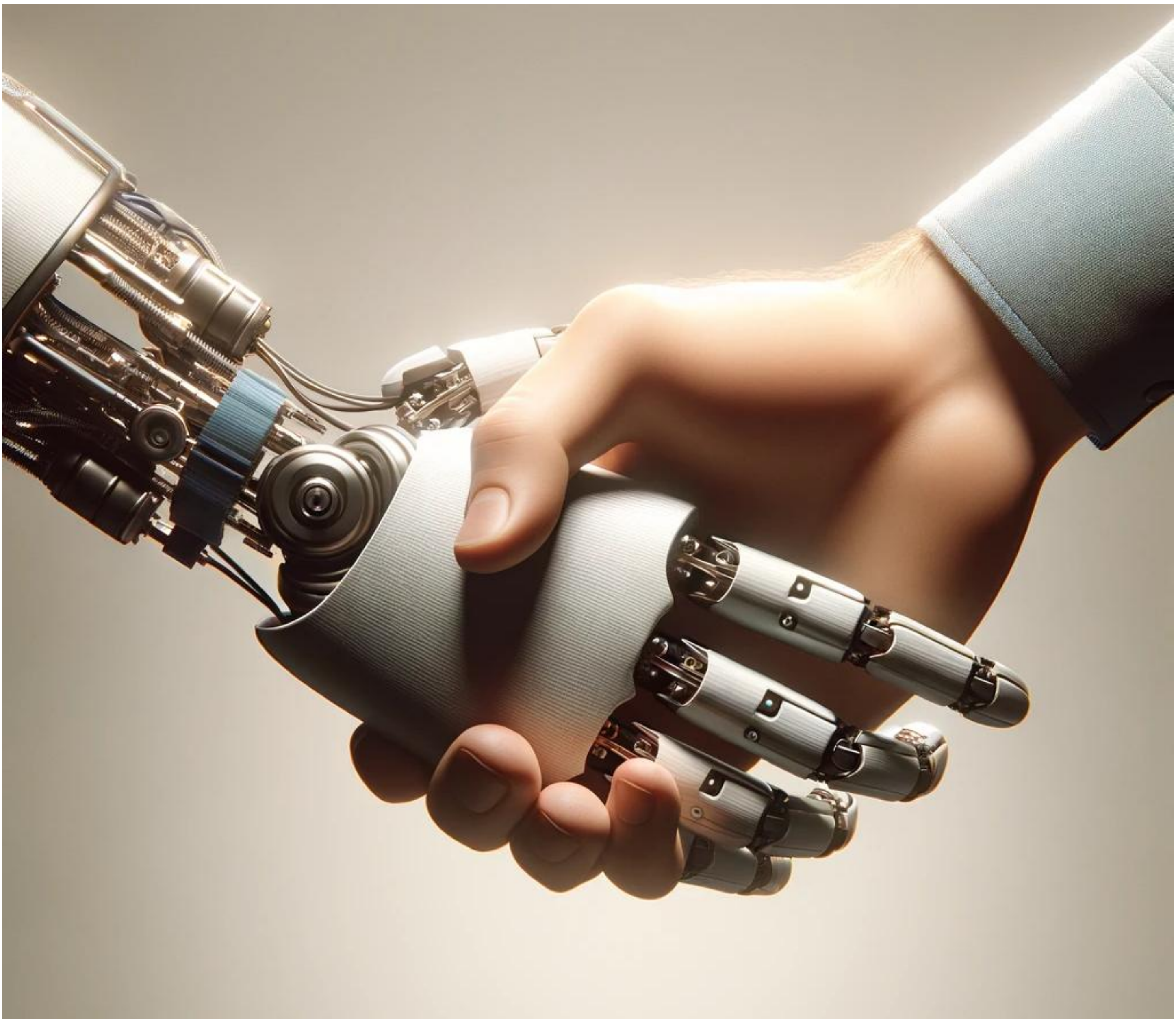


Technology & The Body

A non-invasive, Inexpensive Prosthetic Arm Prototype

(20098910) Shane Whitmore 11/8/24 Bsc in Applied Computing, IoT

Image Generated using ChatGPT

Contents

Abstract

"Technology & The Body: A non-invasive, Inexpensive Prosthetic Prototype" is a project with the goal of investigating the feasibility of developing an inexpensive, environmentally friendly, prosthetic arm alternative by using the United Nations: Sustainable Development Goals as the core foundations to the development of the project, taking into account several considerations such as "Goal 1: No Poverty", "Goal 3: Good Health & Well-being", "Goal 10: Reduced Inequalities" , "Goal 14: Life Below Water" and "Goal 15: Life On Land".

The Intention for this project is to use Inexpensive, easily sourced equipment so that anybody may reproduce the works.

Preface

The project idea came about when I visited the City of Arts and Science in Valencia in 2023. There was a game at the museum which involved two participants sitting at opposite end of a table, each placed their head on a sensor which measures brain activity. A ball in the centre of the table, would move to whomever could concentrate the most effectively causing them to win the game if the ball reached their side of the table. I became incredibly fascinated in how technology could integrate with human functions in such a seamless way.

Delving deeper into the subject, I began investigating the biomedical industry and the numerous ways technology can be used to improve people's lives. I discovered that, on average, prosthetic arms can reach $40,000 and more highlighting both the complexity of the technology and the medical procedures required to operate such a device.

I saw this as an issue primarily for young people from 7-18 as they continue to develop and grow. A prosthetic arm that requires invasive surgery to install sensors to operate, does not seem like a viable option for young people so I wanted to investigate the possibility of making a prosthetic that uses skin contact (surface ElectroMiography) sensors to operate. I then began questioning the possibility of designing a prosthetic that is inexpensive to develop while simultaneously being environmentally friendly by using recycled materials and or, biodegradable materials.

Introduction

United Nations: Sustainable Development Goals

"The 2030 Agenda for Sustainable Development, adopted by all United Nations Member States in 2015, provides a shared blueprint for peace and prosperity for people and the planet, now and into the future. At its heart are the 17 Sustainable Development Goals (SDGs), which are an urgent call for action by all countries - developed and developing - in a global partnership. They recognize that ending poverty and other deprivations must go hand-in-hand with strategies that improve health and education, reduce inequality, and spur economic growth – all while tackling climate change and working to preserve our oceans and forests." – [1]

Goal 1:

Global Goal 1, End Poverty in All its Forms Everywhere. The COVD-19 pandemic caused extreme poverty to increase in 2020 for the first time in decades, reversing global progress by three years.

The significance this holds on my project is to create a prosthetic alternative from inexpensive, easy to source materials, so that a person can have access to such a technology, regardless of their economic and social situation.

Goal 3:

Good Health and Well-Being, aims to ensure healthy lives and promote well-being for all at all ages.

As this project focuses on inexpensive, non-invasive technologies for a prosthetic alternative, it aims to promote well-being for people of all ages, prosthetics that cost $40,000 may not be accessible for families especially considering that children will inevitably grow out of them.

Goal 10:

Global Goal 10, Reduced Inequalities, highlights the empowerment and promotion of the social, economic and political inclusion of all, irrespective of age, sex, disability, economic or other status.

Over the past 5 years, the gap in per capita income growth between the poorest and richest countries has widened.

The challenge for my project is to create a prototype inexpensive prosthetic alternative so that it can be reproduced using easy to source equipment.

Goal 14:

Life Below Water aims to target several issues such as illegal fishing and reducing plastic pollution therefore, another factor that I would like to consider with my project is the use of environmentally friendly and/or recycled materials to cut down on pollution of our oceans.

Goal 15

Life on Land underscores the importance of biodiversity as humanity's life-support system and aims to address the pressing global environmental challenges and crises including pollution. Similar to Global Goal 14, my challenge is to incorporate environmentally friendly and/or recycled materials.

Project Goal

With the Sustainable Development Goals in mind, the goal of this project is to build a prototype non-invasive, inexpensive prosthetic arm alternative using environmentally friendly material and easy to source components.

Project Requirements

Hardware

To achieve the goal of this project, I would require hardware that is lightweight and has the computational power required for the prosthetic to function as intended. As one of the goals of this project is to keep the cost down, these components must also be inexpensive and easily sourced.

I intend on researching the ESP-32 which is a microcontroller roughly half the size of a credit card and capable of handling the many aspects required to make this project possible. I will also be investigating and implementing the MyoWare EMG (electromyography) sensor which will be used for monitoring muscle stimulation and communicating that data to the ESP-32. Both components are lightweight and are low in power consumption.

Other hardware would include wires for connecting the component and a 3D printer. As I already own a 3d printer, I will be using this to print a prototype design using filament (the plastic for the 3D printer).

Software

The software that I will be using is open source, this means that they are free to use and easy to access. This software ranges from CAD software for modelling the prosthetic prototype, to an Integrated Development Environments (IDE) for programming the prosthetic prototype. I would like to make a GitHub Repository publicly available with the software I create so that anyone can download it for their own use.

I have many years of experience with CAD software and the software I intend on using is fusion 360 as it is free and comes with options to export files in formats that are readable by 3D printing software.

I will be using both Creality slicer and Prusa slicer which are slicing software's that turn files into g-code. G-code is a programming language designed for machines and is, in its simplest terms, a set of instructions which tells the machine (3d printer in my case) where to move, how fast it must move and what path it must follow.

Lastly, to program the microcontroller, I will be using the Arduino IDE as it has an extensive list of tools which will allow me to view the output signal of the device which will be extremely useful for debugging my code should I run into difficulties. The Arduino IDE also allows you to program the ESP-32 in this environment.

Aspects to Consider

Functional Aspects

- Non-invasive sensor for operating the prosthetic
- Prosthetic Weight
- Maximum weight Limit (not investigated as I am limited on resources)
- Open/Close Speed

Non-Functional Aspects

- Durability against weather conditions
- Durability against temperature
- Durability against fall damage and impacts
- Cost

Project Objectives

Model a prosthetic

One of the tasks for this project is to either model a prototype arm from scratch which would take a long time or, find an existing design that I like and modify it to function as a prosthetic. This would involve adding cavities or fitting points in the existing design which will hold the components as well as fastening points for threaded inserts and screws so that the individual parts can be assembled. I will be using a site called GrabCad which is a library that offers millions of free CAD designs which also includes prosthetic arm models.

3D print and assemble prosthetic

Once I have found a design that I like and have modified it to accommodate for the components to operate the prototype, the next task is to begin printing and assembling. I intend on using PLA (Polylactic acid) filament as it is not only inexpensive (€25 per 1KG spool) but it is a biodegradable material which ties into the SDG goals that I am trying to align this project with (Goal 14 & Goal 15). [5]

Investigation and Implementation of the MyoWare EMG sensor

The sensor that I intend on using to control the prosthetic is the MyoWare 2.0 sensor. I will need to gain an understanding of this sensor such as how it functions, what is the output signal, compatibility with the ESP32 microcontroller etc. (more on this later)

Control prosthetic using EMG sensor

The biggest step of this project is to successfully open and close the prosthetic using the MyoWare sensor. This is the most significant part of the project and will require all other aspects of the project to function correctly (wireless control, data interpretation etc). When I get this working, I would like to investigate the possibility of recreating several gestures found in most high-end prosthetic arms. These gestures include pointing and, locking the hand to more easily carry things such as shopping bags or water bottles etc.

