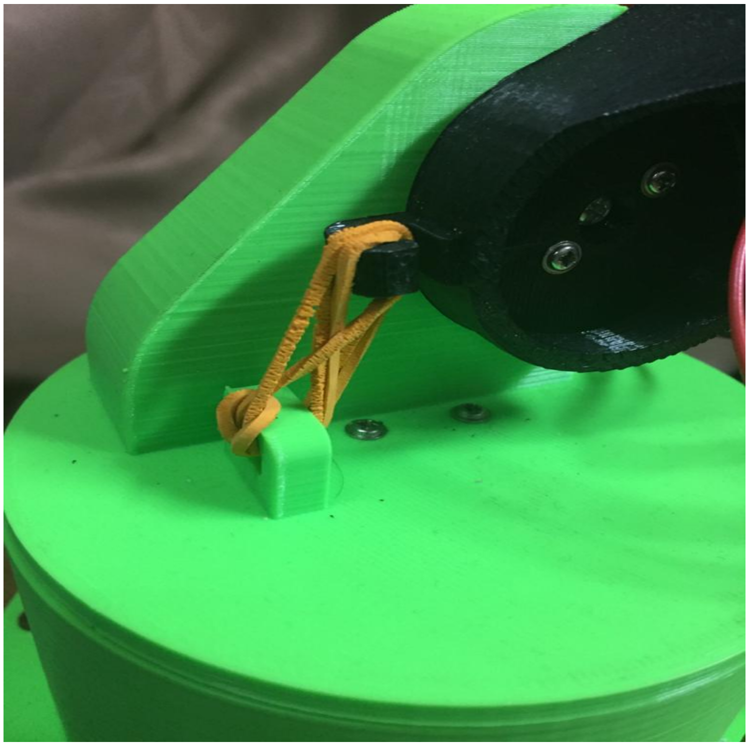


Figure 6 Rubber band

In similar way we continued to assemble the rest of the robot arm. As for the gripper mechanism we used some 4 millimeters bolts and nuts to assembly it.

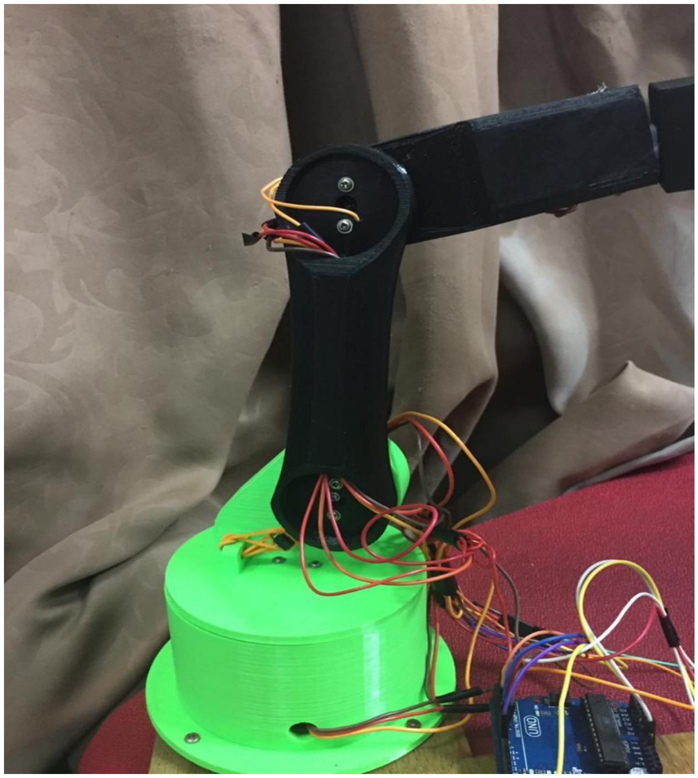


Figure 7 Middle connecting point

Finally we attached the gripper mechanism onto the last servo and the robot arm was completed.

