Design Documentation

Spring 2016

Team 6: Need-A-Hand

Group Members: MarcoAntonio Ledesma

Jesus Espinoza

Omar Guzman

Jorge Gonzales

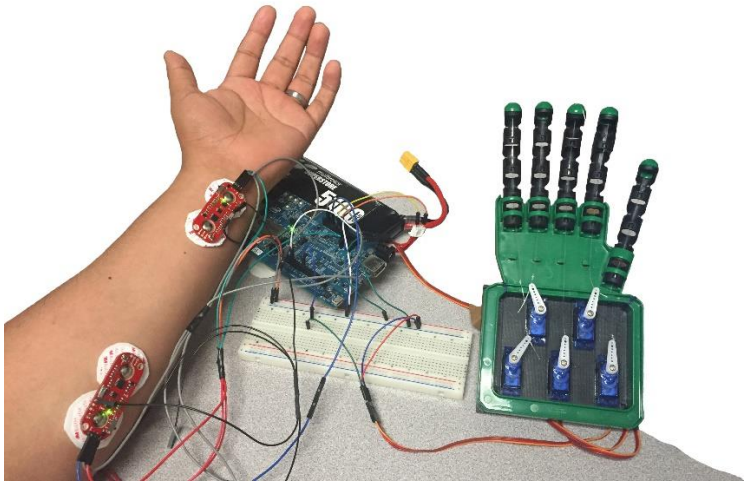
Table of Contents

ABSTRACT

Need-A-Hand: Muscle Sensor Controlled Prosthetic Hand for those with Hand Disabilities

Need-A-Hand is a project to create a prosthetic hand that is controlled using muscle sensors in the forearm. The purpose of this project is to build a prosthetic robotic hand to aid those people who have a hand disability. The prosthetic hand will help those people regain some hand ability to aid them in daily tasks like getting dressed, cooking a simple meal, grasping a glass to drink water. In the end our goal is to make life easier to those with an unfortunate hand disability.

If you look into hand anatomy, we can see that human fingers move with the muscles in the palm and forearm. Instead of using those muscles in the forearm to move real fingers we will use them to move our prosthetic hand fingers. This is where the basic functionality of our prosthetic hand comes from. We will be using muscle sensors to read muscle activity in our forearm. The muscle sensors read muscle contractions to get a signal. We then use those signals to activate servos to individually move the fingers of the prosthetic hand.



Introduction

The motivation behind our project was that there are a lot of people with hand disabilities who have lost their ability to complete daily tasks especially those of physical nature after having one of their hands amputated such as war veterans or people suffering from malignant diseases. Currently, the options on the market for people who have had one of their hands amputated are to buy a plastic prosthetic hand or invest on a robotic prosthetic hand which can cost up to $100,000 per unit.

However, these options are either insufficient to help an individual complete basic daily tasks or too expensive for a common individual to afford. Therefore, with the inspiration to improve people's lives, our project began as an idea to develop a robotic prosthetic hand that would allow people to perform and complete necessary and essential tasks such as clothing themselves more efficiently, driving, or even cooking by controlling each of the fingers in the robotic hand. Our goal was to develop the robotic prosthetic hand that would result in an inexpensive, but efficient product that would be easy for the person the use.

In understanding this project there were 3 major tasks to accomplish. First, we needed to figure out how we would be reading the signals from the arm muscles that belonged to each individual finger. Second, given the signal, we needed to find a way to interpret the signal and pass it to our robotic hand for movement. Third, we needed to design our robotic hand in a way that the fingers could move according to the data being received from the actual muscles allowing the user to perform the desired task.

Product Proposal both Fall and Spring Semesters

CPE 190: Project Proposal

Project Name: Need-A-Hand

Team 6 Members:

Espinoza, Jesus; Gonzales, Jorge; Guzman, Omar; Ladesma, MarcoAntonio

Purpose and Justification

The purpose of this project is to design and build a prosthetic robotic arm to aid those who have lost part of their limbs in accidents or military combat. The prosthetic arm will help improve the quality of life of those who have lost a hand by giving them the ability to once again be able to perform basic daily tasks that may need the ability to use individual fingers in order to accomplish them.

Statement of the Problem or Need

Many people have lost the ability to fully enjoy life due to the loss of an arm, either through unfortunate life events or military combat. This has caused many people difficulties integrating themselves into society because some of the simplest daily tasks such as changing clothes required the use of both hands. This unfortunate disability has created a need for engineers to develop replacement arms. The user will be able to control this robotic prosthetic arm through the use of muscle sensors by reading the electrical signals outputted by the muscles that may still remain in the arm. The robotic arm will simulate the functions of the original limb and will allow its users to better perform their daily tasks by being able to move each finger individually with the correct amount of force necessary to perform the task.

As of now, there are other robotic arms people have access to, but these arms are very costly to the consumer and very few can afford them. However, with the introduction of development microcontroller kits, people have access to the same technology many big firms do for a lot cheaper. This project will allow more people to have access to a robotic arm that will improve their quality of life by making use of devices that are available to the general public.

Project Deliverable and Beneficiaries

- Develop functionality code in C++ - includes code to pick up electrical readings from muscle sensors and map them to the servos.
- Create GUI for electrical signal readings in Java - includes code to read the electrical signals and outputs them to a graph in order to calibrate muscle sensors.
- Test prototype on selected subject – test the first prototype on a person who has full use of both arms.
- Second stage of the project will required a 3D printed hand to better fit the needs of the user.
- Test prototype on final subject – test final design on a person who has lost a partial arm (fall semester).
- At the end of the project, we hope to have a fully functioning prosthetic robotic arm. With the use of sensors, the user will be able to control the arm and will be able to perform simple tasks such as cracking an egg open with both arms.

Knowledge Areas Needed for Project

- Software engineering
- Hardware design
- Circuit Analysis

Time Factors

We would like to start with the research of this project as soon as possible. Documentation of the hardware as well as the software design will need to implemented as soon as possible in order to complete the project. The end of the semester is coming near and we would like to have a completed budget in the following week.

