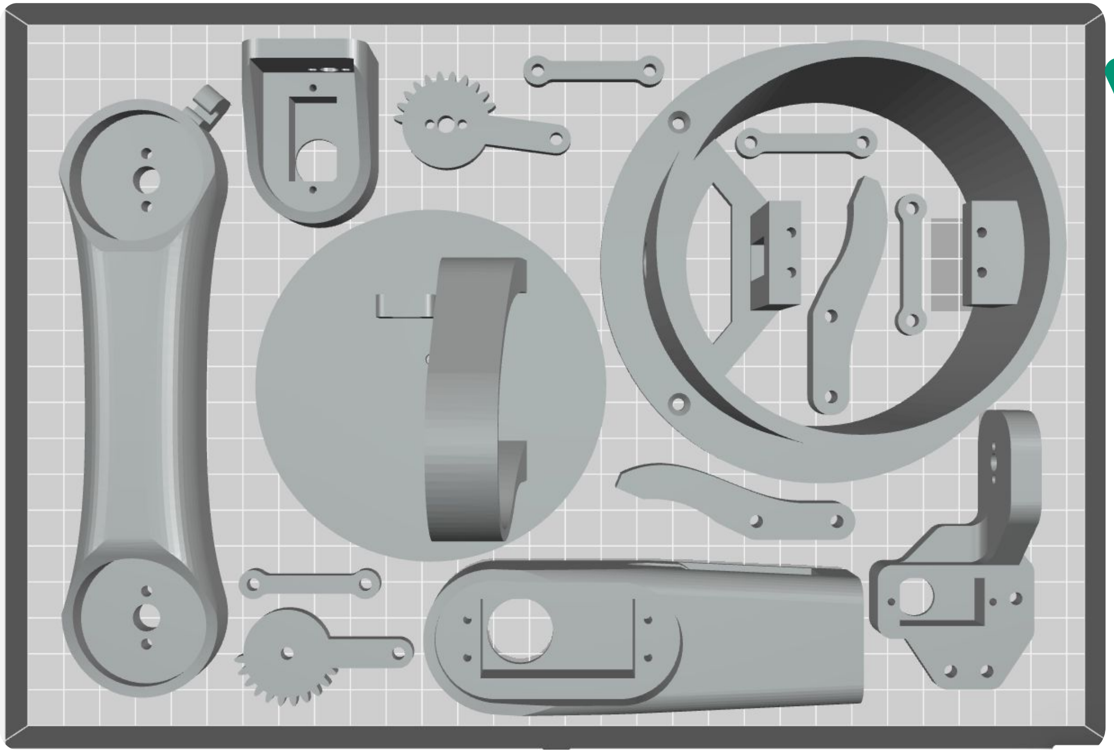
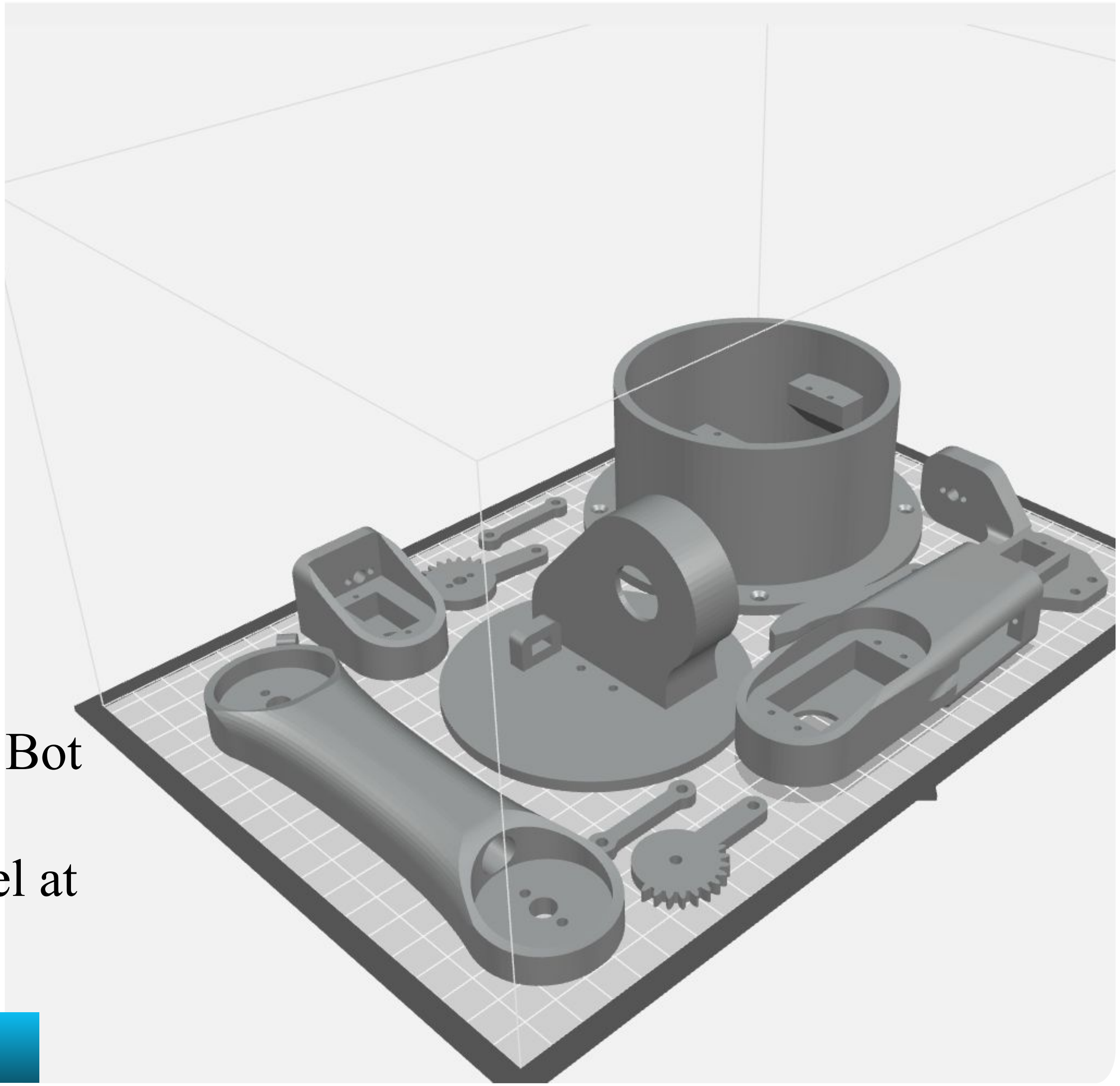
Robotic Systems-1 Final Project: A Smartphone controlled Robot Arm