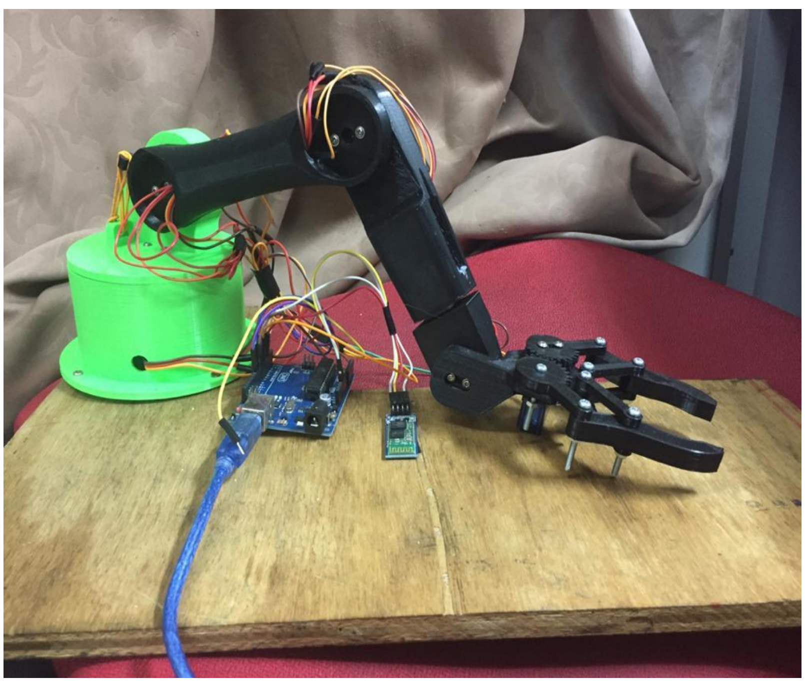


Figure 8 Gripper connected to robot arm

Robot Arm Circuit Diagram

The next stage is connecting the electronics. The circuit diagram of this project is actually quite simple. We just need an Arduino board and a HC-05 Bluetooth module for communication with the smartphone. The control pins of the six servo motors are connected to six digital pins of the Arduino board.

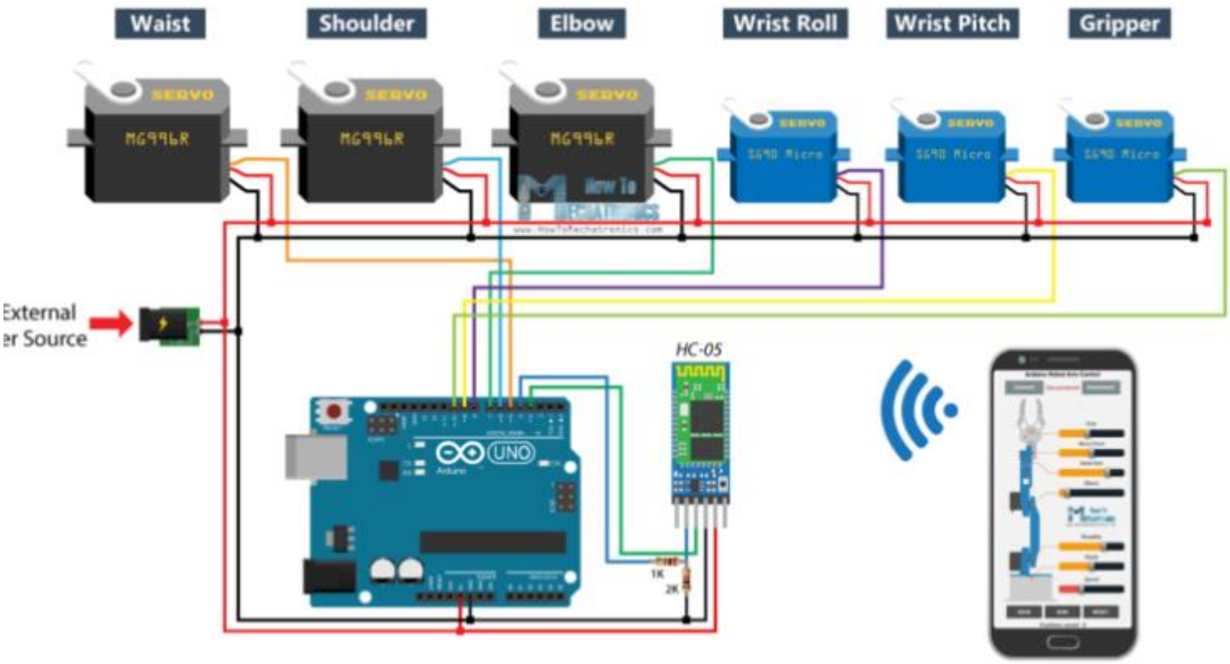


Figure 9 Circuit Diagram

For powering the servos we need 5V, but this must come from an external power source because the Arduino is not able to handle the amount of current that all of them can draw. The power source must be able to handle at least 2A of current.