Project Assumptions and Constraints

Assumptions:

- The prosthetic arm will be able to be tested on a person with both arms
- All members are able to learn the required material in the allotted time
- All hardware will be available for purchase

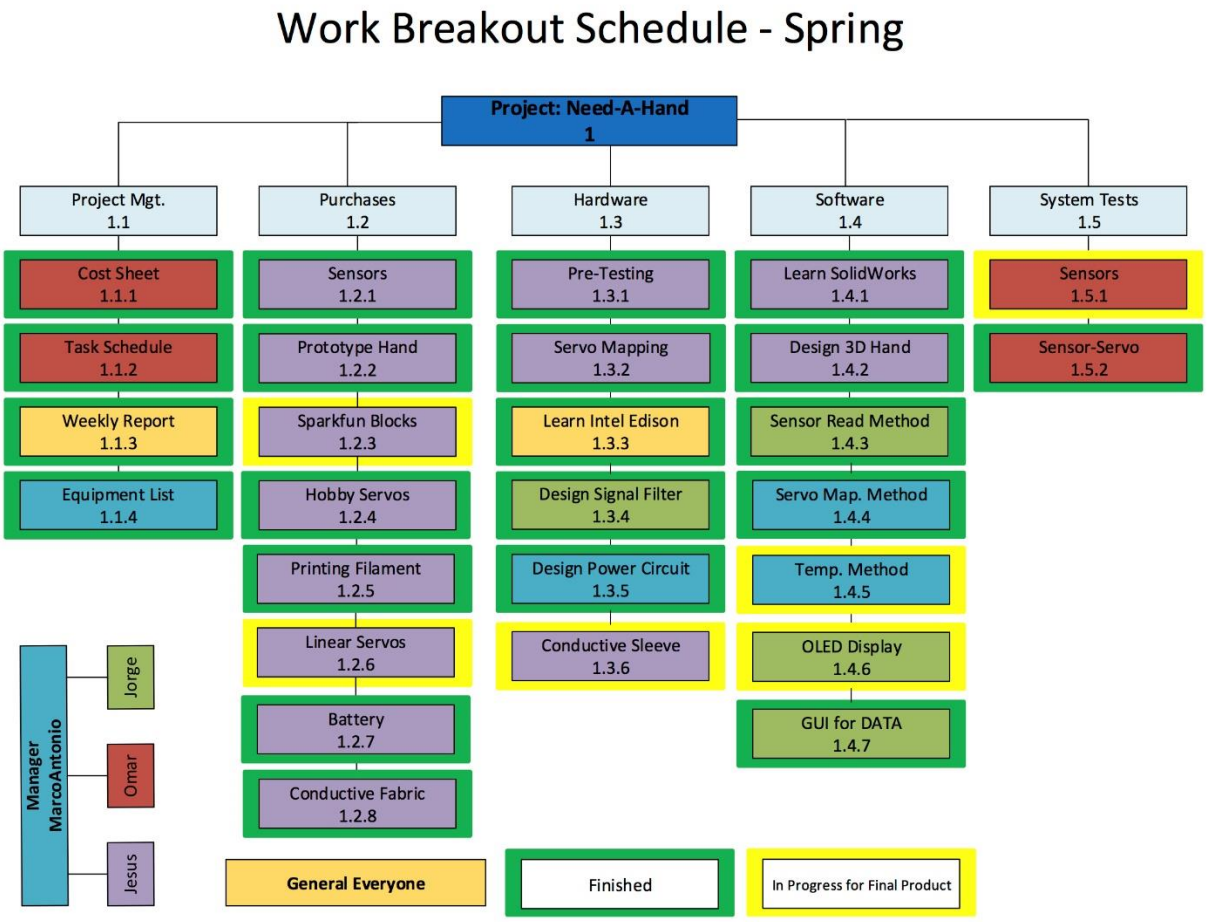
Constraints:

- The prototype must be completed before the end of the Spring 2016 semester

Project Risks

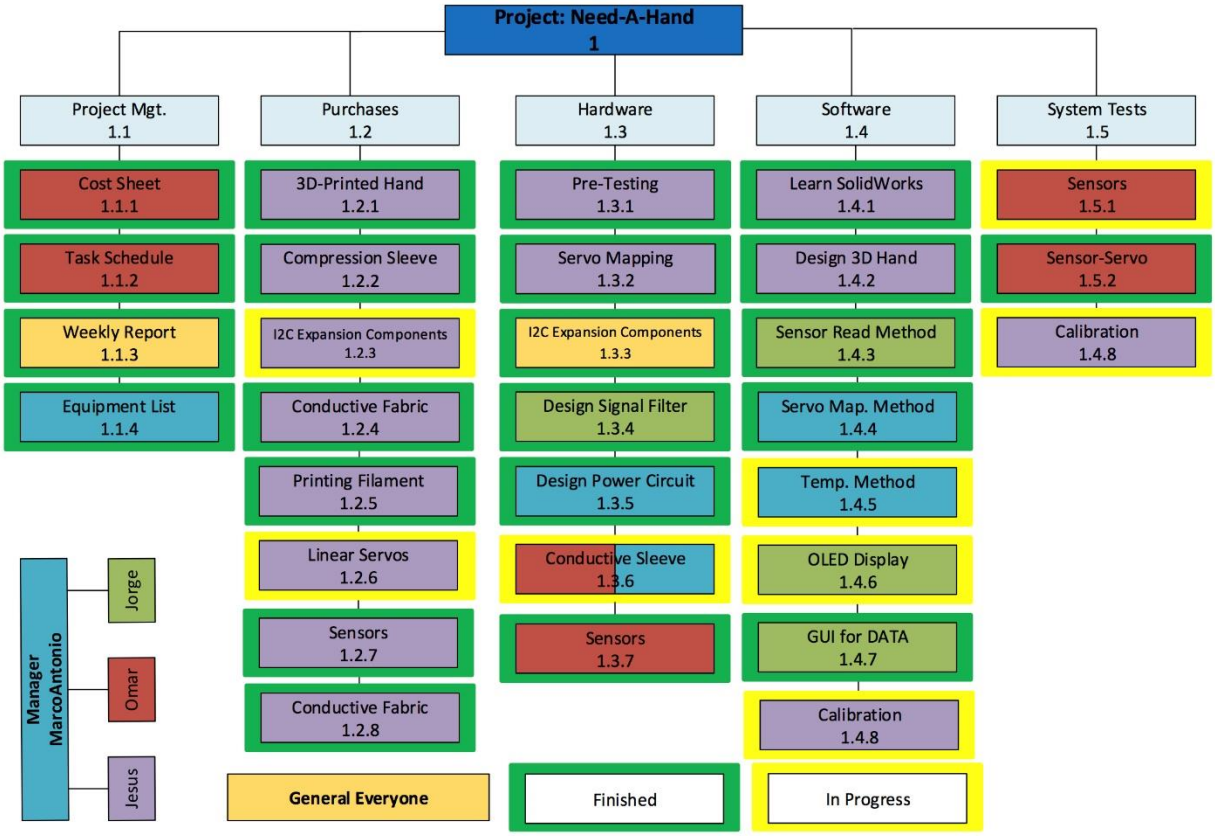
- There is a risk that the hardware may not be available for purchase or that the project may need more skills in the field of mechanical engineering