Shravani Sai Ventrapragada
ASU ID: 1222627419
Arizona State University
Under the guidance of
Dr. Sangram Redkar

3 D Printing





- 3D printed the model.
  - Designed with the help of maker Bot file
  - Used a filament to print the model at maker’s lab at Hayden Library

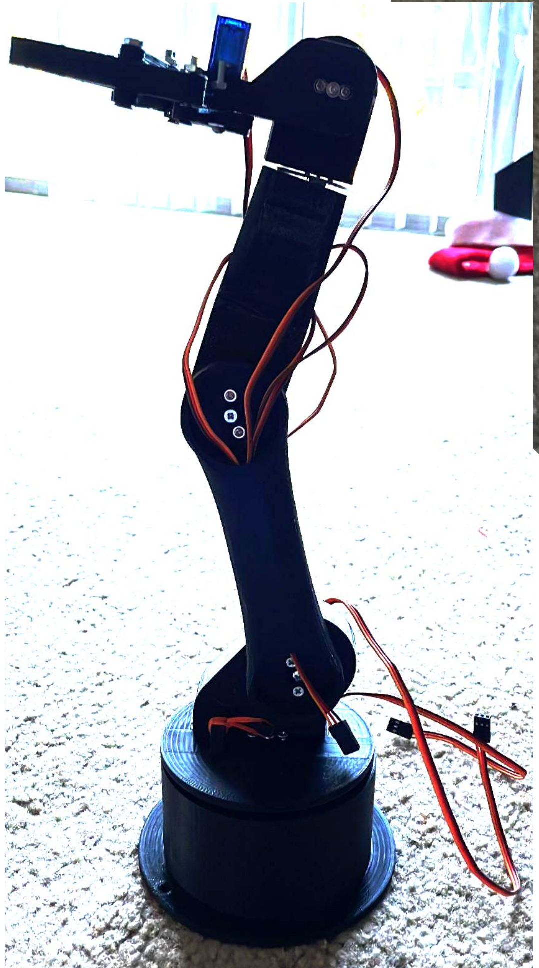
Click here to view the file

3 D printing

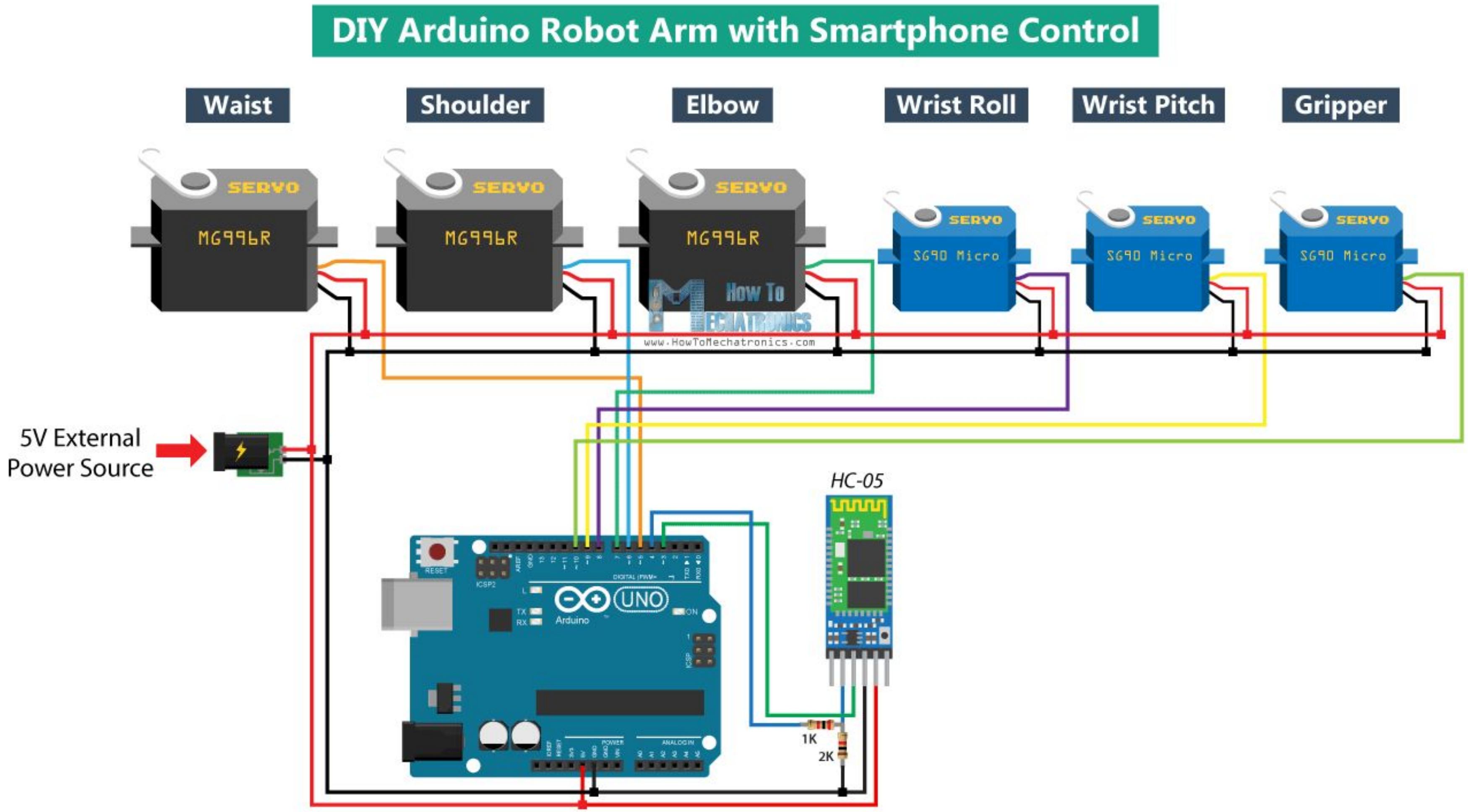
- The product after 3D printing came out on a plate.
- I detached each element from the thin sheet and filed each component to make it presentable.

FIXING