Robot Arm Code

First we included 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 no need 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.

```
#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 position
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50]; // for storing positions/steps
int speedDelay = 20;
int index = 0;
String dataIn = "";
```

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.

```
void setup() {
  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(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);
  servo6PPos = 80;
  servo06.write(servo6PPos);
}
```

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.

```
// Check for incoming data
  if (Bluetooth.available() > 0) {
    dataIn = Bluetooth.readString();  // Read the data as string
```

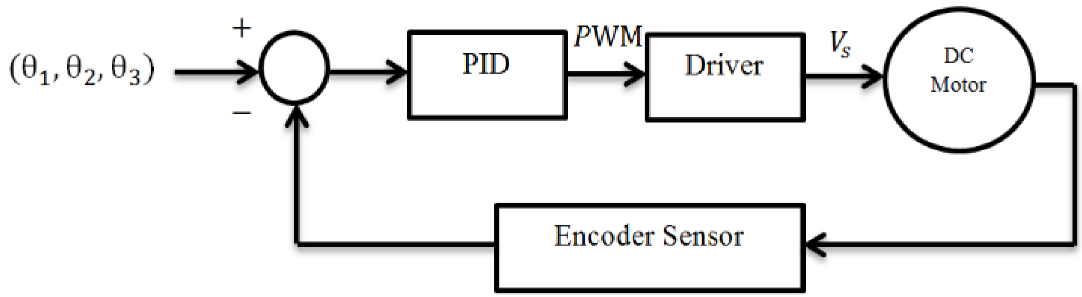
Robot Arm Control Android App

Figure 10 Robot arm controller

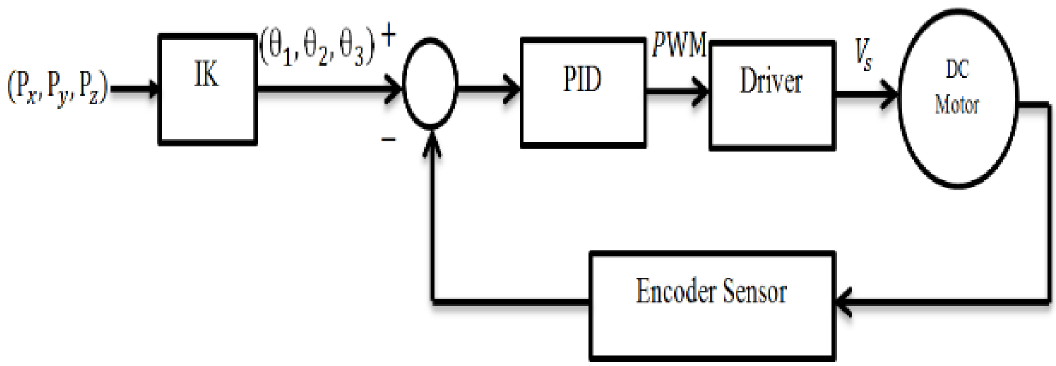
At the top we have two buttons for connecting the smartphone to the HC-05 Bluetooth module. Then on the left side we have an image of the robot arm, and on the right side we have the six slider for controlling the servos and one slider for the speed control. Each slider have different initial, minimum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

