Design Paper

Group C.O.R.E.

Electrical System for Exoskeletal Arm

Department of Electrical Engineering and Computer Science
University of Central Florida
Dr. Lei Wei
Senior Design I

Group Member	Major
Brandon Johnson	CpE
Daniel Reveron	EE/CpE
Kelvin Feliciano	EE
Gavin Bell	EE

TABLE OF CONTENTS

Table of Figures

1: Executive Summary

2: Introduction

3: Administration

4: Research

1: EXECUTIVE SUMMARY

1.1: Abstract

Limbitless Solutions is a non-profit organization that produces biomedical applications at low cost for patients with physical disabilities throughout the country. They are established on the University of Central Florida's main campus, where people with the same state of mind get together and find suitable, affordable solutions for a great variety of issues. Patients come from many different states and request diverse prosthetic parts and other devices that they can use in order to improve their quality of life. Under the same token, students and companies alike help the organization further advance for society and these patients by providing bright design ideas and funding respectively.

During the spring semester of 2016, several groups of mechanical engineering students from UCF came together, with the help from L.S., and thought up an idea that would not only improve the capabilities of a patient with limited limb mobility, but would also allow the patient to be fluidly integrated into society by feeling less hindered by robotic movement and the bulkiness that is prone to occur as a result of designing a physical device. The idea consisted of an exoskeleton arm called "Carapace" that would be fully made of soft materials and would be driven by pneumatic actuators. The exoskeleton would essentially read inputs from the patient's muscle signals and transform them into actuation outputs. Due to the complexity of the electrical framework that would be needed to control the actuators, the electrical design would require to be in its own project in order to develop; thus, the mechanical teams required a team that is specialized in the electrical components that would control the actuators and the software that would run the components. Since our team had planned to present a biomedical senior design at the same time as the mechanical team was to present theirs, Limbitless Solutions decided to join efforts and group both teams together in order to make this system a reality, while still preserving enough independence for both teams to complete their design without risking each other's success in case either team would not be able to deliver their design on time.

Although the fully-pneumatic design showed great promise in the making of a fluid exoskeleton, concerns arose from both teams regarding completion of the project by the December deadline. Additionally, some changes to current design ensued due to the nature of the patient's disability, who suffers from Arthrogryposis Multiple Congenita. This condition is typically caused at birth and affects his joints by having them fused together. His muscles are also deteriorated due to lack of use, making any signals harder to obtain from the muscle. For this reason, the electrical team, also named group CORE (Controller for Organic Range of Exoskeleton), proposed a solution that would guarantee a better adjustment to the patient's measures as well as the completion of the project by the given deadline. The solution consists of a hybridization of both electrical motors for the elbow and pneumatic actuators for the hand. Furthermore, the number of degrees of freedom to be regarded for input and actuation was reduced from eight to three. This design was approved by all the stakeholders involved in the system.

The finalized idea of the exoskeleton arm is to be applied to the upper limb of the patient. The arm will operate on 3 degrees of freedom along artificial joints that will simulate their biological counterparts. 2 degrees of freedom will be of an electrical nature. When the patient applies any minimal force by contracting muscles in the arm, the exoskeleton will read the bio-electrical signals originated by the muscle contraction by means of EMGs, and produce movement. Electrical readings from the muscle will be processed by a control unit that will determine where to redirect the power. For the elbow, the EMG will also provide the motor with information required to move at a set of defined speeds that attempts to compensate for the given input by the patient. The hand will only operate on a digital event, as the hand opens and closes. A feedback-controlled system will be set in place to allow the patient for unhindered movement of the articulations, while ensuring the safety of the user. What makes this exoskeleton arm special in comparison to its market counterparts is that it will be produced at a much cheaper rate than that of its average production cost by as much as 10% of the market value.

2: INTRODUCTION

2.1: Exoskeleton Concept

The exoskeleton is made of two different mechanical components: a solid frame that makes up the elbow section, and a custom pneumatic material that goes over the hand as shown on *Exoskeleton Concept 1* below. This image was obtained from a concept drawn by the mechanical team and its original source is unknown.

The mechanical team plans on applying momentum on the arm with the use of a cable that connects to a motor behind the patient, mounted on his wheelchair. Additionally, the hand would be actuated with two solenoid valves that attempt to create a closed-loop control of the soft-robotics material in combination with an electric DC pump that will send pressure out to the hand to close.

The electrical team's job is to provide control for the arm and hand by taking EMG inputs from the patient and processing those inputs into suitable outputs for the actuators. For the motor that controls the elbow motion, different ranges of inputs will map to discrete output speeds. The pneumatic fluid will be directed to the hand by powering the DC pump and opening the appropriate solenoid valves for opening and closing of the hand. The C.O.R.E. team will also use its resources to ensure that the mechanical actuators adjust to the team's electrical mainframe so as to provide the most efficient integration of a working model.

Exoskeleton Concept 1: This is an image of what the exoskeleton would look like (will change with actual prototype)
Image from mechanical team's research files

2.2: Requirements

2.2.1: Engineering Requirements

In order for the team to create a design that will adjust to the patient's needs, some engineering requirements must be developed. These requirements are of a more technical nature and are directly supported by marketing requirements for the user's expectations from the design. *Engineering Requirements 1* shows the list of requirements that the C.O.R.E. team agreed on based on some research and given input by the mechanical team and the sponsor. Note that the first number of the engineering requirement maps to its corresponding end-user, or marketing, requirement.

