ENSC 405W Grading Rubric for Design Specification

Criteria	Details	Marks
Introduction/Background	Introduces basic purpose of the project.	**/05%**
Content	Document explains the design specifications with appropriate justification for the design approach chosen. Includes descriptions of the physics (or chemistry, biology, geology, meteorology, etc.) underlying the choices.	**/20%**
Technical Correctness	Ideas presented represent design specifications that are expected to be met. Specifications are presented using tables, graphs, and figures where possible (rather than over-reliance upon text). Equations and graphs are used to back up/illustrate the science/engineering underlying the design decisions.	**/20%**
Process Details	Specification distinguishes between design details for present project version and later stages of project (i.e., proof-of-concept, prototype, and production versions). Numbering of design specs matches up with numbering for requirements specs (as necessary and possible).	**/15%**
Test Plan Appendix	Provides a test plan demonstrating the fulfilment of the requirements for the final project version (in an appendix). Note that project success will be measured against this test plan (i.e., does the device/system do what it is supposed to do?)	**/10%**
User Interface Appendix	Summarizes requirements for the User Interface (based upon the lectures and the concepts outlined in the Donald Norman textbook).	**Graded Separately**
Conclusion/References	Summarizes functionality. Includes references for information sources.	**/05%**
Presentation/Organization	Document looks like a professional specification. Ideas follow in a logical manner.	**/05%**
Format Issues	Includes letter of transmittal, title page, abstract, table of contents, list of figures and tables, glossary, and references. Pages are numbered, figures and tables are introduced, headings are numbered, etc. References and citations are properly formatted.	**/10%**
Correctness/Style	Correct spelling, grammar, and punctuation. Style is clear, concise, and coherent. Uses passive voice judiciously.	**/10%**
Comments		

Design Specification

for the Handsome 1.0 Prosthetic Hand

Handsome Technology

Project Team	Bairong Li Cyrus Chan Sarah Huang Yi Luo
Contact Person	Bairong Li bairongl@sfu.ca

Date	July 24, 2017

School of Engineering Science
Simon Fraser University Burnaby
British Columbia V5A 1S6

July 24, 2017

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University Burnaby
British Columbia V5A 1S6

Re: Design Specification for Handsome 1.0

Dear Dr. Rawicz,

The attached document, design specification for Handsome 1.0, outlines the detail of design specification for our ENSC 440 (Capstone Engineering Science Project). Our primary goal is to research the approach of a prosthetic forearm for those who work in environments with safety concerns. Therefore, they can be protected from the contamination and the risks while performing operations with their hands.

The design specification will provide specification for a high-level design for the proof-of-concept device of Handsome 1.0 and review the specifications on each design scheme. Design speculation of Handsome 1.0 is examined in detail for both its software and hardware components. This document includes a variety of methods testing plan, which will serve as a comprehensive reference during the development phase of our product.

Handsome Technology Inc consists of four innovative and intelligent engineering students: Yi Luo, Sarah Huang, Bairong Li, Cyrus Chan. If you have any questions or concerns about our requirement specification, please feel free to contact me by phone at 778-919-9133 or by email at bairongl@sfu.ca.

Sincerely,

CEO
Handsome Technology Inc

Enclosure: Requirements Specification for Handsome 1.0

Table of Contents

List of Figures iv
List of Tables v
Glossary vi
1 Abstract 1
1.1 Introduction 1
1.2 Scope 1
1.3 Intended Audience 2
2 Prosthetic Hand 2
2.1 Physical and Mechanical Design 3
2.2 Electrical Design 4
Motor 4
Voltage Regulator 5
Power Source 6
Circuit Design 6
Microcontroller 7
2.3 Software Design 8
3 Sensor Kit 8
3.1 Hardware design 9
3.2 Software design 10
Signal classification 11
Gesture Prediction 12
4 Wireless Communication 13
5 Conclusions 14
Appendix 1: System Test Plan 14
Component Test 14
Table, EMG sensor test cases 15
Robotic forearm test 15
Servo motor test 17
BLE test 18

Li-Po battery test 21
Raspberry Pi test: 21
Integrated Test 21
Appendix 2: User Interface 23
Introduction 23
User Analysis 23
Technical Analysis 24
Discoverability 24
Feedback 24
Conceptual models 25
Affordances 25
Signifiers 25
Mappings 25
Constraints 26
Engineering Standards 27
Analytical Usability Testing 27
Empirical Usability Testing 29
Natural Movement Test 29
Skin cleaning Test 29
Ease of Use Test: 30
Conclusion 30
Reference 32

List of Figures

Figure 1, Six Hand Gestures 3
Figure 2, forearm [1] Figure 3, palm 3
Figure 4, MG996R Servo Motor [2] 5
Figure 5, 5V 1A Voltage Regulator [7] 6
Figure 6, Li-po Battery 7.4V 1600maAh [9] 6
Figure 7, Circuit Schematic for the Proof-of-Concept Product 7
Figure 8, Circuit Schematic for the Final Product 7
Figure 9, Prothestic hand software 8
Figure 10, components of the sensor kit 9
Figure 11, sensor kit hardware 10
Figure 12, sensor kit software 11
Figure 13, example signal classification 12
Figure 14, Bluetooth Module [15] 13
Figure 15, Xbee module 13
Figure 16, Example Sensor Location [13] 15
Figure 17, 3D print hand [14] 16
Figure 18, Hand wiring [14] 16
Figure 19, Servo motor connects to Arduino [15] 17
Figure 20, BLE connects to Arduino [16] 18
Figure 21, BLE control LED light [17] 19
Figure 22, AT command sketch [18] 19
Figure 23, AT command result [18] 20
Figure 24, button and LEDs mapping of the sensor kit 26
Figure 25, EMG sensor locations 26

List of Tables

Table 1, Comparison of Stepper Motor and Servo Motor [4] & [5] 4
Table 2, Number of Motors 5
Table 3, Number of Voltage Regulators 5
Table 4, Comparison of Arduino Mega and Nano [10] & [11] 7
Table 5, BLE Vs. Xbee 13
Table 6, EMG sensor test cases 14
Table 7, Robotic forearm test cases 15
Table 8, Servo motor test cases 17
Table 9, BLE test cases 18
Table 10, Li-Po battery test cases 21
Table 11, Raspberry Pi testcases 21
Table 12, Discoverable elements 24
Table 13, Feedbacks 24
Table 14, Standards 27
Table 15, Natural movement test 29
Table 16, Skin cleaning test 30
Table 17, Ease of use test 30

Glossary

BLE	Bluetooth Low Energy
EMG	Surface Electromyography
Li-po battery	Lithium polymer battery
IEEE	Electrical and Electronics Engineers
ISO	International Organization for Standardization
InMoov	The first Open Source 3D printed life-size robot
RoHS	Restriction of Hazardous Substances
UI	User interface
Xbee	Form factor compatible radio
Home position	Palms up， all fingers straightened
Prosthetic hand	The robotic hand that partially realize human hand functionality or behaviour

1 Abstract

Imagine yourself as a born maker and tinkerer, who loves engineering and playing basketball. How would losing your hand one day affect your life? For those who work in an environment with potential risks such as builders, roofers, and electric welders, one careless decision made at work will lead to loss of fingers or hands. We came up the approach to design multifunctional and adaptive prosthetic hands that take place in human hand to perform tasks. There is a variety of prosthetic limbs existing in the market. They are primarily designed to function as a person's replacement limb. However, we believe rather than replacing people's missing limbs with artificial limbs, we can protect them from losing their own limbs.

Motivated by this thought, our company designs and manufactures prosthetic forearm named Handsome 1.0, which protects those who work in environments with safety concerns from potential risks while being able to perform operations as using their hands. Due the budget and time constraints, a Handsome 1.0 hand will not be dexterous enough to realize the behaviour of a human hand. However, it will be able to pose several hand gestures, wrist gestures, and grab objects with some constraints. The details of the implementation will be covered in this document.

This document outlines the details of the design specification and provides analysis of the various aspects of our product. The objective of this document is to explain our design approach and describe our current and future implementation of Handsome 1.0. The specifications of each component will also demonstrate that all functional requirements are met according to the requirement specifications document. In addition, this document includes the test plan that includes several test cases that will examine the functionality of each component.

1.1 Introduction

Handsome 1.0 is a prosthetic hand is capable of grabbing objects with certain size and weights and performs some basic hand gestures. To use Handsome 1.0 prosthetic hand, a user needs to attach surface Electromyography (EMG) sensor electrodes, as well as the gravity sensors to specific locations on people's forearm and wrist. The sensors will analyze the signals collected while the user is moving people's hand, and sends instructions to the robotic hand to change its gestures. The force sensitive resistors on the fingertips of the prosthetic hand will ensure the object is held when grabbing objects. The design details of Handsome 1.0 will be covered in this document. Along with the specifications, a detail system test plan will be included in this document.

1.2 Scope

This document discusses the design and implementations of Handsome 1.0 prosthetic hand of each distinct part. The design specification includes are primarily based on the proof-of-concept product with brief discussion on possible improvement on the final product. Some features listed describing on the requirements specification will not covered. This document also lists the test plans designed for our proof-of –concept product.

1.3 Intended Audience

The design specification is intended for all members in the developing team of Handsome 1.0 project. Design details listed in the documentation are used to arrange and distribute tasks to each member through the development. All the members shall keep in mind with the requirements listed to design, test and to assess the completeness of their tasks. Test engineers shall use this document to implement the test plan and to confirm the correct behaviour of the Handsome 1.0.