- I used :
  - 3 Big motors
  - 3 Small motors
  - Hex Nuts
    - Size- M3- 0.50
  - Screws-
    - Size- 4X3/8
    - Size- M3- 0.50x 20
- To fix the robot together.



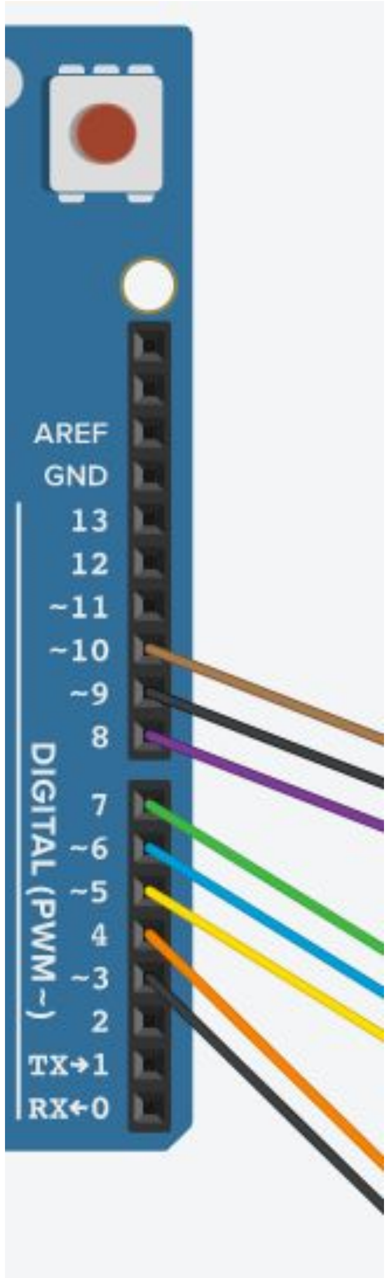
The circuit



Reference:https://howtomechatronics.com/wp-content/uploads/2018/09/Arduino-Robot-Arm-Schematic-Circuit-Diagram.png

Arduino Circuit

- Pin 10- Gripper (Brown)
- Pin 9- Wrist Pitch (Black)
- Pin 8- Wrist Roll (Purple)
- Pin 7- Elbow (Green)
- Pin 6- Shoulder (Blue)
- Pin 5- Waist (Yellow)
- Pin 4- Tx (Orange)
- Pin 3- Rx (Black)