Methodology

Kanban

KanBan is a methodology that focuses on the visualisation of work. It consists of several ‘boards’ representing the ‘To-Do’ tasks, the ‘Doing’ tasks and the ‘Completed’ tasks. This is a methodology I am comfortable with using because I have used it for many projects over the last few years including developing a productivity app which is currently in beta testing. I like this methodology because I can see the work that needs to be complete, what I am currently working on and the work that is completed and it helps me organise my tasks and get an understanding of the time required to complete the project.

Scrum

I am using Scrum for my weekly meetings with my mentor. I feel like it is a useful tool to communicate the work that needs to be done each week. I like the use of sprints to get valuable work complete in short periods of time so I feel that this methodology will keep me pushing the project forward to achieve my tasks on time.

GANNT

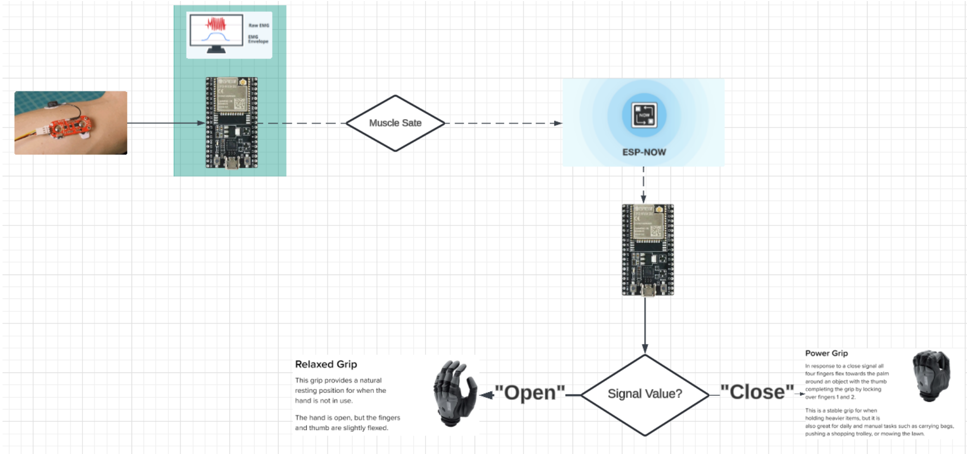
GANNT charts is something I am new to and was introduced to them during my flexible semester. It is a method that allows you to break down each step of the project and graph the time frame required to complete the task. I found this to be a great tool to use as a guideline for what aspect of the project I should be working on.

Productivia

Productivia is an app that I developed over my flexible semester, I will be using this app to set notifications and tasks. It uses a minimal interface to focus my attention on the task at hand. I can also set reminder dates and due dates for my tasks so I will receive notifications when I am nearing the deadline of a task in my project.

Design

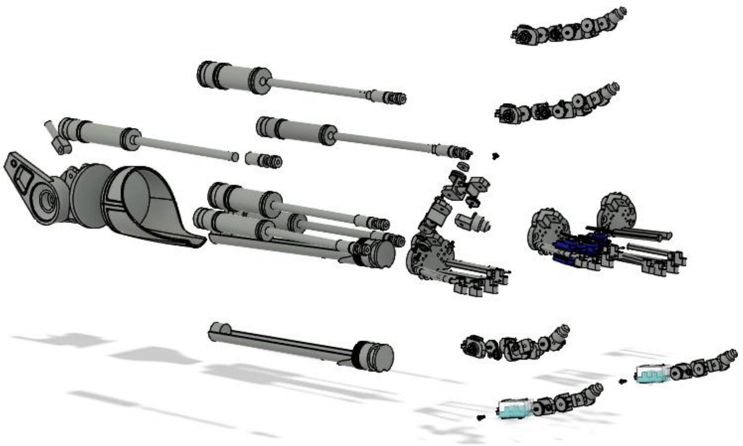
Flow Chart



The above flow chart illustrates how I intend to operate the prosthetic prototype using wireless communication between 2 esp-32 microcontrollers. In the top left-hand corner of the flow chart, The EMG sensor is connected to the GPIO pins of the esp32 which are used to power the sensor and receive the enveloped signal (clean signal) from the sensor. Once the data is received from the sensor, the esp-32 can determine if the muscle is stimulated (representing a closed fist) or if the muscle is relaxed (representing an open fist). This is then translated into a Boolean value of ‘1’ or ‘0’. Using ESP-NOW, the signal is wirelessly transmitted to the esp-32 microcontroller used for controlling the prosthetic prototype. The value is checked once again, and the prosthetic prototype will open or close based on the value.

Arm Model

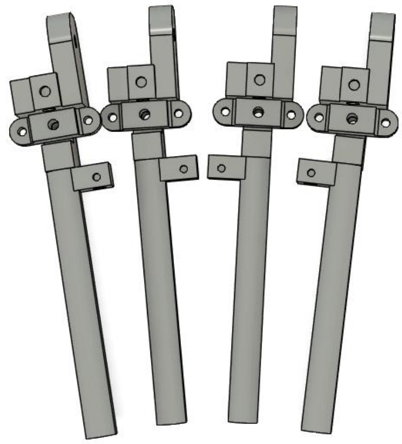
[6]



The above image is an exploded view of the model that I downloaded from GrabCad. I created an exploded view so that I could inspect each individual part and understand what modifications are required so that this could be assembled in real life. The main modifications required across all the parts is to include holes for screws and threaded inserts so modifications mainly consisted of making holes for threaded inserts and screws so that I could fit all of the parts together.

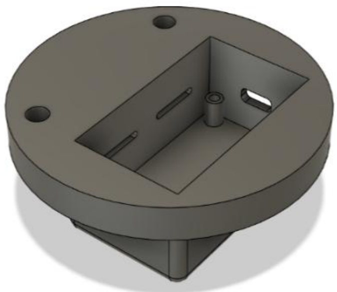
I then began re-assembling the arm with the modified parts to ensure that everything fitted together again before I exported and 3D printed parts. (see in above image)

The largest modification I made to this model is the knuckles as they required a mounting point for the servos which will be used to open and close the fingers. This required me to print sections of the original knuckle so that I could take measurements and understand what the part would look like in real life. Once I did this, I could then re-design the knuckles in fusion 360 to hold a servo.

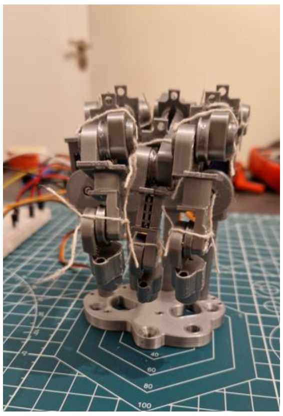


*Original Knuckles *Servo Fitting Knuckle

Along with modifying the knuckles to fit the servos. I needed to create an area for the ESP-32. I decided to put the ESP-32 in the elbow as this has the largest area to work with so there is lots of space to fit the microcontroller and any other potential hardware. When designing this elbow, I wanted to keep in consideration the temperature that the ESP-32 could be operating at and how that may affect the integrity of the plastic around it so I decided to make 4 cutouts which will act as ventilation. These 4 cutouts are positions just at the edges of the micro controller. Along with these cut outs, I added a port for the cable. This port will be used for programming the prosthetic, debugging my code and powering the prototype however, in a real world scenario, an on board battery would be required to operate the device but as this is a prototype, I will be using a plug to power it. Lastly, I included stand offs which will allow for clearance for the GPIO pins as well as acting as a mounting point so the micro controller can be fastened to the device so it won't shake around inside.



I began printing the hand first as this was not only the most intricate part of the prototype but also the most important. The above image shows the knuckles and palm printed and temporarily fitted so I could understand the clearance required for the servo wheels (used to retract the fingers by rotating and pulling a string). I modified the palm to have 2 cut outs which will allow me to route the wires down through the centre of the arm which will prevent them from pinching on anything or getting caught in moving parts and potentially damaging the prototype.



In the above photo, I began temporarily assembling the fingers by using string to tie the parts together. This was again, to understand the clearance requirements and also as a temporary assembly while I waited on the screws and female-to-female stand offs to arrive in the mail so I could fasten them together. One issue that I noticed was a clearance issue with certain parts of the fingers that are non-functional, so I was able to remove these parts without affecting the functionality of prototype.

The above photo shows the fingers being assembled using female-to-female stand offs which will act like a bearing for each joint in the finger.

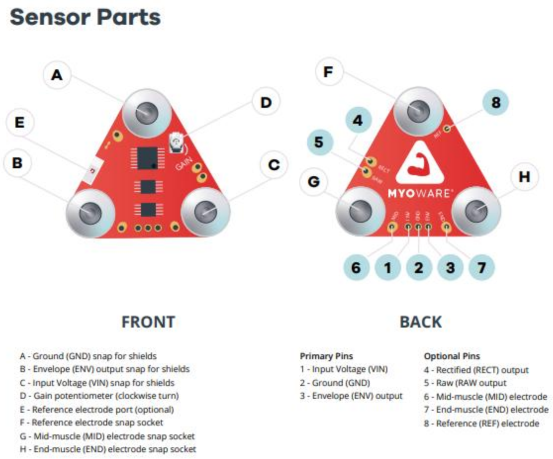
The above image shows one of the piston rod parts (located in the forearm) being printed. The piston rods are the only parts that did not need to be modified for this project and are the original parts found in the files that I downloaded; however, I was required to lightly sand down the ends so that it would correctly fit into the wrist joint and piston block parts (lower forearm close to the elbow). This fitting is a friction fit and did not require screws. I found this to be the hardest part to assemble as the fitting was so tight that the piston rod would constantly snap, and I need to reprint it multiple times with higher densities each time to provide it with the strength required to assemble.

Technologies

Hardware

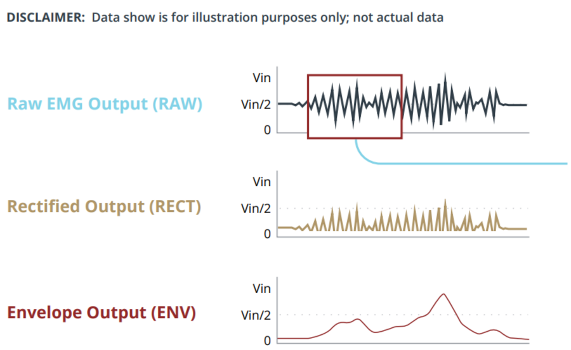
MyoWare Sensor

The MyoWare sensor measures muscle activity through the electric potential of the muscle. This sensor analyses the electrical activity of a muscle and outputs an analog signal that represents how hard the muscle has been flexed. [2]

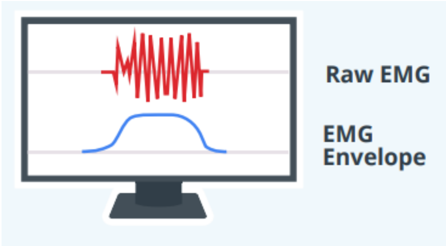


According to the manufacturer of this sensor, it is designed to be used directly with a microcontroller. This is extremely useful for my project as I plan on using the ESP-32 (a microcontroller) to wirelessly send commands to the prosthetic (controlled with another ESP-32).