End-User Requirement #	**Engineering Requirement #**	**Engineering Requirement**
EU.1	ENG.1.1	The system shall provide a fail-safe feature in case of emergency
EU.1	ENG.1.2	The system shall not exceed a temperature of 48 °C for components that are in direct touch with the user
EU.1	ENG.1.3	The power subsystem shall not compromise the user's safety in case of hazard
EU.2	ENG.2.1	The system shall be able to adjust its input signal detection criteria in a user-by-user basis
EU.2	ENG.2.2	The system shall have a response time no longer than 10 ms, as specified by mechanical engineering constraints
EU.3	ENG.3.1	The calibration of the electrical system for a daily basis use shall not exceed 2 minutes
EU.3	ENG.3.2	The system shall be able to shut down after exoskeleton reaches a position that makes arm easy to remove
EU.4	ENG.4.1	The system shall allow the powered system to be used for a period longer than 1 hour on maximum use for a single charge
EU.5	ENG.5.1	The system shall allow the user to reset the controls to their initial state in case a deadlock occurs

Engineering Requirements 1: This is a table of the engineering documents for the system.

2.2.2: End-User Requirements

These are more direct requirements that were obtained by analyzing the user needs. They are in direct contact with what the user wants and are easier to understand by non-technical personnel. *End-User Requirements 1* specifies the five requirements that the team derived engineering requirements from.

Req. #	End-User Requirement Statement
EU.1	The system shall not jeopardize user's safety
EU.2	The system shall provide a fast response time to simulate human motion
EU.3	The system shall be easy to install and remove
EU.4	The system shall be durable to adapt to human lifestyle
EU.5	The system shall be stable

End-User Requirements 1: This is a table of the end user requirements.

2.2.3: Requirement Concepts

This system is to be designed to meet end-user expectations; our end-user is the team that will design the mechanical framework for the exoskeleton. Group 14 has carefully chosen some general constraints that apply to the electrical scope of this system and are essential in providing a product that is compatible with the mechanical system's goals. These end-user requirements are listed as follows:

1. **The system shall not jeopardize user's safety.** As with any technological applications that involve human interaction, the interacting device must guarantee human safety at all times. For the project's purpose, temperature and controls must be considered. The electrical framework should neither allow itself to become overheated nor attempt to manipulate arm past a pressure range, so as to prevent hazard to human safety. Additionally, if the arm's control system were to become unstable to where the user's integrity might be compromised, the system should be able to cease all operations by disconnecting the arm's power supply in a safe and accessible manner.

2. **The system shall provide a fast response time to simulate human motion.** The main goal of the exoskeleton arm is to provide the ability to operate the equipment under seamless motion, as opposed to their more digital, robotic counterparts. It was determined by the mechanical team that the response time between input and output should not exceed 10 milliseconds.

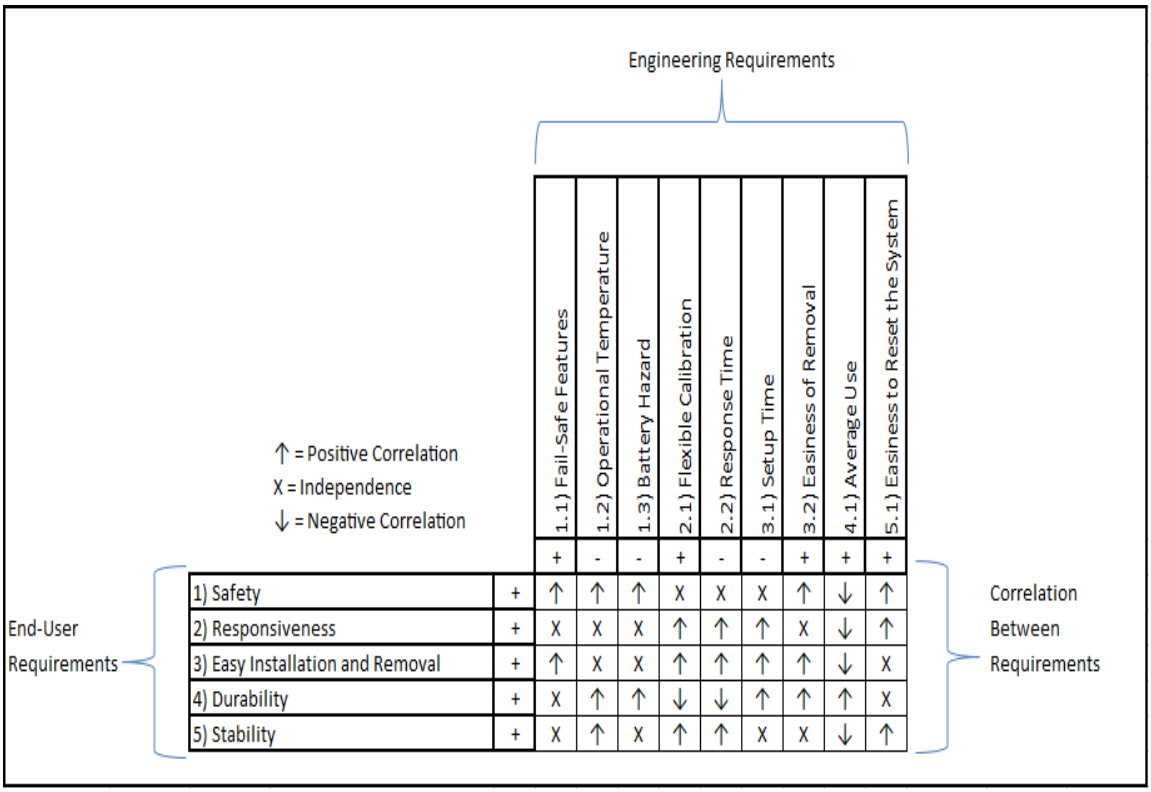
3. **The system shall be easy to install and remove.** For ease of installation and removal, the current design should be as efficient as possible. Calibration procedures and controls' initialization are established to be set up under 2 minutes. Also, controls will manipulate arm to be relaxed upon shutdown, so that it can be removed easily from the user.

