# Design History File

for

## EMG Prosthetic Arm

Prepared by:

Felipe Rosero Castro, Juan Enrique Villacres, Pandey Thaneshwor, Michael Cui

# Table of Contents

# Executive Summary

Product:
EMG Prosthetic limb is an artificial device that can replace a hand of the human body. Nowadays, they are being popular for patients who are amputated due to various reasons. Artificial organs like EMG prosthetics are widely used in countries like the USA, Russia. The device is an EMG reading prosthetic for amputees that can detect electrical signals from muscle movement. The contraction and relaxation of muscle fibers produces electrical signals which are detected by sensors called MyoWare Sensor. Depending on the strength of the signal and the signal waveform, the prosthetic shall close/open claw. The impact force of closing and opening of prosthetics arms is calculated and calibrated. Myoelectric signals using prosthetics like EMG prosthetic Limb are solely run by signals obtained from the patient's own body. Therefore, it stands out from other externally powered or signals EMG arms. Such EMG signal prosthetics provide confidence to patients and help them run day to day function very smoothly. At all times, the device shall display the performed action on a screen. All the information (e.g Battery Label, current actions, warning signs) needed for the user are displayed on the display screen.

Market:
The device is marketed to amputees and those that would require the use of a prosthesis. EMG prosthetic limb is mainly targeted to people whose one of the hands are amputated. People who can follow basic information shall be the scope of our design project. In the USA alone, 2.1 millions people are living amputated and it is only growing every year (According to acessprosthetics.com). The domain and demand of artificial devices like EMG prosthetics limb is increasing rapidly. In financial terms, the global prosthetics market value is around 9.2 billions dollars. So, there is a lot of room for motivation and innovation to develop artificial organs like EMG Prosthetic Limb.

# Project Plan

GANTT CHART

| | | Task Mode | Task Name | Duration | Start | Finish | Predecessors |
|---|---|---|---|---|---|---|---|
| 1 | | | Lab start | 0 days | Fri 1/31/20 | Fri 1/31/20 | |
| 2 | | | Job Allocation and Brainstroming | 5 days | Fri 1/31/20 | Thu 2/6/20 | |
| 3 | | | Requirements And Development | 15 days | Fri 2/7/20 | Thu 2/27/20 | 2 |
| 4 | | | Customer Needs | 0 days | Thu 2/13/20 | Thu 2/13/20 | |
| 5 | | | Team Assessment | 0 days | Fri 2/14/20 | Fri 2/14/20 | 4 |
| 6 | | | Project Plan | 0 days | Fri 2/21/20 | Fri 2/21/20 | |
| 7 | | | Requirement Specification Documents | 0 days | Thu 2/27/20 | Thu 2/27/20 | |
| 8 | | | Hardware and Software Requirements | 4 days? | Fri 2/28/20 | Wed 3/4/20 | 3,7 |
| 9 | | | Design | 26 days? | Thu 3/5/20 | Thu 4/9/20 | 8 |
| 10 | | | Hardware Design Review | 4 days | Mon 4/13/20 | Thu 4/16/20 | |
| 11 | | | Design Specific Documents | 0 days | Fri 4/17/20 | Fri 4/17/20 | 10 |
| 12 | | | Implementation | 6 days? | Fri 4/17/20 | Fri 4/24/20 | |
| 13 | | | Device Finalize/Code Review | 7 days | Sat 4/25/20 | Mon 5/4/20 | 12 |
| 14 | | | Final Presentation | 0 days | Fri 5/8/20 | Fri 5/8/20 | 13 |

# Customer Needs

**Version 2 (4/30/2020)**

<u>Device description & Expected use</u>
The device is an EMG reading prosthetic for amputees that can detect electrical signals from muscle movement. Depending on the strength of the signal and the signal waveform, the prosthetic will do one of the following tasks:

○ close/open claw with varying speeds

Customer Needs:
● Patient:
● Lightweight
● Long battery life
● Rechargeable battery
● Easy to put on
● Non-invasive
● Material should not cause allergies
● Change how fast claw moves
● Should have a display
● Portable
● Aesthetic
● Medical accuracy
● Physician:
● Easy to work and understand interface so physician can change settings on a patient to patient basis
● Large storage space for physician access of device information and patient movement
● Data collection will work via a series of electrodes
● To protect patient information, access to data will be password protected

# Requirements Traceability Matrix

| Requirements Traceability Matrix | | | |
|---|---|---|---|
| Project Name | EMG Prosthetic Claw | Requirements | |
| Last update | 4/23/2020 | | |
| ID | Customer need | Customer need document | Requirement specification document |
| 1 | Lighweight | 1 | 3.1.1.1 |
| 2 | Portable | 9 | 3.1.1.2 |
| 3 | Long battery life | 2 | 3.1.2.1 |
| 4 | Lightweight | 1 | 3.1.2.2 |
| 5 | Rechargable battery | 3 | 3.1.2.3 |
| 6 | Easy to put on | 4 | 3.1.3.1 |
| 7 | Easy to put on | 4 | 3.1.3.2 |
| 8 | User friendly | | DELETED |
| 9 | Medical accuracy | 11 | 3.2.1.1 |
| 10 | Portable | 9 | DELETED |
| 11 | Durable | | 3.2.1.3 |
| 12 | Durable | | 3.2.1.4 |
| 13 | Easy to put on | 4 | 3.2.1.5 |
| 14 | Relatively cheap | | 3.2.2.1.1 |
| 15 | Durable | | 3.2.2.1.2 |
| 16 | Medical accuracy | 11 | 3.2.2.1.3 |
| 17 | Medical accuracy | 11 | 3.2.2.1.4 |
| 18 | Easy to put on | 4 | 3.2.2.2.1 |
| 19 | Durable | | DELETED |
| 20 | Portable | 9 | 3.2.2.2.3 |
| 21 | Durable | | DELETED |
| 22 | Easy to put on | 4 | DELETED |
| 23 | Non-invasive | 5 | 3.2.2.3.3 |
| 24 | Material should not cause allergies | 6 | 3.2.2.3.4 |
| 25 | Lightweight | 1 | 3.2.2.4.1 |
| 26 | Lightweight | 1 | 3.2.2.5.1 |
| 27 | Durable | | 3.2.2.5.2 |
| 28 | Large stoarge space for physician access of device information and patient movement | 13 | 3.2.2.5.3 |

| | | | |
|---|---|---|---|
| 29 | Medical accuracy and non-invasive | 11,5 | DELETED |
| 30 | Medical accuracy | 11 | 3.2.3.1.2 |
| 31 | Medical accuracy | 11 | DELETED |
| 32 | Medical accuracy | 11 | 3.2.3.2.1 |
| 33 | Medical accuracy, change how fast claw moves, should have a display | 11,7,8 | 3.2.3.2.2 |
| 34 | Medical accuracy | 11 | 3.2.3.2.3 |
| 35 | Medical accuracy | 11 | DELETED |
| 36 | Easy to work and understand interface so physician can change settings on a patient to patient basis | 12 | 3.2.3.2.5 |
| 37 | Data collection | 14 | 3.2.3.2.6 |
| 38 | Easy to work and understand interface so physician can change settings on a patient to patient basis | 12 | 3.2.3.2.7 |
| 39 | To protect patient information, access to data will be password protected | 15 | 3.2.3.3.1 |
| 40 | Non-invasive | 5 | DELETED |
| 41 | Can change how fast claw moves | 7 | DELETED |
| 42 | Can change how fast claw moves | 7 | DELETED |
| 43 | Medical accuracy | 11 | 3.3.2.1 |
| 44 | Medical accuracy | 11 | 3.3.2.2 |
| 45 | Medical accuracy | 11 | 3.3.2.3 |
| 46 | Medical accuracy | 11 | 3.3.2.4 |
| 47 | Medical accuracy | 11 | 3.3.2.5 |
| 48 | Medical accuracy | 11 | DELETED |
| 49 | Non-invasive, easy to put on | 5,4 | 3.4.1.2 |
| 50 | To protect patient information, access to data will be password protected | 15 | 3.4.1.3 |
| 51 | To protect patient information, access to data will be password protected | 15 | 3.4.1.4 |
| 52 | Non-invasive, easy to put on, medical accuracy, material should not cause allergies | 4,5,6,11 | 3.4.1.5 |
| 53 | Medical accuracy | 11 | 3.4.1.6 |
| 54 | User friendly | | 3.2.3.2.8 |

| Design | Implementation | | |
|---|---|---|---|
| **Design specification document** | **Component** | **Software module** | **Additional Comments** |
| 4.3.1, 4.3.2, 4.3.3 | PLA Casing, polyurethane arm slip, aluminum alloy 3003 claws | | |
| 4.2.1, 4.2.2, 4.2.3 | | | |
| 5.2.1.10.1 | | | |
| 5.2.1.10.1 | | | |
| 5.2.1.10.1, 3.8.4, 3.6.9.3, 3.6.9.4 | | | |
| 3.6, 4.4.4, 4.4.5 | | | |
| 3.6, 3.4.5.3.5 | | | |
| | | | |
| 3.5, 5.2.1.1, 5.2.1.1.1, 5.2.1.2, 5.2.1.2.1, 6.1.1, 6.4.3 | | | |
| | | | |
| 4.3.3, 4.3.4, 4.3.1 | | | |
| 4.3.1, 4.1, 4.3.2, 4.4.5 | | | |
| 4.4.1, 4.4.3, 4.2.1, 4.4.2, | | | |
| 4.3.3, 4.3.4 | | | |
| 4.3.3, 4.3.4 | | | |
| 4.2.1 | | | |
| 4.3.3, 4.4.2 | | | |
| 4.4.5 | | | |
| | | | |
| 4.4.5, 4.1.1 | | | |
| | | | |
| | | | |
| 3.4.5.3.5, 3.4.5.3.6, 5.2.1.1, 3.8.3 | | | |
| 3.4.5.3.5, 3.4.5.3.6, 5.2.1.1, 3.8.3 | | | |
| 5.2.1.7, 5.2.1.7.1 | | | |
| 5.2.1.3, 5.2.1.3.1 | | | |
| 4.1.1, 4.1.7 | | | |
| 4.3.1, 4.1.1, 4.1.2, 4.1.3, 4.1.4, 4.1.5, 4.1.6, 4.1.7, 4.1.8 | | | |

| | | | |
|---|---|---|---|
| | | | |
| 7.1.1.3 | | | |
| | | | |
| 6.1.1, 6.1.2, 6.1.5, 6.1.5.1, 5.2.1.6, 5.2.1.6.1, | | | |
| 6.1.1, 3.5 | | | |
| 3.1.1.1, 3.1.1.2, 3.1.1.3, 3.1.1.4, 3.1.1.5, 3.1.1.5.1, 3.1.1.6, 3.1.1.7, 5.1.1, 5.1.1.1, 5.2.1, 5.1.1.1.3, 6.1.4, 6.1.3, 6.1.4.1, 6.1.4.1.1, 6.1.4.1.2, 6.1.4.1.3 | | | |
| | | | |
| 5.1.1.1.1, 6.1.1, 3.5 | | | |
| 6.2.1, 6.3.1, 6.4.1, 6.4.2, 6.4.3, 7.1.1, 7.1.1.1, 7.1.1.2,7.1.1.3, 7.1.1.4 | | | |
| 5.2.1.4, 5.2.1.4.1, 5.1.1.1.2, 6.4.1 | | | |
| 3.1.2.1, 3.1.2.1.1, 3.1.2.1.2, 3.1.2.1.3, 3.9.1.1, 3.9.1.2, 3.9.1.3, 3.9.1.4, 3.9.1.5, 3.1.1.7, 3.1.1.5.1 | | | |
| | | | |
| | | | |
| | | | |
| 3.6.3, 3.6.3.1, 3.6.4, 3.6.4.1, 3.6.5, 3.6.5.1 | | | |
| 5.2.1, 6.1.1 | | | |
| 4.1.2, 5.2.1.7 | | | |
| 4.1.2, 5.2.1.7 | | | |
| 3.6.2, 3.6.2.1 | | | |
| | | | |
| 3.6, 3.8, 5.2.1.1 | | | |
| 3.4.5.2 | | | |
| 3.1.1.7, 3.1.2.1.1 | | | |
| 3.4.1, 3.4.2, 3.4.3, 3.4.4, 3.4.5, 3.4.5.1 | | | |
| 3.4.5.3.5 | | | |
| 3.1.2.1, 3.1.2.1.1, 3.1.2.1.2, 3.1.2.1.3, 3.1.2.2, 3.1.2.3, 3.1.2.3.1, 3.1.2.3.2, 3.1.2.4, 3.1.2.4.1, 3.1.2.4.2, 3.1.2.4.3, 3.1.2.4.4, 3.1.2.5, 3.1.2.6, 3.1.2.6.1, 3. | | | |

# Requirements Specifications Document

for

# EMG Prosthetic Arm

Version: 1.2

Prepared by:

Felipe Castro, Juan Enrique Villacres, Pandey Thaneshwor, Michael Cui

Last modified: 3/6/2020

# EMG Prosthetic Arm

Requirements Specifications Document Concurrence

The undersigned acknowledge that they have reviewed and approved the EMG Prosthetic Arm Requirements Specifications Document. Changes to this document must be coordinated and approved by the undersigned of their designated representatives.