The primary output of this sensor is the envelope of the amplified and rectified signal which is ideal to work with a microcontroller's analog-to-digital converter (see diagram below).



I will be using the envelope signal (ENV) for this project. As the signal is far cleaner than the raw and rectified signals and because of the signals shape, I could consider using an upper and lower limit (90% and 10%) to determine if the muscle is flexed or not and based on this, instruct the prosthetic to open or close or remain in the same state.



ESP-32

The microcontroller I will be using is the ESP-32. The ESP-32 is a low cost, development board created by Espressif Systems. There are several reasons as to why I have chosen this board all of which will allow me to achieve my goals of this project while still allowing for future development such as integration of additional sensors.

Espressif Systems are a global leader in wireless communication technologies and support diverse smart connectivity scenarios. They provide extensive documentation on all of their technologies from hardware to wireless communications protocols (WCP). ESP-NOW is the WPC that I will be investigating for this project. ESP-NOW enables direct, quick and low-power control of smart devices and is widely used in remote controlling and sensors. [3]

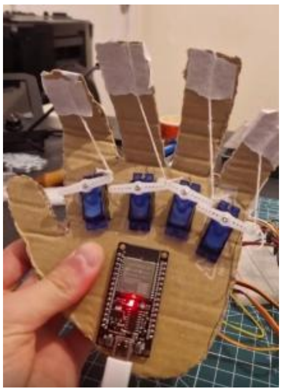
The additional benefits of using the ESP-32 are that it has on-board WIFI & Bluetooth, Analog-to-digital conversion and pulse width modulation. These features play a large role in this project as analog-to-digital conversion will be required to interpret the signals from the EMG sensor, and pulse width modulation is required for controlling the servos which open and close the fingers.

As the ESP-32 is small and lightweight, it makes for a fantastic wearable device. Along with this, the ESP-32 is a ultra-low power consumption device and is even advertised for mobile devices, wearable electronics and IoT applications. [4]

Project Feasibility

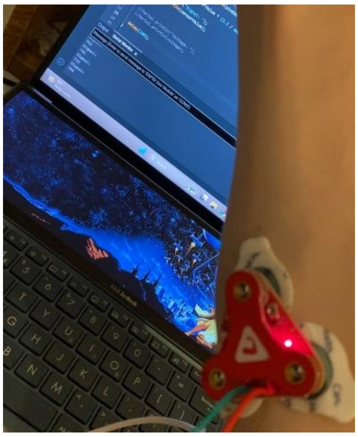
Current progress

Currently I have the entire arm 3D printed, however, I would like to make revisions to this design next semester as improvements can certainly be made on the fingers as well as making the prosthetic more modular should parts fail, it should be much easier to remove and replace parts. I must also wire the prosthetic so that it will operate as I am yet to do this.



The above image shows a demo hand made from cardboard. I did this so I could identify if it is possible to open and close the hand when the servos are positioned in the knuckles which is unlike traditional prosthetics where the servos are generally located in the forearm. I was also able to identify clearance issues with the servos and so I designed a wheel which sits on the servo and "collects" the wire when closing the fingers. Once satisfied with the clearance between components and ensuring that this design was feasible, I began modifying and printing the prosthetic (see below).

MyoWare Sensor



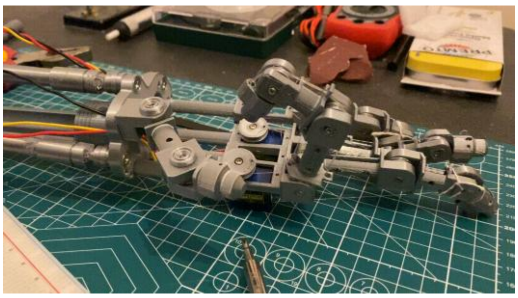
*Number signal of sensor output

*Graphed signal of sensor output

The above images show the output signal of the sensor. When the muscle is flexed, the signal is high and when the muscle is relaxed, the signal is low. What I found when using this sensor is that its position on the arm is extremely important to get a successful reading as in some cases, I did not get any reading and other cases I got a large reading. The other observation I made is that the sensor can be reactive in that the signal can fluctuate if the sensor is pushed too hard onto the arm, not positioned correctly (orientation wise) and/or the sensitivity is not adjusted using the gain potentiometer. This is something that I must keep in consideration when continuing to develop this project as an unpredictable signal may cause the prosthetic to act unexpectedly.

Finger movement using Servos

To ensure that the current positioning of the servos would work, I created a simple loop program that would spin the servos open and closed. This configuration works effectively. What I discovered is that I may need a method to stop the rotation of the servos to prevent it from pulling the string too far and either breaking the wire, breaking a joint in the finger or damaging the gears inside the servo itself. One potential solution is to use a limiting switch which may come in the form of a button, switch or flex sensor that can be monitored and instruct the microcontroller when to stop spinning the servos.



Semester 1 Discussion

I believe that my research went well. I enjoyed the process of modifying the model that I downloaded from GrabCad and then 3D printing it. This process was challenging and time consuming as I had to ensure that the design was possible by modifying the individual parts and reconstructing the entire arm in fusion 360 to confirm that the parts still fit before I could begin 3D printing.

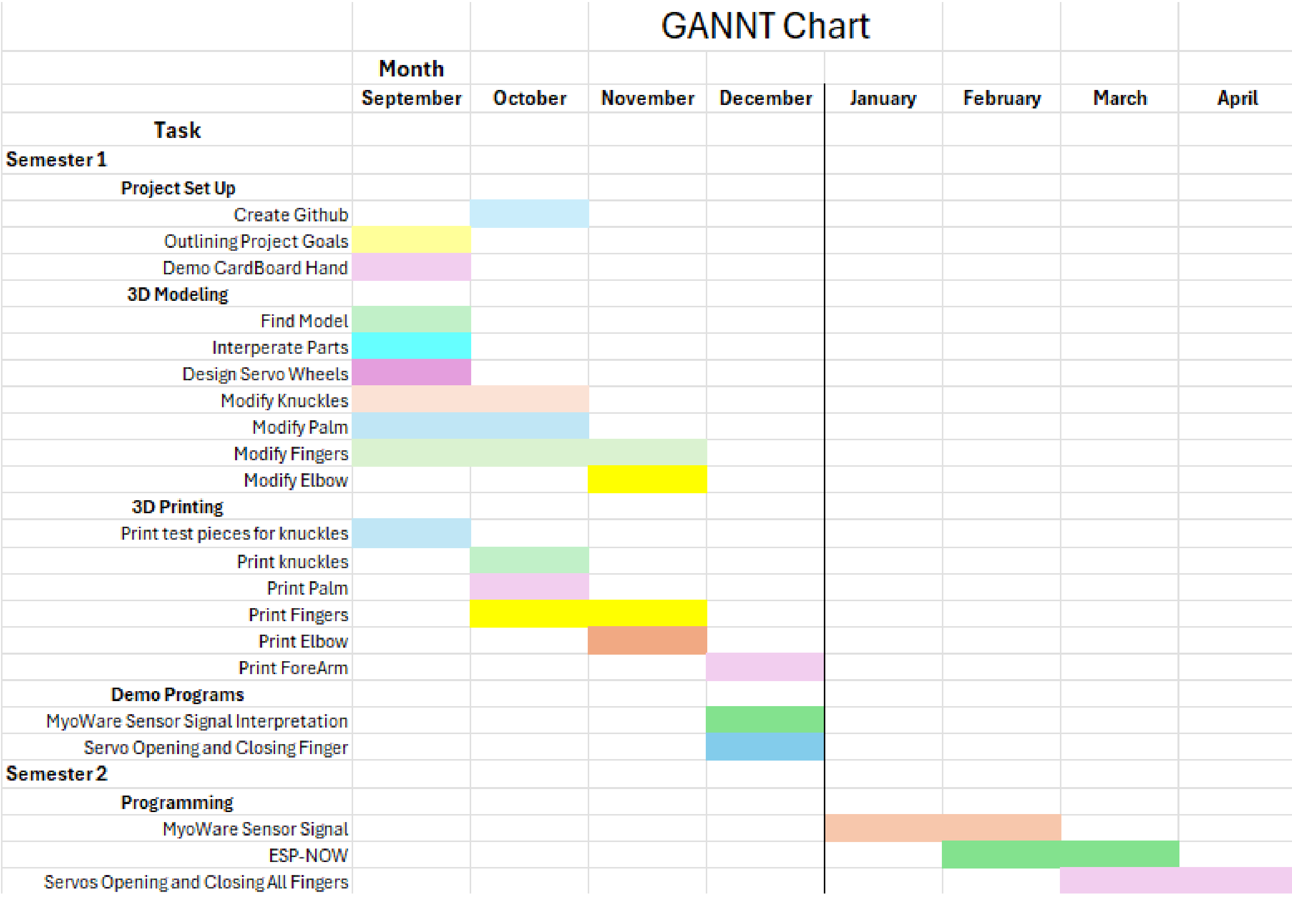
It was also interesting to conduct research on the MyoWare sensor as I have never used a sensor like this before. After testing the sensor with the ESP-32, I gained a far better understanding of how the sensor works and it was very informative that I quickly discovered that the positioning of the sensor on my arm plays an important role on how accurate the sensors output is. I later found that this is mentioned in the advanced documentation which is found on the manufacturer’s website.

Semester 1 Conclusion

In this report, I have identified the scope of the project. Once the scope was identified I researched the technology required to make this project possible and I then developed the prototype arm by downloading an existing design and making the appropriate modifications so that it can functions as a prosthetic.

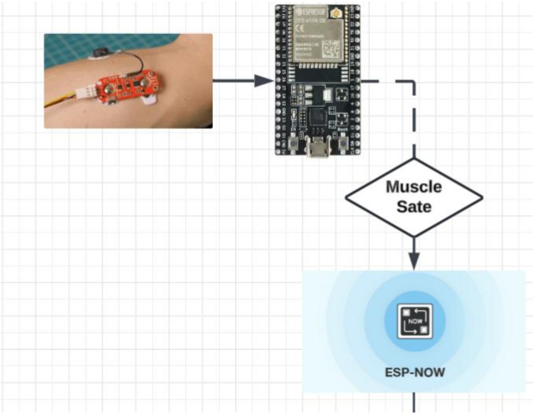
I have gained an understanding of what tasks I must complete next semester to achieve the goal of the project. I did this by creating simple programs to control individual aspects of the arm such as the electromyography sensor and the servos, and from this, I determined possible issue that I may need to address later down the line to improve the prototype.

The primary goal for next semester is to write the code for the wearable device and the prosthetic so that I can wirelessly control the prosthetic. Additional goals would include making improvements to the current design as outlined by my findings from investigating the EMG sensor and the servo rotation, and potentially implementing other gestures such as pointing or locking the arm into a fist.