WBS and Schedule including milestones



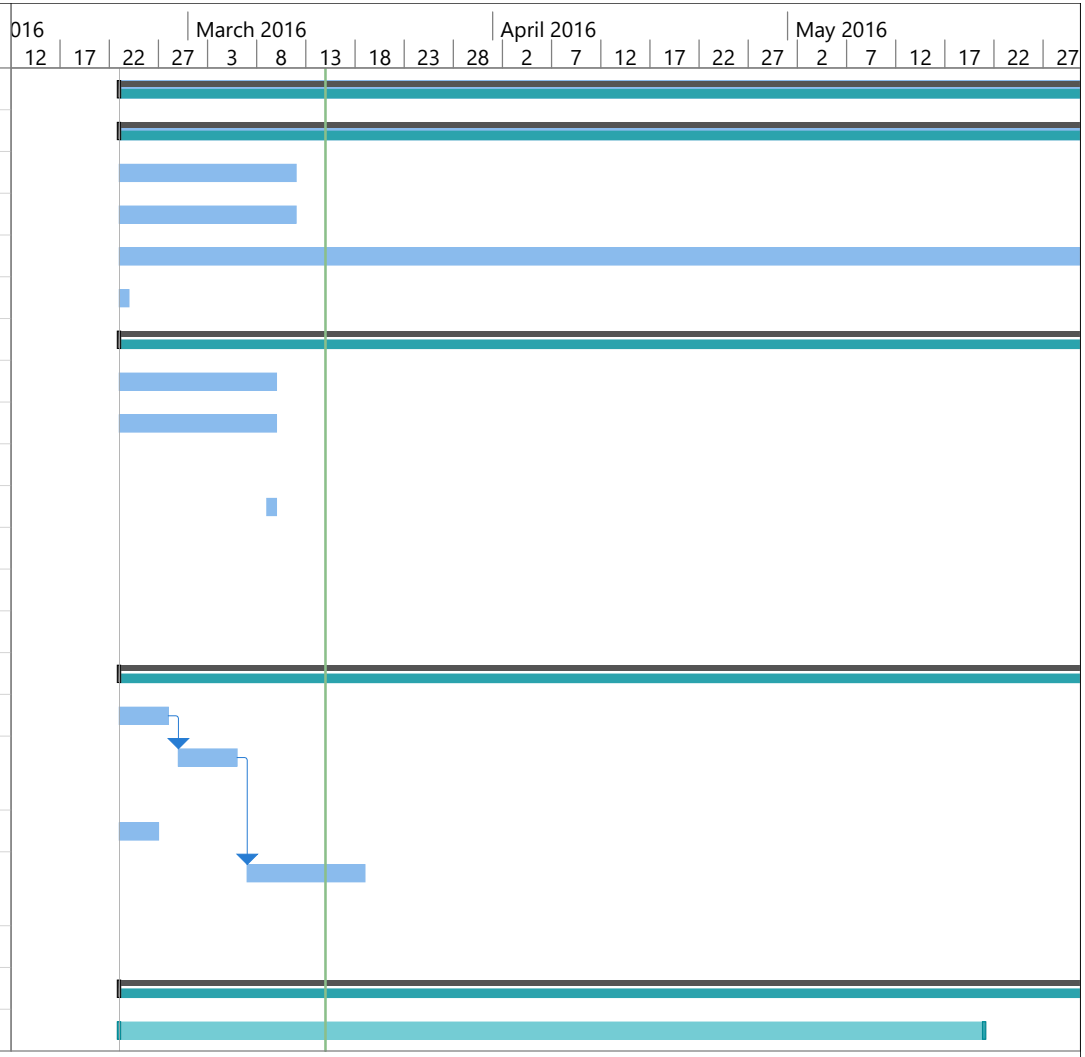
Above we have our WBS for the spring. It is color coded to highlight which team member is assigned which task. It also highlights which tasks have been finished and which ones are in progress or to be implemented in the final product design.

Work Breakout Schedule - Fall



Above we have our WBS for the fall semester. Again just like before it is color coded to highlight which team member is assigned which task. It also highlights which tasks have been finished and which ones are in progress. As you can see it includes many components of the previous first phase design as we are still using some of the components and continue optimizing those parts.

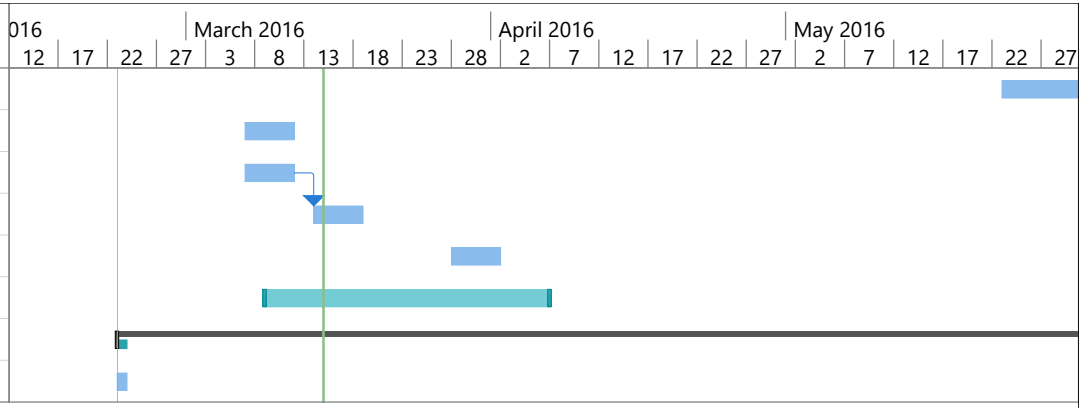
ID	Task Name
1	**1 Need-A-Hand**
2	**1.1 Project Mgt.**
3	1.1.1 Cost Excel Sheet
4	1.1.2 Tasks Schedule
5	1.1.3 Weekly Progress Report
6	1.1.4 Equipment List
7	**1.2 Purchases**
8	1.2.1 Muscle Sensor
9	1.2.2 Prototype Hand
10	1.2.3 Sparkfun Blocks
11	1.2.4 Hobby Servos
12	1.2.5 3D Printing Filament
13	1.2.6 Linear Servos
14	1.2.7 Battery
15	**1.3 Hardware**
16	1.3.1 Pre-Test Muscle Sensor
17	1.3.2 Map Servo with Muscle Sensor Readings
18	1.3.3 Learn Intel Edison Hardware
19	1.3.4 Design Signal Filter for Muscle Sensor Reading
20	1.3.5 Design Power Circuit
21	**1.4 Software**
22	1.4.1 Learn SolidWorks