2 Prosthetic Hand

This section provides the overall design of the prosthetic hand. The prosthetic hand model consists of the palm, five fingers and the forearm. The model has pre-designed slots to hold six motors that control the motion of the hands. Each finger is controlled by a string connecting through the joints the motor. The prosthetic hand receives the encoded gesture information from the sensor kit and moves the motors certain degrees to reproduce the gesture. All the gestures are predefined and stored in a look up table; the hand only needs to decode the information from the Sensor Kit and search for a match in the look up table. Following six gestures excluding the home position are named and registered in the prof-of-concept Product:

- Cylindrical grasp
- Spherical grasp
- Thumb
- Middle
- Index
- Bent-thumb

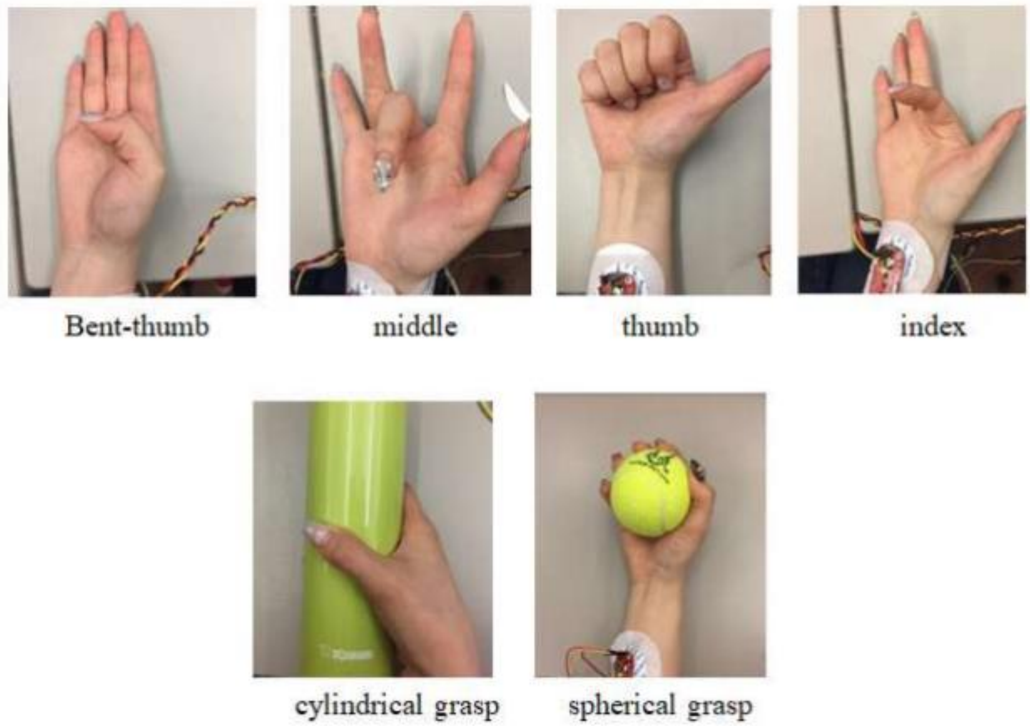

Figure 1, Six Hand Gestures

2.1 Physical and Mechanical Design

We 3D printed and assembled a robotic hand and forearm using the design of InMoov robots. Figure 2 and 3 show the overall mechanical design of the hand model.

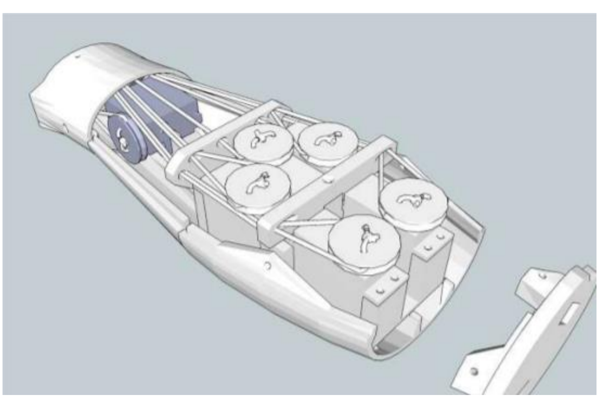

Figure 2, forearm [1]

Figure 3, palm

There are five fishing strings acrossing each finger to be pulled for finger motions. Taking the index finger as an example, when you pull the front side of the fishing line, the index finger will bend towards the direction of the palm. Also, if you pull the string to the other side, the index finger will go back to its original position. We can see in Figure 3, five motors are placed inside the forearm controls. The wrist motor will be presented for the prof-of-concept product but will be added to the final product to produce the wrist motion.

2.2 Electrical Design

Motor

Due to the limited space inside the forearm, we need to choose a motor which is small enough to fit in the diameter of the forearm. On the other hand, it needs to have stall torque as large as possible for grabbing objects. We have two choices for the motor, one is stepper motor and the other one is servo motor. Table 2 shows the advantages and disadvantages of these two options. We want to reduce the delay as much as possible of our system. Motors operate at high speed are better for our design. Stepper motors need external driver to be controlled. Therefore, it requires more space inside the forearm. According to these two considerations, we decided to choose to use servo motors. The best type of servo motors among all the servo motors is high-torque MG996R servo motor [2]. Its maximum stall torque is 12kg/cm at 6V. This type of motor can rotate ±90 degrees from rest [3]. Table 1 shows the numbers of motors that will be installed in the proof-of-concept product and the final product.

	Advantages	**Disadvantages**
Stepper Motor	• Less Expensive • High torque at low speed • Easier drive control	• Draw more power from power source • Need motor driver • A loss of 80% of the rated torque at 90% of the maximum speed
Servo Motor	• High torque at high speed • More efficient • Less power draws • Available in AC and DC drive	• More Expensive • Require gear boxes at low speed operation

Table 1, Comparison of Stepper Motor and Servo Motor [4] & [5]

	Numbers of Servo Motors
Prof-of-Concept Product	5
Final Product	6

Table 2, Number of Motors

Figure 4, MG996R Servo Motor [2]

Voltage Regulator

The minimum operating current for MG996R servo motor is 500mA and the minimum operating voltage is 4.8 V. During the test, one Arduino Mega or Uno is able to power two motors for maximum, so we will need external power source to power motors. The selection of the power source will be explained in the following section. In order to provide stable a 500mA current source for the motors, a voltage regulator with 5V output voltage and 1A output current should be considered. Thus, one voltage regulator will be connected to two paralleled servo motors, so that both motors will be operating at 500mA input current. However, capacitors will be needed to low down the noise of the circuit. As for DC input, 100nF capacitor is enough [6] and we chose 10uF capacitor in the final circuit. Table 2 shows the number of voltage regulators that will be used in the proof-of-concept product and the final product.

	Numbers of Regulators
Prof-of-Concept Product	2
Final Product	3

Table 3, Number of Voltage Regulators

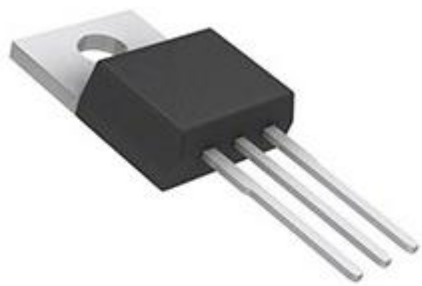

Figure 5, 5V 1A Voltage Regulator [7]

Power Source

For the proof-of-concept product, we decided to use the protabale power source to power the whole prosthtic hand model, because it is more convinient and less time consuming for designing and prototyping the proof-of-concept product. Disposable batteries are not proper for our design because the MG996R servo motors need relative large amount of current to operate, we do not expect the user to replace the batterier too frequently during the operation of the system. Therefore, we chose rechargeable batteries to power all the motors. Li-po batteries will be a good choice. Since the 5V regulator will need at least 7V input voltage to be operating[8], a 7.4V Li-Po battery is chosen.

Figure 6, Li-po Battery 7.4V 1600maAh [9]

Circuit Design

Figure 5 and 6 shows the implementation circuit of the system for the proof-of-concept product and the final product. Since one regulator needs to be connected with two motors, the fifth motor will be powered by the Arduino board.

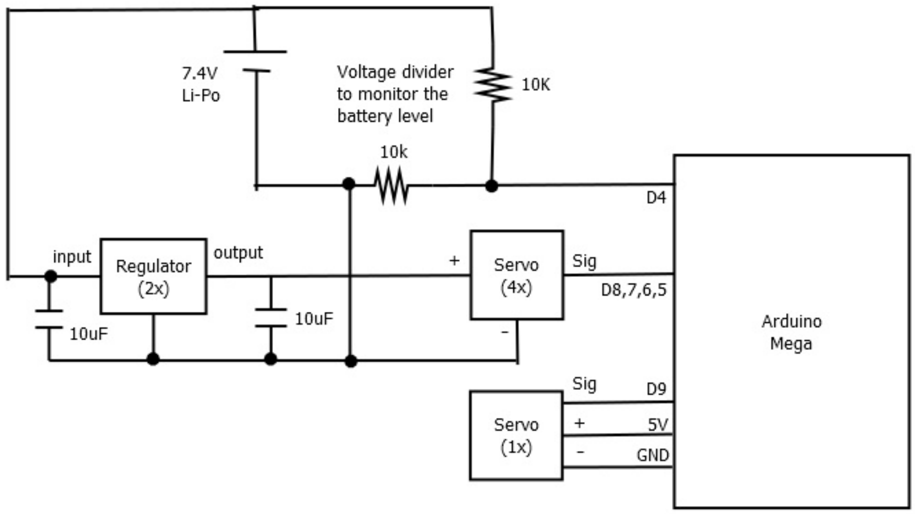

Figure 7, Circuit Schematic for the Proof-of-Concept Product

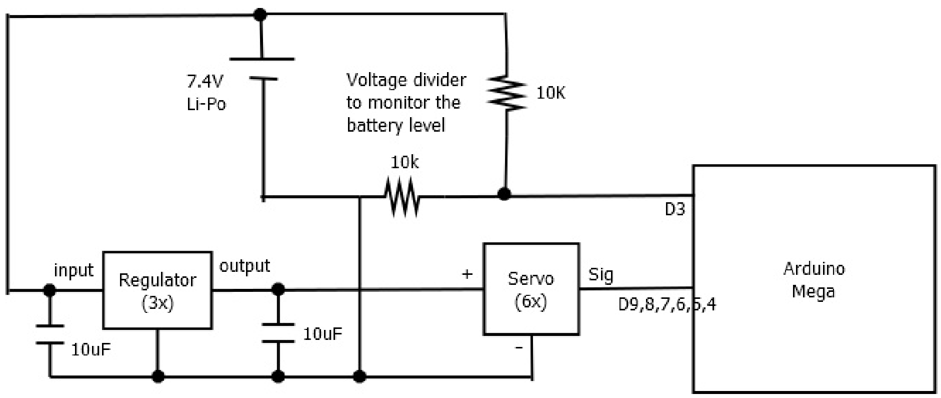

Figure 8, Circuit Schematic for the Final Product

Microcontroller

Our system will be controlled by Arduino Microcontroller. For the final product, we will need a total of 9 digital pins, two for BLE's TX and RX, six for six motors and one for voltage divider. A Mega will be used in the proof-of-concept product because it is convenient for developing and debugging the system. A Nano will be used in the final product to optimize the size of the product. Nano has enough digital pins to support our system.