GANNT Chart
Month
September
October
November
December
January
February
March
April
Task
Semester 1
Project Set Up
Create Github
Outlining Project Goals
Demo CardBoard Hand
3D Modeling
Find Model
Interperate Parts
Design Servo Wheels
Modify Knuckles
Modify Palm
Modify Fingers
Modify Elbow
3D Printing
Print test pieces for knuckles
Print knuckles
Print Palm
Print Fingers
Print Elbow
Print ForeArm
Demo Programs
MyoWare Sensor Signal Interpretation
Servo Opening and Closing Finger
Semester 2
Programming
MyoWare Sensor Signal
ESP-NOW
Servos Opening and Closing All Fingers

SEMESTER 2

MyoWare Sensor Signal



I started the semester by focusing on reading and interoperating the muscle stimulation signal from the MyoWare sensor. Both the ESP32 and MyoWare sensor will be part of the "Wearable" device (this will be referred to as such for the remainder of this report). The objective of the wearable device is to read the muscle stimulation of the forearm, it must then classify the muscle stimulation as either "relaxed/open" or "stimulated/closed" resulting in a binary value of "1" or "0". Lastly, the wearable must use ESP-NOW to wirelessly send this muscle value to the prosthetic arm.

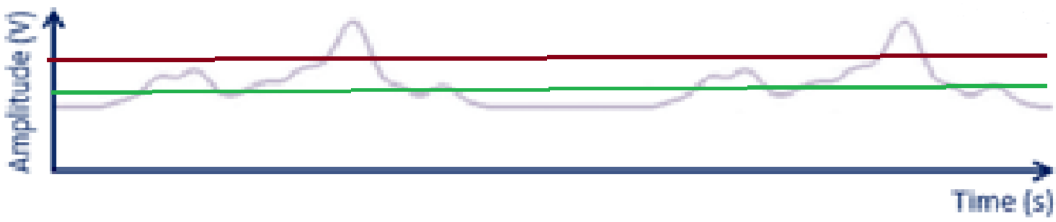
Reading MyoWare Signal

The signal from the MyoWare sensor can be read by using the analogRead function. Once read, I printed the value of the muscle stimulation in the terminal so I could better understand what this signal looks like. I found that the maximum value of the muscle signal is 4095 so I set this value as a variable called "maxVal" which I could use later for the binary classification.

I found that the primary issue with classifying muscle stimulation is that it is not as simple as True (closed) and False (Open) being 50/50 i.e I could not set the muscle stimulation to be true if the signal was greater than half of the maximum value as overtime my strength weakened and despite having a closed fist, the value still read that my hand was open. To resolve this issue, I decided to set an upper and lower limit of the maximum value.

What this does is read the signal from the sensor, if the signal is greater than 55% of the maximum value (2,252.25) , then the signal indicates that the users hand is closed and sets the muscleState variable to True. If the signal is less than 25% of the maximum value (1,023.75), then the signal indicates the user's hand is open and sets the muscleState variable to False (graph seen below under code snippet). This allows for weakening of grip strength over time. The muscleState will remain as “closed” until the signal drops below 25% of the maximum value and it will remain as “open” until the signal is greater than 55% of the maximum value.

```
//Read from sensor
int sensorValue = analogRead(pinENV);
Serial.println(sensorValue);

//interperate sensor data


//maxVal = 4095
float upperLimit = maxVal * 0.55;
float lowerLimit = maxVal * 0.25;

if (sensorValue > upperLimit) {
  muscleState = true;
} else if (sensorValue < lowerLimit) {
  muscleState = false;
}
```



Preparing the Data to be transmitted

```
//message structure
typedef struct struct_message {
  bool t;
} struct_message;

struct_message myData;
```

Once the Data has been classified as ‘True’ or ‘False’, It is then added to a structured message. “**C++ Structures** are user defined data types which are used to store group of items of different data types. A structure creates a data type that can be used to group items of possibly different types into a single type.” – [7]

There are two reasons why I decided to use a structured message to store my data before transmitting it. The first reason is because I can easily update this structure in the future in the event that I decide to include more variables such as the signalValue (Integer value representing the muscle stimulation) or if I introduce a second muscle sensor, I could include unique identifiers for each of the muscle sensors being used. The second and more important reason is so that I can ensure there will not be a data type mismatch, and this can be checked on the receiver of the message (Prosthetic) by creating the same structured message and populating the value with the data from transmission. This ensure that a data type mismatch does not occur and is something that I learned during the development of this project as I incorrectly set up the structure on the receiver and observed as the data would not print on the screen once received by the prosthetic.

Before sending the Data via ESP-NOW, I first add the data to the structured message. To do this, I can assign a Boolean value by using myData.t = {Boolean Value} as seen below.

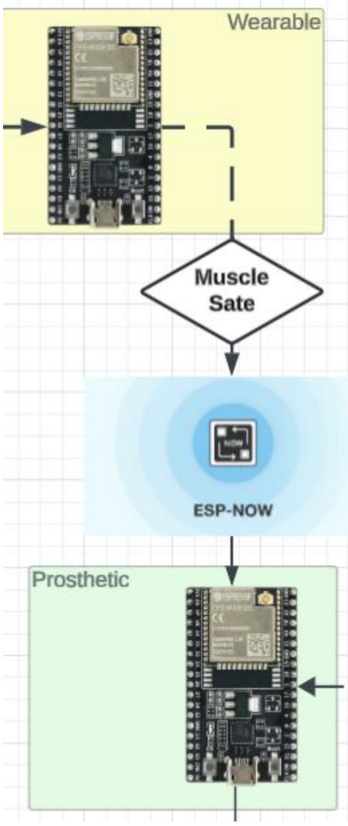
```
if (sensorValue > upperLimit) {
  muscleState = true;
} else if (sensorValue < lowerLimit) {
  muscleState = false;
}

myData.t = muscleState;
```

When the data is received on the prosthetic, it is then copied into a structured message that is the same as the structured message on the wearable.

```
memcpy(&receivedData, data, sizeof(receivedData));
```

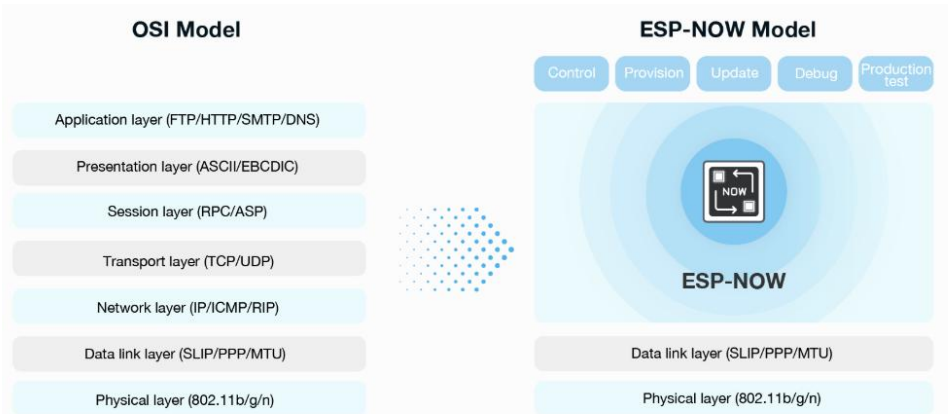
Later, I can then call receivedData.t as a variable in a method called moveHand() which will pass the Boolean value into this method to determine if the hand should open/close or remain in the same position. (More on ESP-NOW and data transmission below).

Sending Data with ESP – NOW



ESP-NOW is a wireless communication protocol developed by Espressif who are the same developers of the ESP32 development board used for this project. The protocol operates as a peer-to-peer protocol, allowing direct communication between ESP32 devices without an access point (such as a Wi-Fi router) being required. It can be used in smart-home applications, remote controlled devices and sensors.

Each esp32 development board has its own MAC address which can be used to identify the 'Master' and 'Slave' boards i.e. The slave board has code to store the mac address of the device it received data from so that it knows it can continue to receive data from that device.

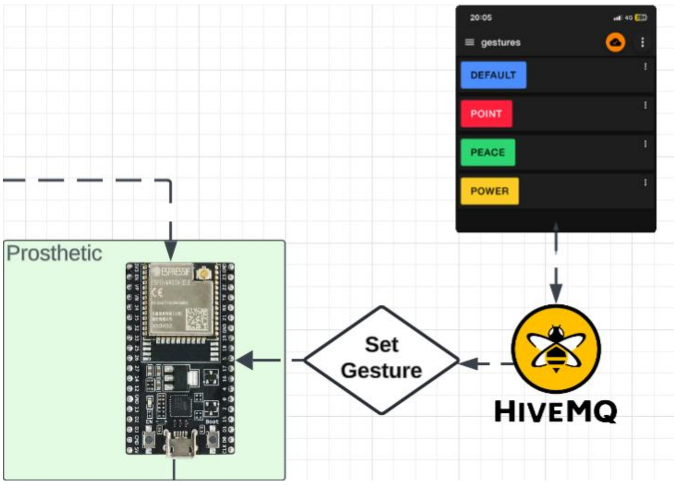


ESP-NOW has One-way communication and Two-way communication. One-way communication is ideal for a scenario where data will be sent at a continuous and rapid rate where packet loss is not a huge issue. This scenario is ideal in my case as muscle state is broadcasted every 1.25 seconds. Two-way communication allows for the 'Slave' (Prosthetic) to communicate back to the 'Master' (Wearable) that is has successfully received the packet.

Initially, I used Two-way communication which, at first, worked perfectly, however, when I tested the communication on campus, I found that a large portion of the packets were lost and as a result, the arm did not receive the muscle state data. This could have been caused from the large network interference, between access points and other wireless devices, packets can be dropped and there can be latency issues reducing the speed at which packets can be received.

Due to this issue, I changed to One-way communication as this would allow me to broadcast the data as opposed to sending it directly to a specified MAC address (the mac address of the prosthetic). Once I made this change and tested it on campus under the same conditions as Two-way communication, I found that the packet transmission was far more successful. The biggest change is that both ESP32 boards (wearable and prosthetic) connect to the same Wi-Fi network and use the same Wi-Fi channel as the router that they are connected to. This 'dynamically set' Wi-Fi channel would ensure that both devices communicate on the same channel.

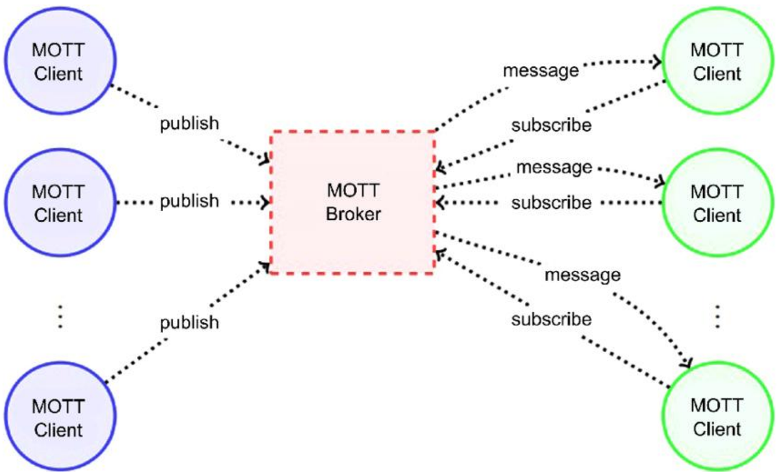
Setting Gestures using MQTT



Message Queue Telemetry Transport (MQTT) is a communication protocol for Internet of Things devices. It is designed for devices with remote locations and was originally developed in 1999 by Andy Stanford-Clark and Arlen Nipper to monitor oil pipelines in the SCADA industrial control system.