Juan Villacres_______________________________________________ BME261L__________
Mr. ORGANIZATION

_________________________________________________ ___________________________
SIGNATURE DATE

Felipe Rosero Castro____________________________________ BME261L__________
Mr. ORGANIZATION

_________________________________________________ ___________________________
SIGNATURE DATE

Pandey Thaneshwor_______________________________________ BME261L__________
Mr. ORGANIZATION

_________________________________________________ ___________________________
SIGNATURE DATE

Micheal Cui_______________________________________________ BME 261L__________
TITLE ORGANIZATION

_________________________________________________ ___________________________
SIGNATURE DATE

# Table of Contents

# 1.0 General Information

## 1.1 Purpose

This specifications document defines the requirements for the EMG Prosthetic Hand to be used by all BME261L employees and clinicians.

## 1.2 Scope

This specifications document will define the EMG Prosthetic Hand requirements in terms of performance, design, and operation to which all manufacturers and clinicians shall conform. This document shall address the device and device components. Acronyms, assumptions, and constraints shall be listed.

## 1.3 Referenced documents

Customer Needs Document

EMG Prosthetic Arm: Manual

## 1.4 Acronyms and Abbreviations

EMG (Electromyograph)

PLA (Polylactic Acid)

ECG (Electrocardiograph)

# 2.0 Overall Description

The device is an EMG reading prosthetic for amputees that can detect electrical signals from muscle movement. Depending on the strength of the signal and the signal waveform, the prosthetic shall do one of the following tasks:

○ Close/open claw

At all times, the device shall display the performed action on a screen.

## 2.1 Assumptions

1. The user is not blind
2. The user is able to read english
3. The user is able to follow basic instructions
4. The user is not paralyzed
5. The user is able to lift at least 5lbs of weight
6. Following instruction from a clinician, the user shall know:
    - How to operate the device.
    - How to wear the device
        - Arm mount
        - Electrode placement

- Care and maintenance for the device.
- Specific device-status instructions provided from the display

## 2.2 Constraints

Device must be portable with its own power source.

Budget of $100

Timeline of 12 weeks

Prototype device

# 3.0 Requirements

## 3.1 User Requirements

### 3.1.1 Bulk Requirements

3.1.1.1 The total weight of the device shall be less than 5 pounds

3.1.1.2 The device volume shall be less than 500 cm^3

### 3.1.2 Power Requirements

3.1.2.1 The device shall have an average battery life of 6 hours

3.1.2.2 The device battery shall not exceed a weight of 10 Oz

3.1.2.3 The device battery shall be a lithium ion battery that is rechargeable by the user

### 3.1.3 User-friendliness requirements

3.1.3.1 Following instruction by a clinician, users shall be able to take device on/off from affected limb with a single hand

3.1.3.2 Following instruction by a clinician, users shall be able to connect/disconnect electrodes in correct configuration with a single hand

3.1.3.3 The device shall come with a manual detailing its operation

## 3.2 System Requirements

### 3.2.1 Performance Requirements

3.2.1.1 The system shall withstand constant motion for 8 hours without incorrect signal detection as defined by:

- Unintended prosthetic motion
- Lack of response from prosthetic following trigger signal

3.2.1.2 Device shall retain functionality at temperatures between -10° and 50°C

3.2.1.3 The claws shall pass an IP66 rating (water shocks, dust tight).

3.2.1.4 The device shall withstand 1.00 m drop without damage as defined by:

- Incorrect signal detection or misfire following drop

- Any integral structural changes following drop

3.2.1.5 The device socket shall be adjustable from 20 to 30 cm (+-) 2cm in circumference.

### 3.2.2 Hardware Requirements

3.2.2.1 Prosthetic Claw Requirements

3.2.2.1.1 Claw material cost shall not exceed $30 per device

3.2.2.1.2 Claw material shall maintain integrity at temperatures below 100° C

3.2.2.1.3 Claw shall be large enough to pick up a balloon with a diameter of 4 inches

3.2.2.1.4 Claw shall be able to pick up objects of weight not exceeding 10lb

3.2.2.2 Claw Attachment Requirements

3.2.2.2.1 The claw attachment shall be via velcro straps

3.2.2.2.2 The claw attachment strap shall support a weight of at least 15 lb

3.2.2.2.3 The claw attachment strap shall be entirely supported by the user's arm

3.2.2.3 Electrode Requirements

3.2.2.3.1 EMG electrode wires shall be electrically insulated

3.2.2.3.2 The prosthetic shall have replaceable electrodes with length chosen by clinician

3.2.2.3.3 EMG electrodes shall be attached to user skin via electrode pads

3.2.2.3.4 EMG electrode pads shall match FDA Class II regulations for electrocardiograph electrodes

3.2.2.4 Motor Requirements

3.2.2.4.1 The motor shall not exceed a weight of 20 Oz

3.2.2.4.2 The motor shall receive an output from the microcontroller

3.2.2.5 Microcontroller and Electronics Requirements

3.2.2.5.1 The microcontroller shall be an Arduino UNO

3.2.2.5.2 The microcontroller and associated electronics shall be encased in a PLA casing

3.2.2.5.3 The device shall include 1TB external storage which will a history of EMG inputs with corresponding outputs

### 3.2.3 Software Requirements

3.2.3.1 Data Acquisition Requirements

3.2.3.1.1 EMG data acquisition shall be via 4 analog inputs coming from the electrodes placed at two sets of antagonistic muscle groups as determined by the clinician

3.2.3.1.2 EMG signal shall be continuously sampled by microcontroller at a frequency of 3000 Hz

3.2.3.1.3 EMG signal shall be amplified using a differential amplifier comparing two analog inputs from a single set of antagonistic muscle groups

3.2.3.2 Code Functionality Requirements

3.2.3.2.1 Code shall send a high voltage signal to motor when EMG reading exceeds threshold

3.2.3.2.2 The code shall include a function to receive each input type, according to the following:

- EMG signal
  - Store information
  - Display current status
- Claw strength toggle
- Claw rotation speed toggle
- Access information
  - Leads to a function asking for a password

3.2.3.2.3 Code shall send high voltage digital signal to indicator LED when EMG readings exceed threshold

3.2.3.2.4 Software shall not function below EMG threshold

3.2.3.2.5 The code shall have comments describing each function

3.2.3.2.6 The code shall store input/output information based one of two conditions:

- EMG successfully reaches threshold
- External button held by user for troubleshooting purposes

3.2.3.2.7 The code shall have a GUI interface displaying the following information:

- Current action
- Current EMG input
- Current device outputs
- Option to access other device information with security password prompt

If password is successful, a second GUI interface will display the following options:

- Access and display EMG inputs
- Average inputs from range x to y
  - x and y can be selected
- Update threshold

- Set by clinician
- Reset memory - will require a second password
  - Erases all stored information without affecting settings
- Access outliers
  - Accesses information stored during troubleshooting

3.2.3.3 Security requirements

3.2.3.3.1 Stored information shall be password protected by a 16-character password including at least one of the following:

- One capital letter
- One number
- One symbol character !@#$%^&*()_+

## 3.3 Interface Requirements

### 3.3.1 Input Requirements

3.3.1.1 Inputs to the microcontroller shall consist of four analog EMG signals (two sets of antagonistic muscle groups)

3.3.1.2 The user shall be able to modulate the rotation speed based on an ordinal scale of 1-5 using two buttons, matching the following speeds:

- 1: 1 degree per second
- 2: 3 degrees per second
- 3: 5 degrees per second
- 4: 15 degrees per second
- 5: 30 degrees per second

3.3.1.3 The user shall be able to modulate the grip strength based on an ordinal scale of 1-5 using two buttons, matching the following forces:

- 1: 20lb
- 2: 40lb
- 3: 60b
- 4: 80lb
- 5: 100lb

### 3.3.2 Output Requirements

3.3.2.1 The Arduino UNO microcontroller shall output 5V to the device motors

3.3.2.2 Output contraction/relaxation shall happen with a delay of no less than 1000 ms

3.3.2.3 Upper claw shall flex up to 160 degrees

3.3.2.4 Lower claw shall flex up to 160 degrees

3.3.2.5 Claws shall remain relaxed when no output is detected

## 3.4 Clinician Requirements

3.4.1.1 The device shall come with an instructional manual for clinicians detailing limitations, usage, and a comprehensive guide to the interface

3.4.1.2 The device shall be compatible with ECG electrodes

3.4.1.3 Access to device data shall be password protected by a password held by the clinician

3.4.1.4 Upon first use, the device shall prompt the clinician to set up their password before continuing onto adjusting the device for patient use

3.4.1.5 The electrodes used shall meet the requirements specified by "Electrocardiograph Electrodes - Class II Special Controls Guidance for Industry and Food and Drug Administration Staff"

3.4.1.6 Use of the device shall be implemented only for those patients meeting the conditions described in "ISO 8548-3:1993 Prosthetics and orthotics — Limb deficiencies — Part 3: Method of describing upper limb amputation stumps"

# Appendix A

## References

**[1] ISO 8548-3:1993 Prosthetics and orthotics — Limb deficiencies — Part 3: Method of describing upper limb amputation stumps**

**[2] International Electrotechnical Commission (2013). *IEC 60529 - Degrees of protection provided by enclosures (IP Code)*. International standard (2.2 ed.). p. 21. ISBN 9782832210864. OCLC 864643678.**

**[3] CFR Title 21, Part 890, Subpart D, Sec. 890.3500**

**[4] 21 CFR Part 801**

**[5] ANSI/AAMI EC12, “Disposable ECG electrodes”**

**[6] 21 CFR 870.2360**

# Requirements Review Minutes

Red Text = Completed

Suggestions to our Document:

- Battery life is controlled by size, weight, and expense.
- Replaceable pads should be considered.
- They can take off the device easily taking showers and not reading.
- What threshold of motion do we need to set off?
- Materials:
    - o PVC materials.
    - o Have we identified our audience?
    - o Future consideration what change we can do?

