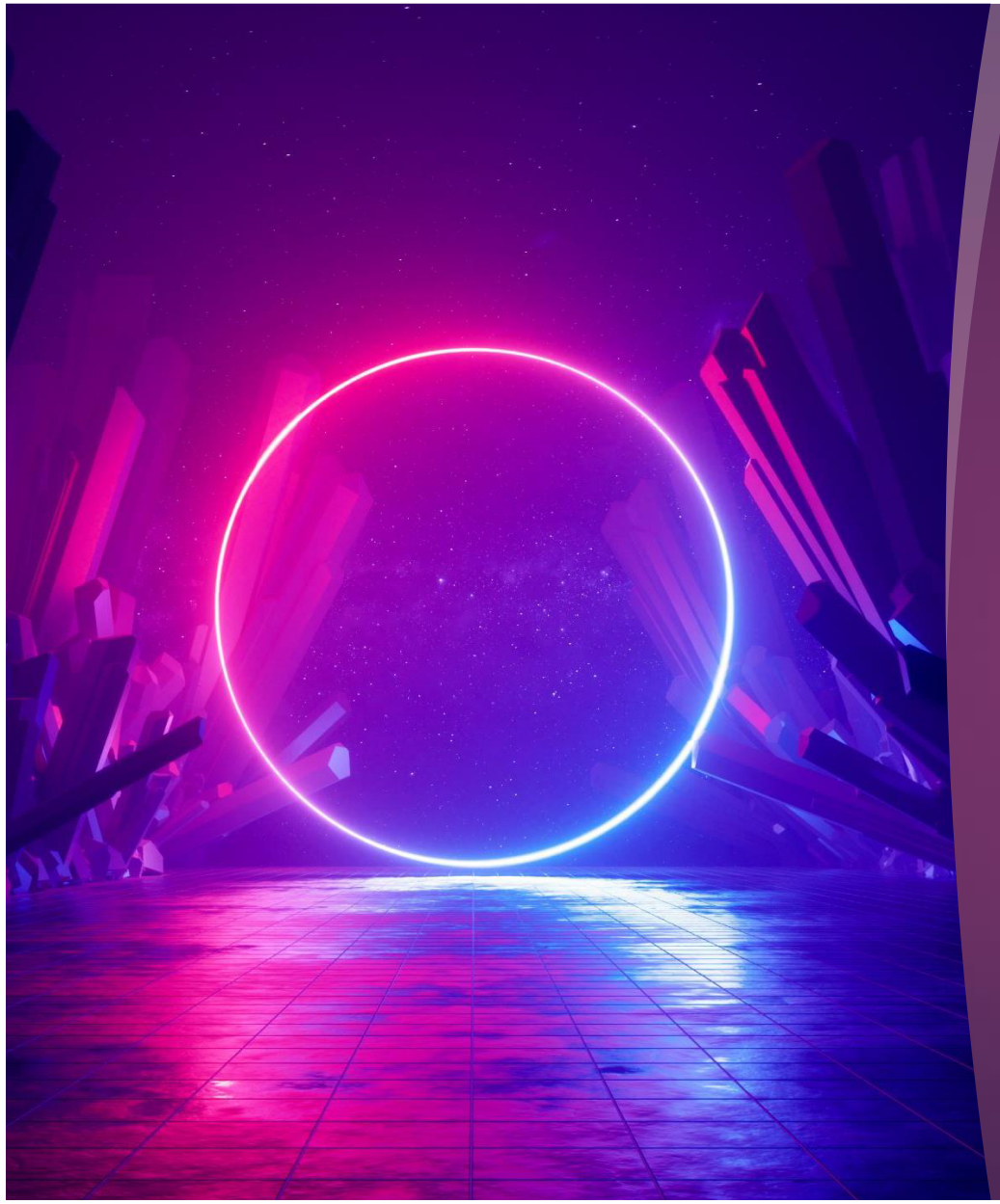


Muscle Sensor

BY: LAMONT MOORE, DANTE HAMPTON AND DANIJEL CORIC

Background / Recognition of need

The goal of this project was to display that prosthetics can be made using off-the-shelf components

Use a device (EMG sensor) to sense myoelectrical activity of a hand opening and closing and by doing that the electrical activity can be turned into a signal; that we can then see display the signal on a grapgh.