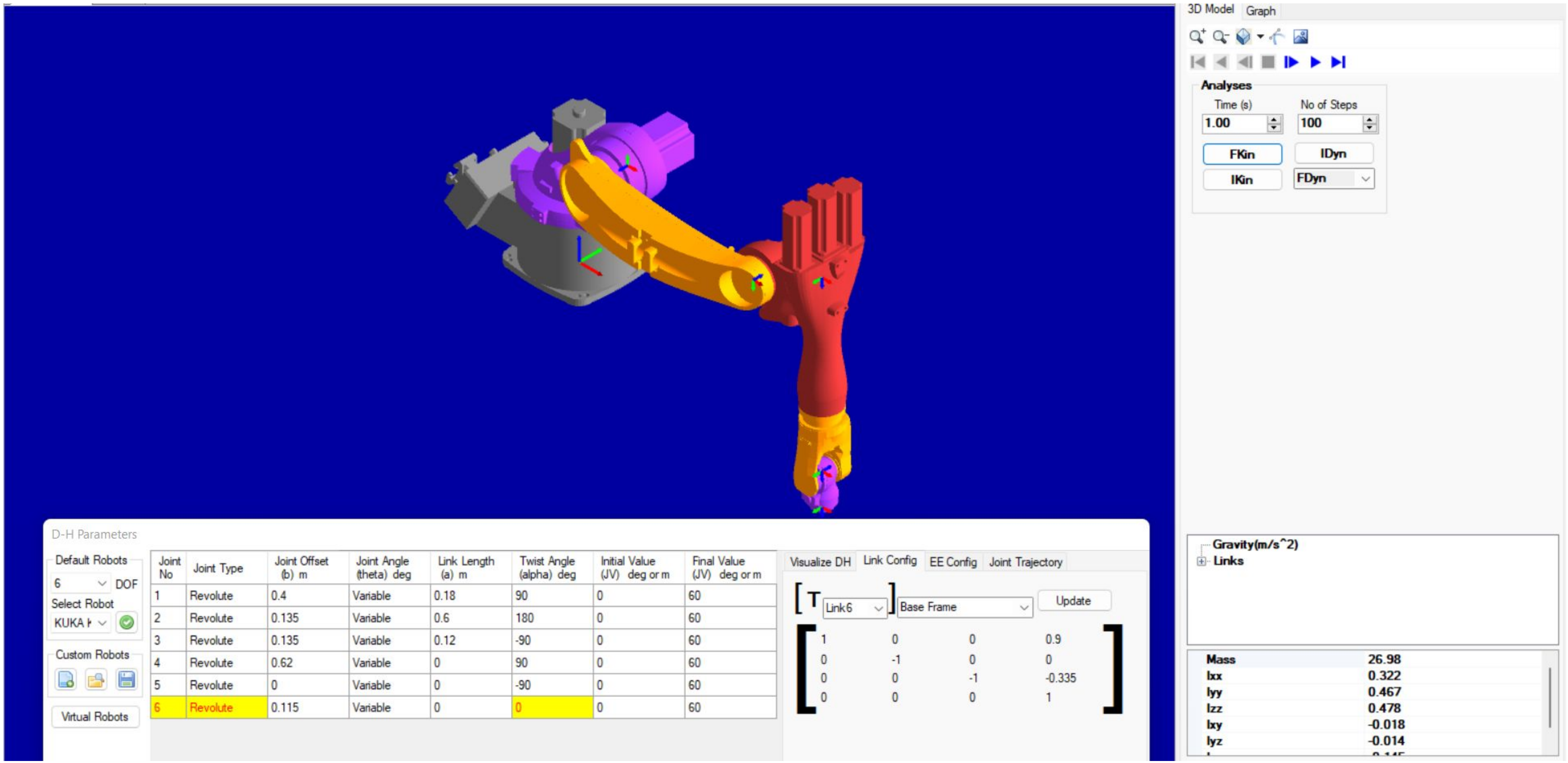
Powering the circuit

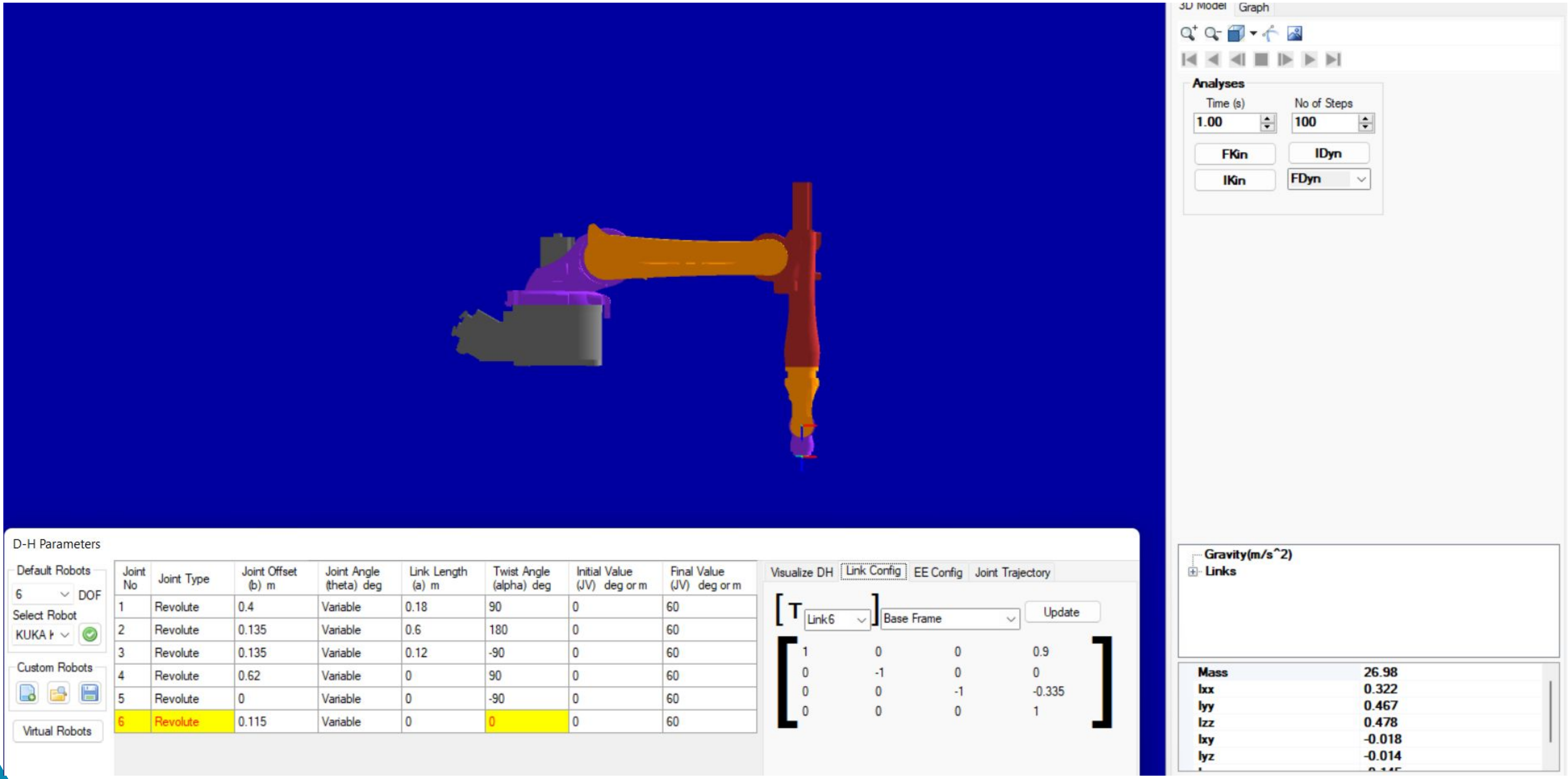
- To power the circuit, I utilize a 5V wall mount power supply

Smooth motion

- For giving the robot a smooth motion:
- We give the lowest speed possible to all the servos

Visualising Kuka robot in RoboAnalyser simulator




3D Model
Graph
Analyses
Time (s)
1.00
No of Steps
100
FKin
IDyn
IKin
FDyn
D-H Parameters
Default Robots
6
DOF
Select Robot
KUKA K
Custom Robots
Virtual Robots
Joint No
Joint Type
Joint Offset (b) m
Joint Angle (theta) deg
Link Length (a) m
Twist Angle (alpha) deg
Initial Value (JV) deg or m
Final Value (JV) deg or m
1 Revolute 0.4 Variable 0.18 90 0 60
2 Revolute 0.135 Variable 0.6 180 0 60
3 Revolute 0.135 Variable 0.12 -90 0 60
4 Revolute 0.62 Variable 0 90 0 60
5 Revolute 0 Variable 0 -90 0 60
6 Revolute 0.115 Variable 0 0 0 60
Visualize DH
Link Config
EE Config
Joint Trajectory
T
Link6
Base Frame
Update
1 0 0 0.9
0 -1 0 0
0 0 -1 -0.335
0 0 0 1
Gravity(m/s^2)
Links
Mass 26.98
Ixx 0.322
Iyy 0.467
Izz 0.478
Ixy -0.018
Iyz -0.014

Algorithm

Step 1: Include the libraries
Step 2: Define the servos
Step 3: Give the motor a speed for the motion.
Step 4: Define each motor for the pin attached in arduino
Step 5: For the data received check the 1st character is a,b,c,d,e or f.

- a- gripper
- b- wrist pitch
- c- wrist roll
- d- elbow
- e- shoulder
- f- waist

Servo Joint	Maximum Angle (in degrees)	Minimum Angle (in degrees)
Gripper	180	0
Wrist Pitch	180	45
Wrist Roll	180	0
Elbow	130	45
shoulder	170	0
Waist	90	0

