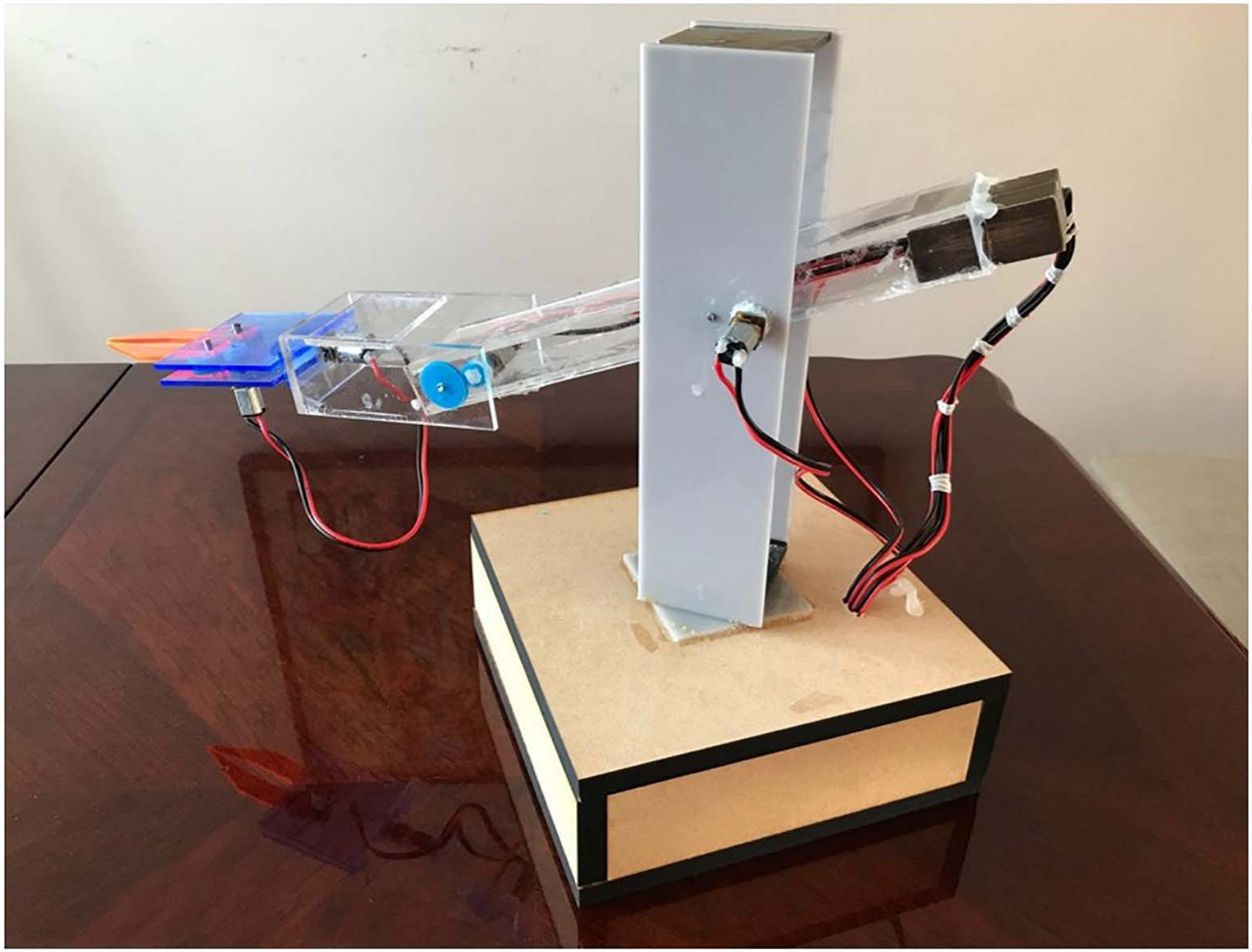
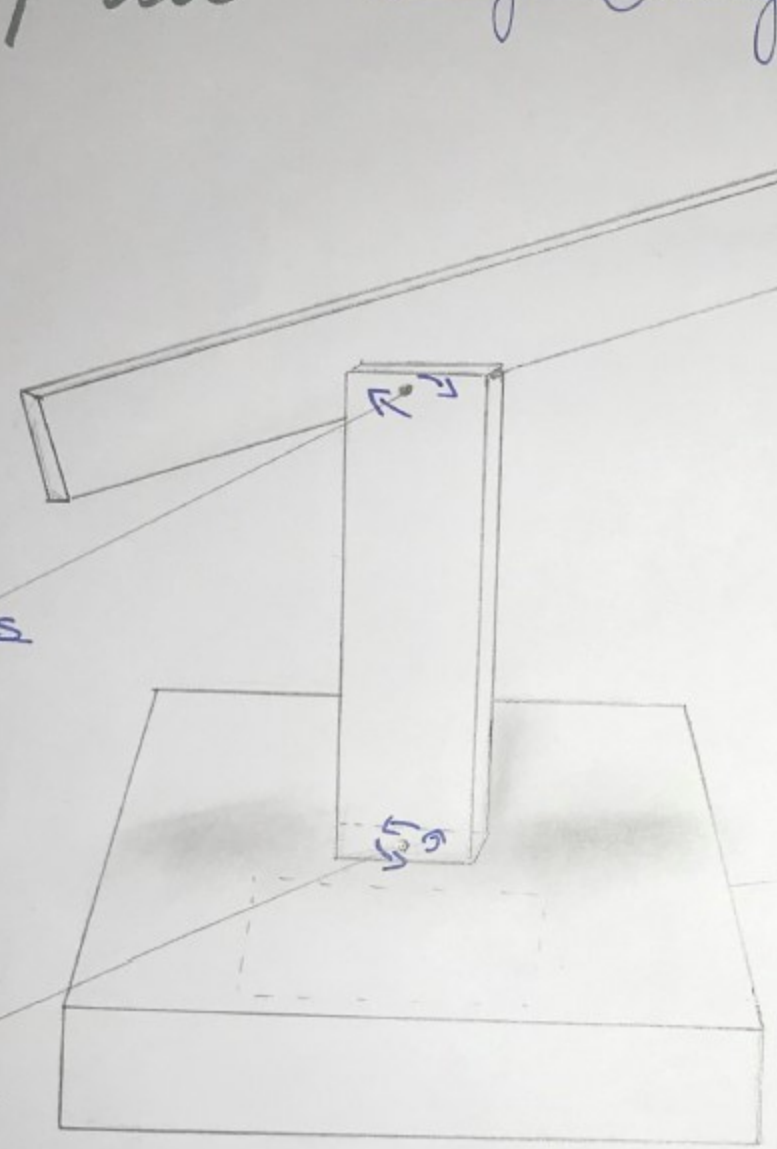
Tariq El-Jumaily
Visual Evidence
Mechanical Robot Arm (Controlled via Wi-Fi)



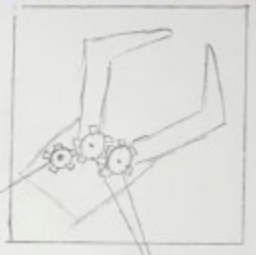
In development -
(Working prototype)

Desk Pad Draft Design



Claws made of high-grip material. eg. Rubber

2 gears for claws controlled by 1 smaller gear with motor



gear on motor - when rotated, claw gear rotates

gears glued to claws - when one moves, the other also moves

Motor Pivots arm on an axis

Heavy base So that arm can extend without falling over

neck (which will be adapted to be cylinder) rotates 360° to base

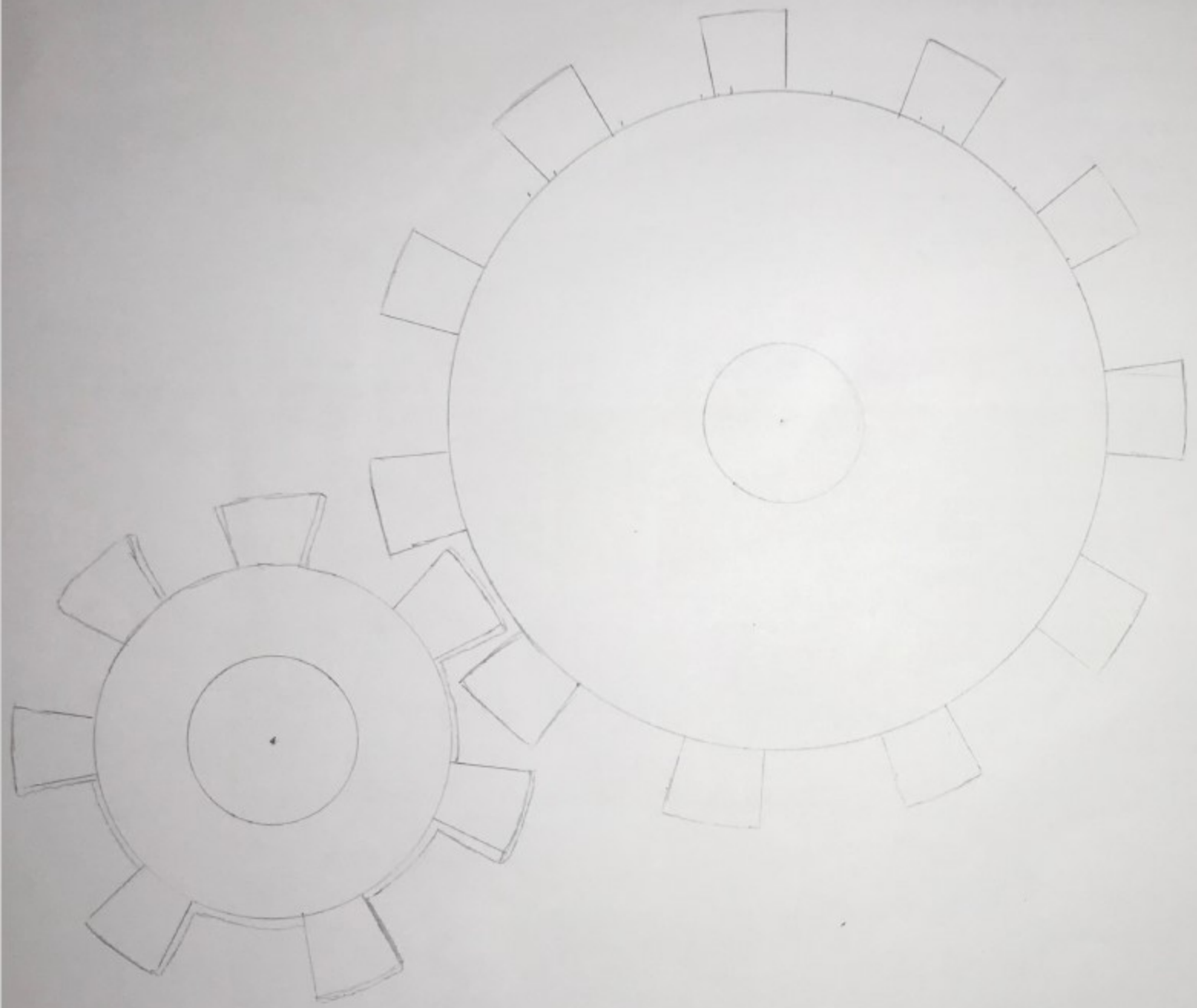
to do list

I am going to design and make a robotic mechanical arm that will be controlled via a mobile device wirelessly. This will be done by connecting a plethora of motors and gears to a programmable embedded computer and I will create an interactive web application to individually control the voltages of the motors which will rotate each part of the arm.