Arduino Microcontroller	Mega (Prof-of-Concept Product)	Nano (Final Product)
Input Voltage	7-12V	7-12V
Digital I/O Pins	54	22
Analog Input Pins	16	8
Size	101.52x53.3mm	18x45mm

Table 4, Comparison of Arduino Mega and Nano [10] & [11]

2.3 Software Design

The platform for the controller application is Arduino Mega. The following figure shows the work flow of the software.

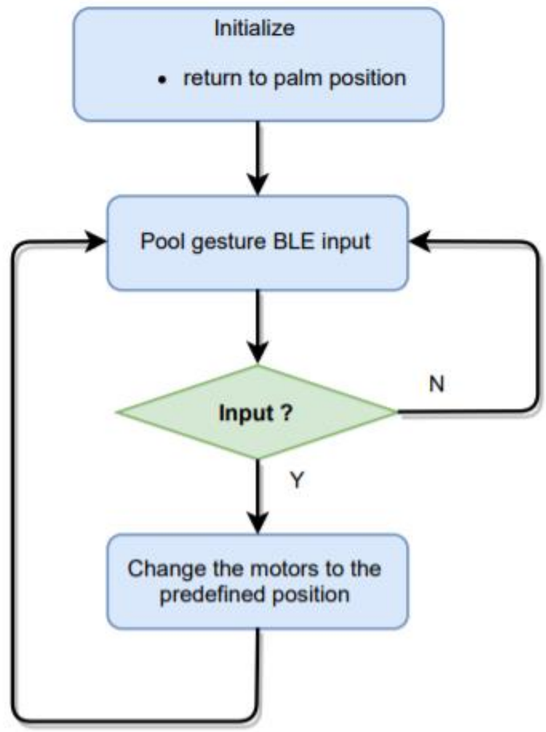

Figure 9, Prosthetic hand software

3 Sensor Kit

The Handsome 1.0 Prosthetic device comes with a sensor kit which collects and processes biological signals from the user. The following graph shows the components and the internal structures of the sensor kit:

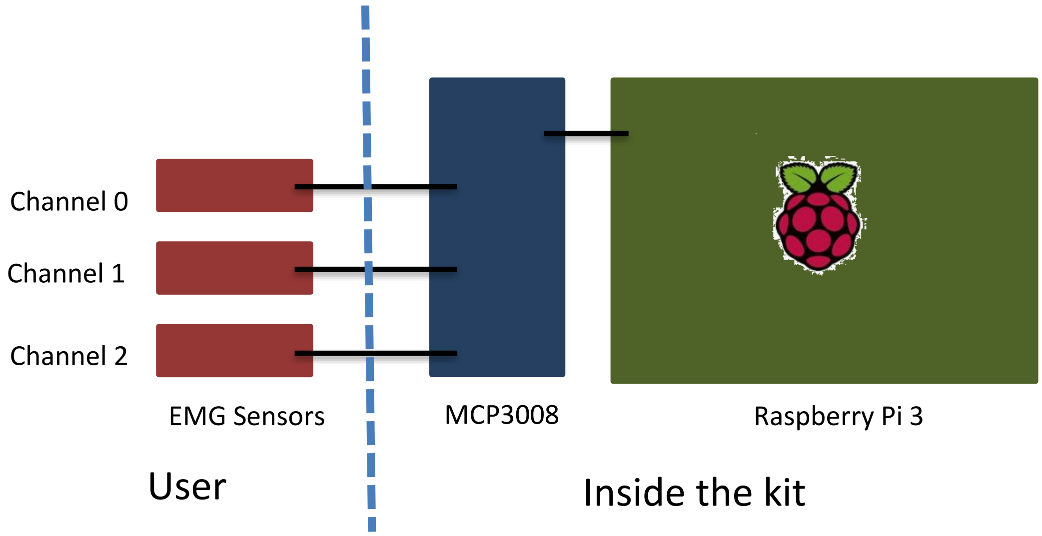

Figure 10, components of the sensor kit

As above graph shows, the system contains the following component:

- surface EMG sensors – collect signals to determine user's finger gestures
- MCP3008 DAC – converts analog signals collected by the sensors to digital signal
- Raspberry pi – classifies the EMG signal collected, predicts the user gestures, and communicates with the prosthetic arm.

The rest of this section will discuss the design and the implementations of the Handsome 1.0 sensor kit in details.

3.1 Hardware design

The hardware design of the proof-of-concept Handsome 1.0 sensor kit will be demonstrated in the following figure:

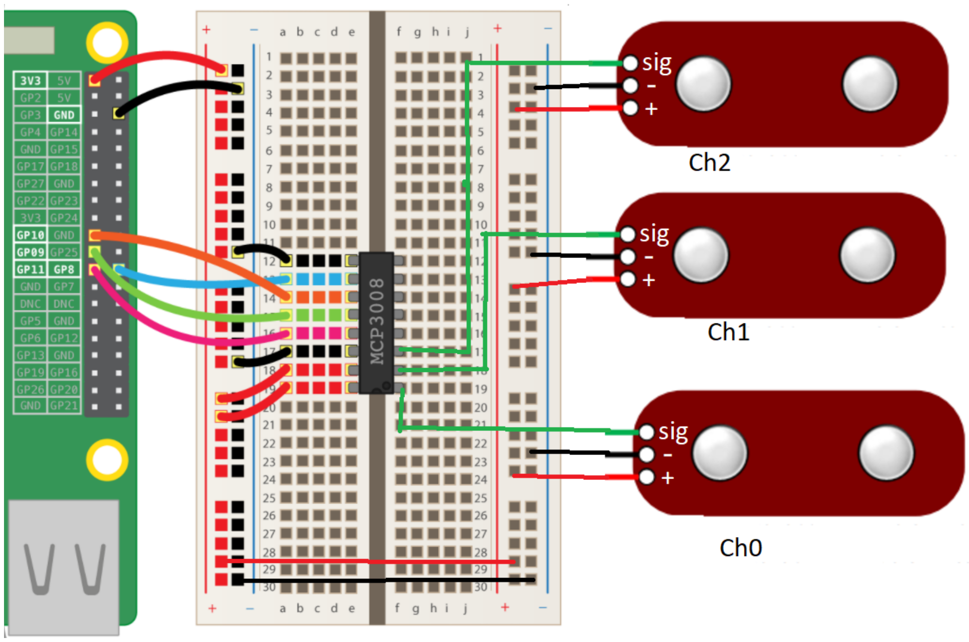

Figure 11, sensor kit hardware

We chose to use Raspberry Pi3 as our main platform not only because the board itself is affordable, the software running on Raspberry Pi 3 are either free or open source. Most importantly, the processing power for Raspberry Pi 3 is good enough for demonstrating purposes.

The Myoware sensors are powered by the raspberry pi at 3.3V, each channel outputs an analog signal indicates the activity level at the certain location. The signals need to be converted to digital signal for Raspberry Pi3 inputs.

3.2 Software design

The platform of the Handsome 1.0 Sensor Kit software is Raspberry Pi 3. In our proof-of-concept design, we used python3 to implement our software due to the Python's advantage of short developing type. The Myoware sensor contains hardware filters and rectifiers to pre-process the EMG data, therefore our software can use the signal directly from the sensor without doing pre-processing. The main work flow of our software application is calibration, signal classification, and gesture prediction.