MQTT is a lightweight protocol that allows communication between multiple devices. It is a TCP-based protocol relying on a publish-subscribe model. The protocol is ideal for devices that are resource-constrained with low power requirements.

The publish-subscribe model allows for multiple devices to publish data and subscribe to receive data. Applications for this protocol would include home automation as it can be used to remotely control lights, air conditioning and automatic vacuums as well as displaying what systems are enabled on your smartphone by making your phone a subscriber to the broker to receive data.



Each device communicates with one another through what is called the MQTT Broker. In my case, I am using a broker called HiveMQ who offer a free tier with a limit of how many messages can be sent each month.

As I have used this protocol in a previous project (voice activated home automation system), I have an understanding of how the data can be sent between two or more devices and I am confident that I can implement this protocol into this project.

I am using this protocol because it will allow remote communication between the prosthetic prototype and my smartphone. I will be able to select the gesture using my phone. The gesture will then be published to the HiveMQ broker which the prosthetic will receive as it is a subscriber to the broker. Once the new gesture is received from the broker by the prototype, it can be stored and used to look up items in a map which will instruct each servo as to the direction and speed that they should move. (more on this later)

Format of JSON data

The data is sent as a JSON (JavaScript Object Notation) payload. This payload is in the form of:

{“value”:<payload>,”is public”: True}

Where <payload> is replace with the gesture value. I have several gestures to choose from which include;

- “default”
- “point”
- “peace”
- “rock”

When one of the gesture buttons are pressed, it is then sent to a topic on the HiveMQ broker.

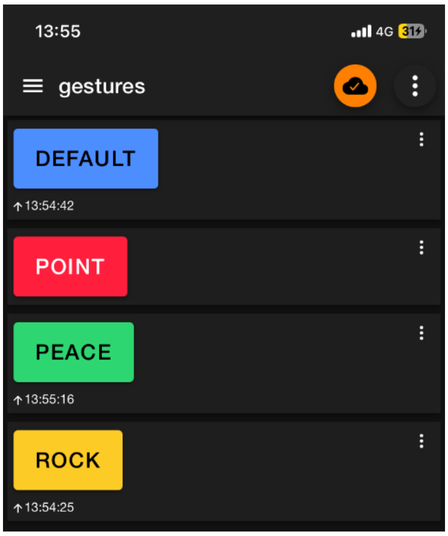
What are Topics?

Each Broker can have its own “topic”. “In MQTT, Topic refers to a UTF-8 string that filters messages for a connected client. A topic consists of one or more levels separated by a forward slash (topic level separator).” – [9]

- [9]

In my case, I am sending the gesture value to a topic called hand/gesture. The prosthetic will subscribe to the same topic and wait for new messages to be published.

Once a button for a gesture is selected, the timestamp is presented below the button indicating when the payload was sent to the MQTT broker.



The corresponding timestamps along with the gesture can be seen in the serial monitor of the Arduino IDE once the payload has been received by the prosthetic from the MQTT broker and parsed to extract the gesture from the payload.

Gesture Map

```
struct Gestures{
  int values[4][2];
};


// { key , { { (pinky Close , pinky Open) , (ring Close, ring Open) , (middle Close , middle Open) , (pointer Close , pointer Open} } }
std::map<std::string, Gestures> GestureMap = {
    {"default", { {{50, 160}, {50, 160}, {50, 160}, {50, 160}}}},
    {"point", { {{50, 160}, {50, 160}, {50, 160}, {90, 90}}}},
    {"peace", { {{50, 160}, {50, 160}, {90, 90}, {90, 90}}}},
    {"rock", { {{90, 90}, {50, 160}, {50, 160}, {90, 90}}}}
};
```

The gesture map is used to store the speeds that the servos must rotate at based on the direction and the gesture selected. The values are stored in a 2D array where there are 4 columns (each column representing 1 of the 4 servos used for controlling the fingers) and the second dimension is used to store the closing speed and the open speed for the finger.

When the hand needs to close, it will use the first value of each index of the array i.e. “default” is selected and hand is closing, {50 , 50 , 50 , 50} and when the hand needs to open {160, 160 , 160 , 160}.

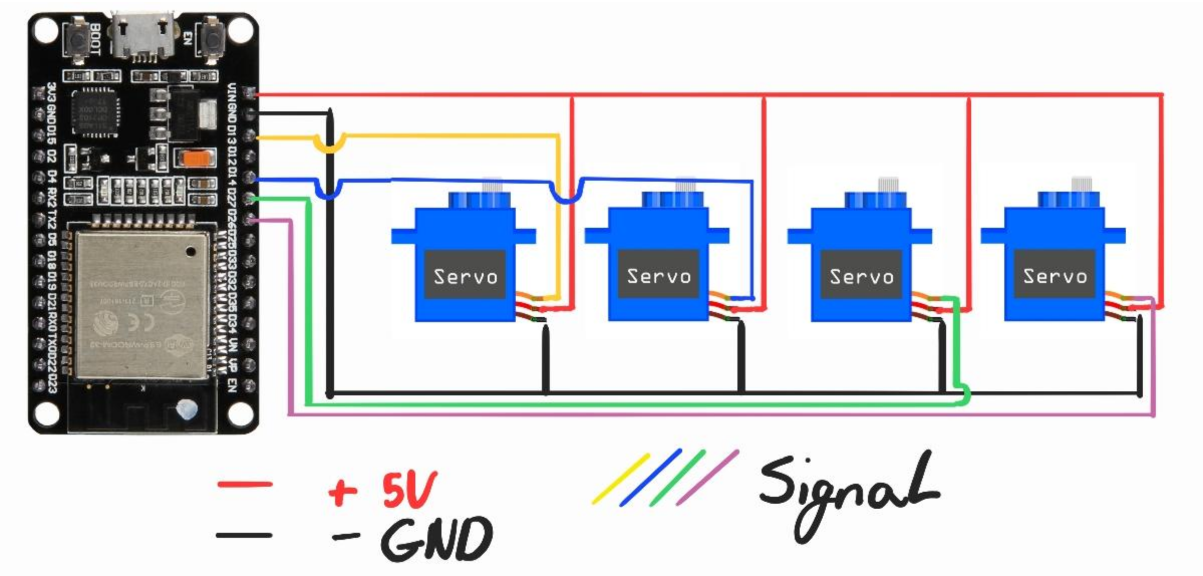
Servo Circuit

To power the four servos that will retract each finger, I am using the 5V pin and ground (GND) pin on the ESP32. I will be using a parallel circuit to ensure that the esp32 can provide sufficient power across all of the servos on the circuit.

Series VS Parallel Circuit

The main issue with servos running in series is that the voltage is not properly distributed for servos as they require a stable voltage to operate. To prove this, I set up one servo and slowly began adding servos in series to the circuit and I noticed that with every servo being added, they moved more slowly until eventually they stopped moving as there was not enough voltage to power all the servos.

I then set up a parallel circuit for the servos and immediately all the servos started working as intended.



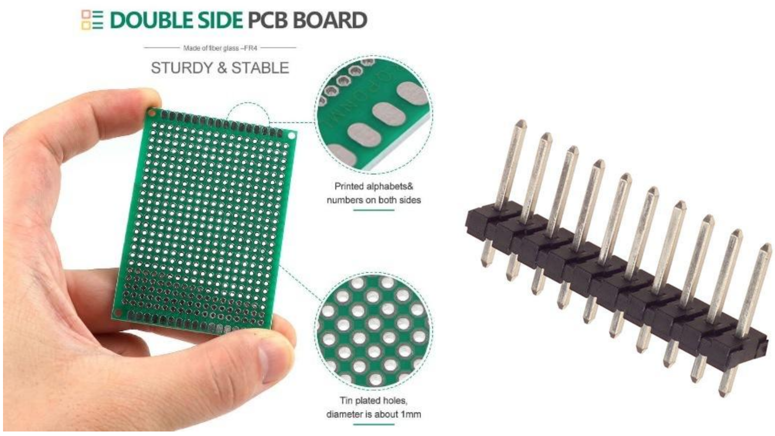
Above, is the circuit layout for the prosthetic prototype. This is a parallel configuration where each Servo is powered from the 5-volt pin (red wire) and connected to the ground pin (black wire). Each servo has its own signal wire which is used to control the direction on speed on the servo and is connected to its own pin on the esp32. I intended on using pin D12 as one of the pins to control a servo but I encountered an issue where the system would not turn on, I later found out through an ESP32 forum that GPIO12 is a bootstrap pin and the ESP32 will not boot if it has a high input (signal) on this pin. [11]

Breakout Board

To power all the servos in parallel, I will need to create a breakout board. “Breakout boards are essential tools that take multi-pin components and "break out" each pin to a set of connectors, making them easier to use. This makes it simpler to connect sensors, chips, and other components to breadboards or development boards, facilitating prototyping and experimentation.” – [12]

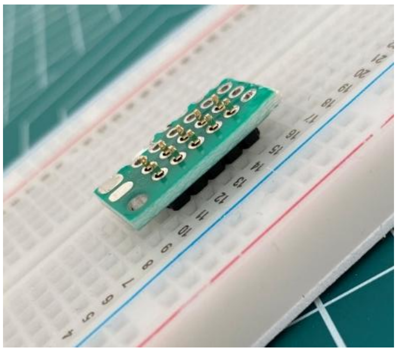
In my case, I am using a breakout board specifically for the 5 volts and ground pins on the ESP32 so that I can power all the servos.

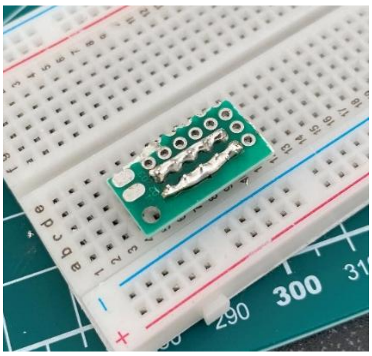
I started with a double-sided PCB board which would allow me to put male header pins on it so that I could easily add and remove the servos from the board, making this system modular.



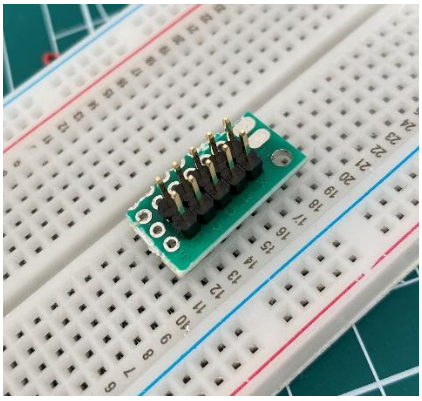
I cut a piece of this board and got 2 rows of 5 male pin headers. One row of pins will be used for the ground connection on the ESP32 and the other row of pins will be used for the 5 volt connection on the ESP32. It is extremely important that the 5-volt connection of a servo is connected to the 5-volt pin on the breakout board because if the 5 volt connector of the servo is connected to the ground of the breakout board, reverse polarity occurs which can burn out the ESP32.