Desk Pad

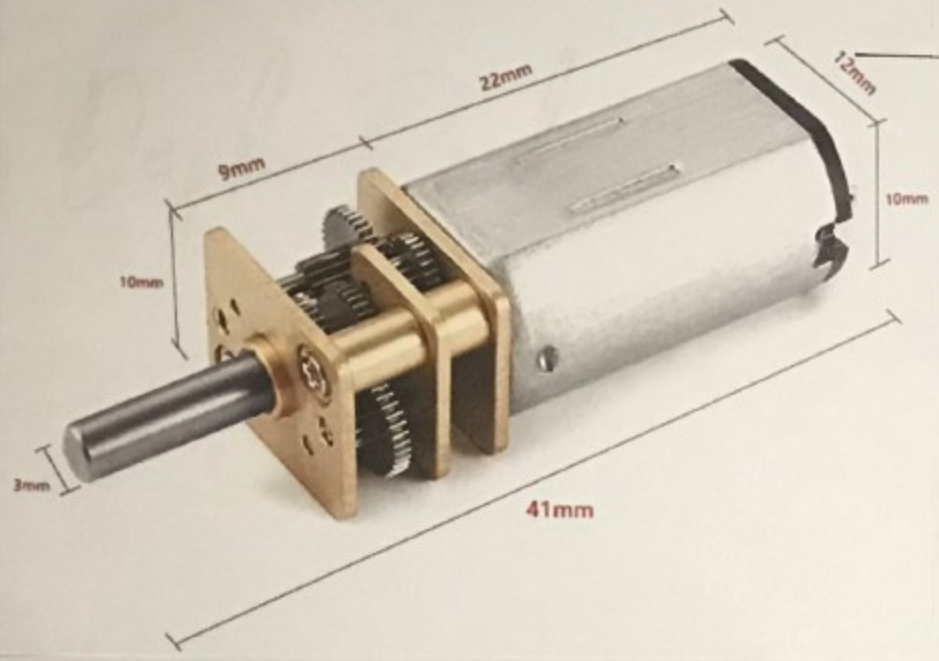
Pivot

(Sketch)
not to scale



to do list

Most Pivot mechanisms in the arm will consist of 1 small gear attatched to motor which will move the larger one which is connected to the body of the arm being moved.

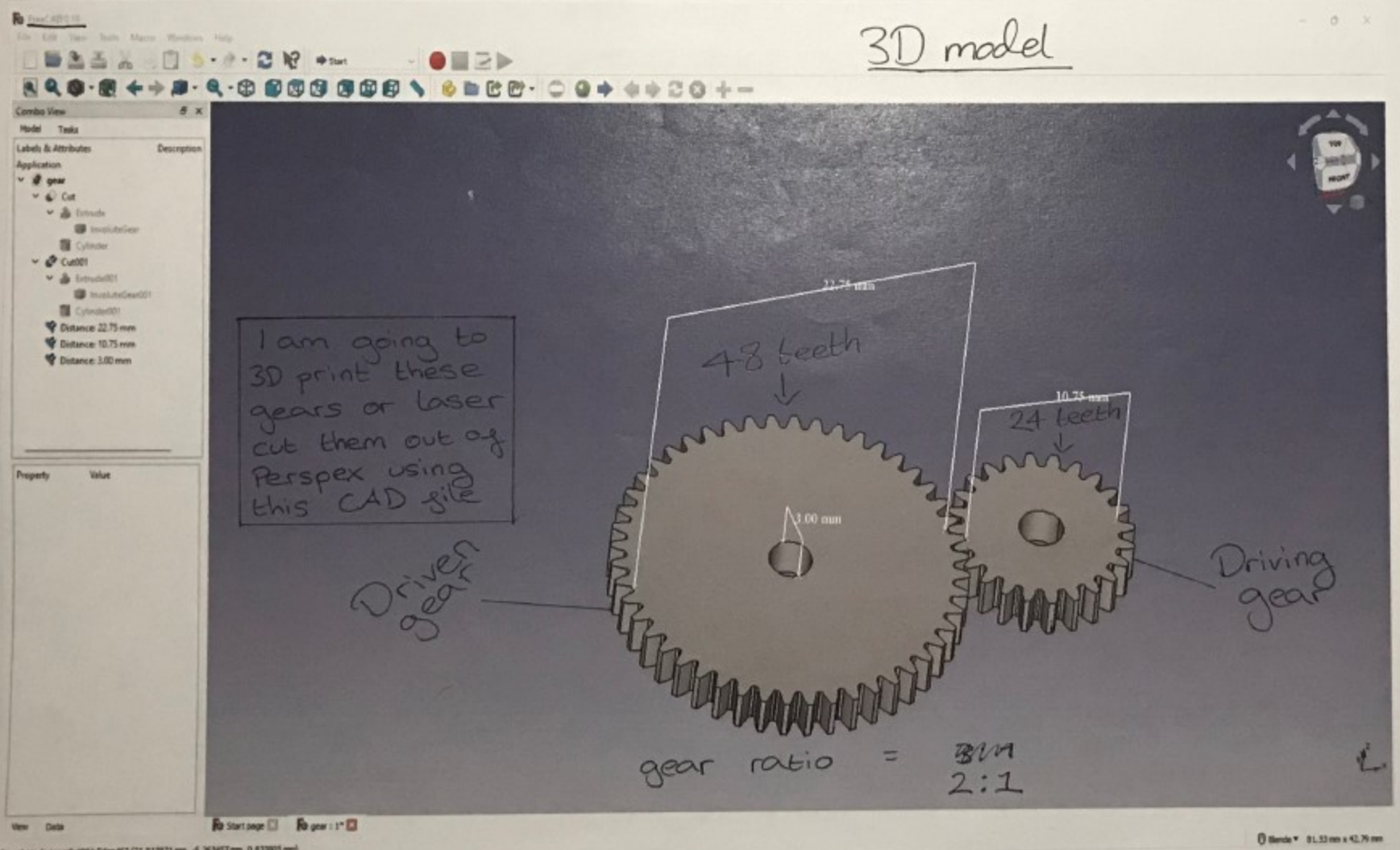


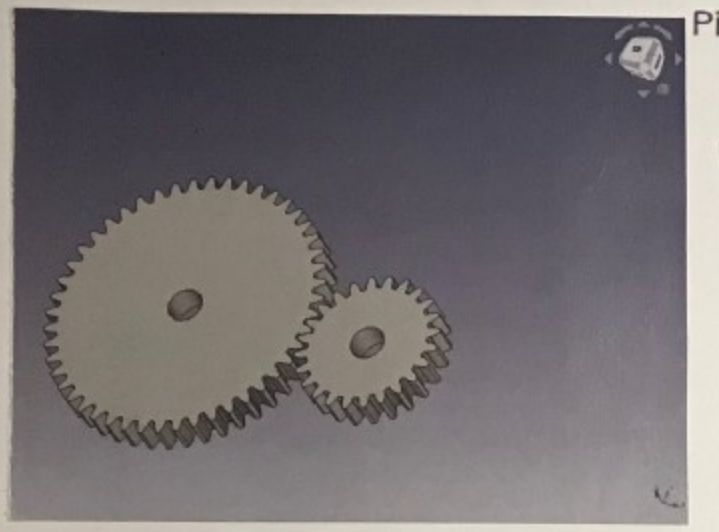
This is a DC ~~3V 9RPM~~ 6V 30RPM gear box motor that I will be using for my project as it is low voltage and low speed with a high torque value which means it is able to move heavy objects at a low speed.
Also, the overall size and weight of the motor is perfect for my project & is very cost effective.

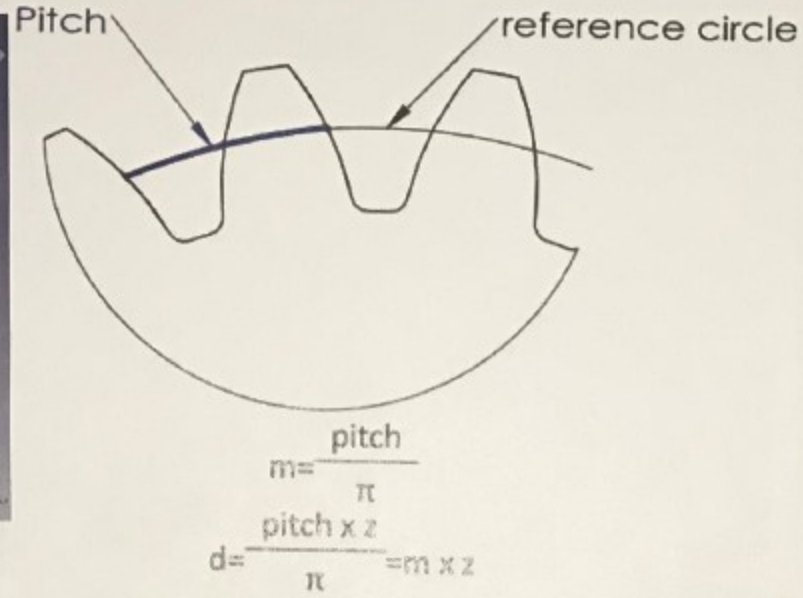
to do list

This is a 3D model of 2 meshing gears designed in FreeCAD. They consist of 3mm diameter holes in the middle (bore) which is wide enough for the motor shaft. This will allow me to securely attach the small gear to the motor with the assistance of a strong adhesive. To also increase the strength of the gear attachment to the motor, I may reduce the bore size very slightly to create a tighter fit to the motor shaft.