4. **The system shall be durable to adapt to human lifestyle.** The main goal for the exoskeleton arm system as a whole is to be used as an extension of the patient. By extension, its power system should allow arm to work harmoniously with day to day tasks. For this reason, a constraint of 9 hours was considered optimal, given a working day.

5. **The system shall be stable.** In other words, the system must handle undesired outputs. Although not necessarily unsafe, these undesired outputs will decrease the user's fidelity toward the exoskeleton arm. Under certain muscle movements, it is possible that the arm might mix the inputs and output incorrect movement controls. To solvent this possible scenario, the system will have to feedback the output and ensure error correction so that the proper controls are applied. Additionally, if the controls were to lock into an unresponsive state from which they cannot exit, the controls should be able to allow the arm to restart into an initial state and resume normal operations.

2.2.4: House of Quality Diagram

A house of quality diagram represents the respective correlation between both engineering requirements and end-user requirements. For *House of Quality Diagram 1* as shown right after this paragraph, both sets of requirements include a level of polarity, which essentially tells the reader whether a higher value of a given requirement is good or bad for the system. Note that some requirements listed below, such as the operational temperature requirement, has an inverse polarity. That means that the less temperature exerted by the system, the better the system fulfills the requirement. Additionally, arrows were used to identify direct correlation between end-user requirements and engineering requirements. For arrows pointing up, fulfillment of both requirements affect each other positively. In the same manner, arrows pointing down signify that the fulfillment of one requirement impacts negatively the fulfillment of its correlation requirement.



House of Quality Diagram 1: Describes correlation between end-user and engineering requirements

2.3: Team Roles

Based on design specifications and each member's technical skillset, Group 14 has decided to break down the project into four sections as follows on figure *Team Roles 1*.

Member	**Role**
Daniel Reveron	Systems and Input Lead
Kelvin Feliciano	Hardware and Power Lead
Brandon Johnson	Software Lead
Gavin Bell	Electro-Mechanical Interface Lead

Team Roles 1: This table shows the roles of the team and who is responsible for what. Each person has specific tasks and we can assist each other when necessary.

2.4: Initial Designs & Block Diagrams

2.4.1: Legend for Block Diagrams

The figure below, *Legend for Block Diagram 1* displays the color schemes for the following block diagrams. It can be used for each diagram in this section.

Legend for Block Diagrams 1: This is a legend that shows what colors correspond to which team members in the following block diagrams.

2.4.2: Initial Electro/Mechanical Design

The design for the electro/mechanical portion of the exoskeletal arm is mainly comprised of four elements which will dictate the locomotion of the pneumatic actuators. The four elements allowing for movement of the actuators are: the compressor/reserve tank, the pressurizing solenoid valves, the vent solenoid valves, and the pressure sensors. The pneumatic actuators translate the pressure of the working fluid, in this case air, into the desired movement. To create the air pressure required to drive the system, there will be a compressor/reserve tank component. The compressor will need the reserve tank to allow for quick movements, reserving a large enough volume for the compressor to be able to keep up with constant movements all while maintaining a pressure adequately above that which is required. The pressurizing and vent solenoid valves will be used to control the flow of air to manipulate different actuators. It is the pressure sensors which will provide feedback ensuring the actuators are at the correct pressure correlating to their desired position.