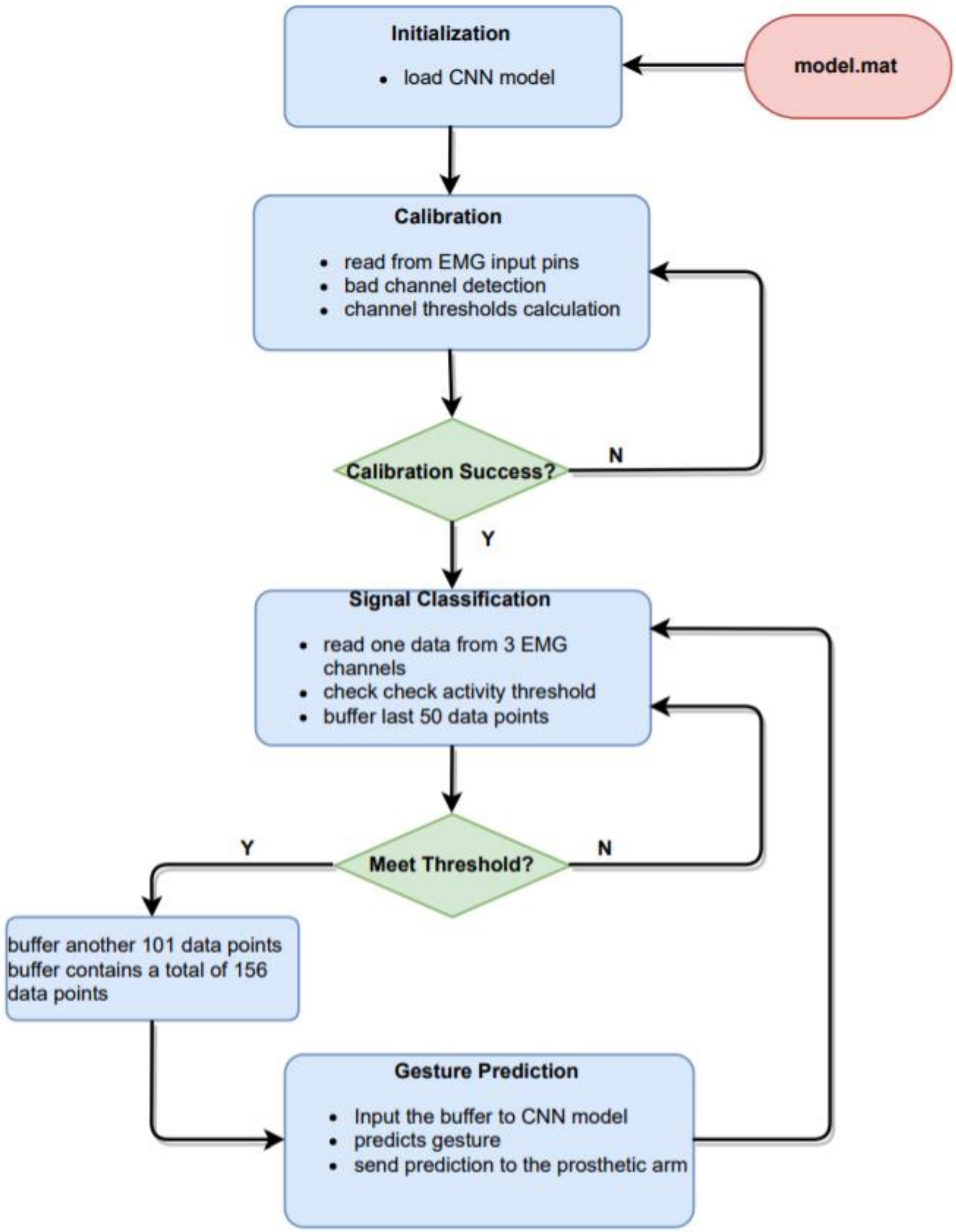

Figure 12, sensor kit software

Signal classification

Signal classification in our implementation defines a procedure of separating the sections of signals that corresponds to the movements of the user's hand. For example, the below graph is a section of three-channel EMG signals revealing a test user changing poses of his hand. The peaks in the plots indicate the movements of the user's hand. Most of information is contained in the region with activities.

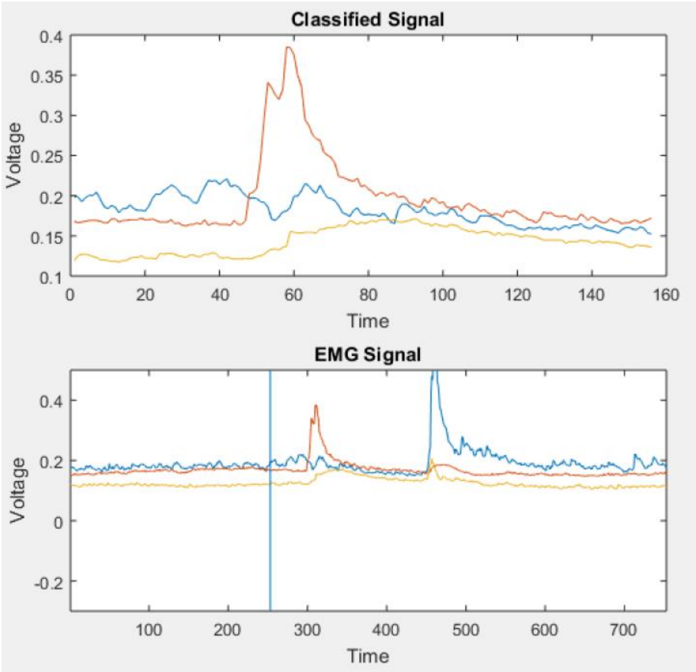

Figure 13, example signal classification

We achieved the functionality of signal classification by first sampling certain length of signals while the user is keeping his hand static at calibration stage, and calculating the thresholds on the values of data points, as well as thresholds measuring the changes in data within certain timeframe. As soon as a data point reaches the threshold, the application takes 50 data points before the detected point and 105 points after the detected points. The 156 data points will be input the CNN model for the prediction.

Gesture Prediction

We use a Convolutional neutral network model to analyze the pattern of the classified data. A Large number is needed to train the CNN model. Due to the development time, we have collected 100 samples for each of the six gestures from our group member. We will need more samples from different test subjects to improve the accuracy of our gesture prediction for the final product of Handsome 1.0.

Our model is trained using Matlab with Matcovnet [12] library. We designed 17 layers in the model. We built 4 dropout layers to avoid the problem of over-fitting. Through testing, we found the accuracy of the CNN model is approximately 95%, which is acceptable for our demonstration.

4 Wireless Communication

For our proof-of-concept model, we are going to use Bluetooth Module HM-10 (Figure 11). HM-10 is a BLE module. The advantage of this module is its small size, low cost and low energy consumption. It also complies with the EIA Standards and has a data rate up to 1 Mbit/s [10]. With these features, it satisfies our BLE module requirements.

Figure 14, Bluetooth Module [15]

Figure 15, Xbee module

For future designs, the use of Xbee module or other BLE module will be considered. This is because the current BLE module (HM-10) may have difficulties to keep a stable wireless connection between Arduino board and Raspberry Pi while performing real time data transmission. By using Xbee module in both Arduino board and Raspberry Pi, we will have a longer distance connection between them, which means that the users can have a better remote-control experience. In addition, Xbee module will have a lower power consumption compared to BLE module, which shown in Figure A.

	BLE module	**XBee module**
Distance	75-100m line of sight	>10m >(33ft)
Latency with Connect	15ms	3-6ms
Power Consumed	30mW	0.01-0.5W
Throughput	0.03Mbps	0.27Mbps
Operating Band	2.400 Ghz-2.4835 GHz ISM band 16 channels, 5MHz apart, 2MHz used Direct Spread Spectrum	2.400 Ghz-2.4835 GHz ISM band 40 2-MHz channels Frequency Spread Spectrum
IEEE Standard	802.15.4	802.15.1

Table 5, BLE Vs. Xbee

5 Conclusions

Handsome 1.0 is a prosthetic hand is capable to perform some basic gestures and to grab objects with certain size and weights. The prosthetic hand and the sensor kit can be remotely connected via Bluetooth. The design meets functional specification of Handsome 1.0 proof-of-concept product. During the development of the production device, the designs will faithfully adhere to meet the functional specification. Through the test plans included in the design specifications, we can ensure that the device is functioning properly.

Appendix 1: System Test Plan

There are two major parts in the test plan. The first part is the individual component testing. The second parts are focusing on integrated testing which involves checking if the device is able to function as expected.

Component Test

This unit test is necessary to make sure that individual component is working properly based on our requirement specifications. Manual testing will be done on the components described below.

The hardware test focuses on individual component in our system. An Arduino program will be written for every single module.

MyoWare Muscle Sensor (EMG) Test:

An Arduino program will utilize the EMG sensor to collect a person's forearm muscle single while the person moving his finger. The signal will be displayed on the screen.

Testing Method	Expected Outcome
Turn on the EMG sensor module and execute the 'Blink LED light' Arduino program.	The LED light of the EMG sensor module will keep blinking.
Execute an Arduino program to read the EMG sensor data. Connect the input channel 1, channel 2 and reference pins together.	The output voltage should be 0V shown on the screen.
Execute an Arduino program to read the EMG sensor data. Set up the EMG sensor shown as Figure 1. Make different gestures while collecting data.	We will see a dramatical change in the EMG signal shown on the screen [13].

Table 6, EMG sensor test cases

Table, EMG sensor test cases

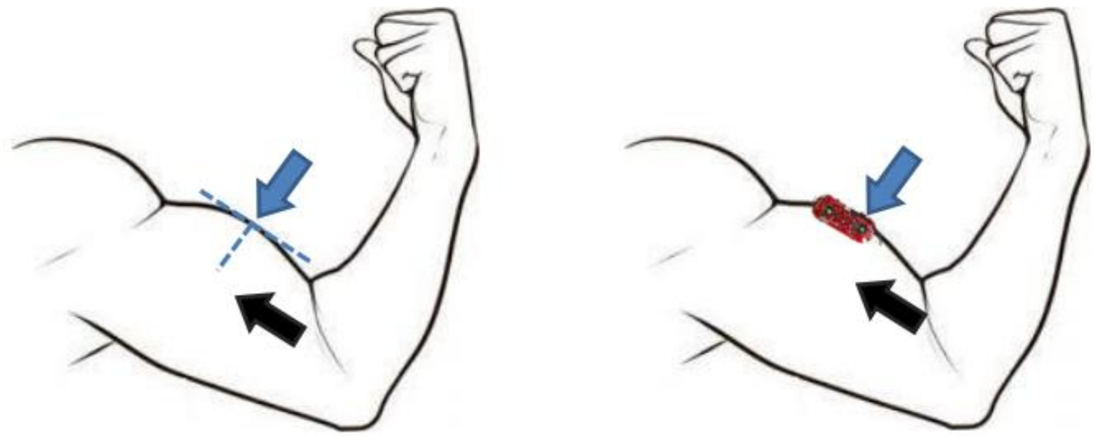

Figure 16, Example Sensor Location [13]

Robotic forearm test

The Robotic forearm is a 3D print item. It will be assembled by our engineers. In order to confirm that a part is adjusted properly, we plan to test each adjustable part separately.