3D model



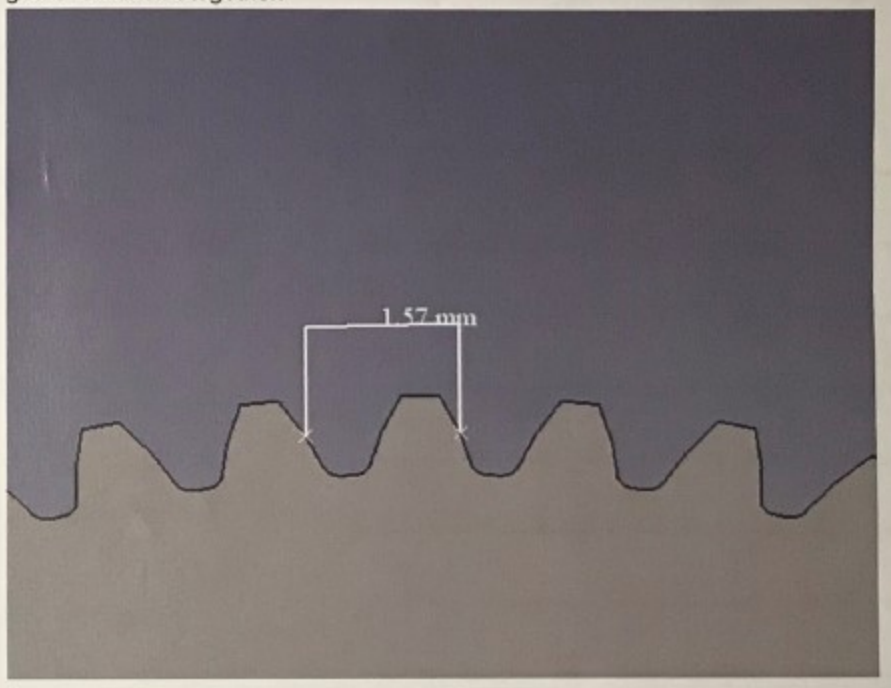


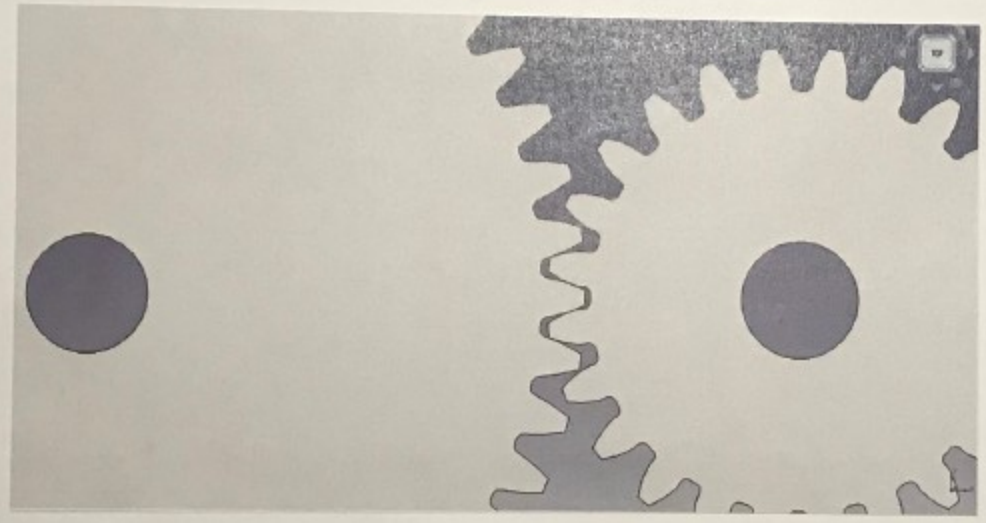


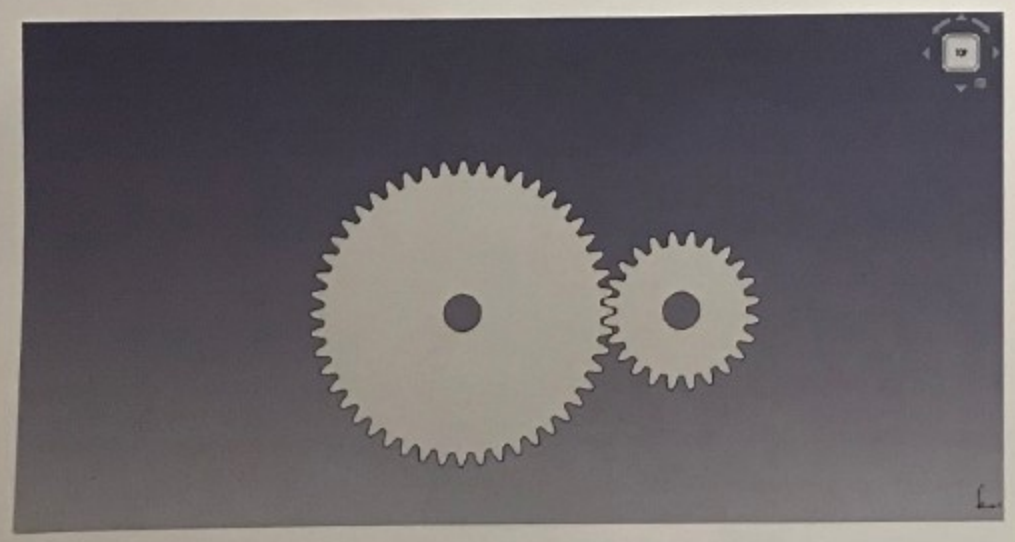
$$m = \frac{\text{pitch}}{\pi}$$

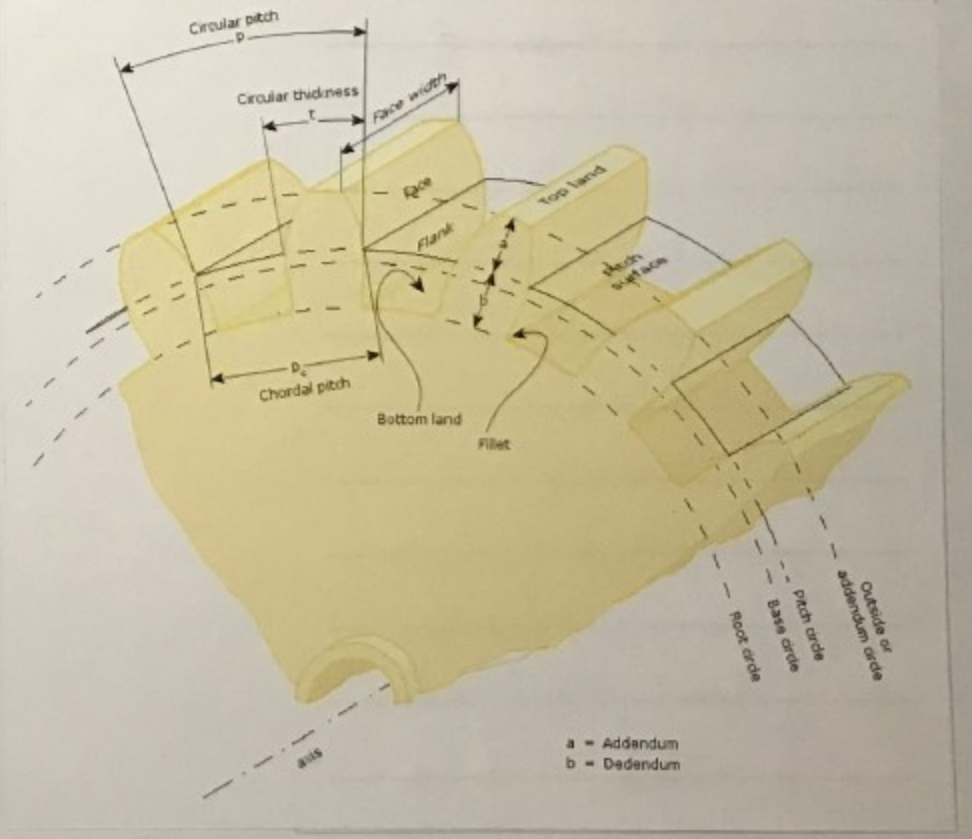
$$d = \frac{\text{pitch} \times z}{\pi} = m \times z$$

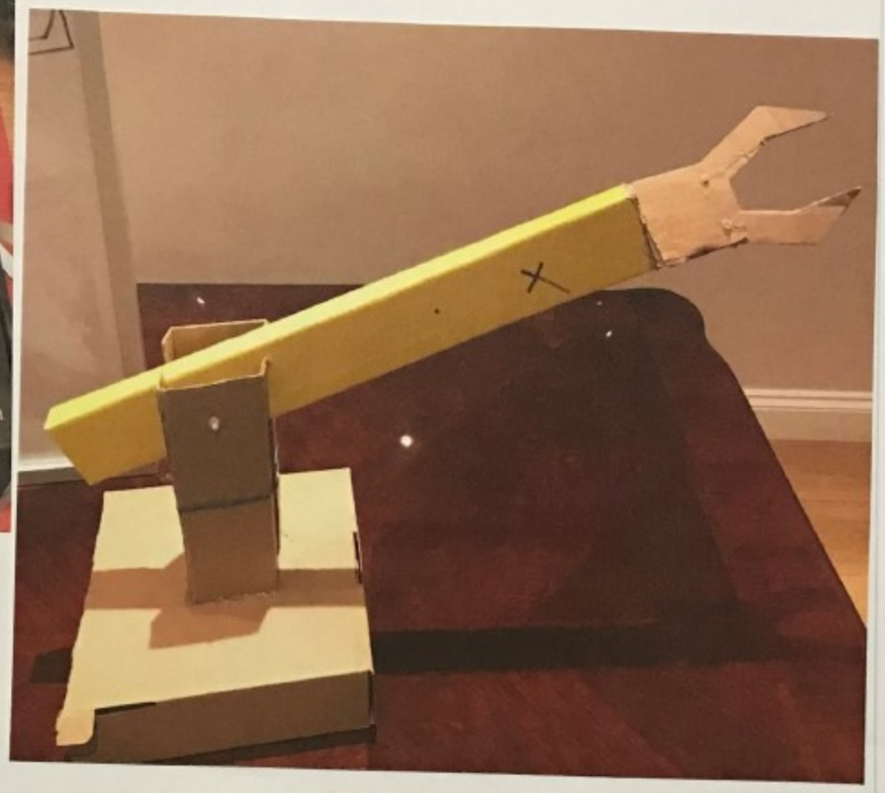
To make compatible gears, you need to match the pitch and shape of the teeth. 2 meshing gears always have the same modules which is pitch/pi. The modules of both gears are set to 0.50mm which would mean that the pitch of both gears are 1.57mm and both gears have a pressure angle of 20º. This means that when printed, both gears are guaranteed to fit together.









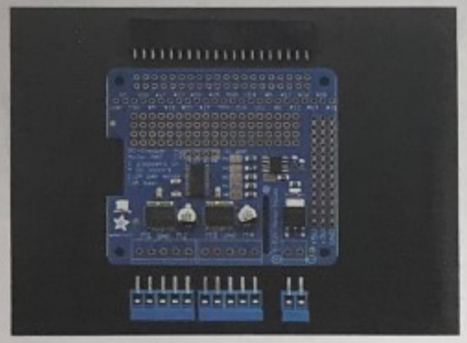


to do list

I took inspiration and ideas from a pre-existing project which uses a 5-way-directional controller I used this to draft a cardboard model without any mechanics and then realised that I wanted it to rotate 360° on the base. Also I want to add another "arm" where the X is on my model to increase flexibility.

The final model will need to have a large mass within the base to prevent it from toppling over when extending.