2.4.3: Initial Hardware Design

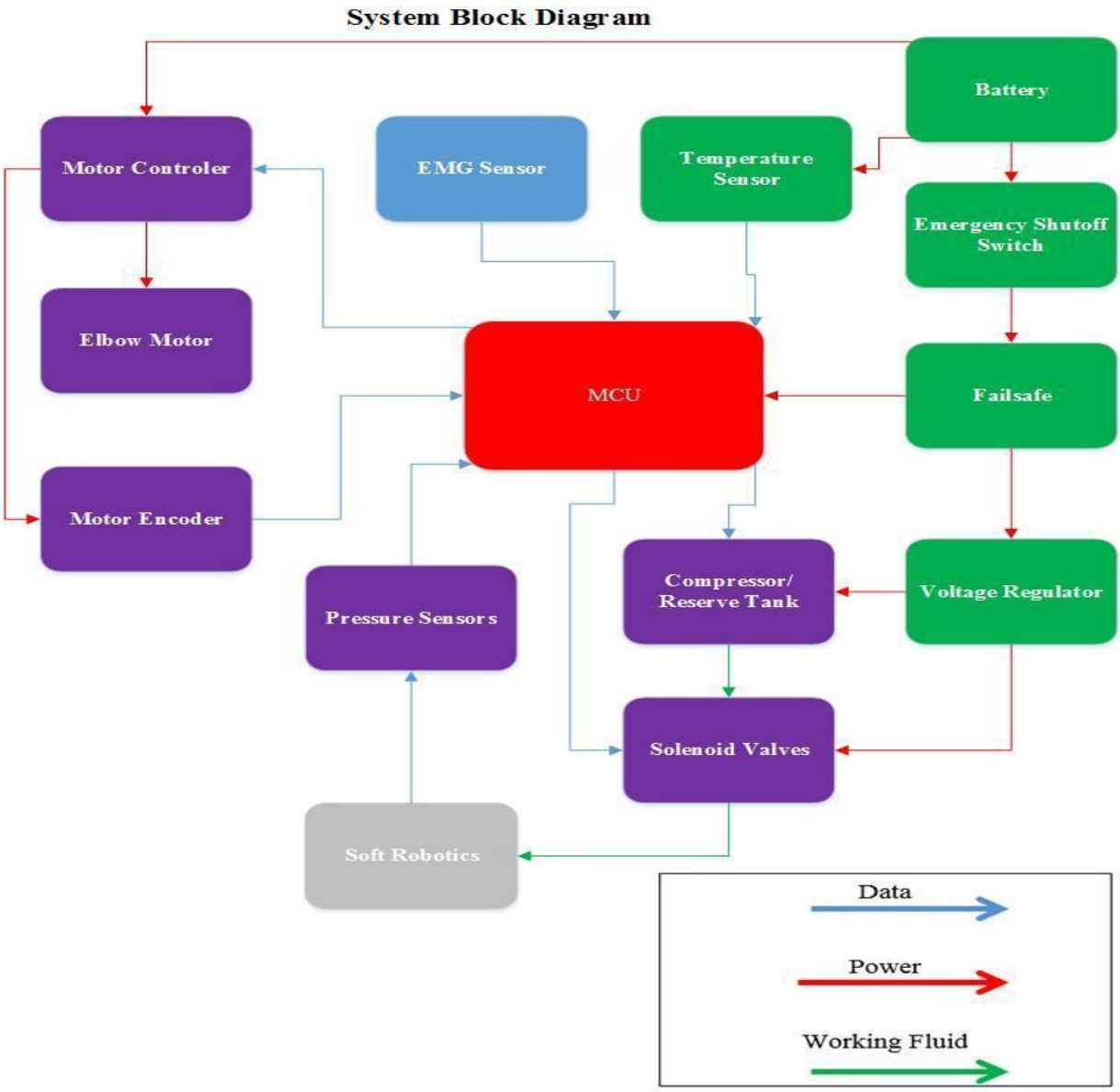
The exoskeleton will have a lithium battery that will power the system for approximately nine hours. The life of the battery will depend on how much and how long the arm is being used. Multiple safety features will be set in place in place; to protect the battery from overheating, a temperature sensor will be installed to determine if the battery is operating at a safe range; if not, the system will shut down. Another safety measure to be installed in the system an emergency shutoff switch that could easily be flipped manually by the user. Lastly, the system will need to operate in a certain psi range. If the sensor is reading a pressure that might be dangerous to the user's integrity, a trigger would deactivate the device.

This design will have multiple ElectroMyoGraphy Sensors (EMG). These are used for diagnostic procedures to assess the health of muscles and the nerve cells that control motor neurons. In this project, EMG sensors are intended to measure electrical response to a muscle contraction by nerve stimulation. The sensors will be attached in specific parts of the muscles, where the signal is strong and clear. In some patients, the signal might not be strong enough; to amplify the signal, additional hardware might be necessary. Additionally, a stronger muscle can be selected from another part of the body to capture the signal. To prevent picking up noisy signals, the channels will have to be as short as possible.

There are also some ideas regarding operation of the compact air compressor. It will likely be connected directly to the battery with a fuse for overcurrent protection. Also, it is worthy to consider that actuators have an in-rush current and any feedback from the motor could potentially affect any of the electrical components in the PCB; therefore, electrical solenoid valves on the actuators may need to be equipped with a diode to prevent any feedback when the coils collapse.

2.4.4: Initial Hardware Block Diagram

After the concept ideas have been drawn out for the project and a set of requirements have been established, the team is now able to create a high level diagram for how the hardware architecture will look after the system has produced a prototype. Based on this diagram, the hardware team will create lower level diagrams that will allow for a proper interface between subsystems and components. *Initial Hardware Block Diagram 1* represents the high-level block design of the system, which the team will be in charge of creating and integrating. It should be noted that only the interfacing mechanical components are being considered in the block diagram.