Testing Method	Expected Outcome
Assembled the hand as shown in Figure 2. Rotate each joint that is marked in red at least 50 times.	The entire 3mm Plastic filament inside the joint in red will not come out or broken. In addition, there is no significant friction while rotating the joint.
Pull each fishing line shown in Figure 3, and keep pulling for 1 minute.	The fishing line will not break.
Pull each pulling line and releasing line shown in Figure 3. Do this motion for 20 times for each finger.	Each finger can bend up while pulling the pulling line; Each finger will lie down while pulling the releasing line.
Pull all the pulling line in the same time.	All the fingers will bend up, and there is no collision between each other.
Hold the hand vertically and shake it slightly.	The parts will not come out from the 3D print hand.
Shake the whole forearm	All parts will not come out from the 3D print forearm

Table 7, Robotic forearm test cases

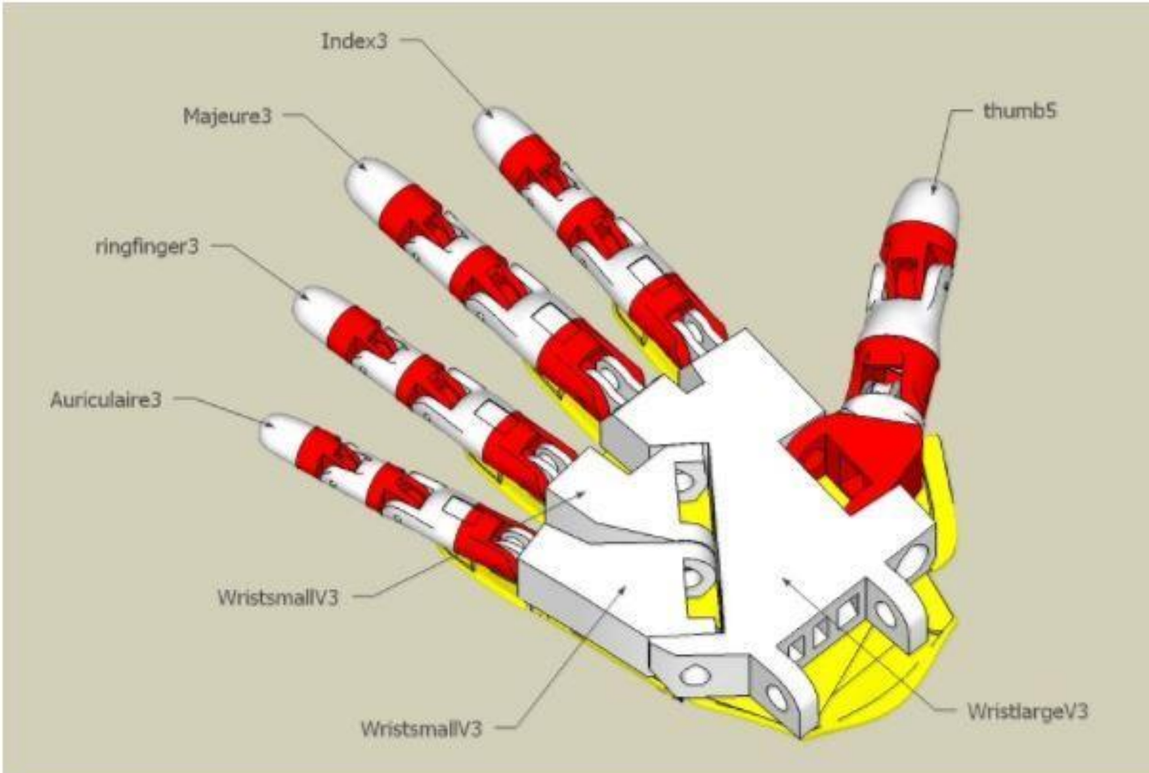

Figure 17, 3D print hand [14]

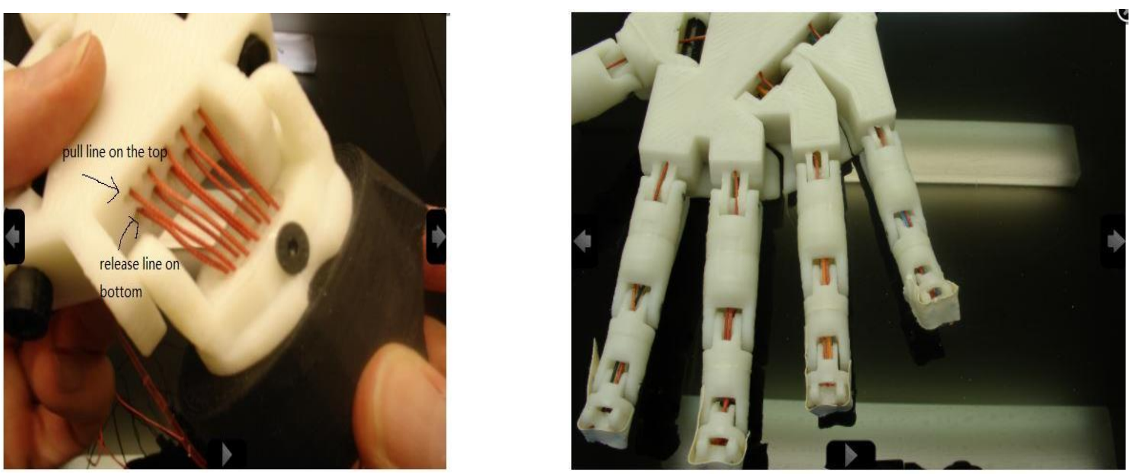

Figure 18, Hand wiring [14]

Servo motor test

There five servo motors are using to control the fingers. One motor will be connected to one finger. Each servo motor will be tested with Arduino board first. If the test succeeds, the servo motor will have a second test that is connecting to both pulling line and releasing line to control a finger.

Testing Method	Expected Outcome
Connect one servo motor with Arduino and use 3V to power the servo motor	The servo motor will not work
Connect one servo motor with Arduino as shown in Figure 4, and execute an Arduino program to control the servo motor	The servo motor is functioning as expected
Connect all servo motor with Arduino and execute an Arduino program	All the servo motors will rotate in the same time

Table 8, Servo motor test cases

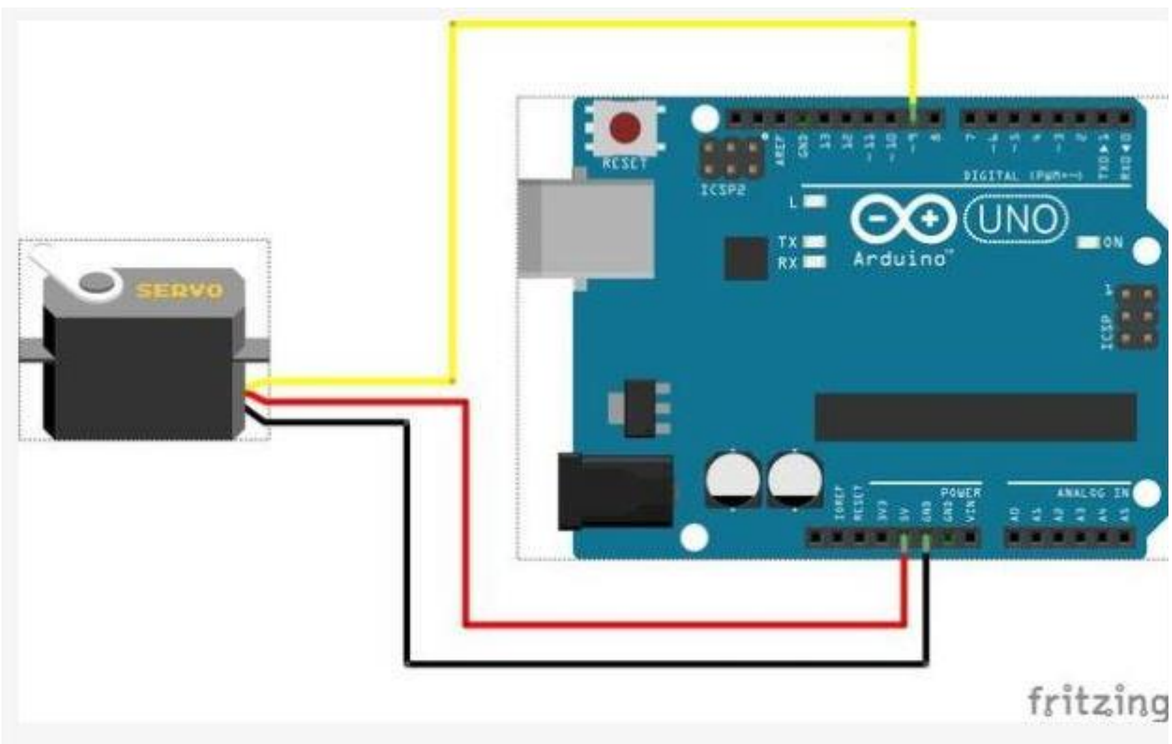

Figure 19, Servo motor connects to Arduino [15]

BLE test

Testing Method	Expected Outcome
Connect the BLE module with an Arduino board as shown in Figure 5 and run a 'Blink' sketch on Arduino IDE.	The LED on the BLE module will keep blinking while the Arduino sketch is executing.
Keep the previous connection, open LightBlue App and search	From the LightBlue App, we can detect the BLE module and get paring.
Upload the 'AT command sketch' shown in Figure 7 and sent AT command to the BLE module	The response of the AT command should be similar to Figure 8.
Set up the BLE module with LED light and the Arduino board as shown in Figure 6, and execute a sketch to turn the LED light on/off.	The LED light can be turned on/off by sending command via the BLE module

Table 9, BLE test cases

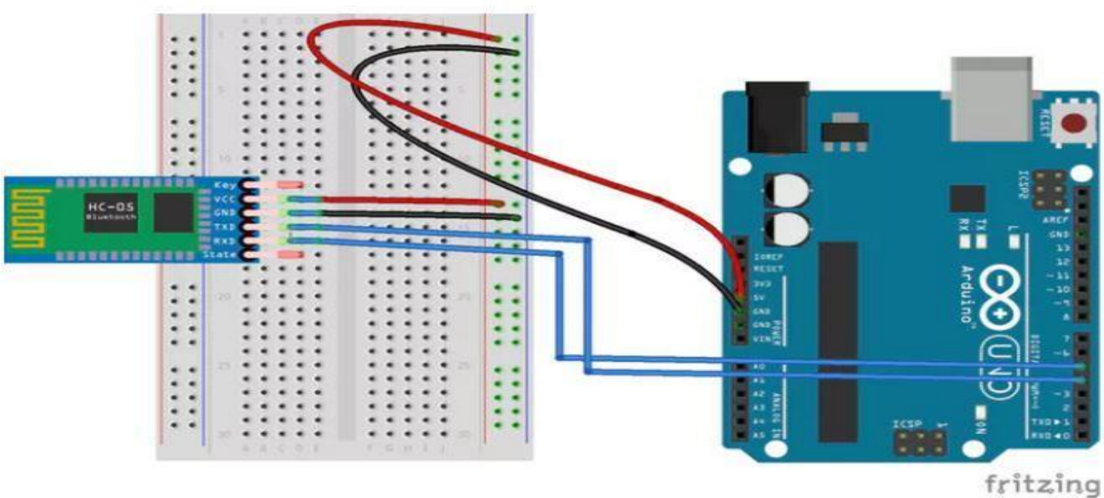

Figure 20, BLE connects to Arduino [16]