Project: Project1
Date: Tue 3/15/16

Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress

ID	Task Name
23	1.4.2 SolidWorks: Design 3D Hand
24	1.4.3 Muscle Sensor Reading Method
25	1.4.4 Servo Mapping Method
26	1.4.5 Temperature Reading Method
27	1.4.6 OLED Display Method
28	1.4.7 GUI for DATA
29	**1.5 System Tests**
30	1.5.1 Test Sensors

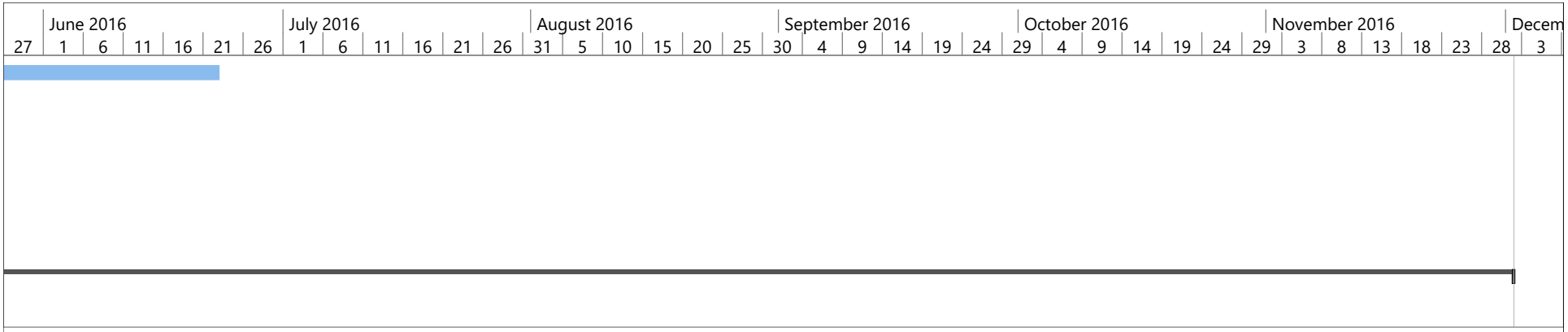


Project: Project1
Date: Tue 3/15/16

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

Project: Project1
Date: Tue 3/15/16

Task	Inactive Summary	External Tasks
Split	Manual Task	External Milestone
Milestone	Duration-only	Deadline
Summary	Manual Summary Rollup	Progress
Project Summary	Manual Summary	Manual Progress
Inactive Task	Start-only	
Inactive Milestone	Finish-only	



Project: Project1
Date: Tue 3/15/16

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

Funding Proposals

The goal of our project is to engineer an affordable prosthetic hand to aid those for hand disabilities. For the reason of making it affordable this project should be affordable for ourselves as well. We are funding the project ourselves not only to emphasize the affordability but also to have full ownership of any biomedical engineering advances we may succeed in.

Below is the total cost of the project broken down by components.

	A	B	C	D
1	**Description**	**Quantity**	**Price**	**Total**
2	Muscle Sensor v3	5	37.99	189.95
3	4M Robotic Hand Kit	1	10.19	10.19
4	MyoWare Electrodes (6-Pack	1	8.99	8.99
5	3M REad Dot Multi-Purpose Monitoring Electrode, Foam, 50/bg	3	13.1	39.3
6	3M REad Dot Multi-Purpose Monitoring Electrode, Foam, 50/bg	1	14.69	14.69
7	Ninjaflex Filament 1.75 mm diameter	1	37.56	37.56
8	Conductive Fabric	1	34.7	34.7
9	3D Hand	1	31.38	31.38
10	Sleve	1	21	21
11			Tax	0.89
12			**Total Cost**	388.65

We have yet to purchase our linear servos and I2C components. We estimate that will bring our cost up to about $600. Even then that cost between 4 people is only $150 each. Which is affordable for a yearlong project. This is why we decided to fund the project ourselves. In the end we also get full ownership so that is good.

Team Member Tasks

Once we determined the work breakdown schedule, we needed to make each member of the group responsible for certain tasks. We split the tasks amongst ourselves by assigning a responsible person as well as a back-up person to a task. The "back-up" was set up in case something happened to the person responsible for the task in order to avoid delays in the development of the project.

We broke down the tasks in the following way:

Task	Assigned to	Backup
Cost Sheet	Omar	Jesus
Keeping track of the cost of every component being purchased for the development of the project. Also, keeping track of how much each member owes		
Weekly Report	MarcoAntonio	N/A
Updating the weekly report with the team minutes from each week's minutes. Also, responsible for submitting the report after each member has updated the report with their respective task progress.		
Equipment List	MarcoAntonio	Jesus
Responsible for tracking the amount of parts being purchased for the project. Also, responsible for keeping track of the equipment being used and for putting in a purchase request if we run of out parts such as gel pads and/or wires		
Purchases of components	Jesus	MarcoAntonio
Responsible for the purchase of the components necessary to complete the project after request has been agreed upon by the member of the group. Also, responsible for updating the cost sheet after each purchase		
Learning Intel Edison tools	General Team	N/A
Each member of the team is responsible for learning the appropriate tools within Intel Edison for the purposes of this project. This includes, but is not limited to learning the IDE by Eclipse which is used to compile our C++ based code.		
Design Signal Filter	Jorge	Jesus