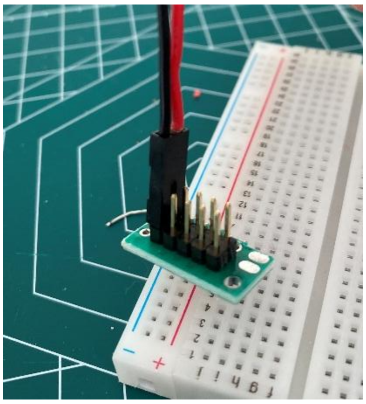
Unfortunately, this was something I experienced by accident when working on the breakout board, I put the servo back on the breakout board the wrong way and I damaged the ESP32 which resulted in me having to replace it with a new ESP32.





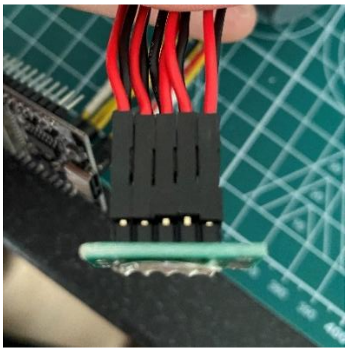
Once I had the board cut and the 2 rows of male header pins ready, I used a breadboard to hold the male pins in place to ensure that they stay aligned while I soldered them to the PCB board. I tacked the ends on the rows to the pcb board so that they would be fixed in place and using some tinned copper wire, I was able to connect the entire row together (seen in the image above on the right). I then proceeded to add more solder to ensure a strong connection for all of the pins.

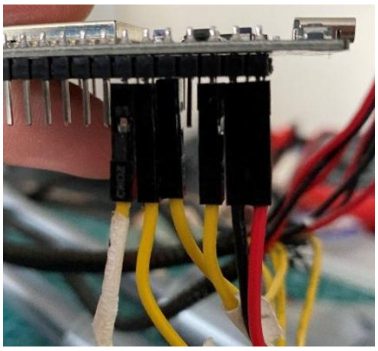




Above shows the male header pins connected to the pcb board. To allow the servos to connect to this board, I needed to use female dupont connectors. I bought a female dupont connector set from amazon and using a crimping tool, I was able to put these connectors on the ends of each wire of each servo (3 wires x 4 servos = 12 wires).

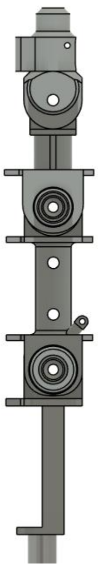
As mentioned previously, this connection allows the prototype to modular which is benificial because in the event that a servo becomes damged or I am upgrading components, they can be quickly & cleanly swapped out. Laslty I used some electrical tape to tape the bottom of the breakout board to prevent it from scratching and damaging the ESP32.



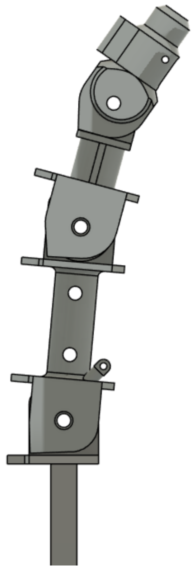


Re-designing Fingers and Elbow

One particular issue I was having when testing the finger movement was that the tips of the fingers would bend backwards as the finger closed over. This was because there was no stopping point preventing the finger from bending backwards so I decided to go back to fusion 360 and edit the fingers to include extra material to stop this from happening. The design difference can be seen in the images below.



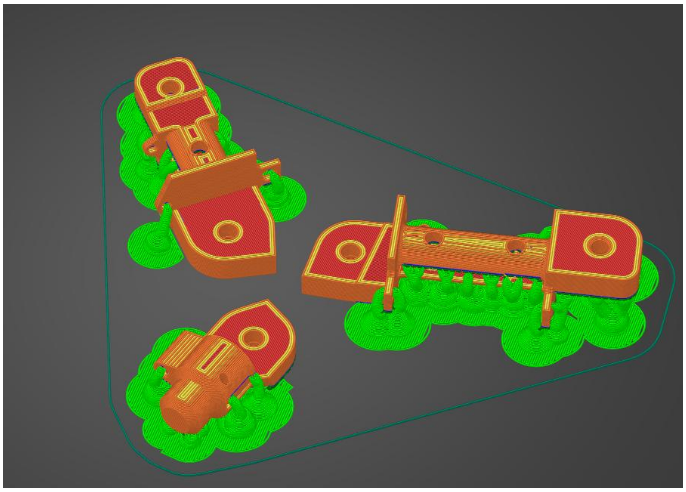
Old Finger (No support)



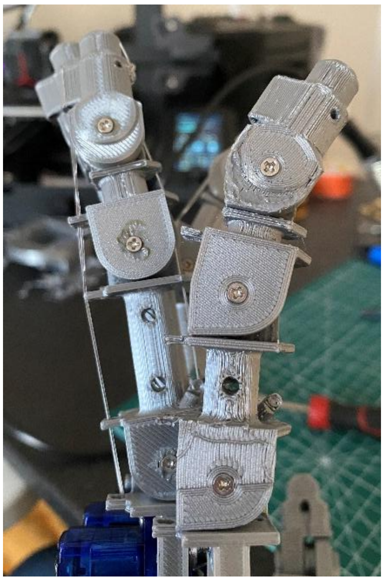
New Finger (Support Prevents finger bending backward)

I included a counter sink for all of the areas where there are screws. This has two improvements on the fingers. The first improvement is that the counter sink will allow for the screws to reach further into the female-to-female standoffs acting as the bearing (or joint) between each joint of the finer. I will still use thread locker to prevent the screws from unscrewing themselves as the prototype is being used as that was one of the main issues I noticed in the original fingers and has persisted in the new design. The second improvement to using the counter sink is that the head of the screws are flush with the surface of the fingers which is both aesthetically pleasing and reduces the chances of the screws catching on anything such as threads from clothing or even the other fingers.

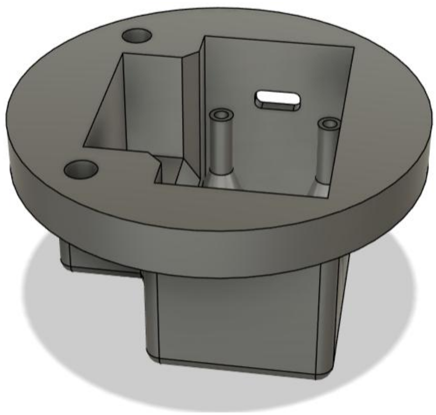
When I printed the original fingers, I used the Creality slicer so this time I decided to print them using the Prusa slicer. The biggest difference is that the Prusa slicer allows for 'organic' supports which uses I minimal amount of support material and makes it very easy to remove the supports from the final object.



The green sections of the print shows the organic supports. The printing time was also much faster as I was able to modify the infill of the fingers to reduce the amount of material being used. Each finger with the new design took approximately 2 hours to print. With each finger being printed, I disassembled the original fingers and began installing the new fingers. I immediately noticed a huge difference in the quality of the print when I had the two version of the fingers side by side.



The new finger design (leaning forward) can be seen next to the old finger design (leaning back) in the image above. Making these adjustments significantly improved the prototype as the original fingers seemed much more fragile and incapable of functioning as intended. Whereas the new design feels more premium and capable of carrying heavier items.



The next update I needed to make was to the elbow cover. I needed to update this part because I originally intended on desoldering the GPIO pins from the ESP32 which would have shortened the ESP32 by roughly 9 millimetres, however, because I decided to make the arm modular (being able to easily replace servos), this requires more space in the elbow. Very little adjustments needed to be made as I parameterised all the values associated with the elbow (length, diameter, etc.) which meant I needed to change the value associated with the parameter called “espHeight” so that it would include this extra height to allow for the GPIO pins and female Dupont connectors. I added additional space near the front of the elbow so that there was room for the breakout board and excess wiring.

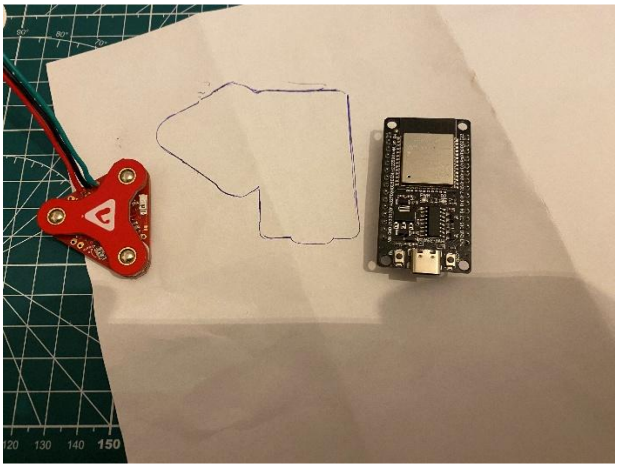
Once I was happy with the updates to the elbow part, I exported it and put it into the Creality slicer to format the part for the 3D printer to print. I used the Creality slicer for this because this is a very simple object and the process of slicing it in the software would be much quicker with Creality as it is a more beginner friendly slicer and does most of the heavy lifting for me.

Wearable Device Enclosure

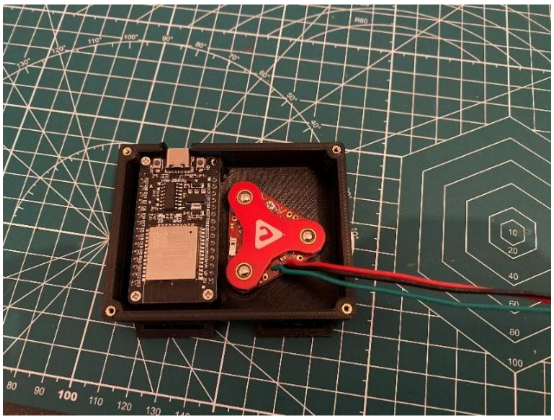
I needed to make an enclosure to hold the components of the wearable device that the user will wear. I started by placing the components on a piece of paper in the position that I would like them to sit in the enclosure and I then used a pen to draw around them.

Next, I drew a square as tight as I could around the components so that I could use it to get a general size for the enclosure while still making considerations for maintenance and wire management. I created parameterised variables in fusion 360 for all of the dimensions of this square because it would be simple and quick to make changes to the dimensions and test them with a quick print.