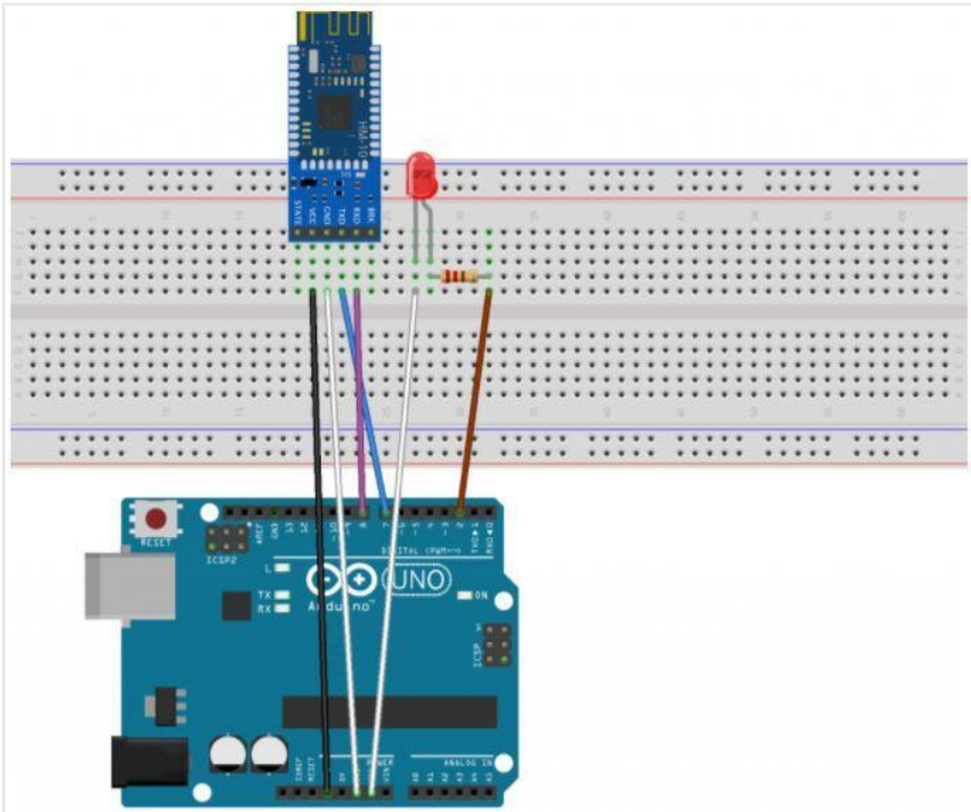

Figure 21, BLE control LED light [17]

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(2, 3); //RX|TX

void setup(){
  Serial.begin(9600);
  BTSerial.begin(9600); // default baud rate
  while(!Serial); //if it is an Arduino Micro
  Serial.println("AT commands: ");
}

void loop(){
  //read from the HM-10 and print in the Serial
  if(BTSerial.available())
    Serial.write(BTSerial.read());

  //read from the Serial and print to the HM-10
  if(Serial.available())
    BTSerial.write(Serial.read());
}
```

Figure 22, AT command sketch [18]