Algorithm

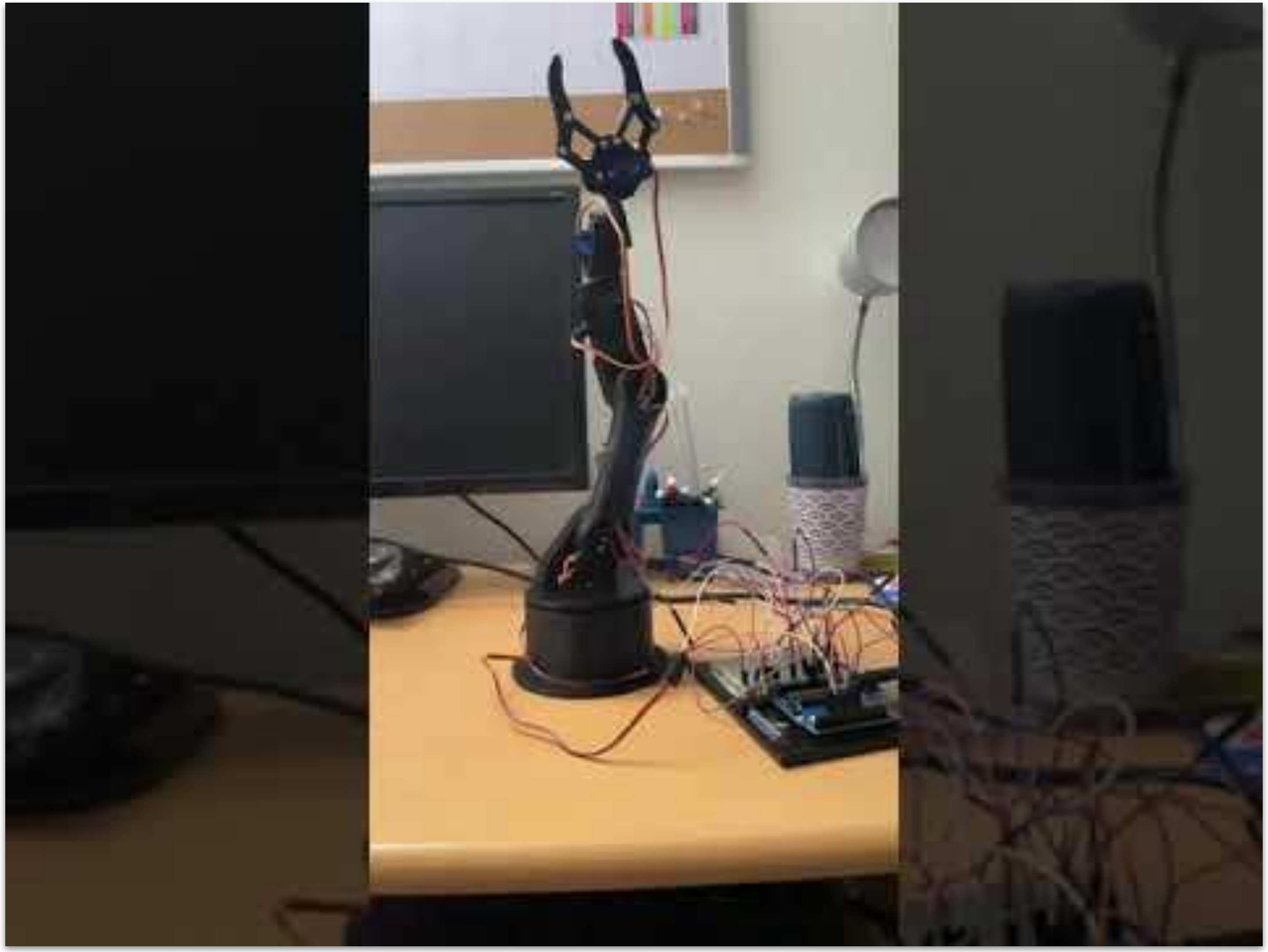
Step 6: Define maximum and minimum angles for each servo motor and write the given value from the value if it is in the range of the max and min angles

Servo Joint	Maximum Angle (in degrees)	Minimum Angle (in degrees)
Gripper	180	0
Wrist Pitch	180	45
Wrist Roll	180	0
Elbow	130	45
shoulder	170	0
Waist	90	0