SYSTEM BLOCK DIAGRAM

Forward Kinematics



Inverse Kinematics



CALCULATION

Forward Kinematic

D-H Algorithm Coordinate

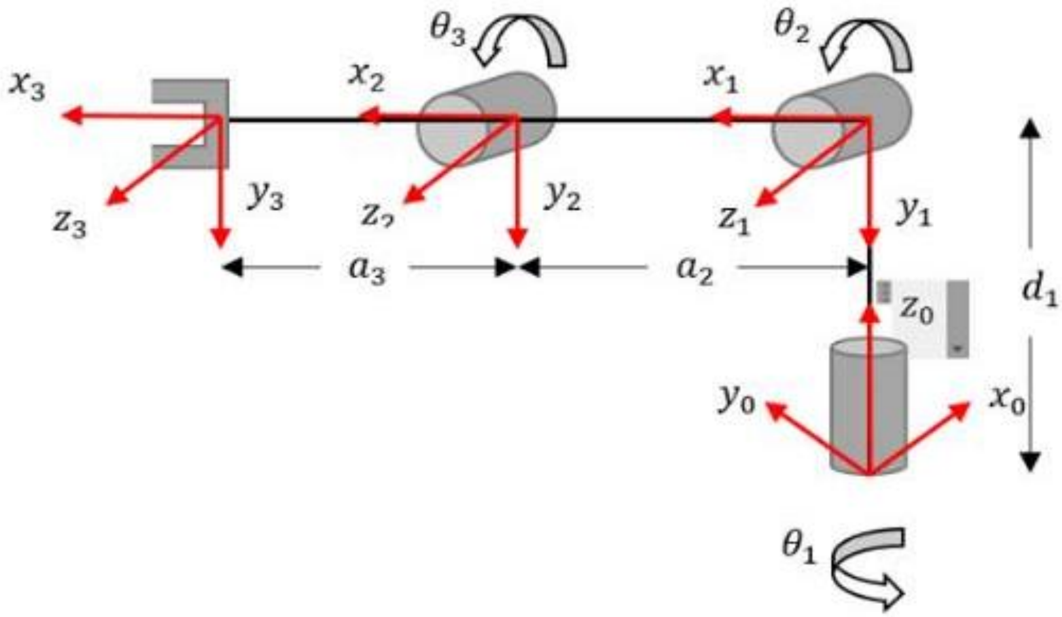


Fig.11 Link Coordinate System

TABLE 1.

i	θ_i	α_i	a_i	d_i
1	θ_1	90°	0	d_1
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0

Forward Kinematic

$$
{}^{0}_{1}A = \begin{bmatrix} \cos\theta_1 & \cos\theta_1 \sin\theta_1 & \sin\theta_1\ sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 \cos\theta_1 & \sin\theta_1 \cos\theta_1 & 0 \\ 0 & \sin\theta_1 & \cos\theta_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{1}_{2}A = \begin{bmatrix} \cos\theta_2 & \cos\theta_2 \sin\theta_2 & \sin\theta_2 \sin\theta_2 & a_2 \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 \cos\theta_2 & \sin\theta_2 \cos\theta_2 & a_2 \sin\theta_2 \\ 0 & \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

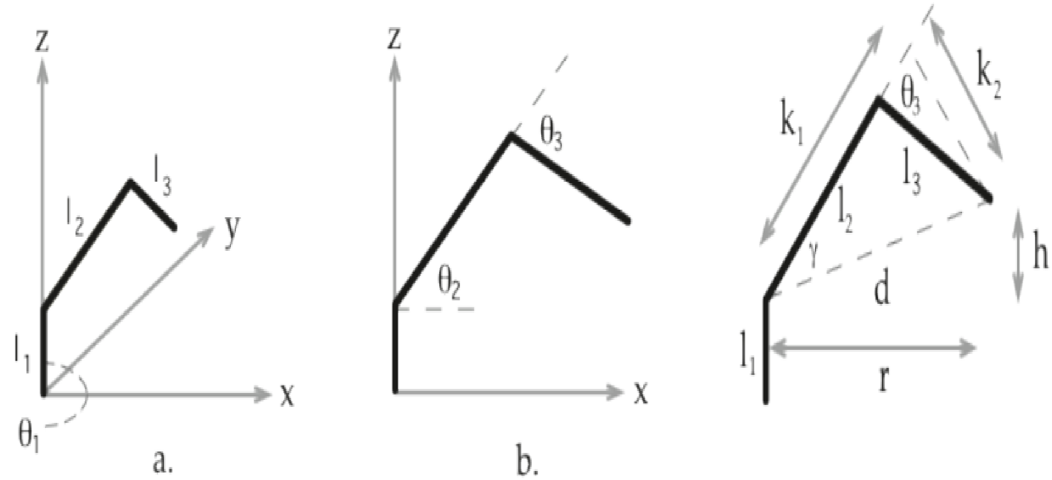
$$
{}^{2}_{3}A = \begin{bmatrix} \cos\theta_3 & \cos\theta_3 \sin\theta_3 & \sin\theta_3 \sin\theta_3 & a_3 \cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 \cos\theta_3 & \sin\theta_3 \cos\theta_3 & a_3 \sin\theta_3 \\ 0 & \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{0}_{3}T = {}^{0}_{1}A\,{}^{1}_{2}A\,{}^{2}_{3}A
$$