```
**********************************************************************
* Command             Description                                    *
* ------------------------------------------------------------------ *
* AT                  Check if the command terminal work normally    *
* AT+RESET            Software reboot                                *
* AT+VERSION          Get firmware, bluetooth, HCI and LMP version   *
* AT+HELP             List all the commands                          *
* AT+NAME             Get/Set local device name                      *
* AT+PIN              Get/Set pin code for pairing                   *
* AT+PASS             Get/Set pin code for pairing                   *
* AT+BAUD             Get/Set baud rate                              *
* AT+LADDR            Get local bluetooth address                    *
* AT+ADDR             Get local bluetooth address                    *
* AT+DEFAULT          Restore factory default                        *
* AT+RENEW            Restore factory default                        *
* AT+STATE            Get current state                              *
* AT+PWRM             Get/Set power on mode(low power)               *
* AT+POWE             Get/Set RF transmit power                      *
* AT+SLEEP            Sleep mode                                     *
* AT+ROLE             Get/Set current role.                          *
* AT+PARI             Get/Set UART parity bit.                       *
* AT+STOP             Get/Set UART stop bit.                         *
* AT+START            System start working.                          *
* AT+IMME             System wait for command when power on.         *
* AT+IBEA             Switch iBeacon mode.                           *
* AT+IBE0             Set iBeacon UUID 0.                            *
* AT+IBE1             Set iBeacon UUID 1.                            *
* AT+IBE2             Set iBeacon UUID 2.                            *
* AT+IBE3             Set iBeacon UUID 3.                            *
* AT+MARJ             Set iBeacon MARJ .                             *
* AT+MINO             Set iBeacon MINO .                             *
* AT+MEA              Set iBeacon MEA .                              *
* AT+NOTI             Notify connection event .                      *
* AT+UUID             Get/Set system SERVER_UUID .                   *
* AT+CHAR             Get/Set system CHAR_UUID .                     *
* -------------------------------------------------------------------*
```

Figure 23, AT command result [18]

Li-Po battery test

The Li-Po battery will be used to power the servo motor.

Testing Method	Expected Outcome
Connect the battery with the Arduino board.	The power light of the Arduino board is on.
Connect the battery with its charger	The charging light is on

Table 10, Li-Po battery test cases

Raspberry Pi test:

The Raspberry Pi will be used to collect the EMG signal and output a result that is a gesture command. The gesture command will send to Arduino board via BLE module.

Testing Method	Expected Outcome
Connect the Raspberry Pi with EMG sensor module	The power light of the EMG sensor module is on
Upload the convolutional neural network(CNN) model to Raspberry Pi	No error message pop up when we finish uploading.

Table 11, Raspberry Pi testcases

Integrated Test

The integrated test will only be conducted when the component tests are successfully finished. These tests will focus on some key ideas shown below:

1. The robotic hand can be control by the servo motors.
2. The BLE module can pair the Arduino board with the Raspberry pi.
3. The Raspberry pi can send string to Arduino board via Bluetooth communication.
4. The Raspberry pi can analyse the EMG signal and have same results while analysing it in MatLab.
5. The robotic hand can perform 6 gestures.

Test Case 1: Testing if the thumb finger can bend up by controlling the connected servo motor

- User Input: The user uploads a command in Arduino IDE
- Expected Result: The servo motor will rotate 180 degrees and only the thumb finger bends up

Test Case 2: Testing if the index finger can bend up by controlling the connected servo motor

- User Input: The user uploads a command in Arduino IDE
- Expected Result: The servo motor will rotate 180 degrees and only the index finger bends up.

Test Case 3: Testing if the majeure finger can bend up by controlling the connected servo motor

- User Input: The user uploads a command in Arduino IDE
- Expected Result: The servo motor will rotate 180 degrees and only the majeure finger bends up.

Test Case 4: Testing if the ring finger can bend up by controlling the connected servo motor

- User Input: The user uploads a command in Arduino IDE
- Expected Result: The servo motor will rotate 180 degrees and only the ring finger bends up.

Test Case 5: Testing if the auricular finger can bend up by controlling the connected servo motor

- User Input: The user uploads a command in Arduino IDE
- Expected Result: The servo motor will rotate 180 degrees and only the auricula ire finger bends up.

Test Case 6: Testing if the EMG signal is able to be analyzed in Raspberry Pi successfully

- User Input: The user should input a EMG signal to Raspberry Pi while the python program is executing
- Expected Result: The analysing result is the same as the signal analyzed in MatLab.

Test Case 7: Testing if the Bluetooth in Raspberry Pi can pair with the BLE module in Arduino board

- User Input: The user should send a pairing command from Arduino board via the BLE module
- Expected Result: The Raspberry Pi receive the pairing request and can successfully pair with the BLE module.

Test Case 8: Testing the Bluetooth communication between Raspberry Pi and Arduino board

- User Input: The user can request sending a string via Bluetooth while executing a python program in Raspberry Pi
- Expected Result: The Arduino board can receive the string via BLE module

Test Case 9: Testing if the Arduino board can control the robotic hand to perform 6 gestures

- User Input: The user can execute the sketch in Arduino IDE to let the robotic hand to perform one of six gestures.
- Expected Result: The robotic hand can perform the gesture while the sketch is executing.

Appendix 2: User Interface

Introduction

The following section provides a high-level User Interface description for the proof-of-concept of Handsome 1.0. The primary goal of UI design is to let the user step in the design process, which make our final product to be efficient and minimalistic. The following principles will be considered when making design decisions for the UI: discoverability, feedback, conceptual models, affordances, signifiers, mappings and constraints.

Furthermore, the user and technical analysis will be outlined in order to spotlight the standard that the UI shall adhere to. Analytical Usability Test plan for users is listed below to ensure the ease-of-use of Handsome 1.0. Correct installation of the UI components ensures proper functionalities mentioned in Requirement Specifications for the Handsome 1.0 Prosthetic Hand.

User Analysis

The user of Handsome 1.0 should have basic knowledge on how to safely power up a use electrical device. The users also need to be capable of reading and understanding the User documentation to operate our device normally.

People with diseases associated with the muscle functions on their right arm should not consider using Handsome 1.0 because our sensor system is only capable of analyzing EMG signals from correct-functioning right forearm and hand.

The users should understand misuse of the device, such as grabbing an object and dropping it, will result in breaking the object or hurting people.

The users should keep Handsome 1.0 in dry locations within the temperature range specified in the Requirement Specification.

Technical Analysis

Discoverability

Discoverability is a very important element to help the users comprehend the instructions. The following table discusses several user cases and how discoverable elements associated with the user cases.

Case	**Solution**
Supplying electricity to the sensor kit	An external power socket is placed on the side of the sensor kit
Turning on the system	A power button is added to the surface of the forearm of the prosthetic hand. A power button is placed on front side of the sensor kit
Resetting the system	A reset button is added to the sensor kit
Recharging the battery on the prosthetic hand	Label the battery slots for replacing the battery
Placing the electrodes	Wires of approximately 0.5 m connecting to the Myoware Sensors for user to add the electrode pads Painted picture on the sensor kit of locations to place the electrode pads

Table 12, Discoverable elements

Feedback

Feedback of our system help the user understand the current state of our device, the following table describes the user cases and the system feedbacks reflecting device state.

Case	**Feedback**
Device is on	Green LED is always on
Device is off	All LEDs are off
System is under reset	Green LED is blinking
System is under calibration	Red LED is blinking
System is capturing and analyzing data	Red LED is always on
System responds to user’s gesture	The prosthetic hand is moving

Table 13, Feedbacks

Conceptual models

Handsome 1.0 is designed to capture the gesture of a human hand and transfer it to the prosthetic hand to do the same gesture. In general, user attaches 3 channels of sensors to his or her forearm, after power on, the system will take a certain amount of data for calibration when user does not have any action on his or her hand at the beginning. When the calibration is done, the user can perform any gestures that already registered in the system. If the user performs an unknown gesture, the system will not report an error; instead it will find the closest match out of the predefined gestures. As long as the gesture is recognized, the prosthetic hand will perform corresponding gesture.

Affordances

Our device has simple a user interface features, which makes our device easy to be used by the users. LEDs allow user to know what the current state of the system is so that the user will understand what the system is doing currently, which will reduce the misuse of the user. There are internal BLE modules inside the sensor kit and the prosthetic hand, which allow the sensor kit side to communicate with the prosthetic hand side. The communication between the two ends will be established as soon as the system is turned on. The prosthetic hand will receive specific gesture code from BLE communication and does the corresponding action. Our prosthetic hand is only allowed to perform a movement from the initial state, which is palm opening. Take the "index" as an example, every time the user performs an "index" action, prosthetic will perform "index" action as well. If user wants to do another gesture, he or she has to open his or her palm to make the prosthetic hand open its palm, then user can do next gesture.

Signifiers

The signifiers for our product are simply LED of different colors. Signifier is another significant element for user interface design. It is a way of communication between the device and user. All the feedbacks should be presented well through all those LEDs.

Mappings

Two buttons and two LEDs are placed at the front surface of the case where easy to be noticed by the user. Two buttons, power button and reset button are also placed at the surface of the sensor kit. Power buttons control the on/off of the device, reset buttons returns the prosthetic hand to its home position. The LEDs indicate the current state of the device. Batteries, wires and microcontroller are placed inside the case. All ports are on the sides of the case. Figure 2 shows the 2D mapping of the case that holding the microcontroller, battery and wires. Two buttons of the prosthetic hand are placed directly to the forearm and they are aligned with each other.

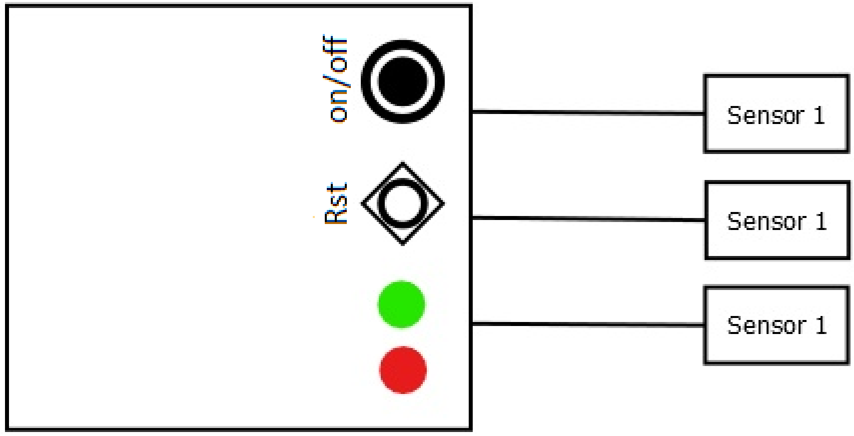

Figure 24, button and LEDs mapping of the sensor kit

The user needs to place the sensors before using the device. The correct locations are demonstrated in the following figure:

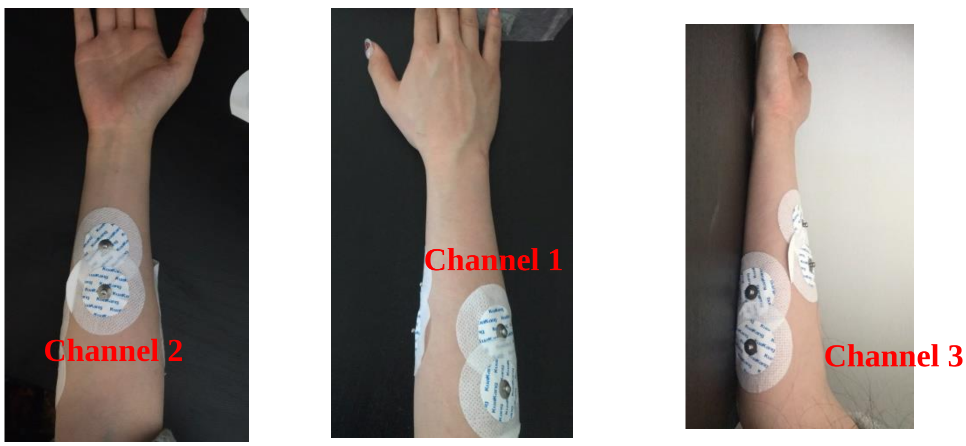

Figure 25, EMG sensor locations

Before attaching the electrodes to the forearm, use should use low concentration ethanol to clean the skin of the forearm to reduce the skin effect to the signal detection. After attaching all the sensors to the forearm, press the power button to turn on the device and then switch on all the sensors respectively.

Constraints

Our EMG sensors are sensitive. A small movement of the fingers will be detected by the sensors. Therefore, if user cannot hold his or her hand at peace during the calibration, the system may take longer time for the calibration. Up to now, only six gestures will be recognizing by our system and it is very limited. In addition, the material of the prosthetic hand and the selection of the motor will limit the total load. Data analyzing and software program algorithm will cause a certain delay for the prosthetic hand responding to user's control.

Engineering Standards

Handsome 1.0 will adhere to the following engineering standards which will be also accounted in the proof-of-concept product and final product stage.

Engineering Standards	Device and System
IEEE802.15.1 specification [19] & [20] Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)	For interoperability, the BLE modules are used for the wireless connection, the usage of BLE module should comply with the IEEE802.15.1 specification.
RoHS Restricted Substances [21] Restriction of Hazardous Substances	All the materials that we used in our device shall be RoHS compliant, so that user will not get harm by touching or wearing our product.
ISO/IEEE 11073 standard [22] Medical/health device communication standards enable communication between medical, health care and wellness devices and with external computer systems	Our EMG sensor is able to capture the signal from user's muscle and transfer the muscle data to our microcontroller. Thus, all EMG related devices shall adhere to this standard.
IEEE 1101.1 standard [23] IEEE Standard for Mechanical Core Specifications for Microcomputers Using IEC 60603-2 Connectors	The total system integration guidelines shall adhere to this standard.