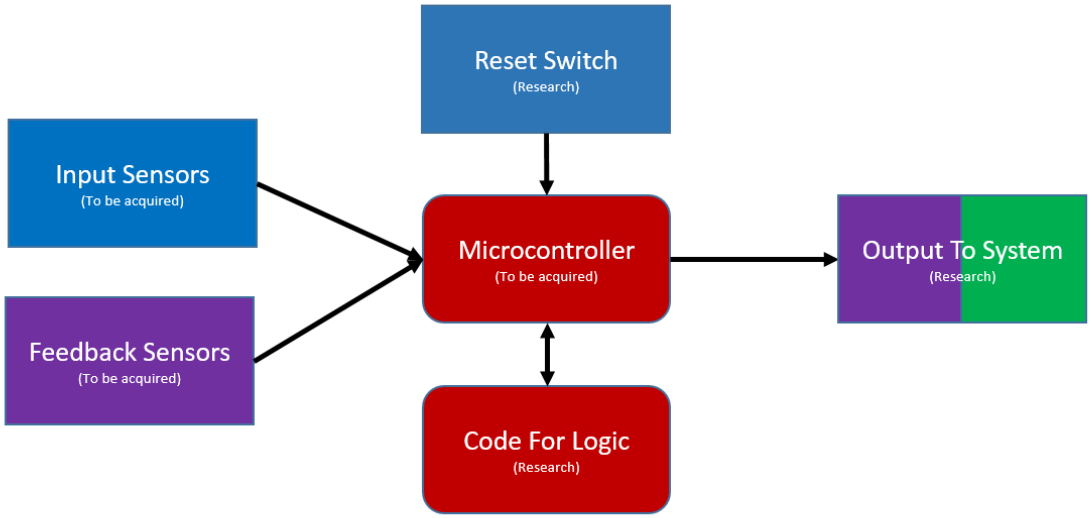
Initial Hardware Block Diagram 1: System Diagram for Hardware Components

2.4.5: Initial Software Design

The basis of the software design is as shown in *Initial Software Block Diagram 1*. The related components include the input sensors, the feedback sensors, the reset switch, the system outputs, the microcontroller, and the code. The idea is to use a sufficiently powerful microcontroller as a central control unit for the entire system. This microcontroller will need to have a large amount of pins to support the various inputs and outputs the system requires and will also need to have a high sampling frequency for good response time. Another good feature would be ease of peripheral addition and removal in case we want to temporarily attach components for testing purposes. The controller will first take inputs from the sensors and then process the code that will convert these inputs into appropriate system outputs. It will also connect to the reset switch, which if activated, should trigger a hard reset of the system. The feedback sensors will also need to be processed. Information from these sensors will be factored into the logic and will adjust the behavior of system.

2.4.6: Initial Software Block Diagram

Initial Software Block Diagram 1 shows the initial design for the software related components. It shows what will be connected to the microcontroller from a high level and how everything will flow.



Initial Software Block Diagram 1:This is a diagram of the software design imagined after some preliminary research.

2.4.7: Design Specifications

The following items have been chosen after careful examination of the system. They will provide a fully developed actuation system from an electrical standpoint. Although not specified by name in this section, particular items were chosen. These particular items will be reflected by their cost in section 3.1 of this design paper.

- Microcontroller Unit
- PCB
- EMG Sensors
- Pressure Sensors
- Compressor
- Solenoid Valves
- Battery
- Temperature Sensor
- Switching Voltage Regulators
- Linear Voltage Regulators
- Switch
- Stepper Motor
- Motor Driver

3: ADMINISTRATION

3.1: Estimated Budget

This section defines the cost that will incur in the success of the project. Parts were carefully chosen after a research was developed to find suitable components; however, since it is a distinct possibility that some other parts might be needed in addition or as a replacement for the system to thrive, the team accounted for a 50% contingency cost. This cost is meant to cover a worst, acceptable case scenario, where half the components need to be replaced. *Estimated Budget 1* shows all components that were chosen, the quantities that were chosen for the components, and their respective costs.

Sponsor:			**Limbitless Solutions**
Component:	**Estimated Quantity:**	**Unit Cost:**	**Total Cost:**
Microcontroller Unit	**1**	**$45.95**	**$45.95**
PCB	**1**	**$30.00**	**$30.00**
EMG Sensor	**3**	**$10.00**	**$30.00**
Electrodes	**3**	**$0.30**	**$0.90**
Pressure Sensor	**2**	**$80.00**	**$160.00**
Compressor	**1**	**$60.00**	**$60.00**
Solenoid Valve	**2**	**$10.00**	**$20.00**
Stepper Motor	**1**	**$100.00**	**$100.00**
Stepper Motor Driver	**1**	**$30.00**	**$30.00**
Battery	**1**	**$210.00**	**$210.00**
Temperature Sensor	**1**	**$1.50**	**$1.50**
Buck Converter	**3**	**$6.60**	**$19.80**
Linear Voltage Regulator	**3**	**$0.56**	**$1.68**
Switch	**1**	**$7.00**	**$7.00**
Total Cost:			**$716.83**
Total Cost With 50% Contingency:			**$1,075.25**

Estimated Budget 1: Estimated total cost for the project including contingency cost

3.2: Milestones

3.2.1: Milestones Overview

Figures *Milestones Overview 1* and *Milestones Overview 2* show the team's milestones for the scope of this project at a high level. Each week is accounted for in order to always have something to work on. Deadlines are spread out to leave appropriate time, while still progressing forward.