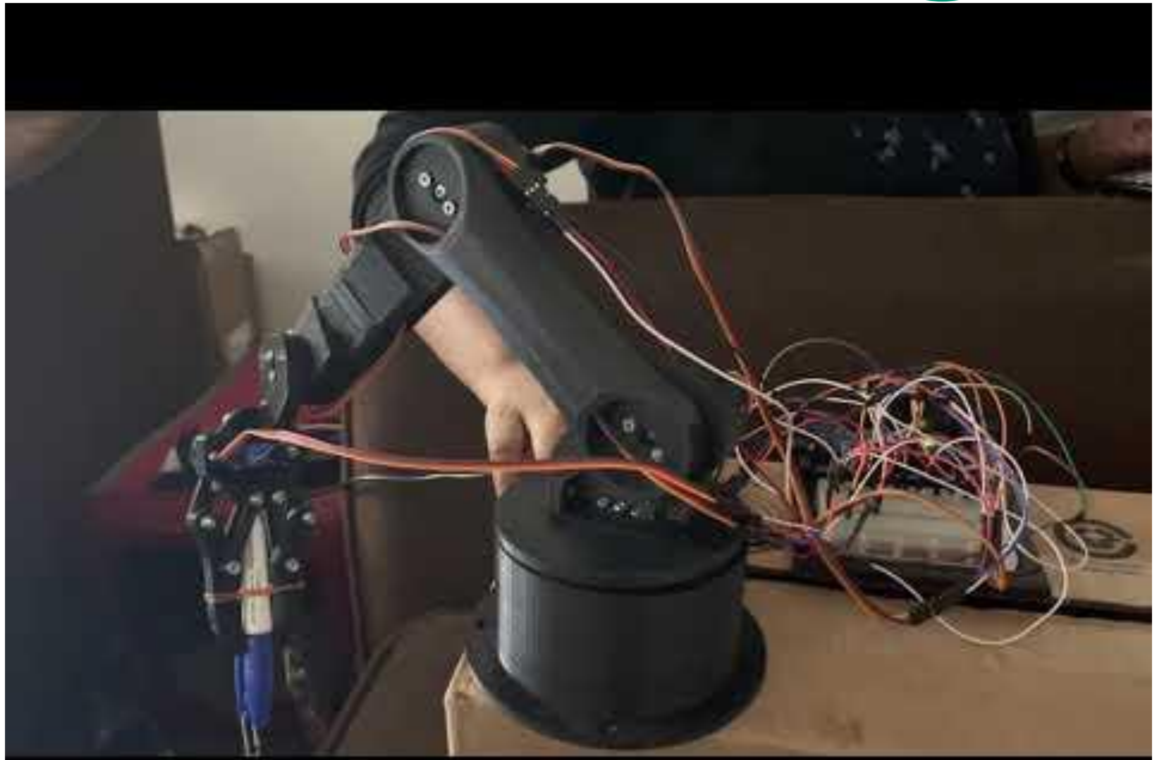
Step 7: Write the entered value if entered in the range.

Code for working of the robot

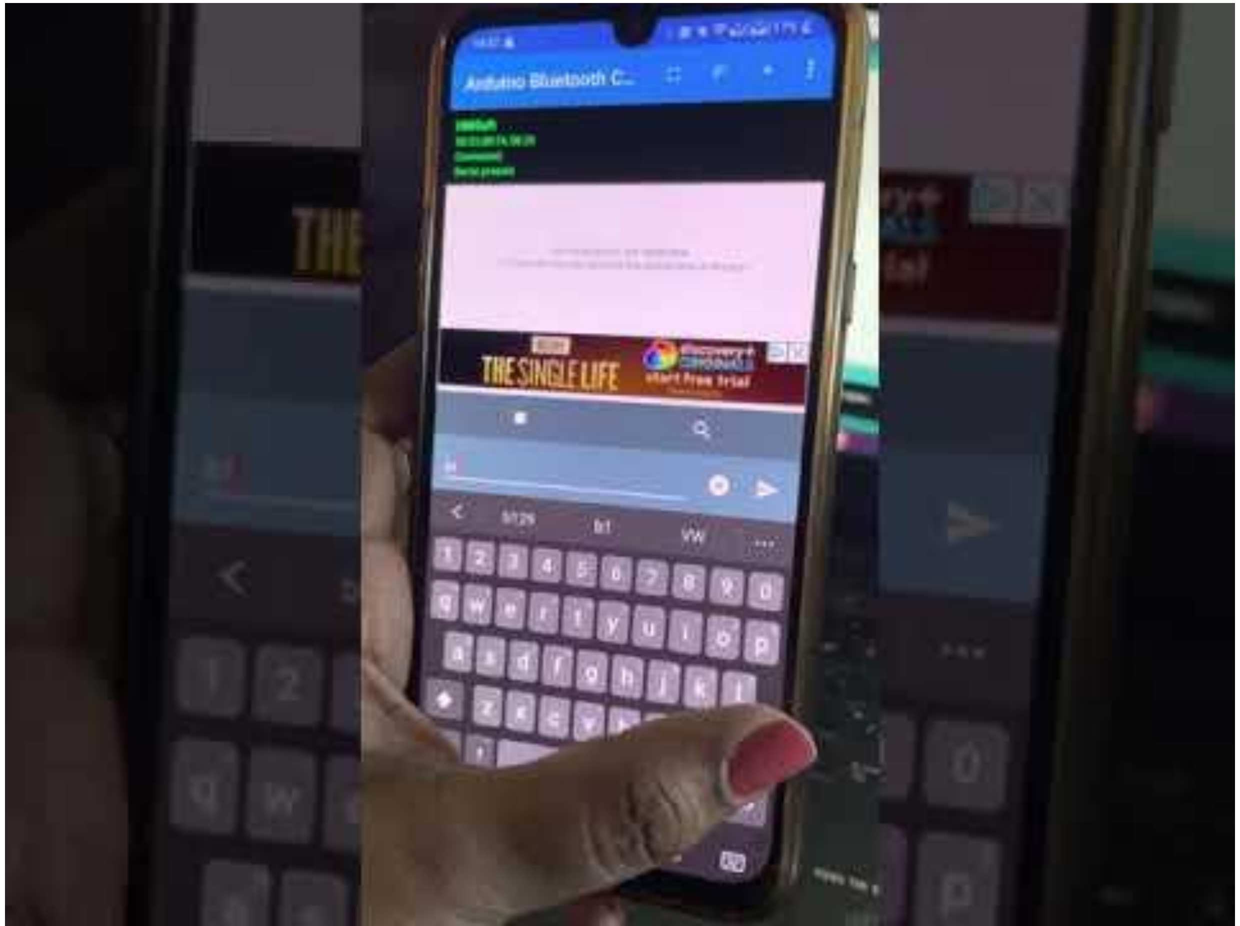
The robot takes initial position on my first run



Insert a video



Servo at the elbow joint moving 40 degrees



Servo at the wrist joint moving

Algorithm- smooth motion

- Read the current angle of the servo and store it in a integer type variable called current.
- Add a rotate function with current and part as arguments
- If current angle is less than actual entered value increment current angle.
- Else current angle is more than actual entered value then decrement the current angle
- eg: If the user chooses a then write the current angle to the servo
- Therefore we create a feedback loop by comparing the previous and the current angles.