ISO/IEC 13251 [24] Collection of graphical symbols for office equipment	The symbol designs of the power button, reset button and LEDs are similar to most of the office equipment. User should not misunderstand the symbols when using our device.
ISO 9241-11 [25] This part deals with the extent to which a product can be used by specified users to achieve specified goals with effectiveness (Task completion by users), efficiency (Task in time) and satisfaction (responded by user in term of experience) in a specified context of use (users, tasks, equipment & environments)	The design of the device allows user to control the prosthetic hand to do what the user wants it to do.

Table 14, Standards

Analytical Usability Testing

This section will provide the detail of analytical usability testing taken by engineering designers. The designers will perform the following test during the development and post-development to ensure the usability of Handsome 1.0.

Phase 1: Unit Test

Unit test will be conducted on individual components of the Handsome 1.0 to examine the functionality.

To test the microcontroller, need to ensure:

- All sensors, circuitry and BLE module is connected to the correct GPIO pins as mentioned above.
- BLE module has been power by 5V while the TX/RX pin has connected with a voltage divider
- Respective components developer code files have been copied and executable files have been created.
- Execute the executable files and check if it is the receiving data from the sensor. It should display proper EMG signal wave.
- Check the LED light of all the sensors
- BLE module's executable code has been created in both Arduino board and Raspberry Pi
- BLE dongle is plugged into Raspberry Pi
- BLE module has been connected to Arduino board with voltage divider in RX/TX pins
- Verify the LED light of BLE module is on when the connection between two devices are succeed.
- All servo motors have been connected to the Arduino board with correct pins
- Arduino IDE has been installed in right folder
- Servo motors have been connected to relative voltage regulators
- Servo motors' executable sketch has been created in Arduino IDE
- Verify all the servo motors are functioning when the sketch is executing

To test the robotic forearm, need to ensure:

- The robotic have five fingers and each finger can rotate independently
- The fingers can not move horizontally
- All the fingers have been connected to correct servo motors
- All the fingers will not rotate when the servo motors are not functioning
- Each finger is connected to one servo motor by one 3mm plastic filament
- Arduino board is inside the forearm and connected with the servo motors

Phase 2: Integrated Test

We will integrate all components and execute the executable program to ensure that all the data are capable of communication as expected and the product meets all the function requirements. Once the unit test is successfully completed, we will integrate them one by one to do this integrated test.

The detail is shown below:

- Ensure all devices are turned on and all unit tests are approved
- Execute the BLE module program and make sure the BLE modules are paring
- Execute the EMG data collection program in the Raspberry Pi
- Turn of the battery that is connected to the motor
- If any of above is not working as expected, please report it to the developer

Empirical Usability Testing

In order to gain the most minimalistic UI, we will have the following tests to help our designers collect necessary data to improve our product. The potential testers will be the those who are 18 years of age or above. To gather more data from those preference choice of users, evaluation report will be created throughout the development stage to optimize our current prototype model.

Natural Movement Test

Test Case Summary: The objective of this test is to examine what the most comfortable way of controlling the prosthetic forearm will be.

Precondition: The tester has been taught how to control the prosthetic forearm.

	Step	**Expected Result**
1	Put the prosthetic forearm on a supporter and ask the testers to make some gestures.	The tester should be capable of making those gestures and feel comfortable
2	Put the prosthetic forearm vertically on a table and ask the testers to make the same gestures as step 1	The tester should be able to finish making some gestures but they will fail to finish the whole task because the prosthetic forearm may fall on the table
3	Let the prosthetic forearm lie on the table and ask the testers to make the same gestures as step 1	
Result	Pass:	Fail:
Comments		

Table 15, Natural movement test

Skin cleaning Test

Test Case Summary: The objective of this test is to examine how necessary to use soap to clean the skin before using the EMG electrode.

Precondition: The tester has been taught how to control the prosthetic forearm. The prosthetic forearm has been set up on a supporter.

	Step	Expected Result
	Ask the testers to use soap to clean the skin, and ask them to use the prosthetic forearm to make some gestures	The tester should can make those gestures easily
1	Ask the testers to use water to clean the skin, and ask them to use the prosthetic forearm to make some gestures as step 1	
2	Ask the testers who do not clean their skin to use the prosthetic forearm to make some gestures as step	
3	Pass:	Fail:
Result		
Comments		

Table 16, Skin cleaning test

Ease of Use Test:

Test Case Summary: The objective of this test is to observe the user's reaction when they use the prosthetic forearm to grab object in certain size. This test should be taken several times for the same testers.

Precondition: All the unit tests shall be finished. The prosthetic forearm has been set up on a supporter. In addition, the tester has not been use Handsome 1.0 before.

	Step	Expected Result
1	Ask the testers to use the prosthetic to grab a tennis ball and then put it on a table	The tester should be capable of finish this task and feel comfortable
2	Repeat step 1 after 1 day	The tester should be familiar with using the prosthetic forearm to grab an object.
3	Repeat step 1 after one week	
Result	Pass:	Fail
Comments		

Table 17, Ease of use test

Conclusion

UI Appendix outlines different aspects of our product. It gives us an overview of the user interface design of our product. It also covers the UI design analysis from our engineers and test experiences and knowledge required from user. Currently, our proof-of-concept product have does not contain all the elements and functions covered in the requirement spec. They will be implemented on our final product for next four months. Buttons on the prosthetic hand will also be added in the final product. User testing has not been covered for this proof-of-concept product, but will be covered in the final product. Therefore, for next four months, this appendix will be referred during the development and testing of our final product.

Reference

[1] "InMoov | open-source 3D printed life-size robot", Inmoov.fr, 2017. [Online]. Available: http://inmoov.fr/. [Accessed: 23- Jun- 2017]

[2] "LTC® 8pcs MG996R Torque Digital High Torque Metal Gear Digital Servo Motor for Futaba JR RC Helicopter Car Boat Model FE: Amazon.ca: Office Products", Amazon.ca, 2017. [Online]. Available: https://www.amazon.ca/gp/product/B01GNOHB58/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1. [Accessed: 24- Jul- 2017]

[3] M. Rotation, "MG996R Robot servo 180° Rotation | Tower Pro", Towerpro.com.tw, 2017. [Online]. Available: http://www.towerpro.com.tw/product/mg995-robot-servo-180-rotation/. [Accessed: 24- Jul- 2017]

[4] "Which Kind of Motor Is Right for You?", Lifewire, 2017. [Online]. Available: https://www.lifewire.com/stepper-motor-vs-servo-motors-selecting-a-motor-818841. [Accessed: 24- Jul- 2017]

[5] "AMCI : Advanced Micro Controls Inc :: Stepper vs Servo", Amci.com, 2017. [Online]. Available: https://www.amci.com/industrial-automation-resources/plc-automation-tutorials/stepper-vs-servo/. [Accessed: 24- Jul- 2017]

[6] "Which capacitors for L7805 regulator?", Forum.arduino.cc, 2017. [Online]. Available: https://forum.arduino.cc/index.php?topic=358322.0. [Accessed: 24- Jul- 2017]

[7] [6]"7805 | Voltage Regulator - Fixed +5V @ 1A, TO220", Rpelectronics.com, 2017. [Online]. Available: http://www.rpelectronics.com/7805-fixed-5v-1a-to220.html. [Accessed: 24- Jul- 2017]

[8] 2017. [Online]. Available: https://www.sparkfun.com/datasheets/Components/LM7805.pdf. [Accessed: 24- Jul- 2017]

[9] "LTC® 8pcs MG996R Torque Digital High Torque Metal Gear Digital Servo Motor for Futaba JR RC Helicopter Car Boat Model FE: Amazon.ca: Office Products", Amazon.ca, 2017. [Online]. Available: https://www.amazon.ca/gp/product/B01GNOHB58/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1. [Accessed: 24- Jul- 2017]

[10] "Arduino Nano", Store.arduino.cc, 2017. [Online]. Available: https://store.arduino.cc/usa/arduino-nano. [Accessed: 24- Jul- 2017]

[11] "Arduino Mega 2560 Rev3", Store.arduino.cc, 2017. [Online]. Available: https://store.arduino.cc/usa/arduino-mega-2560-rev3. [Accessed: 24- Jul- 2017]

[12] 2017. [Online]. Available: http://www.vlfeat.org/matconvnet/matconvnet-manual.pdf. [Accessed: 24- Jul- 2017]

[13] 2017. [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/4/9/1/MyoWareDatasheet.pdf. [Accessed: 24- Jul- 2017]

[14] Instructables.com, 2017. [Online]. Available: http://www.instructables.com/id/Arduino-How-to-Control-Servo-Motor-With-Arduino/. [Accessed: 24- Jul- 2017]

[15] "Bluetooth LE: Using CC-41A (HM-10 Clone)", Arduino Project Hub, 2017. [Online]. Available: https://create.arduino.cc/projecthub/achindra/bluetooth-le-using-cc-41a-hm-10-clone-d8708e. [Accessed: 24- Jul- 2017]

[16] "Control an Arduino via the HM-10 BLE module, from a mobile app on your smartphone | Evothings", Evothings.com, 2017. [Online]. Available: https://evothings.com/control-an-led-using-hm-10-ble-module-an-arduino-and-a-mobile-app/. [Accessed: 24- Jul- 2017]

[17] "HM-10 or CC41-A module? Automatic Arduino BLE module identification | Arik Yavilevich's blog", Blog.yavilevich.com, 2017. [Online]. Available: https://blog.yavilevich.com/2016/12/hm-10-or-cc41-a-module-automatic-arduino-ble-module-identification/https://blog.yavilevich.com/2016/12/hm-10-or-cc41-a-module-automatic-arduino-ble-module-identification/. [Accessed: 24- Jul- 2017]

[18] X. (802.15.4), S. USB, S. Dongle, S. Bi-Directional, S. Serial, S. Shield, X. (802.15.4), S. Kit, 2. Socket and B. Module, "XBee 1mW Trace Antenna - Series 1 (802.15.4) - WRL-11215 - SparkFun Electronics", Sparkfun.com, 2017. [Online]. Available: https://www.sparkfun.com/products/11215. [Accessed: 24- Jul- 2017]

[19] T. ARTICLES, I. ARTICLES, G. ELECTRONICS, C. PROJECTS, E. MICRO, V. Lectures, I. Webinars, P. Search, T. DB, B. Tool, C. Library and M. Christiano, "ZigBee vs Bluetooth and Bluetooth Smart", Allaboutcircuits.com, 2017. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/zigbee-vs-bluetooth-and-bluetooth-smart/. [Accessed: 24- Jul- 2017]

[20] Anon, (2017). [online] Available at: http://www.sj.ifsc.edu.br/~msobral/RCO2/docs/ieee/802.15.1-2005.pdf [Accessed 25 Jul. 2017].

[21] Standards.ieee.org. (2017). IEEE SA - 1101.1-1998 - IEEE Standard for Mechanical Core Specifications for Microcomputers Using IEC 60603-2 Connectors. [online] Available at: https://standards.ieee.org/findstds/standard/1101.1-1998.html [Accessed 25 Jul. 2017].

[22] En.wikipedia.org. (2017). ISO 9241. [online] Available at: https://en.wikipedia.org/wiki/ISO_9241#ISO_9241-11 [Accessed 25 Jul. 2017].

[23] Iso.org. (2017). ISO/IEC 13251:2004 - Collection of graphical symbols for office equipment. [online] Available at: https://www.iso.org/standard/21486.html [Accessed 25 Jul. 2017].

[24] En.wikipedia.org. (2017). ISO/IEEE 11073. [online] Available at: https://en.wikipedia.org/wiki/ISO/IEEE_11073 [Accessed 25 Jul. 2017].

[25] Rohsguide.com. (2017). RoHS Compliance Guide: RoHS 10 Restricted Substances. [online] Available at: http://www.rohsguide.com/rohs-substances.htm [Accessed 25 Jul. 2017].

ENSC 405W Grading Rubric for User Interface Design
(5-10 Page Appendix in Design Specifications)

Criteria	Details	Marks
Introduction/Background	Appendix introduces the purpose and scope of the User Interface Design.	**/05%**
User Analysis	Outlines the required user knowledge and restrictions with respect to the users' prior experience with similar systems or devices and with their physical abilities to use the proposed system or device.	**/10%**
Technical Analysis	Analysis in the appendix takes into account the "Seven Elements of UI Interaction" (discoverability, feedback, conceptual models, affordances, signifiers, mappings, constraints) outlined in the ENSC 405W lectures and Don Norman's text (*The Design of Everyday Things*). Analysis encompasses both hardware interfaces and software interfaces.	**/20%**
Engineering Standards	Appendix outlines specific engineering standards that apply to the proposed user interfaces for the device or system.	**/10%**
Analytical Usability Testing	Appendix details the analytical usability testing undertaken by the designers.	**/10%**
Empirical Usability Testing	Appendix details completed empirical usability testing with users and/or outlines the methods of testing required for future implementations. Addresses safe and reliable use of the device or system by eliminating or minimizing potential error (slips and mistakes) and enabling error recovery.	**/20%**
Graphical Presentation	Appendix illustrates concepts and proposed designs using graphics.	**/10%**
Correctness/Style	Correct spelling, grammar, and punctuation. Style is clear concise, and coherent. Uses passive voice judiciously.	**/05%**
Conclusion/References	Appendix conclusion succinctly summarizes the current state of the user interfaces and notes what work remains to be undertaken for the prototype. References are provided with respect to standards and other sources of information.	**/10%**
CEAB Outcomes: Below Standards, Marginal, Meets, Exceeds	1.3 Engineering Science Knowledge: 4.1 Requirement and Constraint Identification: 5.4 Documents and Graphic Generation: 8.2 Responsibilities of an Engineer:	