Description	**Weeks**	**Soft Deadline**
Senior Design I		
Project Introduction & Group Formation	Week 1	5/20/16
Determine Project Idea	Week 2	5/26/16
Initial Proposal	Week 3	**6/3/2016**
System Design Review with Mechanical Team	Week 3 - Week 5	6/17/2016
Research Input Sensors & Feedback Sensors	Week 4 - Week 5	6/17/2016
Research Power Requirements	Week 4 - Week 5	6/17/2016
Research Printed Circuited Board Design	Week 4 - Week 5	6/17/2016
Research Microcontroller	Week 4 - Week 5	6/17/2016
Begin Design Paper and Continue Research	Week 6 - Week 8	7/8/2016
Generate and Turn in Table of Contents	Week 7	**7/1/2016**
Design Models/Input Test and Continue Design Paper	Week 7 - Week 9	7/15/2016
Turn in Current Draft of Senior Design Documentation	Week 8	**7/8/2016**
Finalize Paper and Begin Ordering Parts	Week 9 - 11	7/29/2016
Submit Final Document	Week 12	**8/2/2016**
Input Testing on Patient to Calibrate Electronics	Week 12 - Break	8/12/2016
Small Break Before Fall Classes		

Milestones Overview 1: This table shows what the team plans to achieve throughout semester 1

Senior Design II		
Regroup and Discuss Plans	Week 1	8/26/2016
Gather All Parts and Finalize Design	Week 2 - Week 3	9/9/2016
Prototype System	Week 4 - Week 8	10/14/2016
Test and Debug system. Begin Final Documents	Week 9 - Week 11	11/4/2016
Integrate with Mechanical Team and Test	Week 11 to 14	11/25/2016
Final Presentation/ Evaluation	Week 15	12/1/16

Milestones Overview 2: This table shows what the team plans to achieve throughout semester 2

3.2.2: Milestones (Detailed)

3.2.2.1: Senior Design I

The following tables show a lower level plan for progress. These tables go into detail explaining exactly what each person is responsible for and providing accountability in the project. Figures *Milestones (Detailed) 1.1-1.15* shows goals for the first semester.

Project Introduction & Group Formation	Week 1	5/20/16
A clear idea of what the course contains and what is expected of the class will be obtained. Working group will be drafted by students. Individuals' strengths and weaknesses will be defined within the group for design roles. Contact information will be exchanged. <u>Entire Team</u> ❏ 5/17 Start forming group ❏ 5/19 Finalize groupmates ❏ 5/19 Know individual strengths and weaknesses ❏ 5/19 Exchange contact information ❏ 5/20 Understand what the class is about ❏ 5/20 Submit Individual Project Idea assignment		

Milestones (Detailed) 1.1. Project Introduction & Group Formation

Determine Project Idea	Week 2	5/27/16
Project idea will be decided based on each member's input. This will be needed in order to write preliminary documentation. <u>Entire Team</u> ❑ 5/24 Team meeting ❑ 5/26 Brainstorm ideas ❑ 5/27 Decide on idea		

Milestones (Detailed) 1.2. Determine Project Idea

Initial Proposal	Week 3	**6/3/2016**
The initial document will be provided in accordance with the provided rubric. It is to be submitted by 12pm on 6/3. Before writing, details of project will be discussed. <u>Entire Team</u> ❑ 5/28 Decide roles ❑ 5/28 Team meeting with mechanical liaison ❑ 5/31 Meet with Limbitless Solutions ❑ 5/31 Start writing Initial Proposal ❑ 6/2 Finalize document ❑ 6/3 Submit Initial Proposal by 12pm		

Milestones (Detailed) 1.3. Initial Proposal

System Design Review with Mechanical Team	Week 3 - Week 5	6/17/2016
Discussions will be held with mechanical team. System functionality will be known as well as constraints. By the end of this phase, design idea should be clear. <u>Entire Team</u> ❑ 6/7 Meet with Limbitless Solutions ❑ 6/11 Meeting with mechanical team liaison ❑ 6/14 Submit updated initial document by 10am ❑ 6/17 Should have full details from mechanical team of what needs to be accounted for <u>Daniel</u> ❑ 6/15 Bill of Testing Materials to be provided to Limbitless Solutions for research purposes and funding ❑ 6/16 Discuss extent of requirements with mechanical team and Limbitless Solutions		

Milestones (Detailed) 1.4. System Design Review with Mechanical Team

Research Input Sensors & Feedback Sensors	Week 4 - Week 5	6/17/2016
Research will be done to understand how input is provided to control system. Test scenarios to be developed at a high level with mechanical team liaison in order to carry out with patient. More information on feedback sensors will be acquired Gavin ❏ 6/16 Contact mechanical team lead to discuss feedback sensor ❏ 6/17 Determine feedback system design at a high level ❏ 6/17 Acquire EMG sensor for testing Daniel ❏ 6/17 Understand how signals are retrieved from muscles and EMG functionality ❏ 6/17 Develop input test plan for patient		

Milestones (Detailed) 1.5. Research Input Sensors & Feedback Sensors

Research Power Requirements	Week 4 - Week 5	6/17/2016
Kelvin ❏ 6/15 Contact mechanical team to discuss DC motors and pneumatic actuators ❏ 6/16 Research batteries suitable for the system ❏ 6/17 Select the battery that will power the control system		

Milestones (Detailed) 1.6. Research Power Requirements

Research Printed Circuited Board Design	Week 4 - Week 5	6/17/2016
Kelvin ❏ 6/16 Getting Familiar with the software to create electrical and PCB schematic ❏ 6/17 Research and review companies that provide PCB service		