Action Items:

- Specify replaceable electrodes to add to document
- Reword following clinician (3/14.2020)
    - o After the fact have them do it their own
- Take a look at assumptions (3/14/2020)
    - o People would want the device
    - o Target audience assumptions
- Specify in introduction/assumption somewhere we can characterize the audience a little bit More. (3/14/2020)
- Future studies thing, aluminum but in the future durable but also pretty lightweight

# Detailed Design Specifications Document

for

# EMG Prosthetic Arm

Version: 1.4

Prepared by:

Juan Enrique Villacres, Pandey Thaneshwor, Michael Cui, Felipe Castro.

Last modified: 4/23/2020

# EMG Prosthetic Arm

Detailed Design Specifications Document Concurrence

The undersigned acknowledge that they have reviewed and approve the EMG Prosthetic Arm Detailed Design Specifications Document. Changes to this document must be coordinated and approved by the undersigned of their designated representatives.

| | |
|---|---|
| Juan Villacres___ | BME216L__________ |
| Mr. | ORGANIZATION |
| ____________________________________________ | ______________________ |
| SIGNATURE | DATE |
| Felipe Rosero Castro | BME216L |
| Mr. | ORGANIZATION |
| ____________________________________________ | ______________________ |
| SIGNATURE | DATE |
| Pandey Thaneshwor________________________________________ | BME216L __________ |
| Mr. | ORGANIZATION |
| ____________________________________________ | ______________________ |
| SIGNATURE | DATE |
| Michael Cui_________________________________________________ | BME216L |
| Mr. | ORGANIZATION |
| ____________________________________________ | ______________________ |
| SIGNATURE | DATE |

# Table of Contents

# 1.0 General Information

## 1.1 Purpose

This specifications document defines the design for the “EMG Prosthetic Hand”, a prosthetic device used for patients with a single amputated hand. The device acts as a replacement hand that provides basic motions including grasping and rotation upon detecting specified EMG signals.

## 1.2 Scope

This specifications document will specify the design, use, and components of the device.

- Bulk design
  - CAD diagrams
- Electronic design
  - Circuit flowchart
  - Pin diagram
- Software design
  - Software flowchart
  - Data content
  - Functions and Libraries

Specifications on materials used and dimensions are also provided.

## 1.3 Referenced documents

Customer Needs Document

EMG Prosthetic Arm: Manual

ISO 8548-3:1993 Prosthetics and orthotics — Limb deficiencies — Part 3: Method of describing upper limb amputation stumps

IEC 60529 CFR Title 21, Part 890, Subpart D, Sec. 890.3500

21 CFR Part 801

ANSI/AAMI EC12

21 CFR 870.2360

## 1.4 Acronyms and Abbreviations

EMG – Electromyography

sEMG - Surface electromyography

PLA - Polylactic Acid

ECG - Electrocardiograph

ISO - International Organization for Standardization

CFR - Code of Federal Regulations

IEC - International Electrotechnical Commission 4

ANSI - American National Standards Institute

# 2.0 Overall Description

The device is a sEMG detecting prosthetic patients with single-hand amputations. The device detects electrical signals from muscle movement and translates them to functional mechanical motion of the hand. Depending on the signal reaching threshold and patient input parameters, the prosthetic shall do one of the following tasks: ○ Close/open claw ○ Rotate claw At all times, the device shall display the performed action on a screen.

## 2.1 Assumptions

2.1.1 The user is not blind

2.1.2 The user is able to read English

2.1.3 The user retains a functioning hand

2.1.4 The user retains a residual limb onto which the device can be fastened

2.1.5 The user is able to follow basic instructions.

2.1.6 The physician has access to a laptop

2.1.7 After instruction by a physician, the user will know how to operate and maintain the device

2.1.8 The device is used for basic everyday tasks that does not include vigorous and strenuous activity that could potentially damage the device

2.1.9 The user is able to lift the weight of the device: Approximately 5 lbs

## 2.2 Constraints

2.2.1 The device is portable and has its own power source

2.2.2 All components used to make the device are constrained to a total budget of 100$

2.2.3 The device's development timeline is restricted to 15 weeks

2.2.4 The device is in design and no prototype or testing is done

2.2.5 The device signal acquisition method (sEMG) is inherently noising

2.2.6 Should the physician forget the password, there is no method for password retrieval

2.2.7 There is an ongoing pandemic coinciding with the development of this device

2.2.8 Sophisticated signal filtering and detection methods implemented by other medical-grade sEMG-detecting prosthetics are not used. This includes:

2.2.8.1 Motor-unit signal decomposition

2.2.8.2 Cohen class signal transformation

2.2.8.3 Autoregressive model

## 2.3 Risks

2.3.1 People who suffer from chronic arthritis or bone weakness can have pain over time due to weight of device.

2.3.1.1 Contact physician about use and reconsider use

2.3.2 The device is not meant for prolonged exposure to water so water exposure could cause damage to the device and/or user.

2.3.2.1 Avoid swimming with or washing the device. Instructions for cleaning the device are provided

2.3.3 The device uses gears to move the claw which could injure the user should they put their hand in between the gears

2.3.3.1 Avoid contact with the gears when the device is powered

2.3.4 Should the rubber padding on the claws deteriorate, the claws have the potential to be sharp

2.3.4.1 Follow proper maintenance instructions

2.3.4.2 Contact physician should the device be damaged in any way

# 3.0 System Design

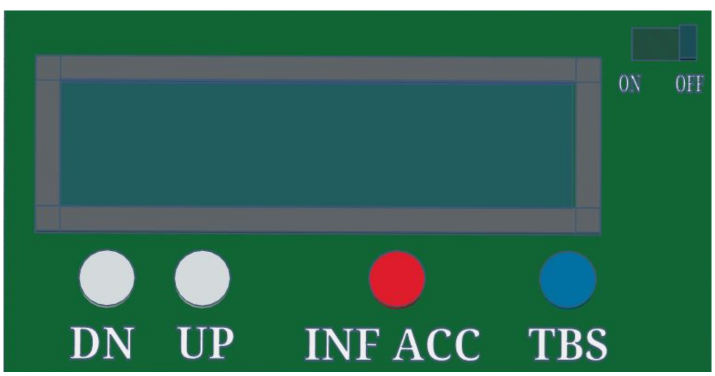
This section details the user interface, product assessment, packaging and labeling, and instructions for setup, use, troubleshooting, and maintenance.

## 3.1 User Interface

This section details the user interface of the device, relevant actions, and display prompts.

### 3.1.1 Prosthetic Display

3.1.1.1




3.1.1.2 The user interface will display the claw speed with bars from 1-5

3.1.1.3 If the user presses the UP button, a bar will be added, up to a maximum of 5

3.1.1.4 If the user presses the DN button, a bar will be added, up to a minimum of 1

3.1.1.5 If the user presses the TBS button, the display will toggle between "troubleshooting" and "please flex muscle".

3.1.1.5.1 Once done, the display will show "Troubleshooting complete" followed by "Please contact physician"

3.1.1.6 At all times, the device will display the grip speed setting, and mode (discrete or graded).

3.1.1.7 Should the user press the INF ACC button, the display will show "Please contact physician"

**3.1.2 Computer Display**

3.1.2.1 Upon first time setup, a computer interface will display the following in sequence:

3.1.2.1.1 "Welcome. Please choose a password. Must contain at least one number, a capital letter, and a symbol character !@#$%^&*()_+"

3.1.2.1.2 Should the password fail to match the requirements, the display will show: "Password does not match criteria. Please try again."

3.1.2.1.3 Should the password match the requirements, the display will show: "Password setup complete. Please put on a device and press ok".

3.1.2.2 Once ok is pressed, the display will show: "Lower Limb Noise Calibration?"

3.1.2.3 If "Yes" is selected, the following prompts will be displayed in sequence, otherwise they will not:

3.1.2.3.1 "Calibrating Noise. Please instruct the patient to jump 10 times" followed by "Done"

3.1.2.3.2 "Calibrating Noise. Please instruct the patient to squat 10 times" followed by "Done"

3.1.2.4 Regardless of the choice, the following prompts will be displayed:

3.1.2.4.1 "Calibrating Noise. Please instruct the patient to raise their arms 10 times" followed by "Done"

3.1.2.4.2 "Calibrating Noise. Please instruct the patient to walk for 30 seconds" followed by "Done"

3.1.2.4.3 "Calibrating Noise. Please instruct the patient to rotate their arm 10 times" followed by "Done."

3.1.2.4.4 "Calibrating Noise. Please instruct the patient to flex their arm 10 times" followed by "Done."

3.1.2.5 Upon completion, the display will show "Calibration complete. Would you like to use graded actuation?" followed by "Yes" or "No" options.

3.1.2.6 During troubleshooting, the display will show a display with the options "Return", "Access troubleshoot information" and "display sampled signal".

3.1.2.6.1 "Access troubleshoot information" will prompt "Please press information access button on device"

3.1.2.6.2 If pressed, the display will show all stored signals over time.

3.1.2.6.3 “Display sampled signal” will display a real-time signal sampled from the MyoWare device

3.1.2.6.4 “Return” will return to the main display.

3.1.2.7 When connecting the device anytime other than first time use, a display will show up with the options: “Troubleshoot”, “Recalibrate”, and “Close”

## 3.2 Packaging and Labeling

### 3.2.1 Packaging

3.2.1.1 Packaging for the EMG prosthetic will be 34cm x 50cm x 20xm white cardboard box.

3.2.1.2 The package will be covered by PVC plastic wrapping.

3.2.1.3 The EMG prosthetic will be held by a styrofoam mold for protection that fits within the white cardboard box.

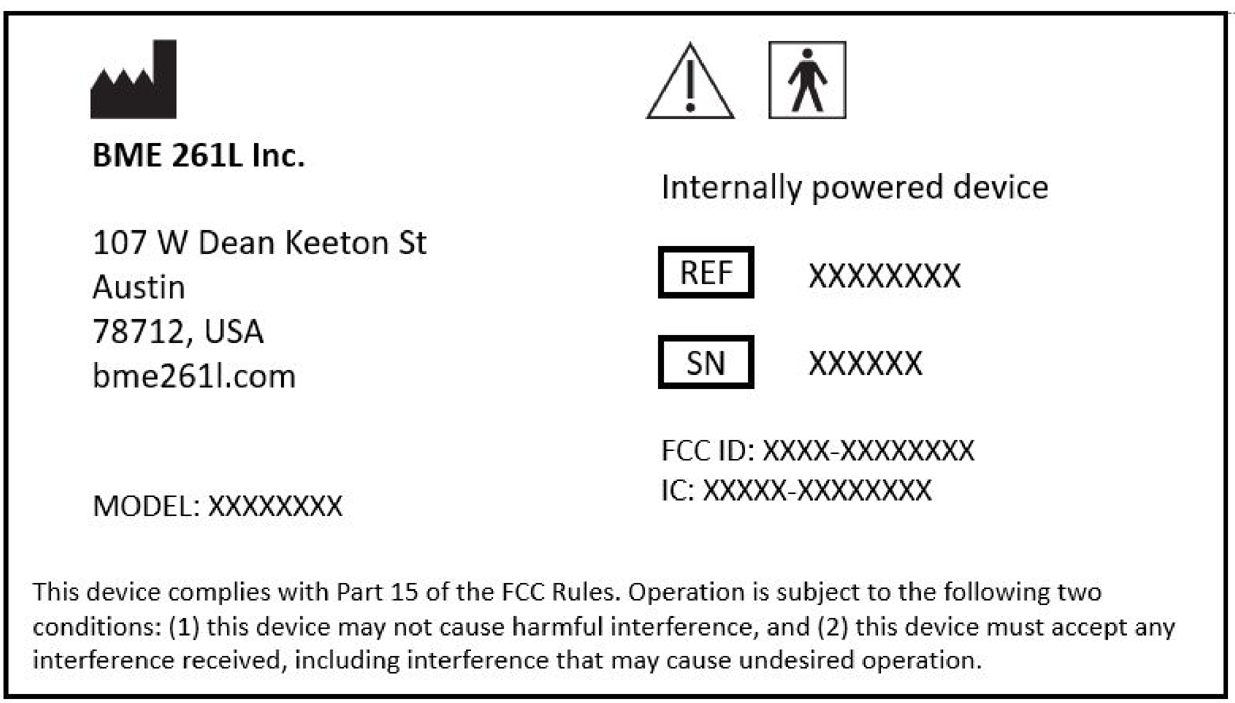
3.2.1.4 The package will resist a 6ft fall with no damage

3.2.1.5 The package will resist a weight of 20lbs with no damage

### 3.2.2 Labeling

3.2.2.1 The logo will be clearly marked on one 34cm x 50cm side of the package in 30pt Impact font

3.2.2.2 The company address and name will be clearly marked on the box




3.2.2.3

3.2.2.4 Manufacturer symbol

3.2.2.5 Company/manufacturer name

3.2.2.6 Company/manufacturer address

3.2.2.7 Company/manufacturer website

3.2.2.8 Model number

3.2.2.9 Reference number

3.2.2.10 Serial Number

3.2.2.11 FCC ID

3.2.2.12 IC ID (for Canada)

3.2.2.13 Internally powered device signifies that the device is battery-powered

3.2.2.14 Exclamation mark in triangle: signifies that the device instructions must be consulted in order for the device to be safely operated

3.2.2.15 Person in square: signifies that this is a type BF device. That is, a device that is partially electrically connected to patient

3.2.3 FCC Part 15 compliance, which applies to electrical devices

## 3.3 Product Assessment

3.3.1 Thoroughly examine package before opening

3.3.1.1 If any damage to the package is present, please contact the company through bme261L.com and return the package to the given address and submit a report. A new one will be sent free of charge.