$$
= \begin{bmatrix} c_1\, c_{23} & -c_1\, s_{2-3} & s_1 & a_3\, c_1\, c_{23} + a_2\, c_1\, c_2 \\ s_1\, c_{23} & -s_1\, s_{23} & -c_1 & a_3 s_1\, c_{23} + a_2\, s_1\, c_2 \\ s_{23} & c_{23} & 0 & a_3\, s_{23} + a_2\, s_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
\therefore\ \left(P_x, P_y, P_z\right)^T = \begin{bmatrix} a_3\, c_1\, c_{23} + a_2\, c_1\, c_2 \\ a_3\, s_1\, c_{23} + a_2\, s_1\, c_2 \\ a_3\, s_{23} + a_2\, s_2 + d_1 \end{bmatrix}
$$

Inverse Kinematic



$P_x = a_3 c_1 c_{23} + a_2 c_1 c_2$ (1)

$P_y = a_3 s_1 c_{23} + a_2 s_1 c_2$ (2)

$P_z = a_3 s_{23} + a_2 s_2 + d_1$ (3)

$r = \sqrt{(x)^2 + (y)^2}$:

$= \sqrt{(a_3 c_{23} + a_2 c_2)^2 c_1{}^2 + (a_3 c_{23} + a_2 c_2)^2 s_1{}^2}$

$= a_3 c_{23} + a_2 c_2$

h = z - d_1

$= a_3 s_{23} + a_2 s_2$

$r^2 + h^2 = (x)^2 + (y)^2 + (\mathrm{z} - d_1)^2$

$= a_3{}^2 c_{23}{}^2 + a_2{}^2 c_2{}^2 + 2a_2 a_3 c_{23} c_2 + a_3{}^2 s_{23}{}^2 + a_2{}^2 s_2{}^2 + 2a_2 a_3 s_{23} s_2$

$= a_3{}^2 + a_2{}^2 + 2a_2 a_3 c_2 \left(c_2 c_3 - s_2 s_3\right) + 2a_2 a_3 s_2 (s_2 c_3 - c_2 s_3$

$= a_3{}^2 + a_2{}^2 + 2a_2 a_3 c_2{}^2 c_3 - 2a_2 a_3 s_2 s_3 c_2 + 2a_2 a_3 s_2{}^2 c_3 + 2a_2 a_3 s_2 s_3 c_2$

$= a_3{}^2 + a_2{}^2 + 2a_2 a_3 c_2{}^2 c_3 + 2a_2 a_3 s_2{}^2 c_3$

$= a_3{}^2 + a_2{}^2 + 2a_2 a_3 (c_2{}^2 c_3 + s_2{}^2 c_3)$

$= a_3{}^2 + a_2{}^2 + 2a_2 a_3 c_3$

$(x)^2 + (y)^2 + (\mathrm{z} - d_1)^2 = a_3{}^2 + a_2{}^2 + 2a_2 a_3 c_3$

$c_3 = \frac{(x)^2+(y)^2+(z-d_1)^2-a_3{}^2-a_2{}^2}{2a_2\, a_3}$

For up elbow, $s_3 = +\sqrt{1-\left[\frac{(x)^2+(y)^2+(z-d_1)^2-a_3{}^2-a_2{}^2}{2a_2\, a_3}\right]^2}$

θ_3

$= atan2$

$(\sqrt{1-\left[\frac{(x)^2+(y)^2+(z-d_1)^2-a_3{}^2-a_2{}^2}{2a_2\, a_3}\right]^2}, \frac{(x)^2+(y)^2+(z-d_1)^2-a_3{}^2-a_2{}^2}{2a_2\, a_3})$