Milestones (Detailed) 1.7. Research Printed Circuited Board Design

Research Microcontroller	Week 4 - Week 5	6/17/2016
Research will be done for microcontroller setup and codebase. This will be the logic control for the system so much effort needs to go into understanding this process. Brandon ❏ 6/7 Decide what controller will be used ❏ 6/8 Decide coding language and download IDE ❏ 6/12 Research the hardware of the controller (# of pins, chip, available peripherals) ❏ 6/17 Learn basic syntax of the controller in chosen language (Should be able to at least flash an LED)		

Milestones (Detailed) 1.8. Research Microcontroller

Begin Design Paper and Continue Research	Week 6 - Week 8	7/8/2016

Research efforts will be continued for the scope of this project. A final design paper will start to take form as a draft.

Entire Team
- ❑ 6/21 Have Final Paper Started
- ❑ 6/28 Have at least 10 pages worth of documentation each (Not including this document).

Brandon
- ❑ 6/24 Be familiar with more complicated code for controller
- ❑ 7/8 Be completely comfortable with the code base and be able to develop real projects

Daniel
- ❑ 6/21 Obtain finalized test plan for input testing
- ❑ 6/21 Meeting with mechanical team to discuss integration
- ❑ 7/1 Finalize requirements' statement

Gavin
- ❑ 6/24 Create Mechanical system overview design schematic
- ❑ 7/1 Create outline for mechanical system actuation

Kelvin
- ❑ 6/20 Obtain specifications of electrical components
- ❑ 6/26 Develop circuit design
- ❑ 7/1 Simulate circuits on electrical design software

Milestones (Detailed) 1.9. Begin Design Paper and Continue Research

Generate and Turn in Table of Contents	Week 7	**7/1/2016**

Table of contents will be submitted based on what has been done on the document so far.

Entire Team
- ❑ 6/28 Have all pieces of document (so far) combined
- ❑ 6/30 Have table of contents generated
- ❑ 7/1 Turn in Table of contents by 12pm

Milestones (Detailed) 1.10. Generate and Turn in Table of Contents

Design Models/Input Test and Continue Design Paper	Week 7 - Week 9	7/15/2016

Models for the system will be designed and continued work will be done on the paper.

Entire Team
- ❑ 7/6 Have at least 20 pages each (Not including this document).

Brandon
- ❑ 7/8 Have pseudocode for test system
- ❑ 7/15 Have pseudocode for control system

Daniel
- ❑ 7/2 Meet with team to discuss status
- ❑ 7/8 Testing procedures for the system to be fully documented in accordance with acquired requirements
- ❑ 7/15 Obtain budget from teammates to order parts for system implementation

Gavin
- ❑ 7/8 Have circuit set up to measure EMG signals
- ❑ 7/15 Determine operating range of EMG signals and extent of sensor noise

Kelvin
- ❑ 7/4 Begin integrating circuits in breadboard
- ❑ 7/8 Measure current, voltages, and compare to theoretical calculated values
- ❑ 7/12 Make necessary adjustments and retest
- ❑ 7/15 Finalize PCB Design

Milestones (Detailed) 1.11. Design Models/Input Test and Continue Design Paper

Turn in Current Draft of Senior Design Documentation	Week 8	**7/8/2016**

Current draft of the design documentation will be turned in.

Entire Team
- ❑ 7/7 Have current pieces of the document assembled
- ❑ 7/8 Turn in current draft of design paper by 12pm

Milestones (Detailed) 1.12. Turn in Current Draft of Senior Design Documentation

Finalize Paper and Begin Ordering Parts	Week 9 - 11	7/29/2016
Document will be finalized in preparation for submission. Parts will be ordered <u>Entire Team</u> ❏ 7/22 Have all pages of final design paper complete ❏ 7/24 Have complete list of required parts and vendors ❏ 7/26 Have document assembled and number of pages checked ❏ 7/29 Peer-review entire document and begin final edits ❏ 7/29 Have all parts ordered (including PCB)		

Milestones (Detailed) 1.13. Finalize Paper and Begin Ordering Parts

Submit Final Document	Week 12	**8/2/2016**
Final document for the course will be submitted. <u>Entire Team</u> ❏ 7/31 Have entire paper completely finished ❏ 8/1 Have paper bounded ❏ 8/2 Submit paper by 12pm		

Milestones (Detailed) 1.14. Submit Final Document

Input testing on patient to calibrate electronics	Week 12 - Break	8/12/2016
Sensor testing must be done on the patient (to be monitored by faculty or Limbitless) to obtain readings and develop thresholds for the arm. <u>Entire Team</u> ❏ 8/12 Have completed input testing on patient ❏ 8/12 Have all required data for thresholds		

Milestones (Detailed) 1.15. Input testing on patient to calibrate electronics

3.2.2.2: Break

Milestones (Detailed) 2 is reserved for the break period. It just shows that there is a space of time between senior design 1 and 2.

Small Break Before Fall Classes		

Milestones (Detailed) 2. Small break before Senior Design 2 starts

3.2.2.3: Senior Design II

Milestones (Detailed) 3.1-3.6 relates to the second semester of senior design. These tables show the work that we have laid out for us in terms of actual implementation of the project.