3.3.2 Remove the plastic wrapping

3.3.3 Open the box and pull out the styrofoam block

3.3.4 Separate the two sides of the styrofoam block holding the prosthetic

3.3.5 Thoroughly examine the prosthetic

3.3.5.1 If any damage is present, please contact the company through bme261L.com, put the prosthetic back in the styrofoam blocks, put the block back inside the box, seal the box with tape, and visibly mark on the box: “OPENED. DAMAGED” and return the device. A new one will be sent free of charge.

3.3.5.2 If any damage is present, please contact the company through bme261L.com. A new one will be sent free of charge.

## 3.4 Setup

3.4.1 Arduino software must be installed prior to use

3.4.2 Please refer to https://www.arduino.cc/en/main/software for download and installation.

3.4.3 Connect the software from the USB port onto the computer and install the program

3.4.4 Open the battery pack and add batteries

3.4.5 Connect the prosthetic via USB serial port onto the computer

3.4.5.1 The software will detect first time use and provide a password entry prompt

3.4.5.2 Choose a password based on the criteria and safely store the password

3.4.5.3 Instruct the patient to put the prosthetic device on the residual limb

3.4.5.3.1 Firmly hold the prosthetic with one arm

3.4.5.3.2 Slide the foam socket of the prosthetic onto the residual limb

3.4.5.3.3 Adjust the velcro strap to comfort

3.4.5.3.4 Place sensor electrodes with top electrode over the muscle belly and middle electrode over muscle end closest to device

3.4.5.3.5 Place reference electrode on skin lateral to the sensor

3.4.5.3.6 Flip the power switch to “ON”

## 3.5 Product Assessment (Done by Orthotist)

3.5.1 The software will provide step-by-step instructions for calibrating the device

3.5.2 There is an option to skip lower limb noise calibration for patients with amputated lower limbs. Click on the “Skip lower limb noise calibration” option.

3.5.3 Lower limb noise calibration

3.5.3.1 Following the display prompt, Instruct the patient to walk for 30 seconds

3.5.3.2 Following the display prompt, Instruct the patient to jump 10 times

3.5.3.3 Following the display prompt, Instruct the patient to squat 10 times

3.5.4 Upper limb noise calibration

3.5.4.1 Following the display prompt, Instruct the patient to raise their arms 10 times

3.5.4.2 Following the display prompt, Instruct the patient to rotate their arm 10 times

3.5.4.3 Following the display prompt, Instruct the patient to stretch and flex their arm 10 times

3.5.5 Signal Calibration

3.5.6 Following the display prompt, instruct the patient to flex the forearm muscle intended for use

3.5.6.1 Repeat the procedure each time the prompt is displayed.

3.5.7 The display will inform the orthotist that calibration is complete.

3.5.8 The display will prompt the orthotist to disconnect the device from the computer

## 3.6 Instructions for Use

3.6.1 The orthotist will provide the following instructions to the patient

3.6.2 Instruct the patient to move their arm around without flexing the connected muscle

3.6.2.1 The device will not grip

3.6.3 Instruct the patient to flex the connected muscle

3.6.3.1 The device will grip

3.6.4 Instruct the patient to press the UP button and flex the muscle

3.6.4.1 The device will grip faster

3.6.5 Instruct the patient to press the DN button

3.6.5.1 The device will grip slower

3.6.6 Instruct the patient to press the TBS button

3.6.6.1 The device will prompt to flex muscle ten times

3.6.6.1.1 Instruct the patient to flex the connected muscle ten times

3.6.6.2 The device will prompt end of troubleshoot and contact physician

3.6.7 Instruct the patient to press INF ACC button

3.6.7.1 The display will prompt to contact physician

3.6.8 Instruct the patient to grab an object

3.6.9 Instruct the patient to access the battery pack

3.6.9.1 Instruct the patient to turn off the device

3.6.9.2 Show the patient the location of the battery pack latch

3.6.9.3 Instruct the patient to pull out the battery pack and remove the batteries

3.6.9.4 Instruct the patient to replace the batteries and close the battery pack

3.6.10 Instruct the patient to turn on the device

## 3.7 Optional Graded Actuation

3.7.1 For patients who would like the device to function without full flexion

3.7.1.1 Turn on device

3.7.1.2 Connect device to computer via serial USB

3.7.1.3 When prompted, type correct password

3.7.1.4 Click “Activate Graded Actuation”

3.7.1.5 Instruct the patient to flex the muscle at various strengths

3.7.1.5.1 The claw speed will match the flexion strength rather than toggle options

3.7.2 NOTE: Graded actuation inactivates UP and DN buttons since they are not used in this mode.

## 3.8 Device Maintenance

3.8.1 Device exterior can be cleaned with disinfectant wipes

3.8.2 Foam pad can be removed and replaced on a need-to basis

3.8.2.1 Contact physician for more pads

3.8.3 Electrodes will be replaced daily and in between removal of device

3.8.4 Battery replacement can be done on a need-to basis

## 3.9 Troubleshooting

3.9.1 Signal problems

3.9.1.1 Connect the device to computer via serial port

3.9.1.2 Following a password prompt, enter the password

3.9.1.3 Click “Access stored information”

3.9.1.4 Press “INF ACC” on the device

3.9.1.5 Click “Display Troubleshoot Signals”

3.9.1.6 Click “Return”

3.9.1.7 Click “Display Signal Samples”

3.9.1.8 If signals match, perform the following:

3.9.1.8.1 Click “Recalibrate”

3.9.1.8.2 Follow calibration instructions

3.9.1.8.3 Instruct patient to test device

3.9.1.8.4 If device does not work, contact bme261L.com and you will be connected with an assistant

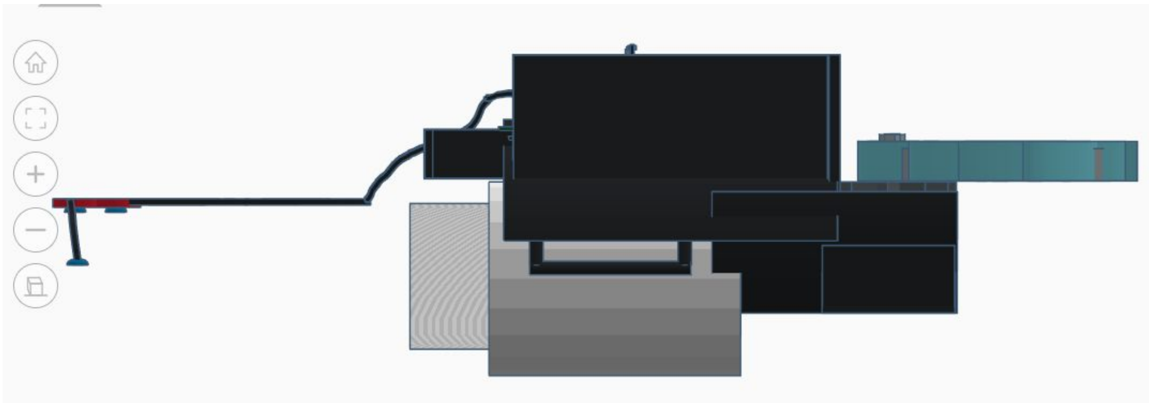
3.9.1.9 If signals do not match, contact bme261L.com and you will be connected with an assistant

3.9.2 Any hardware problems should be directed to bme261L.com

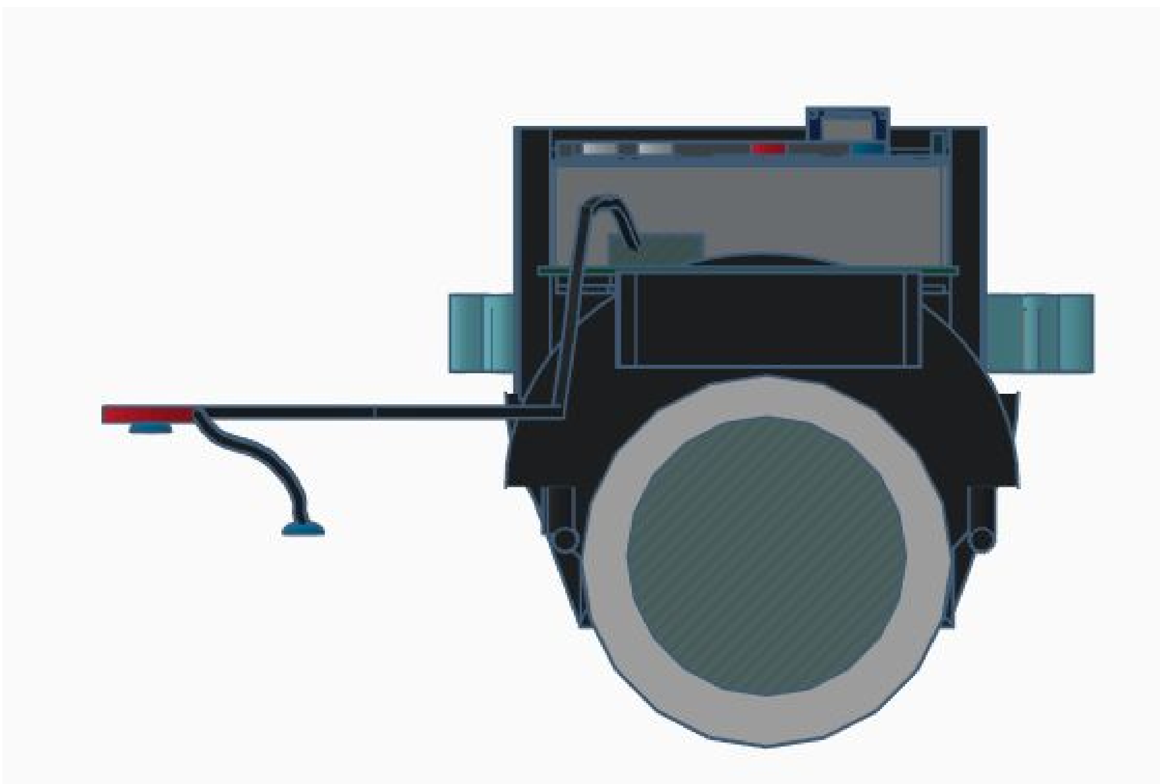
## 4.0 Hardware Design

This section details the external components, dimensions, functions, and materials.