Sketching



Assembly

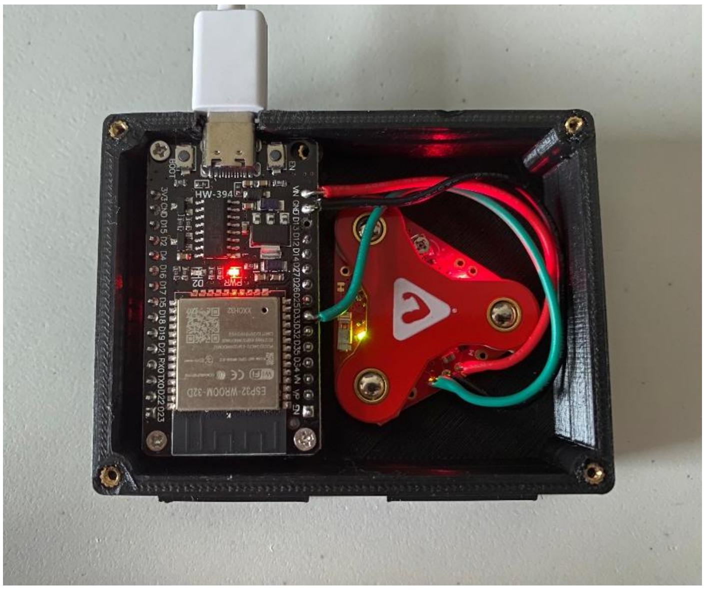


Inside the Enclosure

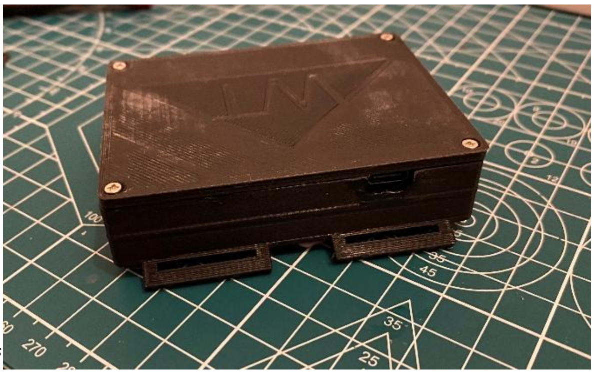
Back (skin contact) side of enclosure

Above is Images of the front and back of the enclosure with the components loosely fitted to ensure that there is enough clearance. The most difficult part of making the enclosure was taking the measurements of the MyoWare sensor (red device) and ensuring that I could clip the electrode pads to the device from the outside of the enclosure. I did this by creating three holes that match the dimensions of the electrode pad connectors and ensuring that the thickness of the enclosure was thick enough so that the enclosure would not break, but thin enough so that it would not obstruct the electrode pads from fitting into the electrode pad connectors. This fitting is tight and I'm happy with how it turned out because it feels like the sensor is secure within the enclosure and cannot loosen and fall around inside the enclosure.

Next, I soldered the flex sensor to the ESP32 and powered on the system to ensure that the connections were solid and the system could operate. I found that I could not use the flex sensor on pin 13 because it is on the ADC2 (Analog to digitial conversion) circuit and ADC is disabled when using ESP-NOW which is why it can be seen in the image below that I origionally soldered to pin 13 but had to move to pin 33. This was a mistake that I made because I did not consult the ESP32 documentation and tested the flex sensor and ESP-NOW seperately and assumed that it would all work as intended.



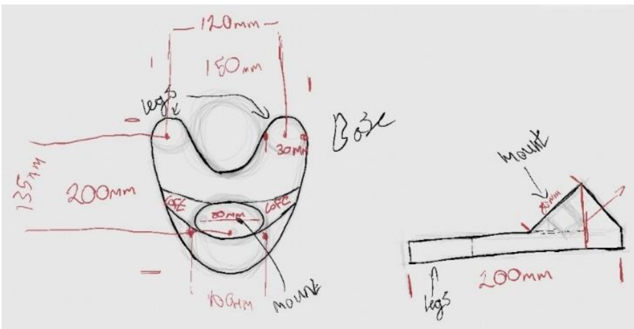
To ensure that the lid would not come off, I used threaded inserts and screws. The threaded inserts are place in each corner of the enclosure and screws will be used to keep the lid on. I did something similar for the ESP32 and it sits on 4 standoffs with screws holding the ESP32 in place so that it would not move around when the user tries to plug it in.



I am happy with how the enclosure turned out, however, the GPIO pins of the ESP32 caused for this enclosure to be much larger than it needs to be. If I could either get an ESP32 which does not have GPIO pins connected or, I desoldering all of the pins, I could shorten the height of this enclosure by roughly 6 millimetres and this would result in the enclosure being more comfortable to wear as it does protrude away from the arm quite far and during testing, I cought it on the edge of my desk a number of times.

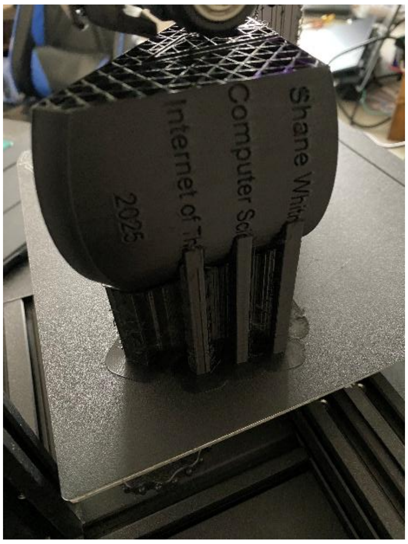
Expo Display Stand

One aspect of this project that I have been particularly excited about is the Computing Expo. I wanted to create a display stand to showcase my project that is minimal so that it does not take away from the prototype itself. I began sketching a design on my tablet and putting some general dimensions to it that I could later change if that was required. I came up with this minimal design consisting of two legs which would stabilize the stand with the prototype screwing into a surface which is position at a 30 degree angle so that it could lean back slightly. Once I was happy with the design, I opened fusion 360 and began created parameterised variables and populated them with the values seen below in the sketched image.

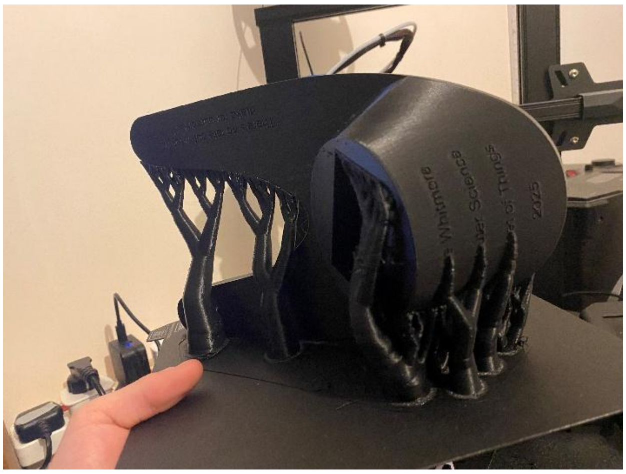


Lastly, I included some text on different points of the stand to display more information about the project etc. I then created a Render of the stand to include in this report (seen below).

The display stand is quite large and barely fit on my 3D printer at home. I had to print the stand on its side rather than sitting flat as seen above. This object takes 286 grams of filament (1 quarter of a role of filament) and a considerably long time to print which is why I decided to do a slightly lower quality print as the lower quality can only be visible on rounded edges and because I am printing the part on its side, the layers should be less visible.



I had an issue with this version of the stand because the esp32 that controls the prototype would not fit into the cavity inside the stand. I adjusted the values such as changing the angle that the prototype would sit at and the height that it was standing up at. Once I made these adjustments I printed the new design this time with the Prusa slicer as this would be a great opportunity to compare the differences between the two slicers.



They new design works well, and I can now power the ESP32 that controls the prototype from inside the cavity of the stand.

I am very happy with how the part turned out. I was concerned about the print quality and density at first because I needed the part to be heavy enough to hold the arm in place, but it worked out well.

Testing

I wanted to ensure that the prototype would function correctly. During my internship, I learned about test scripting. “Test Scripts are line-by-line description that contains information about system functions that must be performed to verify an application or system under test.” [13]

Test Scripts should contain the following information for an individual test;

- Test
- Expected Result
- Actual Result
- Fix (If required)
- Comments
- Date of Test
- Tester Name

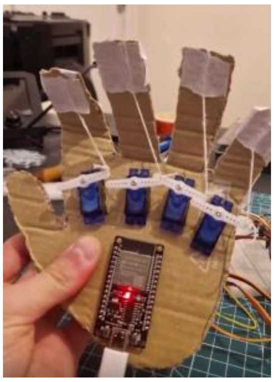
This information would allow me to keep track of all the components that have been tested along with their dates. I documented these tests in a table of an excel spreadsheet.

The testing I did mainly involves loss of connection between components such as loss of power to wearable, loss of power to prosthetic, MQTT broker disconnected, no Wi-Fi available etc. These are the more important tests to ensure that the prosthetic operates as intended however, testing should be performed to ensure that servos operate as intended etc. An example of a test like this would be to open & close a finger continuously until the servo stops working to figure out how durable the servo is. Another test could be performed to understand the weight the prototype can carry before breaking to better understand the limitations of the plastic, infill, structural design etc.

Result of Final Year Project

The purpose of this project was to investigate the feasibility of creating a non-invasive, inexpensive prosthetic arm prototype while using several of the United Nations: Sustainable Development Goals as the core foundation to the project.

I started the project with a piece of cardboard cutout from the outline of my hand to see if I could get general movement with the servos connected to an esp32. I put careful consideration into the material and components I wanted to use such as “can it be easily sourced” , “is it inexpensive to source”, “is it environmentally friendly” trying to keep in line with some of the United Nations: Sustainable Development Goals.



It was extremely exciting not only to see the project take shape over the college term but also to have had so many people take interest in what I was working on, and I enjoyed sharing my progress. The most challenging part about the project was creating new gestures as I was unsure as to how I could do it at the start, however, I got a clearer understanding of how I could create and implement new gestures as the project progressed.

I really enjoyed being able to use all the skills that I have gained during my college term such as Designing and 3D printing parts, creating circuit diagrams and circuits, soldering and programming.

Is it possible to create a non-invasive inexpensive prosthetic arm? I learned that there is a lot required to create a prototype prosthetic that may be used as an alternative to what is currently on the market in the future. I believe that the current design is not a viable alternative, however, I do believe that using the United Nations: Sustainable Development Goals as the core foundations for making an inexpensive prosthetic alternative is a must as it covers several important economical & environmental issues that could make it more affordable for families to purchase a prosthetic arm.

Future Design Concept (Generative Design)

*Images generated using ChatGPT

What is Generative Design?

“Generative design is an advanced, algorithm-driven process, sometimes enabled by AI, used to explore a wide array of design possibilities that meet predefined criteria set by engineers or designers.” [14]

There are several reasons as to why I would investigate the use of generative design in prosthetics. The main reason being that generative design reduces the amount of material used without impacting the structural integrity of the object. Cutting down on the amount of material used would decrease the weight of the arm, making it more comfortable to use for the wearer. Cutting down on the amount of material used may also decrease the manufacturing cost, however, I cannot say that for certain as there are other variables such as how long would it take to print a generative design prosthetic. Using a hollow generative design prosthetic would leave lots of room inside for components such as batteries, stepper motors, servos etc which is something I did not get to do with the current design.