Let $k_1 = a_2 + a_3 c_3$, $k_2 = a_3 s_3$

r $= a_2 c_2 + a_3 c_{23}$

$= a_2 c_2 + a_3 c_2 c_3 - a_3 s_2 s_3$

$= k_1 c_2 - k_2 s_2$

h $= a_3 s_{23} + a_2 s_2$

$= a_2 s_2 + a_3 s_2 c_3 + a_3 c_2 s_3$

$= k_1 s_2 + k_2 c_2$

If $k_1 = d\cos\phi$, $k_2 = d\sin\phi$

$d = +\sqrt{k_1{}^2 + k_2{}^2}$

$\alpha = \text{atan2}\,(k_2, k_1)$

$r = dc_\alpha c_2 - ds_\alpha s_2 = dc_{2-\alpha}$

$h = dc_\alpha s_2 + ds_\alpha c_2 = ds_{2-\alpha}$

$\theta_2 - \alpha = \text{atan2}\,(\frac{h}{d}, \frac{r}{d})$

$= \text{atan2}\,(h, r)$ since d>0

$\theta_1 = \text{atan2}\,(x, y)$

$\theta_2 = \text{atan2}\,(z - d_1, \sqrt{(x)^2+(y)^2}$

$-atan2\,(a_3\sqrt{1-\left[\frac{(x)^2+(y)^2+(z-d_1)^2-a_3{}^2-a_2{}^2}{2a_2\, a_3}\right]^2}, a_2 + a_3\frac{(x)^2+(y)^2+(z-d_1)^2-a_3{}^2-a_2{}^2}{2a_2\, a_3})$

CONCLUSION

This project is not fully function but the last three degrees of freedom has been working well and perfectly from gripper and the two degree of freedom near to it, while the other two degree of freedom from base and near to it, they have some issues due to using different motors and these motor needs high power. As only two members in this group we have faced some problem like it has many works form Designing and wiring to coding but we have learnt a lot from this subject and it made us able to deal with many things that we have not knew them before beside it teach us how to be able to take decision and deal with time management. Lastly, we would like to thank our lecturer **DR. SMAYUZRI BIN ISHAK** for everything that he has done for us during this semester. Below is the main reference of our project.