Responsible for the development of a system to filter the signals being read by the muscle sensors whether by using additional hardware or developing some code		
Design Power Circuit	MarcoAntonio	Omar
Responsible for the design and implementation of the power circuit		
Conductive Sleeve	Jesus	MarcoAntonio
Responsible for the design and implementation of the conductive sleeve to better pick up the electric signals from the muscles		
Learn SolidWorks	Jesus	General Team
Responsible for learning SolidWorks which will be used to design the custom 3D hand for the fall semester		
Design 3D Hand	Jesus	MarcoAntonio
Responsible for the design of the 3D hand to be used on the amputee. This 3D hand is supposed to fit in a real human being.		
Method to read sensor signals	Jorge	Jesus
Develop C++ code to read the signals from the MyoWare muscle sensors and pass them to the Intel Edison		
Method to map the servos	MarcoAntonio	Jesus
Develop C++ code to convert the stored data from the muscle sensors into a signal that can be mapped using the servos attached to the robotic hand in order to produce movement		
Temperature Method	MarcoAntonio	Jorge
Develop C++ code to be able to read the surrounding temperature and report it back to the user		
OLED Display	Jorge	Omar
Responsible for the implementation of the OLED display to report the surrounding temperature		
Graphic User Interface	Jorge	Jesus
Responsible for the graphic user interface which will plot each signal being read from the muscle sensors		
Sensor Testing	Omar	MarcoAntonio
Responsible for the testing and debugging of the sensors and any other hardware/software that may be showing errors		
Sensor-Servo communications	Omar	Jorge
Responsible for the communications between the sensors and servos. This includes, but is not limited to wiring and circuit analysis		

User Manual

The artificial robotic hand uses 5 muscle sensors (Figure 1.1) to obtain an analog signal from the muscle contractions. Which are the filtered and translated to servo positions which controls each finger.

Figure 1.1 (http://www.advancertechnologies.com/p/myoware.html)

Muscle Position and Sensor Placing

The five sensors will be placed around the forearm and a common ground on the elbow. Please see the muscle sensor placement (Figure 1.2) to find the best muscle position to obtain a clean signal.

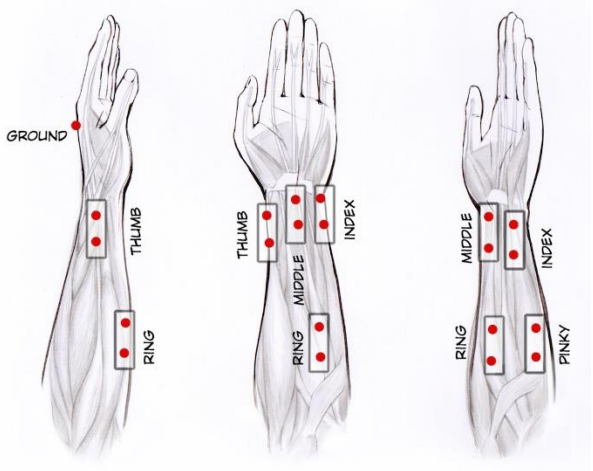


Figure 1.2 (https://backyardbrains.com/experiments/robothand)

The sensors will require two electrode pads each and an electrode pad for the common ground. Before placing each sensor on the forearm, make sure to clean your skin with rubbing alcohol, or conductive gel to obtain better readings.

Each sensor is labeled as P, R, M, I, and T (Pinky, Ring, Middle, Index, and Thumb) place each sensor according to the position from Figure 1.2. Place the common ground electrode near a bone, we strongly suggest using the elbow since there are not many muscles around the elbow. Each sensor will have four wires connected, Signal, Power, Ground, and Reference electrode.

Make the following connections to each sensor.

Muscle Sensor	**Intel Edison pin**
+	5V
-	GND
Sig	A0-A4
R (Reference Electrode	N/A

The reference electrode will connect each sensor to each other using the 1:4 female connector cable. The pinky sensor will be closed to the elbow, use this sensor to obtain the reference electrode from the elbow. Then connect the 1:4 female connector cable from the “R” pin on the sensor to the “R” pin on the other four muscle sensors.

Connect the battery to Intel Edison, you should see the sensors LED’s light up. The Servos should start moving now.

Muscle Sensor Calibration

If the servos are not moving full range and/or not responding to the muscle sensors. You will need to start the calibration process by pressing the calibration button. Once the calibration process starts, you will need to do two actions to calibrate the sensors for each finger individually.

1. To define the baseline, relax your muscle for 10 seconds.
2. To define the Max value, you will need to pulse the muscle by contracting the muscle and relaxing multiply times for 10 seconds.
3. Repeat these steps for each sensor.

Once the calibration process is finished, the servos should be responding. If they are not, you can attempt to calibrate again.

Design Documentation (Spring & Fall)

The purpose of this design is to aid people without a hand, whether they lost their hand in combat, birth defects, or an unfortunate tragedy. Our design will use the Intel Edison to take analog signals generated from muscle contraction and translate/convert the signals to hand movement that will be used to control a 3D printed artificial hand. The user will mount all the sensors and the Edison module on their forearm.