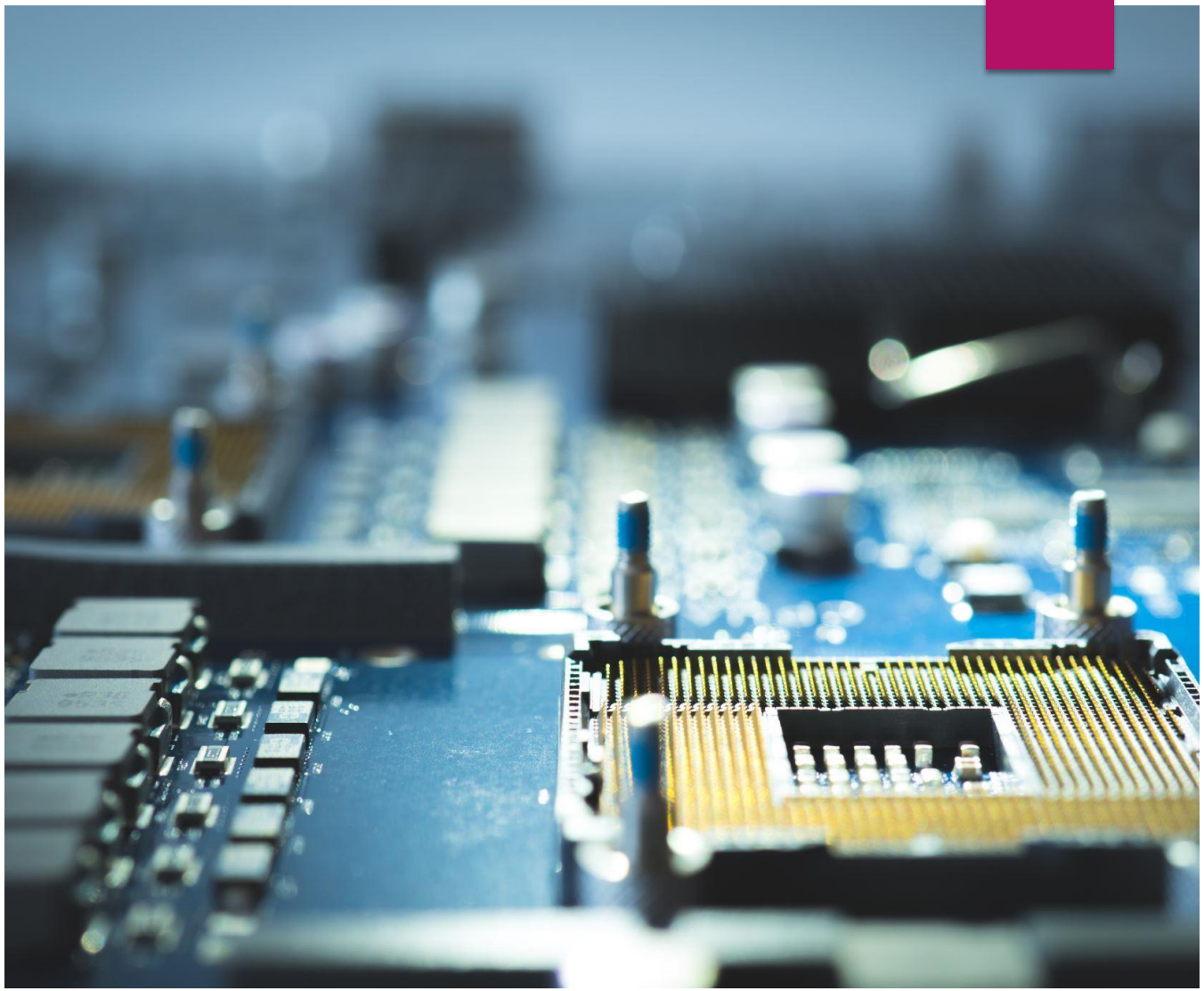
Many prosthetic apendiges on the market can cost from $15,000 to $23,000

FMEA

Function Affected	Potential Failure modes	Serveraity rating
Sensor Pads placed wrongly	Sensor pads won't read correctly	5
TRS Cable unplugged	TRS Cable won't properly read and give information to pads	5
LED breadboard won't light	LED will be irresponsive	0
Computer output display won't turn on	Will be unable to read information provided	3
EMG pad cables unplugged or has an electric short	Pads won't be able to retrieve proper data from arm.	2
Power(Cable) unplugged	Device won't turn on	7
Power(Battery) Dead	Device won't turn on	1

Methods/ Device iteration

- The first version of the unit was going to include a breadboard of LED's that would display the intensity of the myoelectric signal by illuminating more bulbs.
- The reason for the change is because the Arduino module for the muscle sensor occupied every available pint port on the Arduino UNO
- The current design of the unit is a muscle sensor that connects to the pin sockets on the Arduino Uno board and that Arduino connects to the laptop through USB and displays the signal on the screen.