Computation & Logic System



Adagruit DC & Stepper Motor HAT for raspberry Pi

- Uses PWM (Pulse-width modulation) to control motor direction and speed.
- Connect up to 4 DC motors or 2 stepper motors
- Powered by 5-12V voltage input
- Can stack up to 32 HATs for a total of 128 DC motors or 64 Stepper motors controlled by a singular Pi

Waveshare Motor Driver pHAT for Raspberry Pi

- Controls 2 DC motors
- 6V-12V input VIN terminal
- uses I2C Control pins to work with other control boards

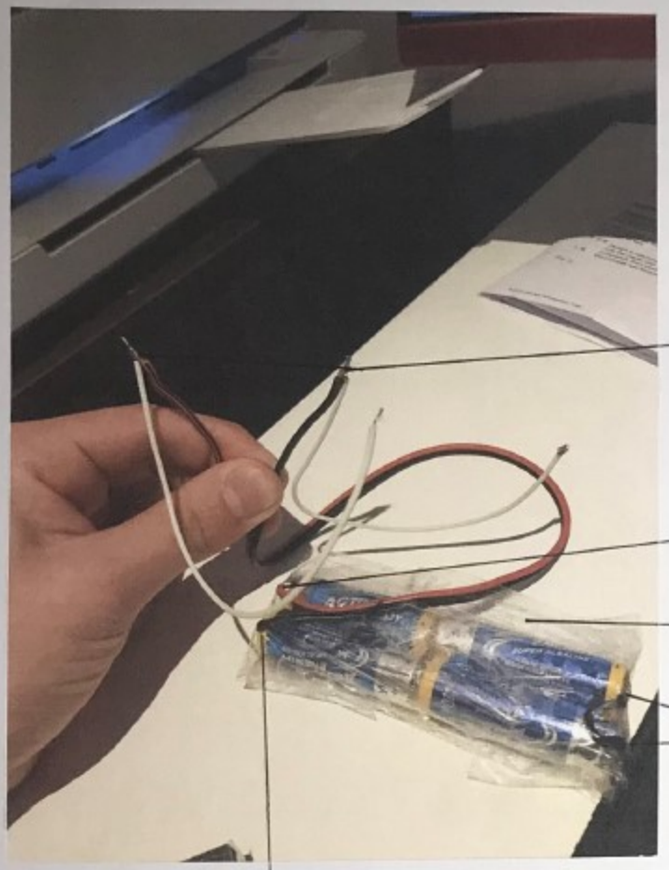
Raspberry Pi 3

- Cheap, small computer running Linux with a set of GPIO (General purpose input/output) to control electrical components
- 1.2 GHz Quad Core CPU
- 1GB RAM
- 40-pin extended GPIO
- 4 USB 2.0 Ports
- Full-size HDMI
- wired, and wireless internet connectivity with Bluetooth

to do list

In order to digitally control the motors to move each part of the arm, I need to be able to control the voltages and polarity that I send and a Raspberry Pi is the perfect, low-profile computer to use within my project as it is also very portable. The wireless functionality of the Pi will allow me to communicate with it from my phone. However, I need motor control boards in order to deliver the desired signals to the motors.

Power



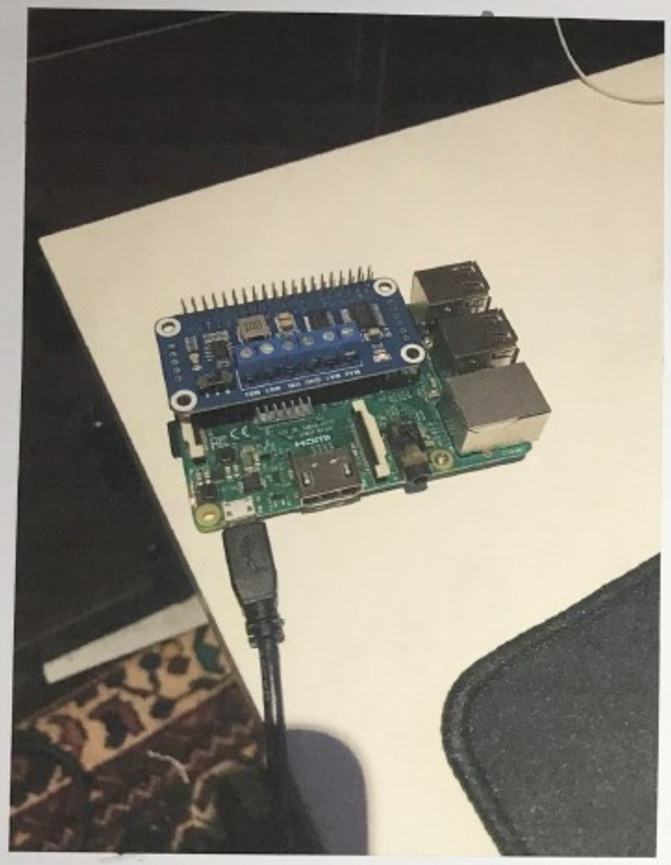
SAFETY WARNING

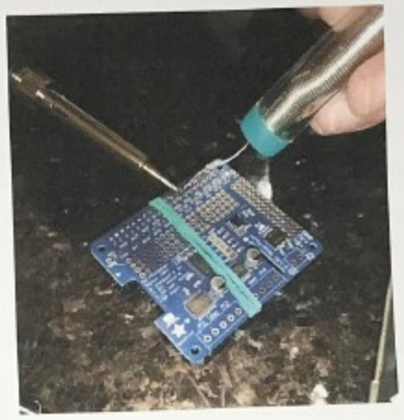
The exposed wires are live and will create a spark when they touch - this can be dangerous and so must be separated and covered with insulation when not connected to component

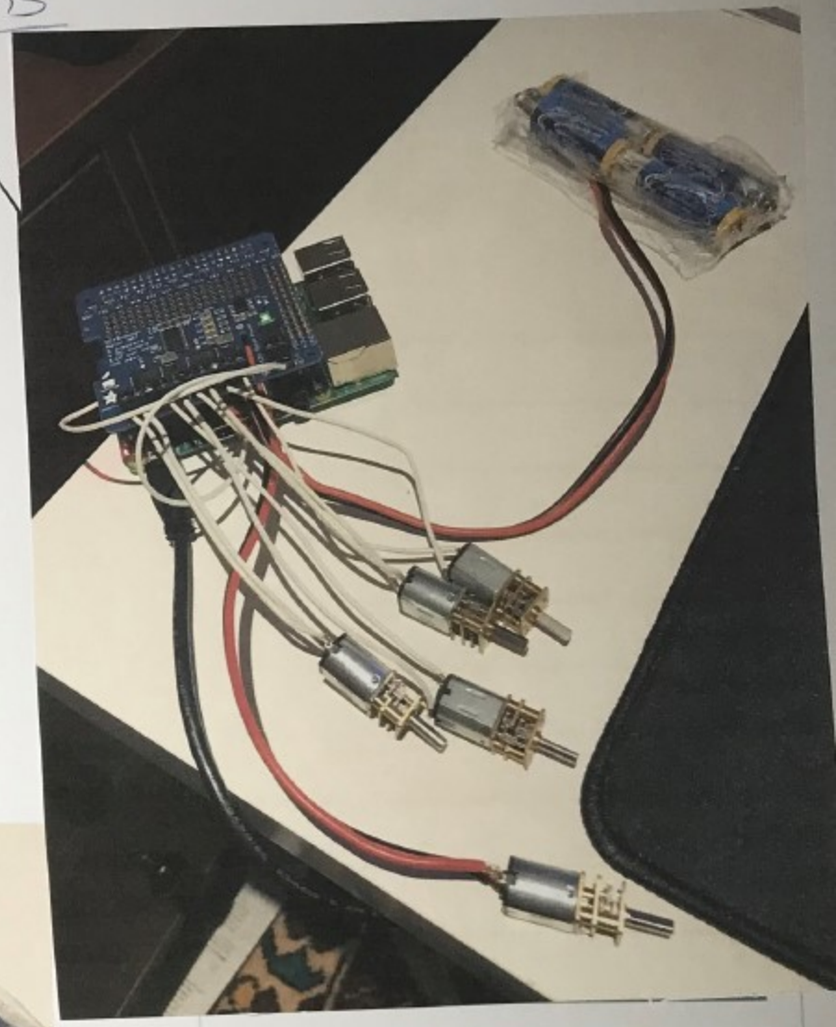
to do list

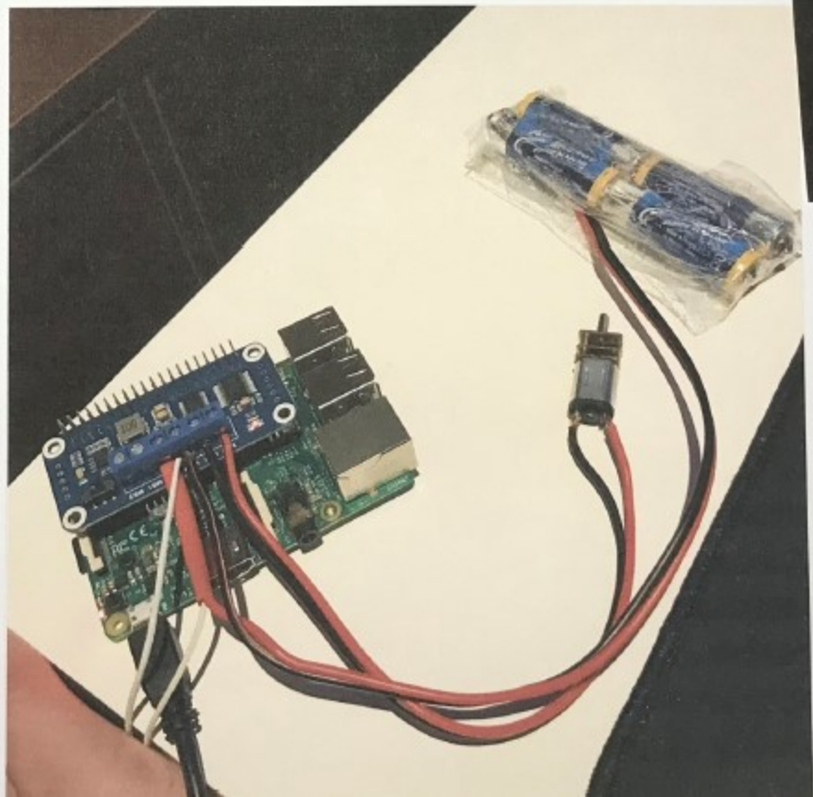
Both motor controllers need at least 6V external power and I designed a temporary solution (prototype) - I attatched 4 AA ~~7.4~~ 1.5 V batteries with a positive and negative soldered onto the poles. This was held together by insulating tape and was only used to test the functionality of the control boards. Since I was also powering 2 boards, I added extra wires onto the ends to connect the circuit in parallel.