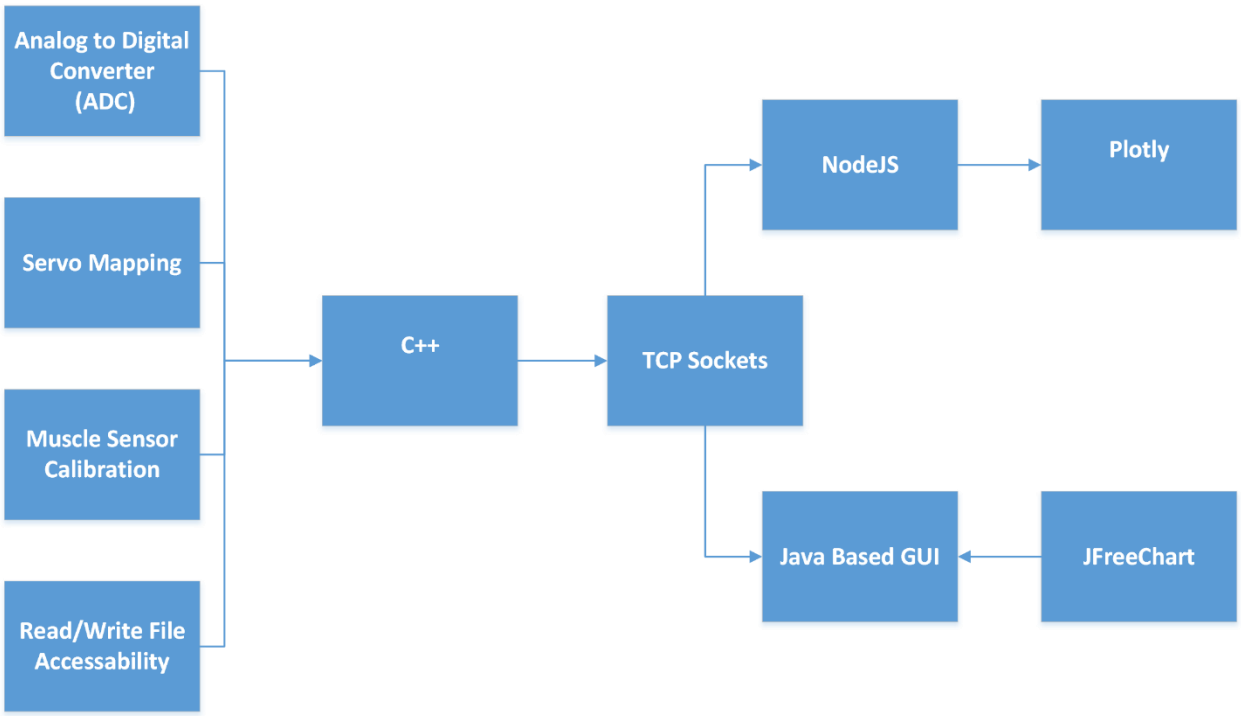
In the spring semester we were able to design and built a prototype design to prove the idea and logic behind our design. Currently we have four sensors reading muscle contraction signals, the logic for five sensors is implemented and tested but due to a hardware limitation on the Intel Edison Arduino board, which is only capable of driving four PWM signals at a time. This will be addressed in the fall semester.

During the spring semester we came across a couple of obstacles that we successfully overcame. Our first obstacle was finding a power supply to drive five servos. Since the servos can draw up to 2A each, we are using a Li-Po battery that is able to continuously output 150A which is more than enough. This gave us the idea to use a smaller battery for the final design, since the Edison module is low power the battery will last longer. We can design an algorithm that will help decide to turn off the power being applied to the servos in order to save battery. Another challenge that we overcame was getting the muscle sensor signals. Since the muscle sensors contract the signal varies between 1 – 2 Volts. The signals were originally mapped to the servos without being modified. What we decided to do, was design the calibration algorithm to calibrate the sensors, essentially what it does is find the baseline and max value

for the sensors. Then we take around 1 hundred samples and find the average to rule out the outliers.

For the Fall semester we plan to migrate our design to the Intel Edison breakout board, which will allow us to condense our design into a smaller form factor that will be mounted on the forearm of the user. We also intend to implement an I2C OLED display to provide the user visual feedback from calibration process and/or battery voltage.

Software Block Diagram & Documentation



The software designed in our project is based on C++ and NodeJS, both which are simultaneously running on the Intel Edison Development Platform. The Intel Edison board is running an embedded version of Linux based on the Yocto Project which has Wi-Fi capabilities.

The C++ contains the following logic.

- Read analog signals and converting to digital.
- Mapping the sensor readings to an angle to be written to the servo
- Client TCP socket
- Muscle sensor calibration logic
- I/O Interfacing and Initialization
- Read and Write to file where calibration values are stored.

The NodeJS contains the following logic.

- Server TCP socket
- Plotly API to send data to Graph

The Java based GUI contains the following logic.

- Server TCP socket
- JFreeChart to graph data

Hardware Block Diagram & Documentation

The Intel Edison Development platform was our best option for our project for various reasons. The fast prototyping can then be easily transferred into a smaller board to design wearable technology. In this case, the Edison module will be mounted onto the design that will be on the users forearm along with all the sensors. The muscle sensor reads EMG signals that can be read with an ADC chip.

The current Arduino Expansion board has an analog to digital (ADC) chip which allows us to sample a analog signal, a pulse width modulation driver which allows us to send a pulse signal

to the Servos. A Bluetooth and Wi-Fi module are built in to the module that allows us to connect to a network to send data to the GUI or Plotly to graph the muscle sensor readings.

The Intel Edison module will allow us to transfer our current project and enhance the hardware by using external PWM and I2C chips to expand the capabilities but at the same time condencing our project to allow the comfort of being on the forearm of a person.

Software

In order to develop the software necessary to operate the robotic prosthetic hand, we decided to write the code in C++ language using the Eclipse IDE.

The following are some excerpts of the code we wrote in order for the hand to operate.

```
int mmap(int, int, int, int, int);
void my_socket(int, int, int, int, int);
void readCalibration();
void Calibration();
```

In the portion of the code shown above, we simply declared the mmap function used to map the signals to the servos. We declared the socket as well as the calibration methods that we will be demonstrating later in this report.

```
int s0 = socket(AF_INET, SOCK_STREAM, 0);
char const *peerHost = "127.0.0.1";
short peerPort = 8589;
struct sockaddr_in peeraddr;
```

In this portion of the code we defined the socket and assigned the IP address the socket will be communicating with. We also assigned the port number.

```
upm::ES08A *servo0 = new upm::ES08A(3);
upm::ES08A *servo1 = new upm::ES08A(5);
upm::ES08A *servo2 = new upm::ES08A(6);
upm::ES08A *servo3 = new upm::ES08A(9);
servo0->setAngle (180);
servo1->setAngle (180);
```

```
servo2->setAngle (180);
servo3->setAngle (180);
```

Next, we set up the pins from which the servos would be acquiring the signal first acquired from the muscle sensors to produce movement in the fingers. We also set the servos to an initial angle of 180 degrees that signified the hand was relaxed and fingers spread out.

```
int count = 0;
int samples = 30;
for (;;) {
usleep(3000);
pinky_val += a0->read();
...
index_val += a3->read();
my_socket(pinky_val,ring_val,middle_val,index_val,0);
if(count == samples){
pinky_val = (((pinky_val+5)/10) * 10)/samples;
…
```