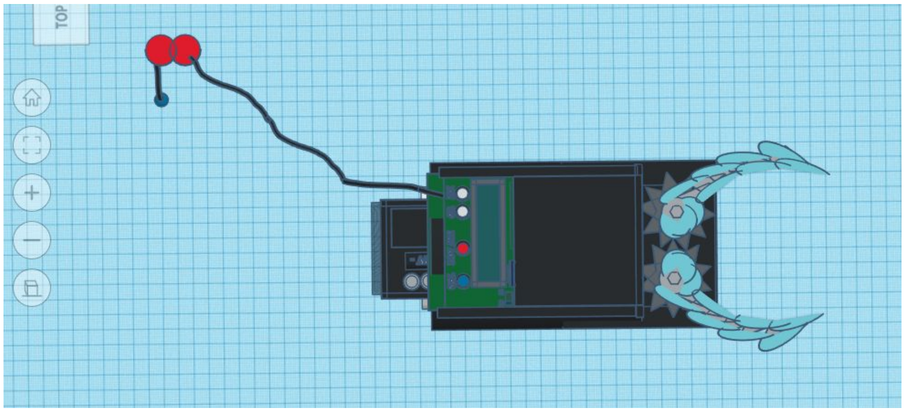
## 4.1 CAD Diagrams

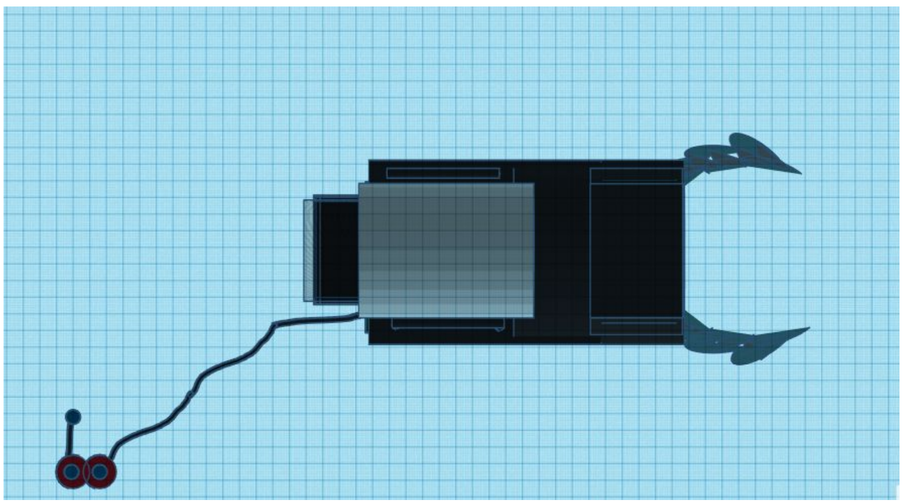


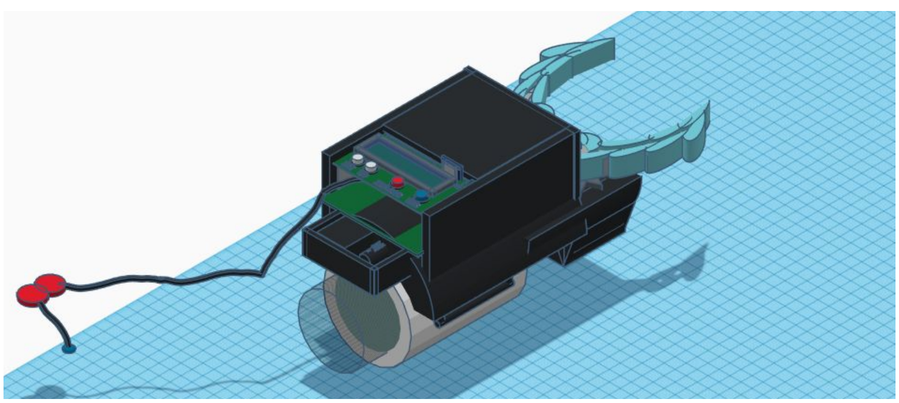
4.1.1

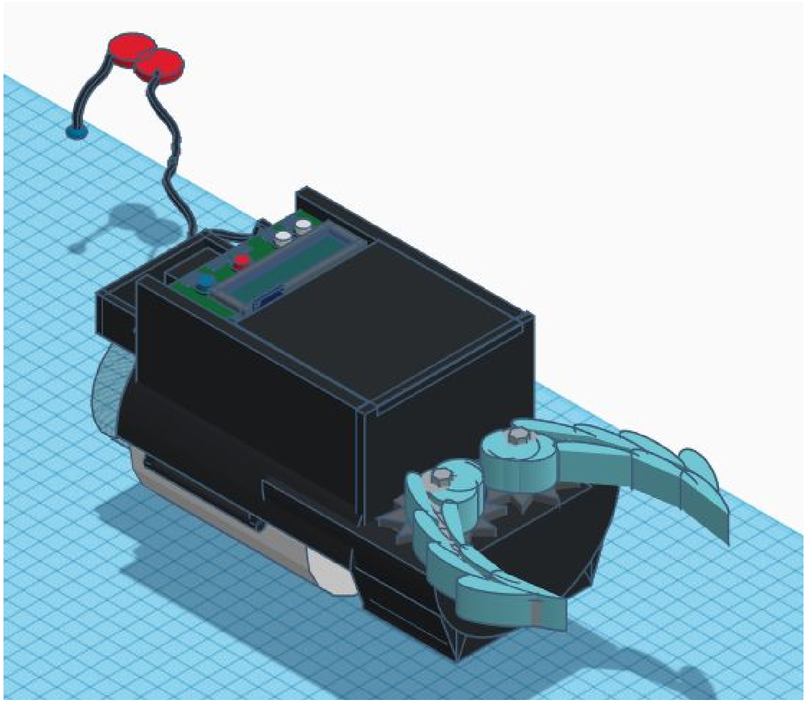


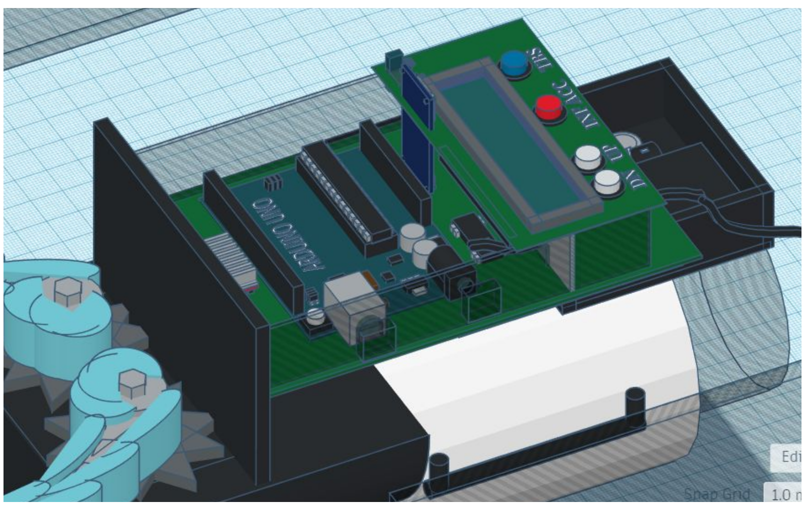
4.1.2

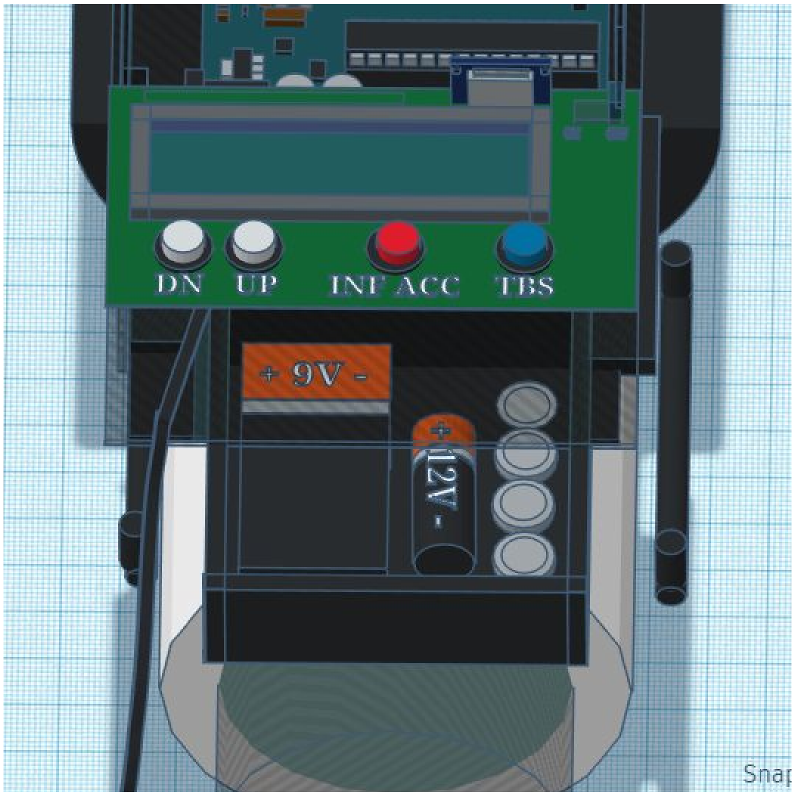
4.1.3 

4.1.4 

4.1.5 

4.1.6 

4.1.7 

4.1.8 


**4.2 Dimensions**

4.2.1 Each claw extends to a square length of 11.2cm, square width of 1.8cm, and thickness of 2cm

4.2.2 The arm socket is 8cm in diameter, 10.5cm long, and 1cm thick

4.2.3 the battery pack is 64 long and 65 cm wide

**4.3 Materials**

4.3.1 All casing (black) is made of PLA

4.3.2 The arm slip (white) is made of polyurethane foam

4.3.3 The claws are made of aluminum alloy 3003

4.3.4 The claws are covered with nitrile rubber

**4.4 Functions**

4.4.1 The claws provide the prosthetic's main action - they close to grip objects

4.4.2 Rubber padding helps maintain friction and electrical isolation

4.4.3 Gears synchronize both claws by using a single motor

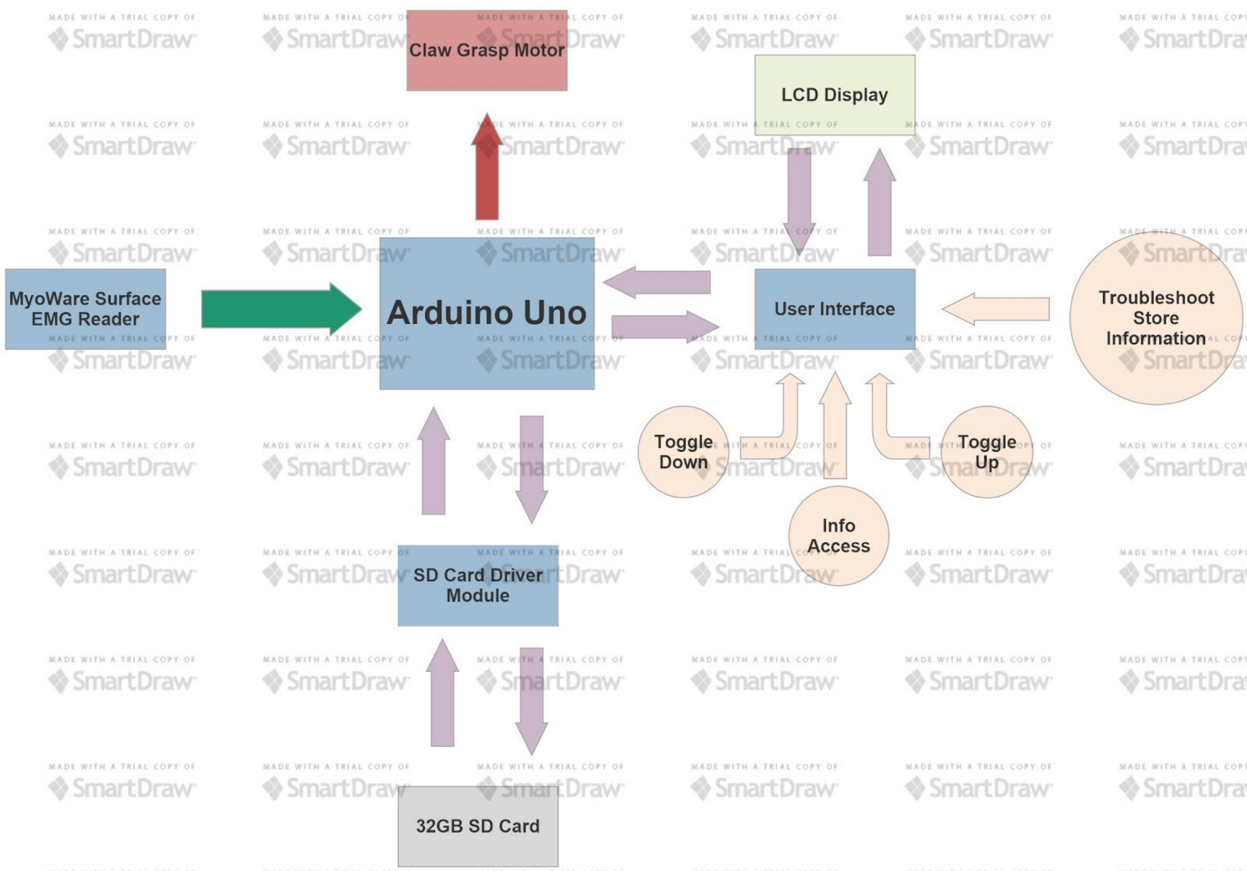
4.4.4 The arm slip provides comfort for the user

4.4.5 Fastened tight to the residual limb with an outer cuff of the patient's choosing

## 5.0 Circuit Design

This section details the overall electronic design, circuitry, and electrical components.

### 5.1 Circuit Overview




5.1.1

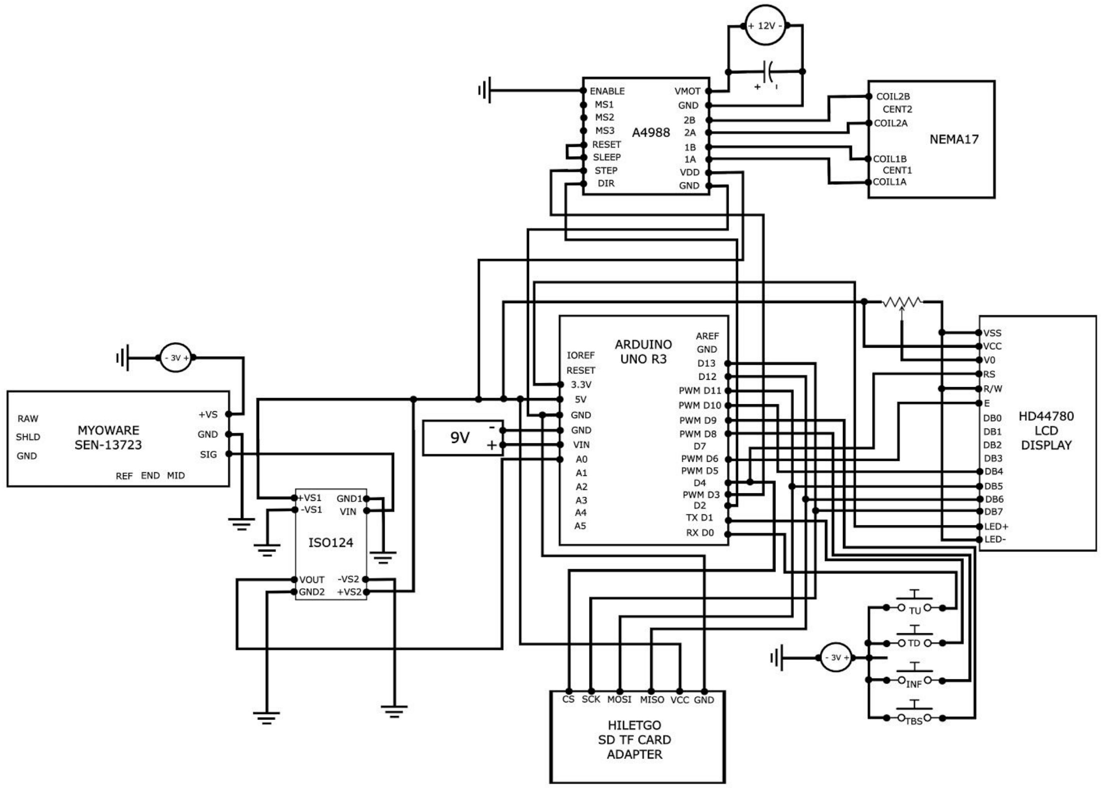
5.1.1.1 The arduino uno microcontroller is the computational centerpiece onto which all components interface

5.1.1.1.1 The Myoware Surface EMG reader provides the signal input from electrical activity of skeletal muscle

5.1.1.1.2 The SD card driver module provides an interface onto which a 32GB SD storage card can be connected and r/w with the arduino