Problem formulation Inputs

Interface: The Patient would be able to use an app created by the prosthetic manufacturers to better control and understanding of their prosthetics.

Performance Requirements: Device works decently well and has a response time in almost an instant of being used. The muscle sensor will be attached to the patient carefully to not cause irritation of the patient.

Interface Requirements: Screen that can help show how good the patient's muscle is after being attached to the muscle sensor.

Solution Formulation

- The completed product is a muscle sensor that can for display can pick-up the myoelectric signals from the muscle in the forearm and depending if the hand is closed (meaning that muscles are contracted) and a line graph will appear. The graph will show a stronger signal if the patient's hand is open (muscles aren't contracted the screen will display a weaker signal. The myoelectric signal will change depend on the strength of the signal, determine how many finger is holding up.

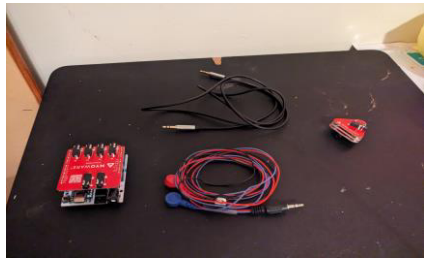
Solution Formulation (software)

The software to run this project is written in two parts:

Part A is code written for the Arduino UNO and collects the signal from the muscle sensor

Part B is a python script that plots the change in the signal to a graph on a screen and it uses the pyrexial and matplotlib modules

Results(but with pictures) parts



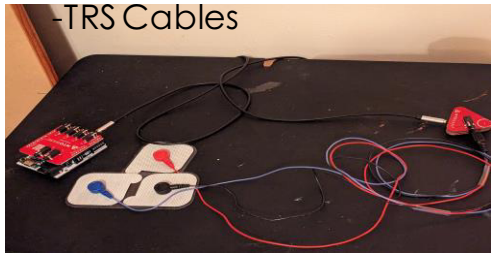
-Muscle sensor: convert myoelectric signal from muscles to analog signal.

-Link & cable module: send signal from sensor to Arduino module

-Arduino board module: Allows the myoelectric signal to be sent to the Arduino.

-Dual EMG patches: pick up muscular contractions.

-TRS Cables



Results(but with pictures) software

```
int data = 0;
char userInput;

void setup()
{
  Serial.begin(115200);
  while (!Serial); // optionally wait for serial terminal to open
  Serial.println("MyoWare Example_01_analogRead_SINGLE");
}

void loop() {

  if(Serial.available()> 0){
    userInput = Serial.read();
    if(userInput == 'g'){
      int sensorValue = analogRead(A0);
      data = analogRead(sensorValue);

  Serial.println(sensorValue); // print out the value you read

  delay(50); // to avoid overloading the serial terminal
    }
  }
}
```

```
Done Saving.

Sketch uses 1992 bytes (6%) of program storage space. Maximum is 3
Global variables use 224 bytes (10%) of dynamic memory, leaving 18

33
```

Part A: arduino code

Part B: python script

```
import time
import serial
import matplotlib.pyplot as plt
import matplotlib.animation as animation

def animate(i, dataList, ser):
    ser.write(b'g')
    arduinoData_string = ser.readline().decode('ascii')
    #print(i)
    try:
        arduinoData_float = float(arduinoData_string)
        dataList.append(arduinoData_float)
    except:
        pass

    dataList = dataList[-50:]

    ax.clear()
    ax.plot(dataList)

    ax.set_ylim([0, 1200])
    ax.set_title("Muscle Sensor Data")
    ax.set_ylabel("Value")

dataList = []

fig = plt.figure()
ax = fig.add_subplot(111)

ser = serial.Serial("/dev/ttyACM0", 115200)
time.sleep(2)



ani = animation.FuncAnimation(fig, animate, frames=100,

plt.show()
ser.close()
```

Questions?