In the above code, we set up some parameters before the signal was passed on to the servo to be mapped. We decided to take 30 data samples. After these samples were taken and added together, the number was rounded to its closes multiple of 10 and the average was taken. This allowed us to reduce the shaking of the hand and resulted in smoother and more accurate movement.

```
if((pinky_low) < pinky_val && pinky_val < (pinky_high + 1)){
pinky_servo = mmap(pinky_val, pinky_low, pinky_high, 180, 0);
servo0->setAngle (pinky_servo);
}
```

After the calculations previously mentioned were done, we compared our new value to the high and low threshold and if it was in between these values, it was then passed unto the servo to rotate to certain degree in order to move the robotic fingers.

```
void readCalibration(){
ifstream file("/home/root/calibration.txt");
if(file.is_open()){
for(int i = 0; i < 10; i++){
file >> cali_val[i];
}
}
file.close();
}
```

In the above method, we set up the way in which we would read the high and low threshold values stored in a text file.

```
void Calibration(){
int tmparray[10] = {0,0,0,0,0,0,0,0,0,0};
int count;
int tmpvar = 0;
int samples = 1500;
printf("Order of calibration: Pinky, Ring, Middle, Index, Thumb\n");
printf("Calibrate each finger twice: Baseline, then Max\n");
printf("Calibration process will start in 3 seconds\n");
printf("\n");
sleep(3);
printf("Calibrating Baseline for Pinky\n");
```

```
        for(count = 0; count < samples; count++){
        tmpvar += a0->read();
        usleep(5000);
        }
        tmparray[0] = tmpvar/samples;
        tmpvar = 0;
        printf("Calibrating Max for Pinky\n");
        for(count = 0; count < samples; count++){
        tmpvar += a0->read();
        usleep(5000);
        }
        ...
        ofstream file("/home/root/calibration.txt");
        if(file.is_open()){
        for(int i = 0; i < 10; i++){
        file << tmparray[i];
        file << " ";
        }
        }
        file.close();
        printf("Calibration Finished...\n");
        }
void my_socket(int sval1, int sval2, int sval3, int sval4, int sval5){
const void *data;
        std::string sensor1 = SSTR( sval1 );
        ...
        std::string sensor5 = SSTR( sval5 );
```

```
        std::string my_string = sensor1 + "," + sensor2 + "," + sensor3 + "," +
sensor4 + "," + sensor5;

        data = my_string.c_str();

        write(s0, data, 20);

        }
```

In the Calibration method, which is one of the most important parts of this code, we acquire the values for the high and low threshold for each individual finger. The above code only shows the code for one finger as we found it redundant to post the code for each individual finger since the code is the same. Nevertheless, we acquired the threshold values by reading 1500 samples and averaging them out. For the low threshold value we maintained our hand relaxed, and for the high threshold value, we contracted our finger fully several times until the calibration process was completed.

Hardware

MyoWare Muscle Sensors

The Myoware muscle sensor measures muscle activation via electric potential, referred to as electromyography (EMG). This muscle sensor is place around our forearm to read finger muscle activity in our arm due to finger movements.

http://www.mayoclinic.org/tests-procedures/electroconvulsive-therapy/basics/definition/prc-20014183

Hobby Servos

For the prototype, we used hobby servos which are mapped using PWM ports from the Intel Edison platform. These servos are great for prototype purposes. Each servo has a 0 to 180 degree rotation movement. We tied fishing line to each of the servos and to each of the fingers to implement the movement of each of the fingers.

Intel Edison

The Intel Edison is an open source development environment that supports many different programming languages such as C, C++, Java, Python, and Arduino Sketches just to name a few. It also comes with integrated Wi-Fi capabilities that makes it very convenient for uploading source quickly.

Testing Hand

The Robotic hand from KindzLabs makes a great prototyping hand for our project. This hand comes as a kit that includes the hand, string, and five rubber fingers. We tied fishing line to each of the fingers, which are also tied to the servos.

3D Printed Hand

For our final product we will be using a 3D printed hand. This hand has a more human based design compared to the testing hand. Each of the fingers has 3 joints that uses flexible material called NinjaFlex. This file was found online from enablingthefuture.org and then printed at a 3D printing shop. The joints were then printed using own our equipment.

Software Testing Plan

As we progressed through our project there were many stops we had make in order to test the multiple software components within our project. Among the components that we tested were the code used to for main functionality of the robotic hand, the Java based code to graph the data from the sensors that will serve as our main graphic user interface next semester, and the code used to get the data graphed using Plotly - a web based graphing interface.

First, as we wrote the code that would be uploaded to Intel Edison, we encountered many difficulties, some of which included misconnections of the sockets, corruption of the eclipse IDE (caused by the download of a set of libraries that were not meant to be used), and lag caused by all the functionalities we had included within the code. Therefore, we had to test each of this components until they worked properly.

Also, when coding our graphic user interface in Java, we encountered some additional difficulties. Instead of running indefinitely or until stopped by the user, the sockets ran only once before terminating the process despite being in a forever for-loop. This meant that if Intel Edison stopped sending data at any one point, the Java based GUI would stop working completely and would ignore any further data received. We had to test and debug this piece of code and although somewhat successful, we decided to leave this portion of our project for the following semester as we plan to incorporate some additional functionalities to our GUI.

Lastly, we wrote a code to use a web-based graphing interface, called "Plotly". With this interface, we were able to graph every signal coming from the sensors. However, these was also encountered with some difficulties. Our initial attempts at graphing our data in Plotly were

somewhat successful because the time series graph was latency. We figure out it was because too many resources were being use in the Intel Edison, so we have to figure out a way to decrease this, and decided to not use the graphing interface at the same time as the arm and only a method to calibrate the prosthetic hand. On the good side, however, our testing of Plotly allowed to easily plot all 4 signals from the muscles sensors and helped us gain a better understanding of the threshold values from the muscle sensors. In the future, our approach to testing our software will be to program blocks of code and start testing them before moving on another block hopefully reducing the amount of errors we will get.

Hardware Testing Plan

1. Test each Myoware sensor to make sure we can get adequate muscle readings from them. Test that the electrodes can stick well and get good readings. Test for faulty sensors.

2. Test Intel Edison. Make sure the wireless communication is working properly. Test that each port necessary is working properly.

3. Test Hobby Servos. Run test on hobby servos to make sure they are fully functional. Test for delay and smooth movement of the servos. No faulty Servos. Test to make sure some don't get too hot. They might be faulty.