Filamentum is a company that recycles ocean waste such as bottles and fishing nets. Their filament collection ranges from PLA (the same material used in this project) to CPE CF112 Carbon which is a carbon fibre reinforced PETG filament which would be far more durable than PLA. Using these filaments would be more environmentally sustainable as they are made from recycled materials. [15]

Brown out

There are some potential issues with the current configuration of the circuit that is powering the Servos. According to data sheets of the SG90 9G servos that I am using for this project, each servo requires a current of 550 milliamps (mA), however, if the servo is blocked or is prevented from moving, the current could potentially spike to 800 milliamps (mA) which is known as the stall current. In the event that all of the servos are blocked and spike to their stall current, it may result in a brown out which would cause the system to reset itself as the required current exceeds the current that the ESP32 can provide.

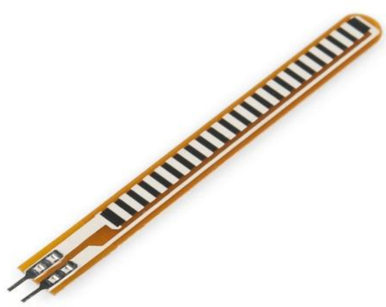
One solution to this is to give the ESP32 and the Servos their own power supply. This power supply would take the form of a battery which could be inserted into the prototype. Having the servos and the ESP32 on separate circuits would mean that each component has the required current to operate.

Limit switch

In this current version of the prototype, I am using a direction speed and time delay to open and close the fingers. The problem with this design is that there is a small possibility that a servo could overspin causing damage to either the finger, the knuckle or even damage the servo itself which would require the servo to be replaces.

To resolve this issue, a limit switch can be introduced. Limit switches are a fundamental tool in the IoT industry. 3D printers use limiting switch to calibrate the positioning of the nozzle. In my case, I can use a limiting switch to detect when a finger is fully closed over. This is a much more effective solution for closing and opening the fingers as it would allow for each individual finger to spin for as long as required until the finger closes or opens.

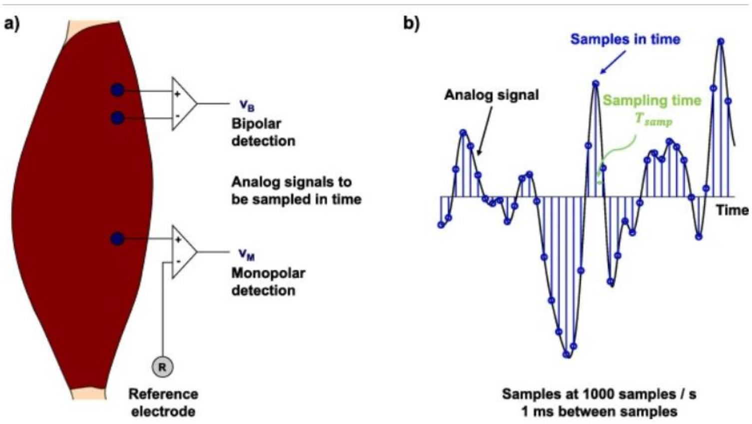
This limit switch could be in the form of a flex sensor. (Image seen below)



[16]

The flex sensor operates as a resistor which changes the potential difference (voltage) of the output as it bends. This sensor could be fitted to either one or all of the fingers on the prosthetic to detect the position of the finger and when the flex sensor is at its highest resistance (fully bent), it would indicate that the finger is closed and thus, stopping the servo from spinning any further. Similarly, the flex sensor could be used to check if the finger is fully open by checking if the resistance of the flex sensor is at its least.

Fast Fourier Transform (FFT)



[17]

“**Fourier transform** is a mathematical model that decomposes a function or signal into its constituent frequencies.” [18]

Fourier transform would allow me to map the frequency of a given gesture (pointer finger closed, making a fist, half closed hand etc) so that the prosthetic prototype could mimic this gesture based on the given frequency.

How this differs from the current implementation is that I am currently sending a true or false signal representing open or closed which does not allow for multiple positions that each finger can stop at. The other difference is that I am using my phone and MQTT to select the gesture that I want to use. Using Fourier transform could potentially allow me to set the gesture based on the position of my hand rather than choosing from a select few gesture on my phone.

Using Fourier transform, I could observe the frequency of the muscle as I turn my wrist which would allow me to introduce a stepper motor into the wrist of the prosthetic to rotate the hand.

Eprom

"EPROM stands for **Erasable Programmable Read-Only Memory** is a kind of nonvolatile memory used in computers and other electronic devices for storing data that must be retained even when the power is turned off. Unlike standard ROM (Read Only Memory) which is permanently programmed at the time of manufacturing. EPROM can be programmed again and erased a number of times making it a versatile choice for storing firmware and other important software." [19]

Using Eprom would be beneficial to the project for several reasons. Eprom would allow me to store the state of the prototype (open/close & selected gestures). The reason I would need do this is that in the event of power loss, the prosthetics state has been saved so as soon as the power is restored, the state can be checked, and the program can resume as normal. Currently, if there is power loss while the prosthetic is closed and the power is restored, it assumes that the prosthetic is open and that the gesture is "default" which may not be the case.

Another application for Eprom would be to store all the available gestures that the prosthetic can do. This can be achieved by fetching the gesture map on start-up using MQTT and storing the information so that if the prosthetic cannot communicate with the MQTT broker, it can still see the last available gestures that it can do and continue to operate as intended until it can fetch & validated the gestures from the MQTT broker again.

Smaller Wearable Device


*Image generated using ChatGPT

Another feature that I would like to change is the current size of the wearable device. It is too large, and I found that when testing the prototype, I would constantly hit the wearable off the edge of my desk by accident. This is a problem because it has a noticeable weight which feels somewhat uncomfortable to wear. I could cut down on the size of the wearable by using an esp32 that does not have GPIO pins soldered to it. Using an esp32 with this specification would reduce the height of the wearable but would not reduce the width and length.

To reduce the width and length of the wearable, I could investigate different layouts of the components and pick the one with the smallest footprint or investigate other components that are much smaller but still allow for this system to operate as intended.

Semester 2 Discussion

The purpose of this project is to investigate the feasibility of creating a non-invasive, inexpensive prosthetic arm prototype all while using the United Nations: Sustainable Development Goals as the core foundation to the direction of the project. This project was heavily inspired by an artifact I came across at the City of Science and Arts in Valencia in 2023 which interested me in the many ways that people can interface with computers beyond the traditional methods such as touch and sight and brought me to the conclusion of investigating the feasibility of creating an inexpensive alternative to traditional prosthetics.

Is it possible to create a non-invasive inexpensive prosthetic arm? I learned that there is a lot required to create a prototype prosthetic that may be used in the future as an alternative to what is currently on the market. This project highlighted some of the many challenges involved in creating a simple prototype and I learned a lot from my findings that I can improve on in future iterations of this project. I do not believe that the current design would be a viable alternative as a prototype prosthetic, however, there are several aspects of this project that should be considered for the next iteration. These aspects mainly regard the United Nations: Sustainable Development Goals as they could tackle economical and environmental issues which, in turn, may lead to an inexpensive prosthetic alternative that is a more viable solution to what is currently available.

Implementing the MyoWare 2.0 sEMG sensor was a fundamental aspect of the project and can be reused in future versions of the project. Reducing the size of the wearable device would be a huge benefit to the project as there were times where I would knock the wearable off the edge of my desk when reaching for my laptop.

There are several limitations to this current iteration of this project. The biggest limitation is the PLA filament used to print the prototype as it degrades when subject to cold, damp temperatures and is not the strongest of materials when put under heavy load. I would also like to begin investigating power supply options such as different types of batteries to provide a sufficient amount of power to the system for daily use while keeping the weight of the prototype to a minimum as not to causes discomfort to the wearer.

If I had more time and resources, I would have been extremely interested in printing many versions of the same prototype using different materials such as nylon filaments, PETG filaments, carbon fibre reinforced filaments etc, at different print densities so that I could stress test all of them and understand the weak points of the design so I could further improve on it.

I would also like to investigate the use of generative design in prosthetics as it would allow me to cut down on the material being used while maintaining the same structural integrity required for the prototype to operate as intended under heavy stress.

Semester 2 Conclusion

In conclusion, I am happy with what I achieved this semester. I achieved my goal of developing a prototype arm which uses non-invasive biosensors (MyoWare 2.0 sEMG sensor) to monitor muscle stimulation in the wearer and open and close the prototype arm. I also implemented MQTT to select from a range of gestures on a smartphone which would control which gesture the prototype will use.

I would like to continue developing this project as I outlined several changes I would make to the current iteration. I am interested in investigating generative design and if it can be used for a prosthetics.

I used many of the skills that I have gained over the past four years at college such as creating objects in fusion 360 and 3D printing them, to designing circuits to control many servos.

I gained new knowledge as I developed this project, and I enjoyed the challenges that occurred such as unicasting the muscle data and observing packet loss in areas with high network interference as this allowed me to better understand limitations with certain protocols and how broadcasting the data can work more effectively.

I also gained valuable experience in project management. I enjoyed organizing tasks for the week to ensure that I would complete this project on time. The methodology I found most helpful was KanBan as it allowed me to see what tasks needed to be done, what tasks I was doing that week and what tasks I had completed. This helped to keep me motivated knowing that I was organised and capable of focusing on the select few tasks that needed to be completed that week while still taking into consideration the time required to complete the tasks that I had planned for later weeks.

References

[1] https://sdgs.un.org/goals

[2] https://myoware.com/wp-content/uploads/2022/03/MyoWare_v2_AdvancedGuide-Updated.pdf

[3] https://www.espressif.com/en/solutions/low-power-solutions/esp-now

[4] https://www.espressif.com/en/products/socs/esp3

[5] https://www.cd-bioparticles.net/polylactides

[6] https://grabcad.com/library/t-800-terminator-endoarm-1

[7] https://www.geeksforgeeks.org/structures-in-cpp/

[8] https://www.geeksforgeeks.org/introduction-of-message-queue-telemetry-transport-protocol-mqtt/

[9] https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices/

[10] https://forum.arduino.cc/t/how-much-power-supply-do-i-need-for-controlling-5-sg90-9g-microservo/627666

[11] https://esp32.com/viewtopic.php?t=9449

[12] https://jlcpcb.com/blog/breakout-board-importance-in-electronics

[13] https://www.geeksforgeeks.org/software-testing-test-script/

[14] https://www.autodesk.com/solutions/generative-design

[15] https://fillamentum.com/collections/abs-extrafill-filament/

[16] https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Fcdn.littlebird.com.au%2Fimages%2Ffiles%2F000%2F054%2F042%2Flarge%2FSF-SEN-10264.jpg%3F1535775179&f=1&nofb=1&ipt=3383c0107934a746a0352cfe63c99ca12e8249efaa6ce71c78942f98bb3d6a3a

[17] https://www.sciencedirect.com/science/article/pii/S1050641124000816

[18] https://www.geeksforgeeks.org/fourier-transform/

[19] https://www.geeksforgeeks.org/eprom-full-form/