5.1.1.1.3 The user interface is comprised of toggle up and down buttons for selecting grip speed, information access button, a button for storing troubleshoot information, and an LCD screen to display information

### 5.2 Circuit Details and Components




5.2.1

5.2.1.1 MyoWare SEN-13723

5.2.1.1.1 An fully integrated sEMG sensor chosen for its reliability, accuracy, and simple implementation

5.2.1.2 ISO124

5.2.1.2.1 An isolating amplifier used to prevent the overall circuitry from interfering with the EMG signals ready by the Myoware sensor

5.2.1.3 Arduino Uno R3

5.2.1.3.1 A very reliable, cost-effective microcontroller with sufficient i/o pins to drive the design

5.2.1.4 HILETGO SD TF Card Adapter

5.2.1.4.1 An SD adaptor module used for the Arduino. Facilitates reading and writing to memory and is compatible with 32GB Sd Cards

5.2.1.5 HD44780 LCD Display

5.2.1.5.1 An LCD display compatible with the hitachi driver. Chosen for its easy implementation and low cost

5.2.1.6 A4988

5.2.1.6.1 This stepper motor driver was chosen for its ability to control stepper motor rotation speeds and direction. By allowing control of direction, the A4988 cuts down on code complexity and computational power.

5.2.1.7 NEMA17

5.2.1.7.1 A stepper motor. Chosen because it is durable, cost-effective, bipolar, and provides consistent torque.

5.2.1.8 100 uF Capacitor

5.2.1.8.1 Required for decoupling the powerline and driving the NEMA17 motor with the A4988 driver

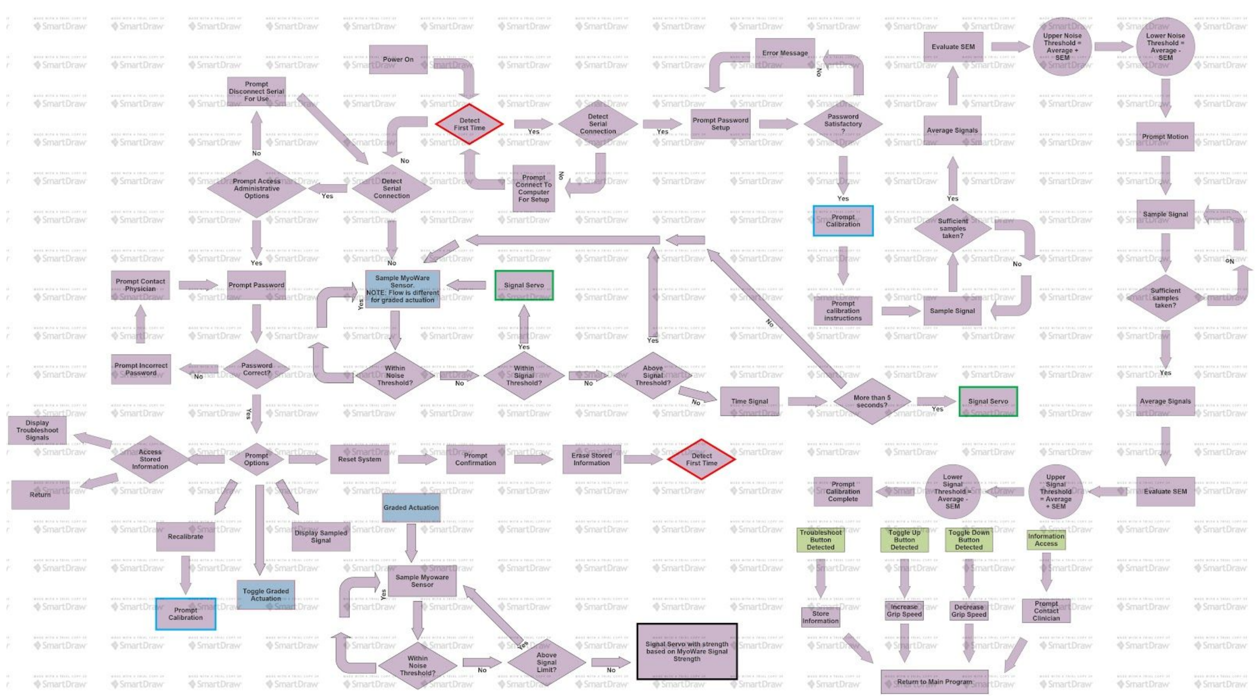
5.2.1.9 Potentiometer

5.2.1.9.1 Allows the user to adjust the LCD display contrast to their liking

# 6.0 Software Design

This section details the software overview used for every step of the device, including startup, calibration, actuation, and troubleshooting. The entirety of the code runs on the Arduino board, however the code also interfaces with the A4988 driver for the NEMA17 motor and HITACHI driver for the LCD display.