4. Test battery. Test to see if battery can power the Intel Edison and the 5 servos simultaneously.

5. Test Oscilloscope. In our case we used our Analog Discovery Kit to use the oscilloscope tool. Make sure the scope is working and we know how to operate it.

6. Once we have done all components testing. Now integrate parts together and test to make 1 finger operate properly. Map sensor to servo and servo to robot finger to get proper movement.

7. Test to get multiple fingers operating at the same time. (There will be interference between the fingers.).

8. Test 3D-Printed Hand for functionality and durability. Test that all finger parts fit together and joints fit properly giving the hand good movement.

9. Test the conductive material. Test the conductive material to make sure it conducts efficiently enough. Make simple led circuit using conductive material to turn led on.

10. Integrate conductive material into sleeve design and test for one muscle sensor working properly with the sleeve.

11. Once you have one muscle sensor working with the sleeve integrate the rest of the sensors and test the functionality.

Integration Plan

1. The first step in our integration plans is to make sure all the hardware and software components are functioning properly individually.

2. The second step is to set up our project to get one functional finger working. We will use one Myoware muscle sensor set to our forearm. We need to wire that sensor to the Edison platform board so get power for the sensor and from there map the sensor reading to a servo. The Analog signal is converted to a digital signal that the servo can read. The servo then gets activated and is routed to a finger on the testing hand to simulate movement.

3. Once we have one functional finger we will integrate more fingers adding 1 more at a time. For this first phase our current Edison platform only has 4 PWM ports so we will only use 4 fingers for the first phase. The integration of multiple fingers causes interference between the fingers. This is where calibration software is integrated to get more clear readings from each finger.

4. We will also integrate the GUI at this point to help us see all finger signals. This helps us see how finger signals interfere with each other. We then adjust sensor placement and sensor ranges to reduce the interference.

5. At this point phase 1 will be finished and we will move on to working on the final prototype. This includes the 3D printed hand design, smaller Edison expansion board, and a conductive sleeve.

6. The 3D hand is used in our final product to give the prosthetic hand some actual functionality. The 3D hand will look nicer and actually be able to pick objects up.

7. The smaller expansion board will be integrated to make our design smaller. We need a small compact wearable technology. Our current testing board only has 4 PWN ports and is too robust to be worn comfortably. Smaller expansion components will be very useful to integrate in our final design.

8. The conductive sleeve will be integrated for two purposes. The first reason is to function as reusable electrodes for the muscle sensors. Having to stick electrodes to your arm after every use is really inefficient. The conductive sleeve will replace the electrodes make the sleeve easy to put on and take off. The sleeve will also help clean up the final build of the prosthetic hand design. It will help hide all the wires.

9. Last we will integrate a better bracket for the 3D printed hand and the servos to be mounted on. The bracket will be 3D printed to hold the servos in the best convenient way.

Accomplishments

Without doubt this semester was full of challenges but it was also full of accomplishments. At the beginning of the semester we had a vision of how our project would look like and how it would operate but we had little idea on how this would have to be done. To make the whole project more manageable and approachable, we learned about Work Breakdown Schedules (WBS) and Scheduling. These both concepts helped the team divide the workload and assignments for better progress of the project. Having weekly reports also contributed to the success of this project. These weekly reports assisted the team with the appropriate updates of each of the tasks that were accomplished for a given week. Following these guidelines made the process through this project much easier.

In terms of the project itself, the team made many accomplishments throughout the semester which concluded with the first prototype of our design. At the beginning of the project, all of us had minimum knowledge on muscle sensors, Electromyography (EMG), prosthetic hands, and muscle placement. We all started doing our research on all of these concepts and starting figuring things out. Our first accomplishment was selecting the appropriate muscle sensors that would serve our needs. We came across the MyoWare Muscle sensor and decided to go with this sensor because it small, convenient, and reliable. Once we had the MyoWare in our hands, we tested it and became familiar with its functionalities. We have also accomplished getting familiar with the Intel Edison Platform. We acquired knowledge on all the different functionalities that the Intel Edison can provide. Using its Wi-Fi functionality, we are able to code the board wirelessly which was very helpful for testing purposes. Our major accomplishment this semester was being able to properly read four out of the five finger

muscle signals and mapped them to its respective servos. For next semester we plan on resolving all the issues that we faced this semester. First of all, we will be solving the interference between the index and the middle finger. Secondly, we will be building a compression sleeve in which we will be placing the muscle sensors. Lastly, we will also be switching our prototype robotic hand for a 3D printed hand.

Summary

Our goal is to design and build a robotic hand that will aid those who have lost a hand in an accident. This robotic hand will be capable of picking up small objects and perform simple daily tasks such as holding a ball. In this first semester of senior design class, we have completed all the tasks that were set at the beginning of the semester. Our team was able to successfully design and build a robotic hand prototype that successfully reads muscle signals from the forearm. The team began this project with minimum knowledge about muscle sensors and EMG signals but we obtained the necessary knowledge through research. Moreover, our current prototype can successfully map four out of the five fingers from the muscle sensors to the servos. For better results, we have also created an algorithm which calibrates the muscle signals taken from the forearm. The current prototype also displays each of the muscle signals on Plotly, which is an online charting service. Some of the major problems we faced was the interference between the middle and the index finger and finger contraction problems. To further explain, when a finger is flexed the signal spikes up but when it is left contracted the signal gets weaker. Our plans for next semester consist of solving the interference issue and finger contraction issues, build a compressive sleeve that will replace the use of electrodes, and switch from a prototype hand to a 3D printed hand. At the end of our second semester, we expect to have a fully functioning robotic hand that will be tested on someone who has lost one of their hands to regain simple capabilities such as picking up small objects.

References

Advancer Technologies – Myoware Muscle Sensor

http://www.advancertechnologies.com/p/myoware.html

Backyard Brains – Signal Classification – Need an Extra Hand

https://backyardbrains.com/experiments/robothand

3D Print Design – Flexy-Hand 2

http://www.thingiverse.com/thing:380665

Intel Edison Kit for Arduino Hardware Guide

http://www.intel.com/content/dam/support/us/en/documents/edison/sb/edisonarduino_hg_331191007.pdf

Intel Edison MRAA Library

https://github.com/intel-iot-devkit/mraa

Intel Edison UPM Library

https://github.com/intel-iot-devkit/upm

JFreeChart Charting Library

http://www.jfree.org/jfreechart/

Plotly (Online Charting)

https://plot.ly/