  Code for smooth working of the robot

Smooth Motion

Trajectory and Forward Inverse Kinematics- Algorithm

Here We use forward inverse kinematics to solve the trajectory problem.

Trajectory and Forward Inverse Kinematics- Algorithm

Step 1: Import the math library

Step 2: Define each links length

Step 3: Create an inverse kinematic function for link 2 which returns the angles of the first two links in the robotic arm as a list when provided with x and y coordinates of the end effector.

angleMode - tells the function to give the angle in degrees/radians. Default is degrees

output:

th1 - angle of the first link w.r.t ground

th2 - angle of the second link w.r.t the first

Trajectory- Algorithm

Step 4: Create another function which takes
- x - The x coordinate of the effector
- y - The y coordinate of the effector
- z - The z coordinate of the effector
- angleMode - tells the function to give the angle in degrees/radians. Default is degrees

Returns the angles of the first three links and the base drum in the robotic arm as a list. Since only three degrees of freedom are required to make the arm go to any location in 3D space this link's job is to keep the gripper at a constant angle relative to the ground. This is useful in situations where the arm is carrying objects like glasses of fluid. This link makes sure that the arm doesn't tip over the glass and spill the fluid
returns -> (th0, th1, th2, th3)

Trajectory- Algorithm

```
#stuff for calculating th2
r_2 = x**2 + y**2
l_sq = l1**2 + l2**2
term2 = (r_2 - l_sq)/(2*l1*l2)
term1 = ((1 - term2**2)**0.5)*-1
#calculate th2
th2 = math.atan2(term1, term2)
#optional line. Comment this one out if you
#notice any problems
th2 = -1*th2
#Stuff for calculating th2
k1 = l1 + l2*math.cos(th2)
k2 = l2*math.sin(th2)
r = (k1**2 + k2**2)**0.5
gamma = math.atan2(k2,k1)
#calculate th1
th1 = math.atan2(y,x) - gamma
```

Trajectory- Algorithm

```
th3 = th1 - th2 - l3ang
except ValueError:
print "ERROR: Given coordinates are outside arm's reach!"
print "Arm will return to starting position"
if(angleMode == RADIANS):
return math.radians(0), math.radians(100),\
math.radians(90), math.radians(10)
else:
return 0, 100, 90, 10
if(angleMode == RADIANS):
return th0, th1, th2, th3
else:
return math.degrees(th0), math.degrees(th1),\
math.degrees(th2), math.degrees(th3)
if __name__ == "__main__":
print invkin2(0, 200, DEGREES)
```

Trajectory

Documenting Bluetooth module

- Tried using a newer android phone
  - While fixing the Bluetooth module detection issues were there.
- While connecting Bluetooth module to the Arduino the android phone could not recognize the Bluetooth module HC-10

To fix this I tried:

- Using ESP32's Bluetooth module instead of the Arduino board- the code was not uploading to ESP 32
- Tried replacing HC 05 module instead of HC 10 Bluetooth module- HC 05 was not getting detected due to hardware issues
- The app we use here is Serial bluetooth module instead of the MIT app inverter as the bluetooth module 10 is not compatible with the app.
  - Therefore even after the bluetooth gets connected to the module the app is not responsive.

The error with ESp32 was the power was sufficient to load the code on to the microcontroller snd the motors were not working on 3.3 V

```
#include <Arduino.h>
#include <BluetoothSerial.h>
#include <ESP32Servo.h>

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

BluetoothSerial SerialBT;

Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
```

An error occurred while uploading the sketch

```
Sketch uses 1003458 bytes (31%) of program storage space. Maximum is 3145728 bytes.
Global variables use 32112 bytes (9%) of dynamic memory, leaving 295568 bytes for local variables. Maximum is 327680 bytes.
esptool.py v3.0-dev
Serial port COM6
Connecting...
Traceback (most recent call last):
  File "esptool.py", line 3682, in <module>
  File "esptool.py", line 3675, in _main
  File "esptool.py", line 3330, in main
  File "esptool.py", line 512, in connect
  File "esptool.py", line 492, in _connect_attempt
  File "esptool.py", line 431, in sync
  File "esptool.py", line 369, in command
  File "esptool.py", line 332, in write
  File "site-packages\serial\serialwin32.py", line 323, in write
serial.serialutil.SerialTimeoutException: Write timeout
Failed to execute script esptool
An error occurred while uploading the sketch
```

Thank you