https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control/

Full code

```
#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 position
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50]; // for storing positions/steps
int speedDelay = 20;
int index = 0;
String dataIn = "";

void setup() {
servo01.attach(5);
servo02.attach(6);
servo03.attach(7);
servo04.attach(8);
servo05.attach(9);
servo06.attach(10);
Bluetooth.begin(38400); // 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);
servo6PPos = 80;
servo06.write(servo6PPos);
}

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;
}
// Move Servo 6
if (dataIn.startsWith("s6")) {
String dataInS = dataIn.substring(2, dataIn.length());
servo6Pos = dataInS.toInt();
if (servo6PPos > servo6Pos) {
for ( int j = servo6PPos; j >= servo6Pos; j--) {
servo06.write(j);
```

```
delay(30);
}
}
if (servo6PPos < servo6Pos) {
for ( int j = servo6PPos; j <= servo6Pos; j++) {
servo06.write(j);
delay(30);
}
}
servo6PPos = servo6Pos;
}
// If button "SAVE" is pressed
if (dataIn.startsWith("SAVE")) {
servo01SP[index] = servo1PPos; // save position into the array
servo02SP[index] = servo2PPos;
servo03SP[index] = servo3PPos;
servo04SP[index] = servo4PPos;
servo05SP[index] = servo5PPos;
servo06SP[index] = servo6PPos;
index++; // Increase the array index
}
// If button "RUN" is pressed
if (dataIn.startsWith("RUN")) {
runservo(); // Automatic mode - run the saved steps
}
// If button "RESET" is pressed
if ( dataIn == "RESET") {
memset(servo01SP, 0, sizeof(servo01SP)); // Clear the array data to 0
memset(servo02SP, 0, sizeof(servo02SP));
memset(servo03SP, 0, sizeof(servo03SP));
memset(servo04SP, 0, sizeof(servo04SP));
memset(servo05SP, 0, sizeof(servo05SP));
memset(servo06SP, 0, sizeof(servo06SP));
index = 0; // Index to 0
}
}
}

// Automatic mode custom function - run the saved steps
void runservo() {
while (dataIn != "RESET") { // Run the steps over and over again until "RESET" button is pressed
for (int i = 0; i <= index - 2; i++) { // Run through all steps(index)
if (Bluetooth.available() > 0) { // Check for incomding data
dataIn = Bluetooth.readString();
if ( dataIn == "PAUSE") { // If button "PAUSE" is pressed
while (dataIn != "RUN") { // Wait until "RUN" is pressed again
if (Bluetooth.available() > 0) {
dataIn = Bluetooth.readString();
if ( dataIn == "RESET") {
break;
}
}
}
```

```
}
// If speed slider is changed
if (dataIn.startsWith("ss")) {
String dataInS = dataIn.substring(2, dataIn.length());
speedDelay = dataInS.toInt(); // Change servo speed (delay time)
}
}
// Servo 1
if (servo01SP[i] == servo01SP[i + 1]) {
}
if (servo01SP[i] > servo01SP[i + 1]) {
for ( int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {
servo01.write(j);
delay(speedDelay);
}
}
if (servo01SP[i] < servo01SP[i + 1]) {
for ( int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {
servo01.write(j);
delay(speedDelay);
}
}

// Servo 2
if (servo02SP[i] == servo02SP[i + 1]) {
}
if (servo02SP[i] > servo02SP[i + 1]) {
for ( int j = servo02SP[i]; j >= servo02SP[i + 1]; j--) {
servo02.write(j);
delay(speedDelay);
}
}
if (servo02SP[i] < servo02SP[i + 1]) {
for ( int j = servo02SP[i]; j <= servo02SP[i + 1]; j++) {
servo02.write(j);
delay(speedDelay);
}
}

// Servo 3
if (servo03SP[i] == servo03SP[i + 1]) {
}
if (servo03SP[i] > servo03SP[i + 1]) {
for ( int j = servo03SP[i]; j >= servo03SP[i + 1]; j--) {
servo03.write(j);
delay(speedDelay);
}
}
if (servo03SP[i] < servo03SP[i + 1]) {
for ( int j = servo03SP[i]; j <= servo03SP[i + 1]; j++) {
servo03.write(j);
delay(speedDelay);
}
```

```
}

// Servo 4
if (servo04SP[i] == servo04SP[i + 1]) {
}
if (servo04SP[i] > servo04SP[i + 1]) {
for ( int j = servo04SP[i]; j >= servo04SP[i + 1]; j--) {
servo04.write(j);
delay(speedDelay);
}
}
if (servo04SP[i] < servo04SP[i + 1]) {
for ( int j = servo04SP[i]; j <= servo04SP[i + 1]; j++) {
servo04.write(j);
delay(speedDelay);
}
}

// Servo 5
if (servo05SP[i] == servo05SP[i + 1]) {
}
if (servo05SP[i] > servo05SP[i + 1]) {
for ( int j = servo05SP[i]; j >= servo05SP[i + 1]; j--) {
servo05.write(j);
delay(speedDelay);
}
}
if (servo05SP[i] < servo05SP[i + 1]) {
for ( int j = servo05SP[i]; j <= servo05SP[i + 1]; j++) {
servo05.write(j);
delay(speedDelay);
}
}

// Servo 6
if (servo06SP[i] == servo06SP[i + 1]) {
}
if (servo06SP[i] > servo06SP[i + 1]) {
for ( int j = servo06SP[i]; j >= servo06SP[i + 1]; j--) {
servo06.write(j);
delay(speedDelay);
}
}
if (servo06SP[i] < servo06SP[i + 1]) {
for ( int j = servo06SP[i]; j <= servo06SP[i + 1]; j++) {
servo06.write(j);
delay(speedDelay);
}
}
}
}
```

	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15
Team Forming												
Group Discussion and brainstorm idea												
Project Planning												
First Prototype												
Forward Kinematic												
Arduino Coding												
Circuit Schematic Design												
GUI Design												
Robotic Arm Assembly												
Final Prototype Assembly												
Prototype Finalize												
Documentation And Presentation												