## 6.1 Software Overview




6.1.1

6.1.2 The software flowchart steps through every part of the code in detail

6.1.3 For simplicity, elements outlined in matching colors are connected

6.1.4 Buttons provide interrupts which are denoted as green boxes

6.1.4.1 Action by any of these buttons interrupts the main program, executes the interrupt program, and returns to the main program

6.1.4.1.1 Troubleshoot button stores information

6.1.4.1.2 Toggle up and down buttons change grip speed

6.1.4.1.3 Information access requests access to stored information

6.1.5 The software supports two modes, graded actuation and discrete actuation

6.1.5.1 Discreet actuation is detailed in the main section of the flowchart and follows the blue filled MyoSensor sampling step

6.1.5.2 Graded actuation is detailed in the lower flowpath subsequent to the bottom blue filled box

**6.2 Software Libraries**

6.2.1

| **Library** | **Purpose** |
|---|---|
| Stepper.h | Library required for driving the bipolar stepper motor via the A4988 |
| SPI.h<br><br>SD.h | Library allowing communication between the SD card module and arduino<br>Library used by the arduino to read and write to an SD card |
| LiquidCrystal.h | Library used by the arduino to interface with the LCD display |

**6.3 Software Variables**

6.3.1

| **Type** | **Variable** | **Purpose** |
|---|---|---|
| int | MyoSignal[] | holds sampled information from MyoSensor signal |
| int | FirstTime | A flag that is 0 upon first use and is set to 1 following first |

| | | |
|---|---|---|
| | | use. Is used by the software to detect whether to set up a password and immediately move into calibration or provide the display. |
| int | Graded | A flag that is set to 0 should the user choose discrete claw speeds or graded claw speeds. |
| int | MotorAct | A flag that is set to 1 whenever a PWM signal needs to be sent to the A4988 driver. |

## 6.4 Software Functions

### 6.4.1 SD Functions

| | |
|---|---|
| SD.begin(#sspin) | Starts up the SD library and card |
| SD.exists(filename) | Detects the presence of a file on the SD card |
| SD.open(filepath, mode) | Opens the file of interest from the SD card. Mode denotes read or write |
| file.close() | Saves and closes the file being worked with |
| file.available() | Determines if the file is empty |

### 6.4.2 LCD Functions

| | |
|---|---|
| lcd.begin() | Starts up the LCD interface |
| lcd.print() | Prints out text to the LCD |
| lcd.setCursor() | Positions the cursor for the next LCD text output |

### 6.4.3 Signal Processing Functions

| | |
|---|---|
| Serial.begin() | Starts up the communication between the MyoWare sensor and the Arduino |
| double uppernoiselimit(Myosignal) | Determines the upper noise threshold |
| double lowernoiselimit(Myosignal) | Determines the lower noise threshold |

| | |
|---|---|
| double uppersignallimit(Myosignal) | determines the upper signal threshold |
| double lowersignallimit(Myosignal) | determines the lower signal threshold |
| double StandardError(signal*) | Implements the standard error formula<br>$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu_x)^2}{n-1}}$$<br>$$\mathrm{SE}_{\mu_x} = \frac{s}{\sqrt{n}}$$ |

## 7.0 Data Format

7.1.1 Data collected includes the voltage potentials that the myosensor picks up.

7.1.1.1 Raw data is an sEMG signal without filtering

7.1.1.2 The MyoSensor uses integrated circuits for filtering and integrating the raw sEMG

7.1.1.3 The integrated signal is sampled by the Arduino at 3000 Hz

7.1.1.4 Data collected includes sampled signal amplitude and timeframe

7.1.2 Example of sample data.

| Time (secs) | Myosensor Voltage (V) |
|---|---|
| 0 | 40 |
| 1 | 41 |
| 2 | 43 |
| 3 | 43 |
| 4 | 43 |
| 5 | 42 |
| 6 | 41 |
| 7 | 90 |
| 8 | 91 |
| 9 | 41 |
| 10 | 42 |
| 11 | 100 |
| 12 | 105 |
| 13 | 99 |
| 14 | 38 |
| 15 | 39 |
| 16 | 40 |
| 17 | 100 |
| 18 | 101 |
| 19 | 98 |
| 20 | 40 |

# Appendix A

# References


**[1] ISO 8548-3:1993 Prosthetics and orthotics — Limb deficiencies — Part 3: Method of describing upper limb amputation stumps**

**[2] International Electrotechnical Commission (2013). *IEC 60529 - Degrees of protection provided by enclosures (IP Code)*. International standard (2.2 ed.). p. 21. ISBN 9782832210864. OCLC 864643678.**

**[3] CFR Title 21, Part 890, Subpart D, Sec. 890.3500**

**[4] 21 CFR Part 801**

**[5] ANSI/AAMI EC12, “Disposable ECG electrodes”**

**[6] 21 CFR 870.2360**

# Design Review Meeting Minutes

Red Text = Completed

Suggestions: We mentioned the number of jumps or squats. Arbitrary

- What we need to do: Determine if the motions for calibrating the device are sufficient
- How we will do it: Look up myoelectric prosthetic interference problems and how different designs address then. We can then take the issues they found into account and implement it to our design (4/27/2020).

· User specific problem: Clamps moving and stuff.

- The motor moves the clamps, the clamps themselves are solid. This is more of a structural issue since we want the claws to be as sturdy yet lightweight as possible.

· For the display, how it's formatted it could be interpreted as a touch screen.

- What we need to do: Clarify that the display in fact uses buttons
- How we will do it: Provide a different angle of the CAD interface and state that the buttons are physical pressure sensors (4/27/2020).

· Universal symbols to reach a wider demographic.

- What we need to do: Implement universal symbols rather than English abbreviations
- How we will do it: Determine what symbols we need to use and change the display labels accordingly (4/27/2020).

· Look into coating the material with something hydrophobic. Getting something better than just plain exposure.

- What we need to do: Determine how we can coat our device in a cost effective manner. One member mentioned that polyurethane could be used as a clear coat over the device parts that can help to seal the parts and reduce exposure. For claws too probably.

· Our power source is inconvenient

- What we need to do: Implement a different power source that does not require changing six batteries of three different types
- How we will do it: A group member mentioned that there is a power management IC that we can use for power supplies. Look into it and see if we can implement it to our device (4/27/2020).

· Someone was wondering about how often recalibration happens.

○ What we need to do: Define a sort of expected calibration timeline that takes into account changes of intrinsic EMG signals with age.

○ How we will do it: look into how often A patient needs to recalibrate. That is a really good suggestion in terms of getting down what a patient would do. How do we account for EMG changes within one sitting, and periodic changes over time (4/27/2020).

# Code Review Meeting Minutes

Red Text = Completed

Review Partner: Sanju

**Suggestions:**

-Large loops might slow down speed of Arduino

What we need to do: Determine what size of loops in Arduino platform can cause device to slow down.

How we will do it: Research online the baseline performance of Arduino and find its optimal loop size (5/2/2020)

-Recalibration section placement was confusing

What we need to do: Clear up calibration vs. recalibration within the code

How we will do it: Comment at the start of the code the difference between calibration and recalibration (5/2/2020)

-Well commented but possibly clean up code a little

What we need to do: Organization of the code could be better

How we will do it: Group all variables together at start and clearly define the difference between setup and the main program (5/2/2020)