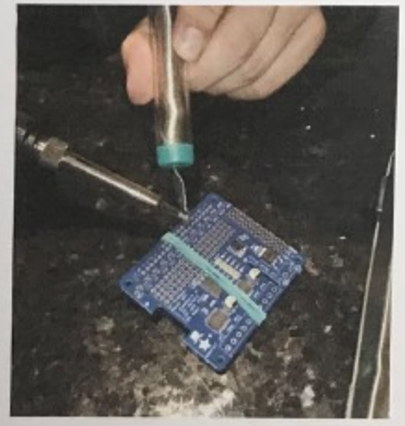
Assembly + Connections



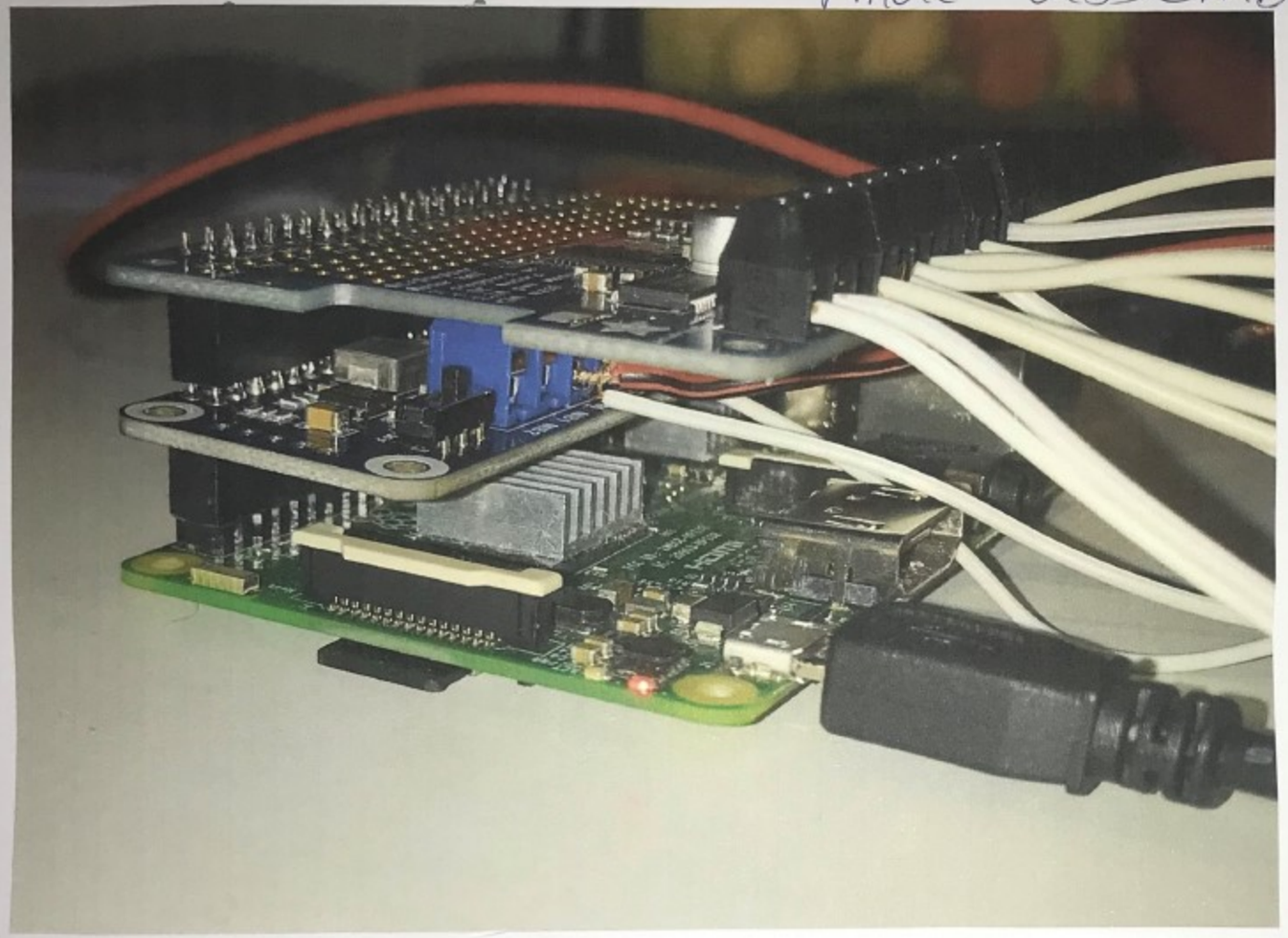


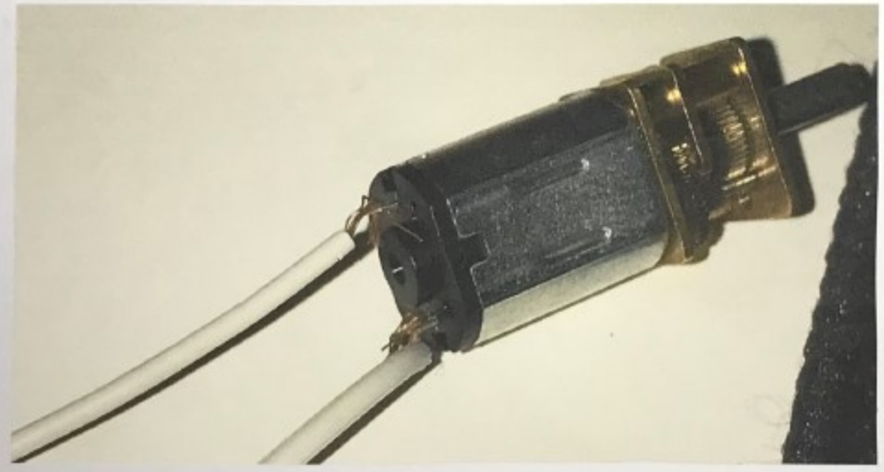






final assembly





to do list

After soldering and connecting the boards to the computer, I loosely connected the power supply and motors. This is because I needed to program and troubleshoot the functionality before measuring wire lengths for the completed build and so I threaded the copper wire through the holes in the motor and folded them so it made good contact while also being easy to remove.

Website

```html
<!doctype html>
<head><title>Test</title>
    <meta charset=utf-8> </head>
    <body>
        <h1>Robot Website</h1>


        <form action="/my-link/">
            <input type="submit" value="Motor 1 spin Anti-Clockwise" />
        </form>
        <form action="/my-link2/">
            <input type="submit" value="Motor 1 spin Clockwise" />
        </form>


        <form action="/my-link3/">
            <input type="submit" value="Motor 2 spin Anti-Clockwise" />
        </form>
        <form action="/my-link4/">
            <input type="submit" value="Motor 2 spin Clockwise" />
        </form>


        <form action="/my-link5/">
            <input type="submit" value="Motor 3 spin Anti-Clockwise" />
        </form>
        <form action="/my-link6/">
            <input type="submit" value="Motor 3 spin Clockwise" />
        </form>


        <form action="/my-link7/">
            <input type="submit" value="Motor 4 spin Anti-Clockwise" />
        </form>
        <form action="/my-link8/">
            <input type="submit" value="Motor 4 spin Clockwise" />
        </form>


        <form action="/my-link9/">
            <input type="submit" value="Motor 5 spin Anti-Clockwise" />
        </form>
        <form action="/my-link10/">
            <input type="submit" value="Motor 5 spin Clockwise" />
        </form>

    </body>
```

8 – 9:
Form action redirects the page to the specified address. I am using the Python library "flask" to "listen" to the website address. When somebody redirects to "/my-link/" it will execute the python code attached to its locator (Line 34 (def motor_1AC():)

This is repeated for every button with the altered address location.

Not secure .45:33

Public IP address

Port 33

My Website

- Motor 1 spin Anti-Clockwise
- Motor 1 spin Clockwise

motor 1

- Motor 2 spin Anti-Clockwise
- Motor 2 spin Clockwise

motor 2

- Motor 3 spin Anti-Clockwise
- Motor 3 spin Clockwise

motor 3

- Motor 4 spin Anti-Clockwise
- Motor 4 spin Clockwise

motor 4

- Motor 5 spin Anti-Clockwise
- Motor 5 spin Clockwise

motor 5

Backend Code

Server

```python
from flask import Flask, render_template
app = Flask(__name__)
import board
import keyboard
from adafruit_motorkit import MotorKit


@app.route('/')
def index():
  return render_template('index.html')

kit = MotorKit(i2c=board.I2C())
kit2 = MotorKit(0x40)

kit2.motor1.throttle = 0.0
kit.motor2.throttle = 0.0
kit.motor3.throttle = 0.0
kit.motor4.throttle = 0.0
kit.motor5.throttle = 0.0

motor1AC = False
motor1C = False
motor2AC = False
motor2C = False
motor3AC = False
motor3C = False
motor4AC = False
motor4C = False
motor5AC = False
motor5C = False
```

1 – 5:
Importing python libraries required for software to communicate with hardware

9 – 10:
Function to load index.html website

12 – 13:
Variables set:
"kit" is for Adafruit motor controller
"kit2" is for Waveshare motor controller

This is because they use different interfaces to control the motors

15 – 19:
When program starts, all motor speeds are set to 0 – they are not moving

Speed:

Negative (-0.5) = Clockwise
0 = stopped/no speed
Positive (0.5) = Anti-Clockwise

21 – 30:
Each motor has 2 Boolean variables, a true or false, one for clockwise rotation, and one for anti-clockwise rotation. This will be changed by the user input when the specified function is pressed. E.g. if motor1AC is set to False, when the user initiates the "motor_1AC()" function, it will be set to true and start spinning anti-clockwise

AC = Anti-Clockwise
C = Clockwise

```python
@app.route('/my-link/')
def motor_1AC():
    global motor1AC
    global motor1C
    motor1AC = not motor1AC
    if motor1C == True:
        motor1C = False
    if motor1AC == False and motor1C == False:
        kit2.motor1.throttle = 0.0
    if motor1AC == True and motor1C == False:
        kit2.motor1.throttle = 0.5
    return render_template('index.html')


@app.route('/my-link2/')
def motor_1C():
    global motor1AC
    global motor1C
    motor1C = not motor1C
    if motor1AC == True:
        motor1AC = False
    if motor1C == False and motor1AC == False:
        kit2.motor1.throttle = 0.0
    if motor1C == True and motor1AC == False:
        kit2.motor1.throttle = -0.5
    return render_template('index.html')
```

33 – 34:
I am using the Python library "Flask" which is a web framework at I've set it to listen to when the website calls "/my-link/" in the form action. It will then send a signal to the Python script to run the code underneath (executing the "motor_1AC" function)

35 – 36:
Calls the Boolean variables of the clockwise and anticlockwise rotations of motor 1 from lines 21 & 22

37:
When "motor_1AC" is called, the Boolean value is switched (if on, turns off. If off, turns on.)

38 – 39:
Checks to see if the motor is moving clockwise. If it is, it will stop moving clockwise

40 – 41:
Checks the Boolean values of the motor rotations. If both are set to false, the motor will not move.

42 – 43:
Checks the Boolean values of the motor rotations. If only anti-clockwise (AC) is true, it will spin at 0.5 speed

47 – 48:
When the website calls "/my-link2/" in the form action, python will execute the function "motor_1C"

51:
When "motor_1C" is called, the Boolean value is switched (if on, turns off. If off, turns on.)

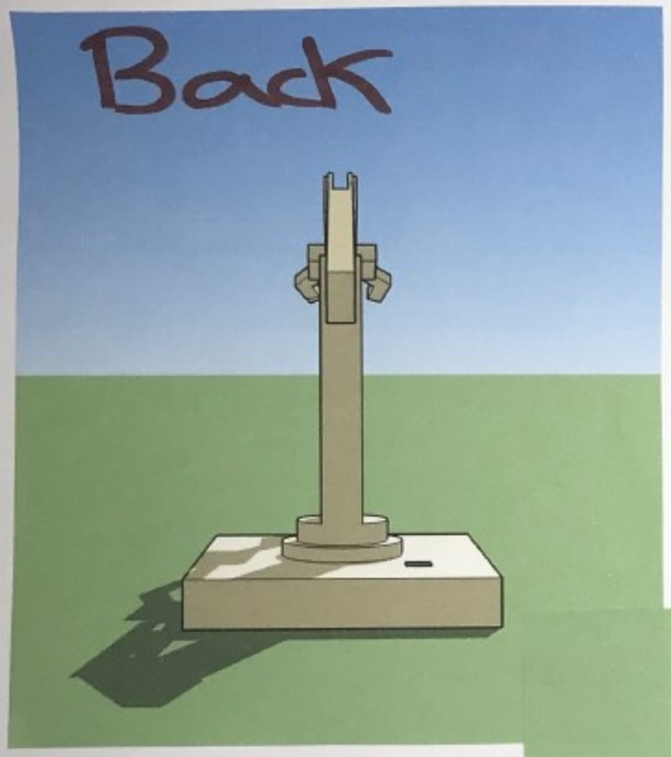
These 2 functions are repeated together 5 times with the other motors in order to be able to individually control the 2 directions of each of the 5 motors. This results in 10 total functions and buttons.

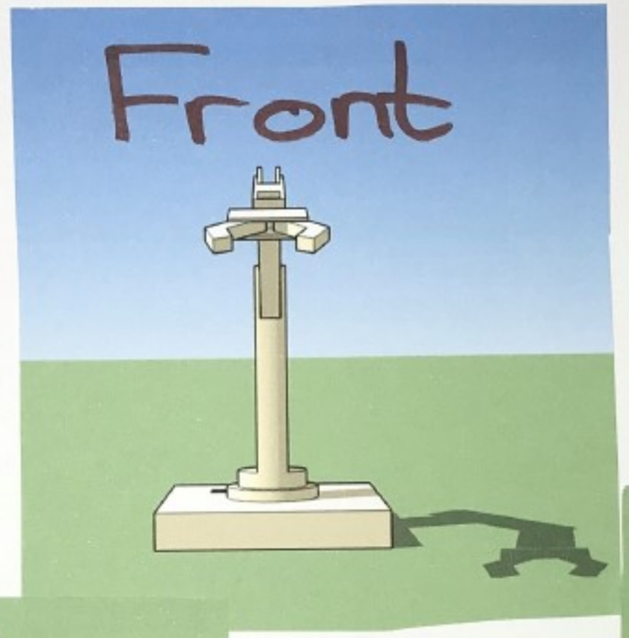
56 – 57:
If motor clockwise is set to true and motor anti-clockwise is set to false, motor will spin at -0.5 speed (Reverse polarity (clockwise))

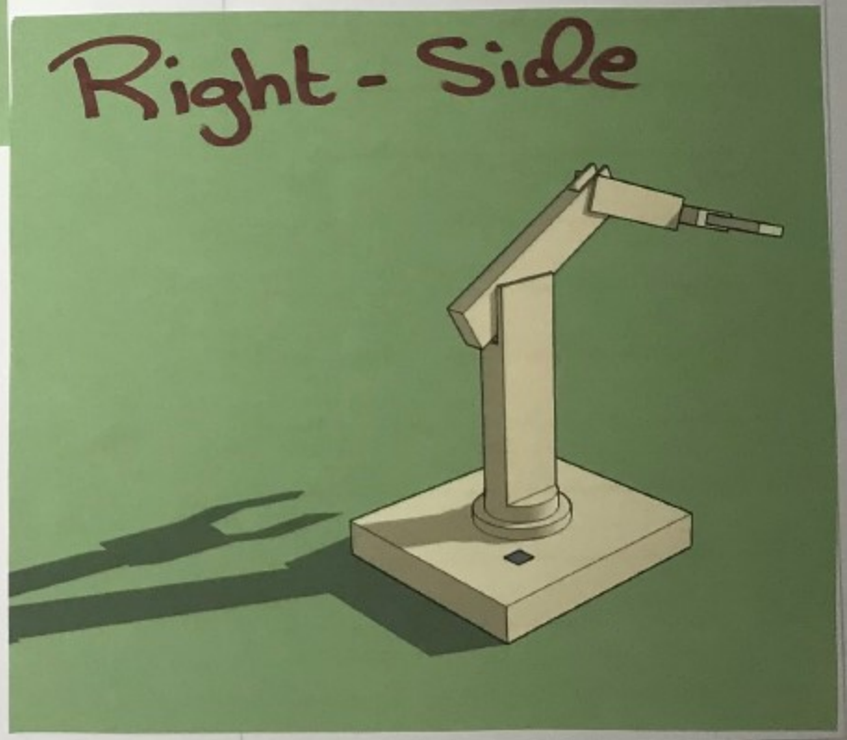
```python
if __name__ == '__main__':
    app.run(debug=True, host="0.0.0.0", port="33")
```

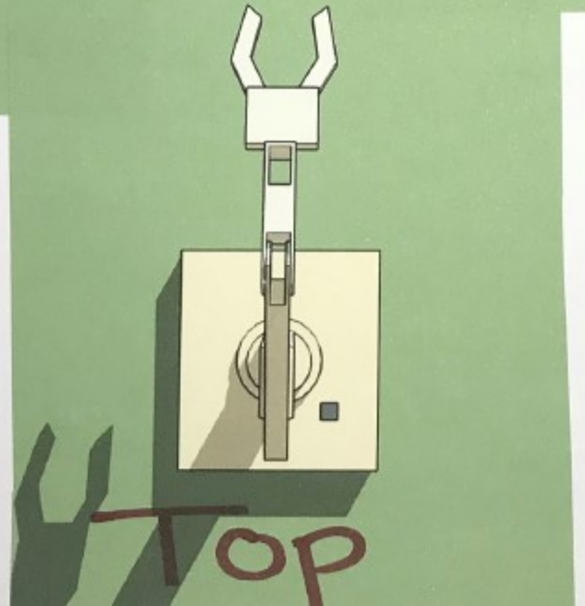
174:
Sets the host to "0.0.0.0" which is the public ip of the raspberry pi and the port to "33"