**Section of Code Reviewed: (Calibration Section and Graded Actuation plus Servo Movement)**

# Code

**Code in Bold Text**
Comments in Regular Text

```
/*Four major sections in this code.
  1. Inputs from the myoware get sent to arduino
  2. Arduino checks to see if threshold value is met for myoware sensor.
  3. Arduino outputs to servos
  4. Arduino will continuously display the data to the display screen.
  5. We will have 4 buttons
    -Toggle Up
    -Toggle Down
    -Troubleshooting
    -Accessing Information
*/
/*Setup involves connecting all of the components to the arduino.
  1. Myoware Sensor
  2. Servo
  3. Display screen to output
*/

//include library for display
#include <LiquidCrystal.h>;
//Initialise the LCD  with the arduino. LiquidCrystal(rs, enable, d4, d5, d6, d7)
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
//include library for servos
#include <Servo.h>
//include library for serial
#include <SoftwareSerial.h>

//Set up servo object 1
Servo servo;

//Set up myoware
int myoware = 0;

//Threshold variable to be determined from calibration process
int threshold = 0;
int upperThreshold =0;
int lowerThreshold = 0;

//Password
String password= "";

//position values for servos
int pos_1 = 0;

//delaytime is default 15ms
int delayTime = 15;

//Set up the buttons we will use with their corresponding pins
const int toggleUp = 6;
const int toggleDown = 7;
const int troubleShoot = 8;
const int accessInfo = 13;
```

```
//determine buttonstates for each button
int toggleUpState = 0;
int toggleDownState = 0;
int troubleShootState = 0;
int accessInfoState = 0;

void setup() {
//initialize serial communication at 9600 bits per sec
  Serial.begin(9600);

//attach servos to pins 9
  servo.attach(9);

// initialize the button pins as inputs:
  pinMode(toggleUp, INPUT);
  pinMode(toggleDown, INPUT);
  pinMode(troubleShoot, INPUT);
  pinMode(accessInfo, INPUT);

/////Connected to Computer/Password Setup/////////////////////////////////////////////////
//checks if serial port is connected
if (Serial.available()){

//If password is default, blank, you know it's first time
if (password == ""){
//Prints to the computer screen
Serial.println("Please setup your password");

//Gets the password set up
if(Serial.available()){
        password = Serial.readString();

    }

//Calibration Process happens only on first time use/////////////////////////////////////////////////

  //set up all variables
  int thresholdWalk[] = {};
  int thresholdJump[] = {};
  int thresholdSquat[] = {};
  int thresholdRaise[] = {};
  int thresholdRotate[] = {};
  int thresholdFlex[] = {};

  int thresholdWalkSum = 0;
  int thresholdJumpSum = 0;
  int thresholdSquatSum = 0;
  int thresholdRaiseSum = 0;
  int thresholdRotateSum = 0;
  int thresholdFlexSum = 0;

  int thresholdWalkAvg = 0;
  int thresholdJumpAvg = 0;
  int thresholdSquatAvg = 0;
  int thresholdRaiseAvg = 0;
  int thresholdRotateAvg = 0;
```

```
int thresholdFlexAvg = 0;

int thresholdWalkSE = 0;
int thresholdJumpSE = 0;
int thresholdSquatSE =0;
int thresholdRaiseSE = 0;
int thresholdRotateSE = 0;
int thresholdFlexSE = 0;


int upperThresholdWalk = 0;
int upperThresholdJump = 0;
int upperThresholdSquat = 0;
int upperThresholdRaise = 0;
int upperThresholdRotate = 0;
int upperThresholdFlex = 0;

int lowerThresholdWalk = 0;
int lowerThresholdJump = 0;
int lowerThresholdSquat = 0;
int lowerThresholdRaise = 0;
int lowerThresholdRotate = 0;
int lowerThresholdFlex = 0;



 //Prompt start of calibration process
 Serial.println("Please start the calibration process");

 //This is the normal value myoware sensor detects when completely still
 int normal = analogRead(myoware);

 //First exercise is walking for 30 seconds
 Serial.println("Please walk for 30 seconds");
  unsigned long currentMillis = millis();
  int i =0;
   while((millis() - currentMillis) <=30000){
      thresholdWalk[i]= analogRead(myoware);
      i++;
   }
   for (int i =0; i<sizeof(thresholdWalk);i++){
   thresholdWalkSum = thresholdWalk[i] + thresholdWalkSum;

   }
   //Average
   thresholdWalkAvg = thresholdWalkSum / sizeof(thresholdWalk);
   //SEM
   for (int i =0; i<sizeof(thresholdWalk); i++){

    thresholdWalkSE = sqrt(((thresholdWalk[i] - thresholdWalkAvg)^2)/(sizeof(thresholdWalk)-1));

   }
   //UpperThreshold
    upperThresholdWalk = thresholdWalkAvg + thresholdWalkSE;
   //LowerThreshold
    lowerThresholdWalk = thresholdWalkAvg - thresholdWalkSE;
```

```
//Second exercise is 10 jumps
Serial.println("Please jump 10 times");


for(int i =0; i<10;){
if(analogRead(myoware)-normal >= 20){

  thresholdJump[i] = analogRead(myoware);
  i = i+1;

}
for(int i =0; i<sizeof(thresholdJump);i++){
  thresholdJumpSum = thresholdJumpSum + thresholdJump[i];
}
//Avg
thresholdJumpAvg = thresholdJumpSum/(sizeof(thresholdJump));
}
  //SEM
  for (int i =0; i<sizeof(thresholdJump); i++){

    thresholdJumpSE = sqrt(((thresholdJump[i] - thresholdJumpAvg)^2)/(9));

  }
   //UpperThreshold
    upperThresholdJump = thresholdJumpAvg + thresholdJumpSE;
  //LowerThreshold
    lowerThresholdJump = thresholdJumpAvg - thresholdJumpSE;




//Next exercise is 10 squats
Serial.println("Please squat 10 times");

for(int i =0; i<10;){
if(analogRead(myoware)-normal >= 20){

  thresholdSquat[i] = analogRead(myoware);
  i = i+1;

}
for(int i =0; i<sizeof(thresholdJump);i++){
  thresholdSquatSum = thresholdSquatSum + thresholdSquat[i];
}
//Avg
thresholdSquatAvg = thresholdSquatSum/(sizeof(thresholdSquat));
}
  //SEM
  for (int i =0; i<sizeof(thresholdSquat); i++){

    thresholdSquatSE = sqrt(((thresholdSquat[i] - thresholdSquatAvg)^2)/(9));

  }
  //UpperThreshold
    upperThresholdSquat = thresholdSquatAvg + thresholdSquatSE;
  //LowerThreshold
    lowerThresholdSquat = thresholdSquatAvg - thresholdSquatSE;
```

```
//Next exercise is 10 arm raises
Serial.println("Please raise both arms 10 times");
 for(int i =0; i<10;){
if(analogRead(myoware)-normal >= 20){

  thresholdRaise[i] = analogRead(myoware);
  i = i+1;

}
for(int i =0; i<sizeof(thresholdRaise);i++){
  thresholdRaiseSum = thresholdRaiseSum + thresholdRaise[i];
}
//Avg
thresholdRaiseAvg = thresholdRaiseSum/(sizeof(thresholdRaise));
}
  //SEM
  for (int i =0; i<sizeof(thresholdRaise); i++){

    thresholdRaiseSE = sqrt(((thresholdRaise[i] - thresholdRaiseAvg)^2)/(9));

  }
  //UpperThreshold
    upperThresholdRaise = thresholdRaiseAvg + thresholdRaiseSE;
  //LowerThreshold
    lowerThresholdRaise = thresholdRaiseAvg - thresholdRaiseSE;



//Next exercise is 10 arm rotates
Serial.println("Please rotate both arms 10 times");
 for(int i =0; i<10;){
if(analogRead(myoware)-normal >= 20){

  thresholdRotate[i] = analogRead(myoware);
  i = i+1;

}
for(int i =0; i<sizeof(thresholdRotate);i++){
  thresholdRotateSum = thresholdRotateSum + thresholdRotate[i];
}
//Avg
thresholdRotateAvg = thresholdRotateSum/(sizeof(thresholdRotate));
}
  //SEM
  for (int i =0; i<sizeof(thresholdRotate); i++){

    thresholdRotateSE = sqrt(((thresholdRotate[i] - thresholdRotateAvg)^2)/(9));

  }
  //UpperThreshold
    upperThresholdRotate = thresholdRotateAvg + thresholdRotateSE;
  //LowerThreshold
    lowerThresholdRotate = thresholdRotateAvg - thresholdRotateSE;

//Next exercise is 10 arm flex
```

```
  Serial.println("Please flex arm 10 times");
   for(int i =0; i<10;){
  if(analogRead(myoware)-normal >= 20){

    thresholdFlex[i] = analogRead(myoware);
    i = i+1;

  }
  for(int i =0; i<sizeof(thresholdJump);i++){
    thresholdFlexSum = thresholdFlexSum + thresholdFlex[i];
  }
  //Avg
  thresholdFlexAvg = thresholdFlexSum/(sizeof(thresholdFlex));
  }
    //SEM
    for (int i =0; i<sizeof(thresholdFlex); i++){

      thresholdFlexSE = sqrt(((thresholdFlex[i] - thresholdFlexAvg)^2)/(9));

    }
    //UpperThreshold
      upperThresholdFlex = thresholdFlexAvg + thresholdFlexSE;
    //LowerThreshold
      lowerThresholdFlex = thresholdFlexAvg - thresholdFlexSE;


  //SOLVE FOR THRESHOLD VALUES THAT WILL BE USED IN MAIN PROGRAM
  upperThreshold = (upperThresholdWalk + upperThresholdJump + upperThresholdSquat + upperThresholdRaise +
upperThresholdRotate + upperThresholdFlex)/6;
  lowerThreshold = (lowerThresholdWalk + lowerThresholdJump + lowerThresholdSquat + lowerThresholdRaise +
lowerThresholdRotate + lowerThresholdFlex)/6;
  threshold = (upperThreshold + lowerThreshold) / 2;
  Serial.println("Calibration is complete.");
}

}
//End of Calibration Section/////////////////////////////////////////////////////


}

//End of Set up Section/////


//the main program that continuously loops
void loop() {

//initalize troubleshoot info
int troubleshoot[] = {};
//initalize actuation info
int minstrength =0;
int medstrength =0;
int maxstrength = 0;
int actuation = 0;

//Only happens in main program if serial connected
if (Serial.available()){
```

```
//If the correct password is not given, then the program will not continue
int check =0;


while (check == 0){
//Prompts for actual password
Serial.println("Would you like to access admin operations? If so, please enter password if not, please disconnect serial port");
if (Serial.readString() == password){
  check = 1;
  }

  ///Actual code beings for entering ADMIN OPERATIONS//
  Serial.println("Here are your options: Access stored information, Recalibrate, Reset, Display Sampled Signal, Toggle Graded Actuation");
  Serial.println("Please type exactly the option you would like to choose");

  //Access stored information
  if(Serial.readString() == "Access stored information"){

   for(int i = 0; i<30; i++){
   Serial.print(troubleshoot[i]);
   }

  }


  //To recalibrate, simply the password will be reset so upon restarting the device, it will act like first time
  if(Serial.readString() == "Recalibrate"){
  password = "";
  Serial.println("Please Restart device");
  }

  //Also resets password but also erases all stored info
  if(Serial.readString() == "Reset"){
  password = "";
  Serial.println("Please Restart Device");
  for(int i=0; i<sizeof(troubleshoot); i++){

   troubleshoot[i] =0;
  }

  //Display the signal
  }
  if(Serial.readString() == "Display Sampled Signal"){
  Serial.println(threshold);

  }

  //Toggle Graded Actuation
  if(Serial.readString() == "Toggle Graded Actuation"){

   Serial.println("Flex your arm minimum strength");
   minstrength = analogRead(myoware);
   Serial.println("Flex your arm with medium strength");
   medstrength = analogRead(myoware);
   Serial.println("Flex your arm with maximumum stregnth");
```

```
      maxstrength = analogRead(myoware);

      toggleUpState = 0;
      toggleDownState =0;
       actuation = 1;

    }

  }
}


//Main program that functions without serial plugin
//Initialize myoware sensor
  int myoware= 0;


//check to see if buttons are pushed
// read the state of the pushbutton value:
  toggleUpState = digitalRead(toggleUp);
  toggleDownState = digitalRead(toggleDown);
  troubleShootState = digitalRead(troubleShoot);
  accessInfoState = digitalRead(accessInfo);

  //When graded actuation is turned on
  if (actuation ==1){
    if(analogRead(myoware)>minstrength && analogRead(myoware)<medstrength){
      delayTime = 10;
    }
    if(analogRead(myoware)>medstrength && analogRead(myoware)<maxstrength){
      delayTime = 7.5;
    }
    if(analogRead(myoware)>maxstrength){
      delayTime = 5;
    }
  }

  // check if the toggle up is pressed. If it is, the buttonState is HIGH:
  if (toggleUpState == HIGH) {
    // change delay time to be shorter
    delayTime = 10;
  } else {
    // change delay time to default
    delayTime = 15;
  }

  // check if the toggledown is pressed. If it is, the buttonState is HIGH:
  if (toggleDownState == HIGH) {
    //
    delayTime = 20;
  } else {
    //
    delayTime = 15;
  }

   // check if the troubleshoot is pressed only if Serial port isn't connected. If it is, the buttonState is HIGH:
  if (troubleShoot == HIGH and !Serial) {
```

```
    lcd.print("Please submit errors for troubleshooting");
    //Store your information
    for(int i=0; i<30; i++){
    troubleshoot[i] = analogRead(myoware);
    }
  }

   // check if the accessInfo is pressed only if Serial port isn't connected. If it is, the buttonState is HIGH:
  if (accessInfoState == HIGH and !Serial) {
    //display for info
    lcd.print("Please contact your physician");

  }

//determine if value has met threshold value in myoware sensor
if(myoware < upperThreshold && myoware >lowerThreshold){
//activate servo
  for (pos_1 = 0; pos_1 <= 180; pos_1 += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    servo.write(pos_1);              // tell servo to go to position in variable 'pos'
    delay(delayTime);                       // waits 15ms for the servo to reach the position
  }
} else {

   for (pos_1 = 180; pos_1 >= 0; pos_1 -= 1) { // goes from 180 degrees to 0 degrees
    servo.write(pos_1);              // tell servo to go to position in variable 'pos'
    delay(delayTime);                       // waits 15ms for the servo to reach the position
  }
}
}
```

# Bill of Materials

| Item # | Item | Quantity | Description | Price per unit | Item cost |
|---|---|---|---|---|---|
| 1 | MyoWare Sensor | 1 | Operating voltage (2.9 V - 5.7 V), Raw EMG output | $35 | $ 35.00 |
| 2 | Display Screen | 1 | High Speed MPU Display (upto 2MHz) | $4 | $ 4.00 |
| 3 | Arduino Uno | 1 | 5 V operating Voltage, 32KB flash Memory, Cheap | $15 | $ 15.00 |
| 4 | Servos | 2 | Holding Torque (2000 g-cm), Detend Torque ( 220 g-cm) | $3.50 | $ 7.00 |
| 5 | Buttons | 4 | Cheap, Easy to use | $0.10 | $ 0.40 |
| 6 | Outer Casting | 1.00 | Strong, Easy to design | $1.25 | $ 1.25 |
| 7 | Claws | 2.00 | Alumnium 3003 | $1.00 | $2.00 |
| 8 | Polyurethane Socket | 1.00 | | $0.50 | $ 0.50 |
| 9 | Packaging and Label | 1.00 | | **$0.75** | $ 0.75 |
| 10 | Battery Pack | 2.00 | Rechargable, Less weight | $5 | $10 |
| | | | | Project cost | $ 75.90 |

Table: Materials Used and Current Market Price.

# Supplemental Data

## Pugh Matrix for Number of Sensors

| Criteria | Weight of Importance | One Sensor | Two Sensors |
|---|---|---|---|
| Effectiveness | 5 | 3 | 4 |
| Long Term Benefit | 4 | 3 | 3 |
| Time to Implement | 3 | 4 | 3 |
| Ease to Implement | 2 | 4 | 3 |
| Cost to Implement | 4 | 3 | 1 |
| | | | |
| | Total | 59 | 51 |

At a point during the design process, our group had to decide whether to include two myoware sensors or one. With only one sensor, an additional button would need to be added to toggle between grasping and rotation. With two sensors, the biggest problem was cost and electrode placement. There was no ideal spot for a second muscle sensor. In the end, there were not enough pins on the arduino for two sensors so only one was used. However, this is the Pugh matrix surrounding that decision.

## Pugh Matrix for Casing Material

| Criteria | Weight of Importance | PLA | Stainless Steel | Ceramics |
|---|---|---|---|---|
| Effectiveness | 5 | 5 | 3 | 3 |
| Long Term Benefit | 4 | 3 | 3 | 2 |
| Time to Implement | 3 | 4 | 3 | 5 |
| Ease to Implement | 2 | 3 | 2 | 3 |
| Cost to Implement | 4 | 4 | 2 | 3 |
| | | | | |
| | Total | 71 | 48 | 56 |

Another critical design decision was the casing material. There were many options for casing materials but we narrowed it down to 3 and used a Pugh matrix to tabulate a score. The biggest problem with stainless steel was the cost and weight and for ceramics, it was too brittle. Thus, we ended up choosing PLA as the material for our final casing.