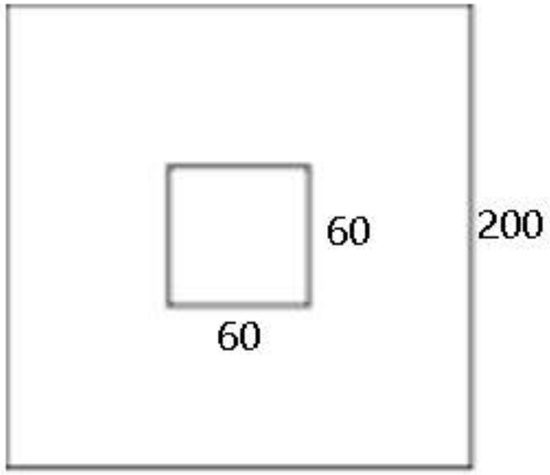
Model Design and Manufacture

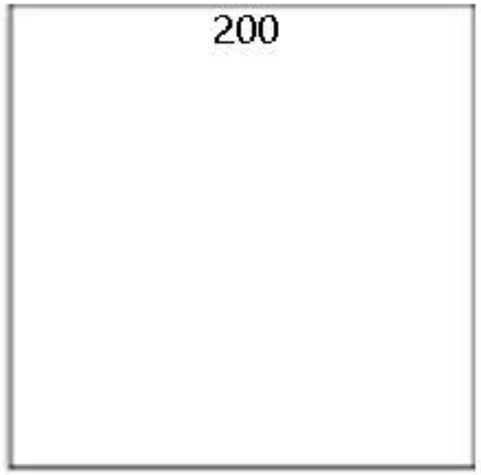




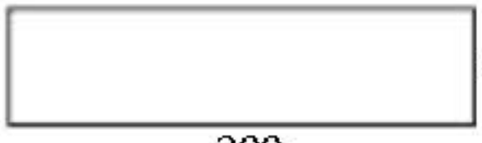





60
60
200
200


200
200

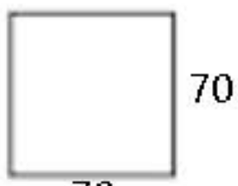

50
200

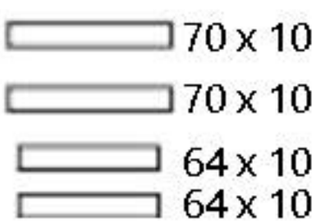

50
200

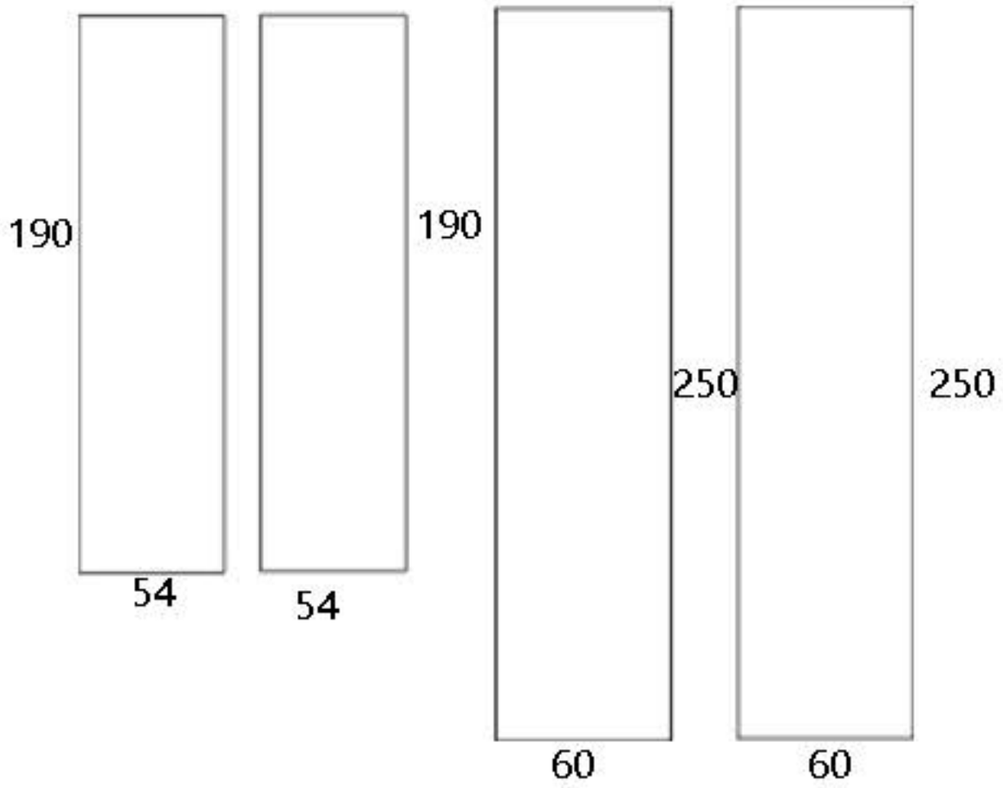

50
176


50
176

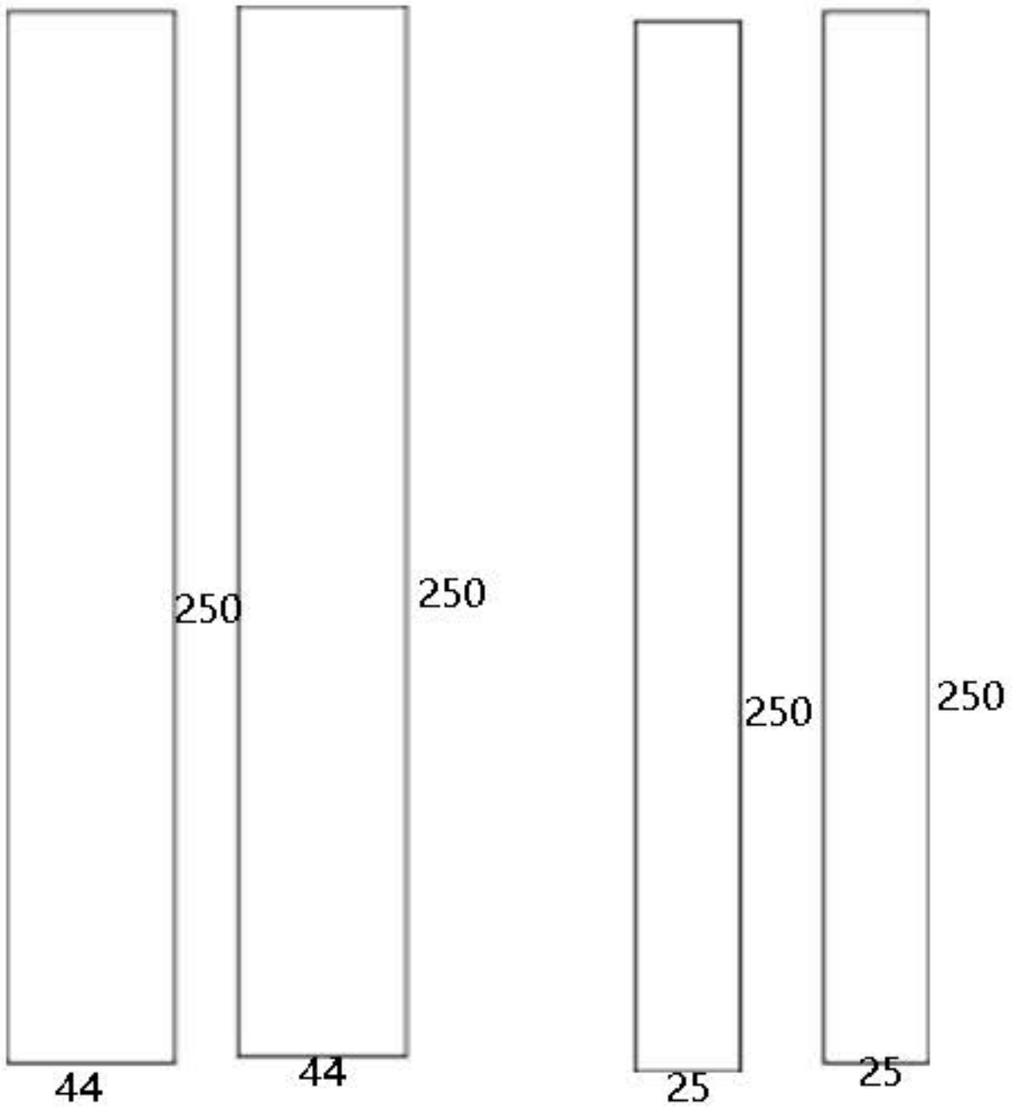
Base


70
70

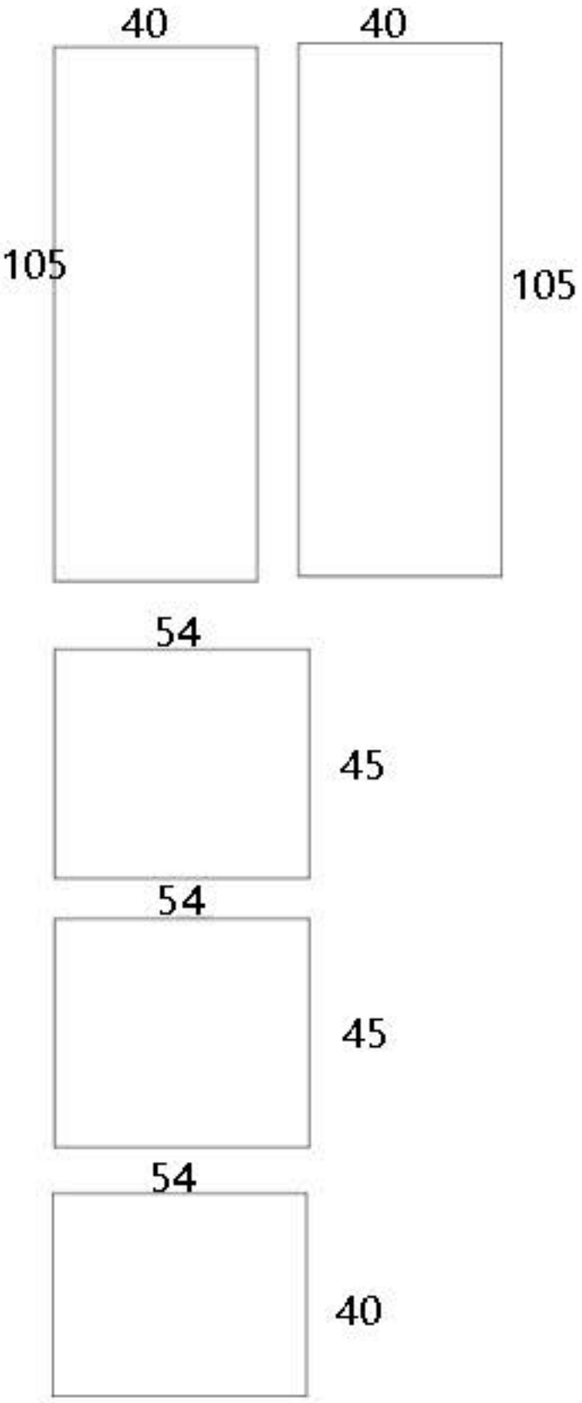

70 x 10
70 x 10
64 x 10
64 x 10



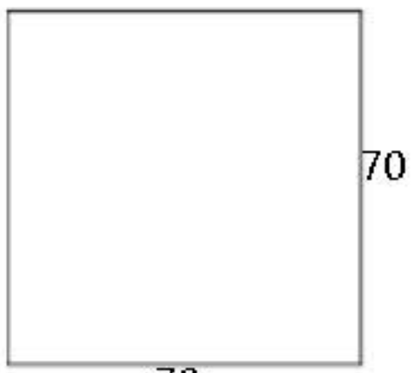
Neck

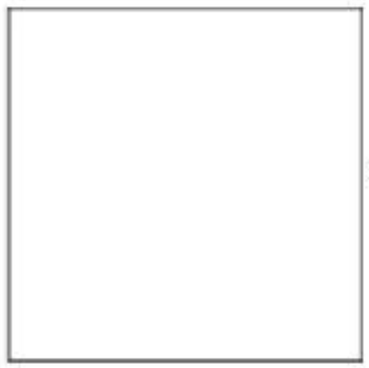


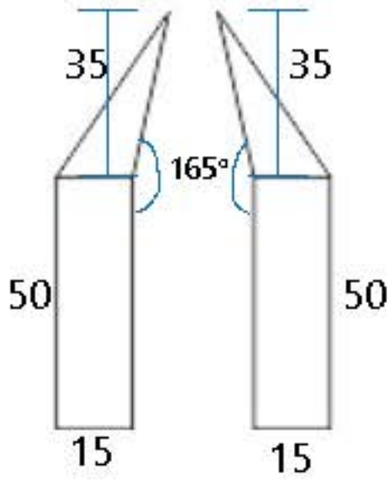
Arm



Joint





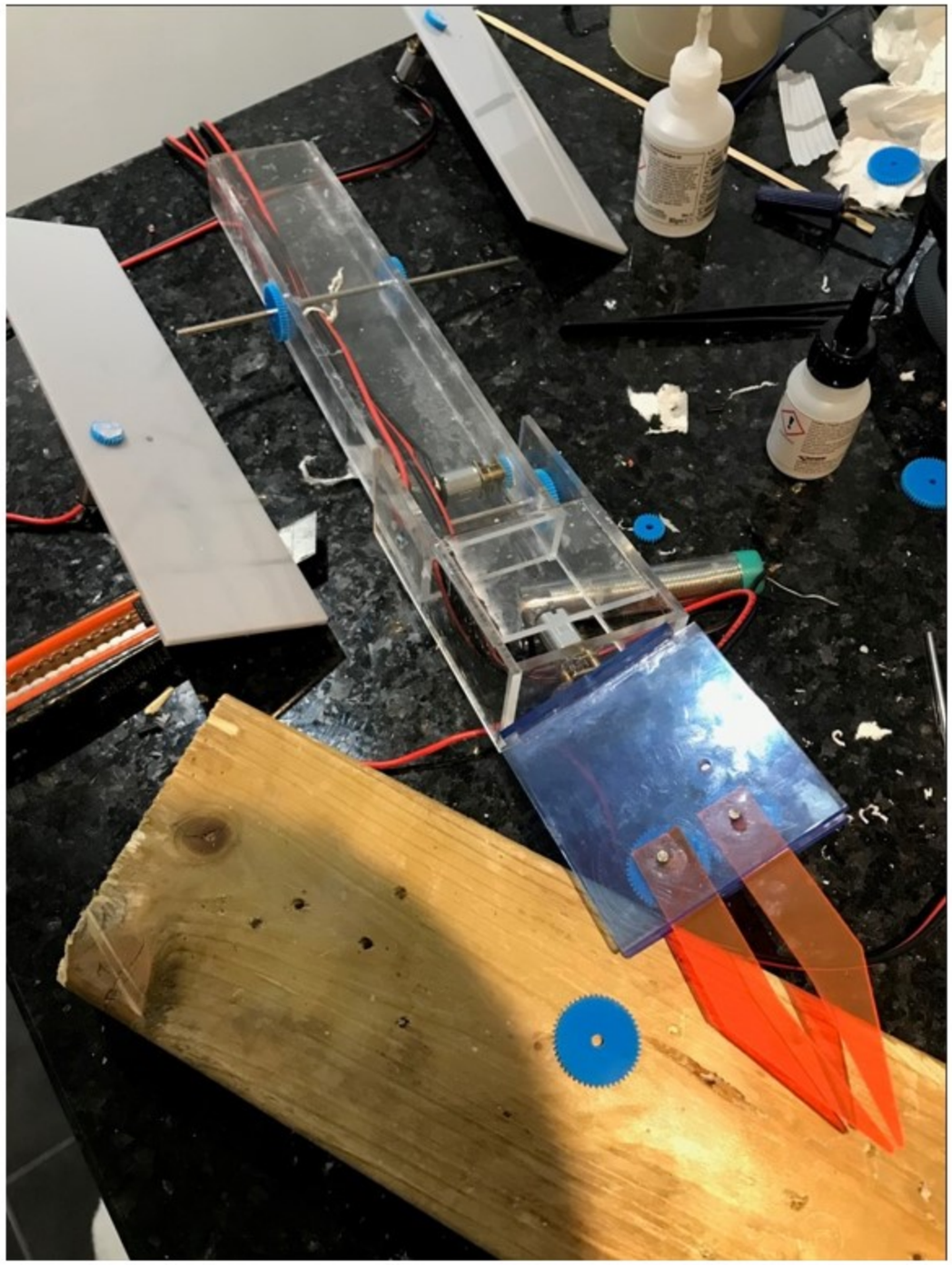


Grabber

This is scale cardboard model that I designed and built in order to have an accurate and precise reference when developing my working prototype. I used the measurements from the pages before which were designed in CorelDraw

Laser Cutter

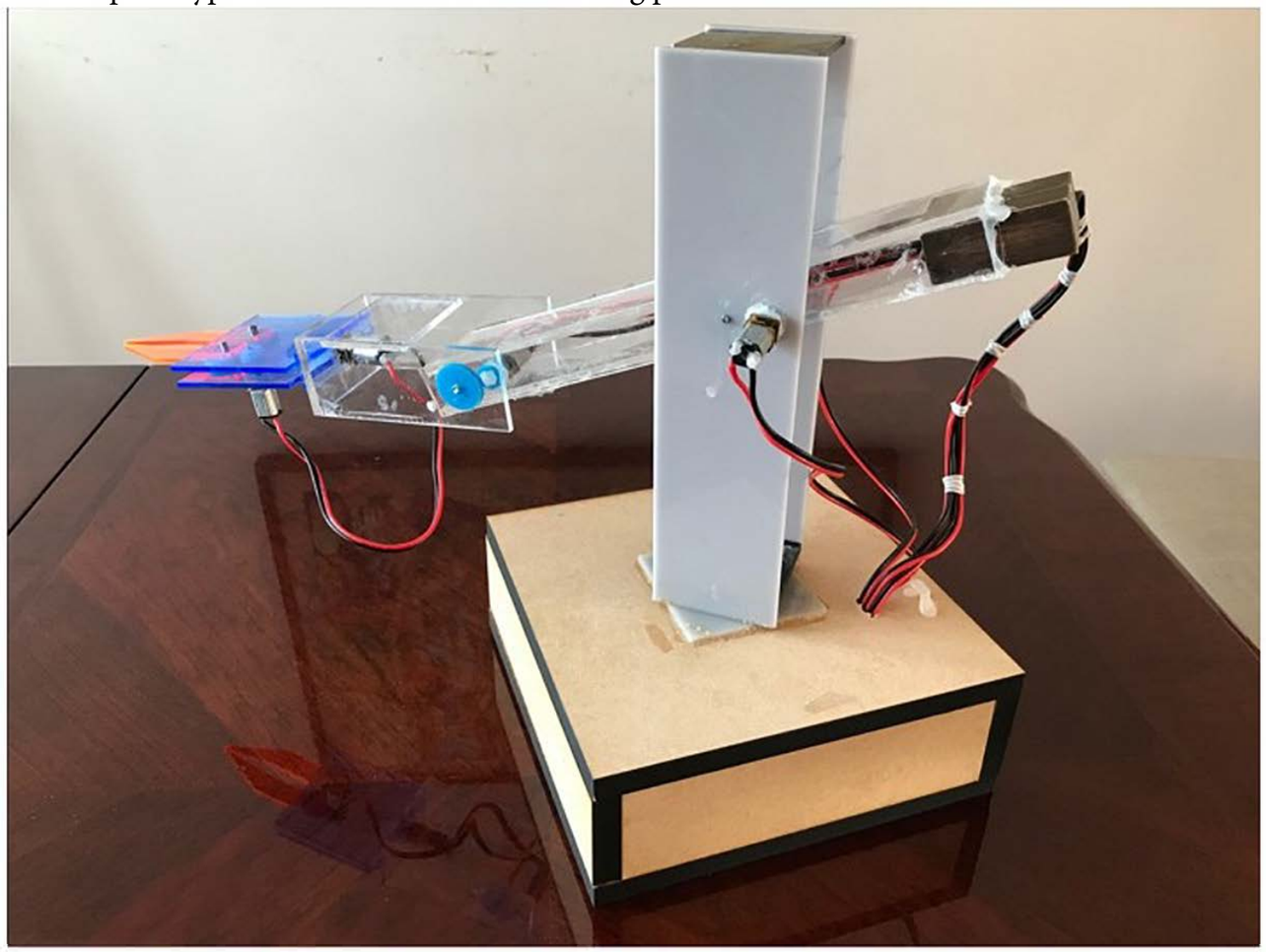
Manufacturing process

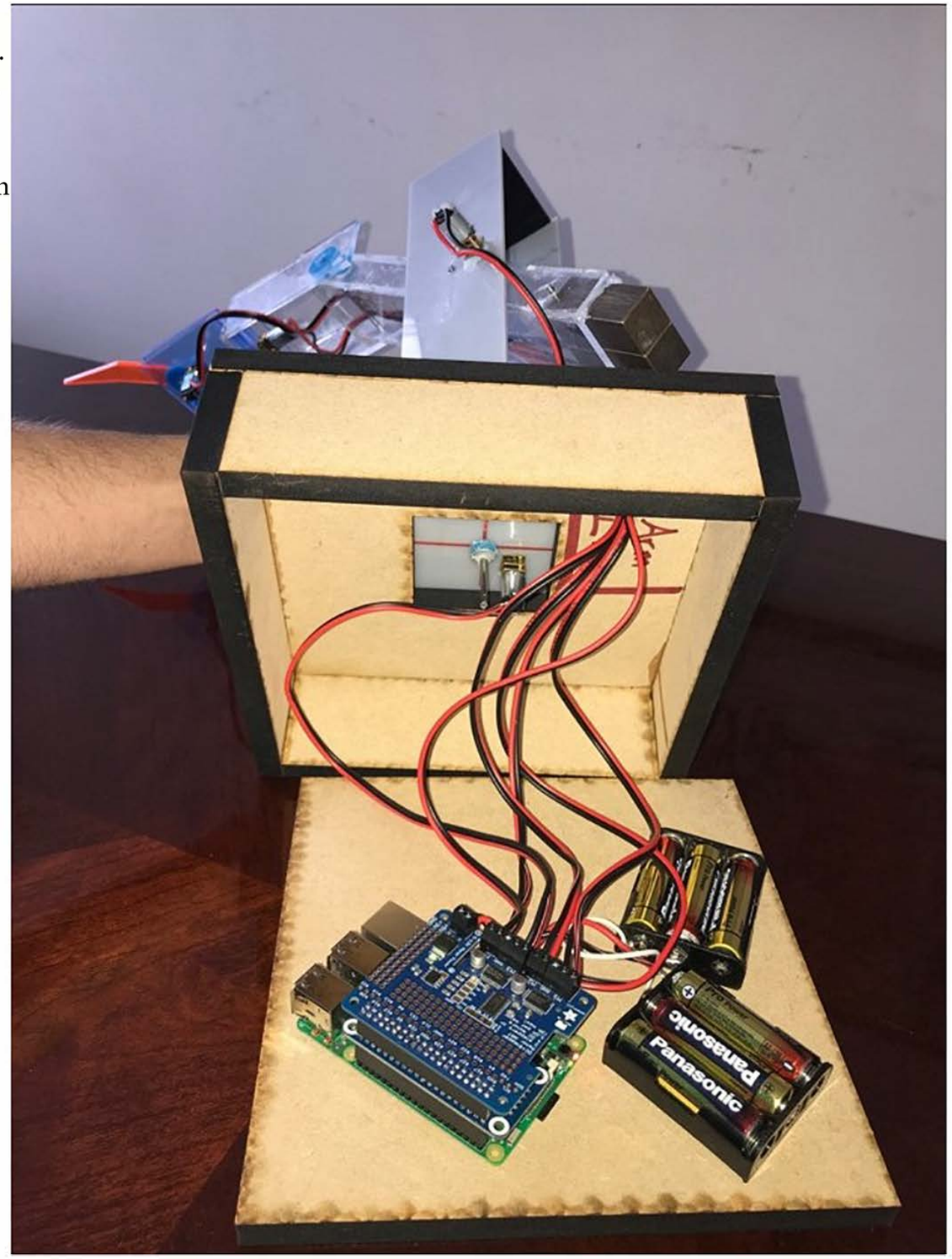


Gluing the parts and assembling the motors and gears to the body

Finished Working Prototype model

I used the dimensions shown in the previous slides when laser-cutting the pieces in order to get a perfect scale model, however, while building this, I was faced with many setbacks and learning opportunities which meant that I had to manually adjust my design to accommodate other things. For example, I had designed my robot to lift the main arm with one single motor but after testing, I noticed that it could not handle it's weight and as a result, I added another motor to the other side which spun in the opposite direction for double the motor strength. Observations like these will be taken into account as a learning opportunity and from them, I will be able to improve upon further prototype models until the final working product.





Working prototype robot with web server

Feel free to watch a short video on how my robot works and the work that went into it.

Click the picture below to be forwarded to a Youtube link. Otherwise, enter this into your browser url:

www.shorturl.at/lmyzH