Regroup and Discuss Plans	Week 1	08/26/2016
We will regroup and go over a detailed plan of how the semester should go. This is important to start the semester on a good foot. Entire Team ❏ 8/22 Discuss new schedules and availability ❏ 8/26 Have discussed any problems that have come up so far ❏ 8/26 Have a detailed plan of the semester		

Milestones (Detailed) 3.1. Regroup and Discuss Plans

Gather All Parts and Finalize Design	Week 2 - Week 3	9/9/2016
We will need to determine what parts we have and set everything in stone to start building our final prototype. Entire Team ❏ 8/31 Perform check of inventory to determine what we are missing ❏ 9/2 Discuss final design as a team and adjust if necessary ❏ 9/7 Perform second inventory check. All pieces should be in (hopefully) Brandon ❏ 8/31 Have pseudocode adjusted for patient thresholds and specifications ❏ 9/9 Have primitive code mockup and adjust bad logic Gavin ❏ 8/31 Have first generation of hardware prototype in hand ❏ 9/9 Determine bugs in initial design Daniel ❏ 8/31 Assist teammates with subsystem testing ❏ 9/3 Coordinate integration testing with mechanical team Kelvin ❏ 8/31 Have schematic/wiring diagram completely revised ❏ 9/5 Verify all components and compare to specifications ❏ 9/9 PCB Tested		

Milestones (Detailed) 3.2. Gather All Parts and Finalize Design

Prototype System	Week 4 - Week 8	10/14/2016

We will build the prototype for the system. It is important that we get it done in this time so that we will have time to test and integrate with the actual arm from the mechanical team.

Entire Team

- ❑ 10/7 Have all individual components complete
- ❑ 10/7 Start integrating parts of the system
- ❑ 10/14 Have prototype built

Brandon

- ❑ 9/21 Controller should be able to at least read valid input and do simple processing
- ❑ 10/4 Controller should be able to convert input to output (even if unstable)
- ❑ 10/7 Have code complete and in working condition

Daniel

- ❑ 10/13 Have designed method of demo in case mechanical team should fail
- ❑ 10/14 Order demo parts if necessary (to be evaluated by Gavin)

Gavin

- ❑ 09/21 If revisions needed order the redesigned components
- ❑ 10/04 Test system using breadboard
- ❑ 10/07 Integrate the multiple sensors and actuators
- ❑ 10/13 Evaluate mechanical team's progress

Kelvin

- ❑ 9/15 Have backup design for system to limit risk
- ❑ 10/3 Evaluate the pros and cons of different power sources. See if current design works.
- ❑ 10/7 Have power system built

Milestones (Detailed) 3.3. Prototype System

Test and Debug system. Begin Final Documents	Week 9 - Week 11	11/4/2016

This is the time that we will debug and finalize our system. By the end of this, we should have an A worthy standalone project. We will also begin our final documentation.

Entire Team

- ❑ 10/17 Start debugging system
- ❑ 10/28 Have operational system for base cases (Ok if edge cases need work)
- ❑ 10/28 Have final documentation started
- ❑ 11/4 Have completed electrical system ready for integration (or submission)

Brandon

- ❑ 10/17 Begin troubleshooting bugs
- ❑ 10/28 Optimize code if needed
- ❑ 11/4 Have bug free code

Gavin

- ❑ 10/17 Begin attempts to integrate system with mechanical team
- ❑ 10/28 Determine areas for change and adjust accordingly
- ❑ 10/28 Order any components needed for change
- ❑ 11/4 Debug and test available system with mechanical team

Kelvin

- ❑ 10/10 Measure current, voltages, and compare to theoretical calculated values
- ❑ 10/17 Make necessary adjustments
- ❑ 10/28 Have stable system

Milestones (Detailed) 3.4. Test and Debug system. Begin Final Documents

Integrate with Mechanical Team and Test	Week 11 to 14	11/25/2016

In this phase, we will integrate our design with the mechanical team and test it as a whole.

Entire Team
- ❑ 11/8 Know if integration with mechanical team will be feasible based on their progress
- ❑ 11/8 Continue to work on documentation
- ❑ 11/22 Have All documentation complete

Brandon
- ❑ 11/18 Adjust code as necessary

Gavin
- ❑ 11/8 Determine if project can be successfully combined with mechanical teams design

Daniel
- ❑ 11/11 Adjust requirements as necessary
- ❑ 11/18 Adjust testing plan as necessary

Kelvin
- ❑ 11/11 Determine that the system has enough power to run components efficiently
- ❑ 11/18 Relocated wiring if necessary for performance/aesthetics/easiness

With Mechanical Integration

Entire Team
- ❑ 11/18 Have system integrated and begin debugging (if not started)
- ❑ 11/25 Have system fully integrated and stable

Gavin
- ❑ 11/11 If design can be successfully integrated, continue integration
- ❑ 11/18 Fully integrated system complete
- ❑ 11/25 Fine tune system

Without Mechanical Integration

Entire Team
- ❑ 11/18 Have initial demo system setup and begin debugging
- ❑ 11/25 Have demo system fully working and stable

Gavin
- ❑ 11/11 If Design cannot be fully integrated with mechanical team's design test rig for presentation
- ❑ 11/18 Build test rig
- ❑ 11/25 Fine tune system

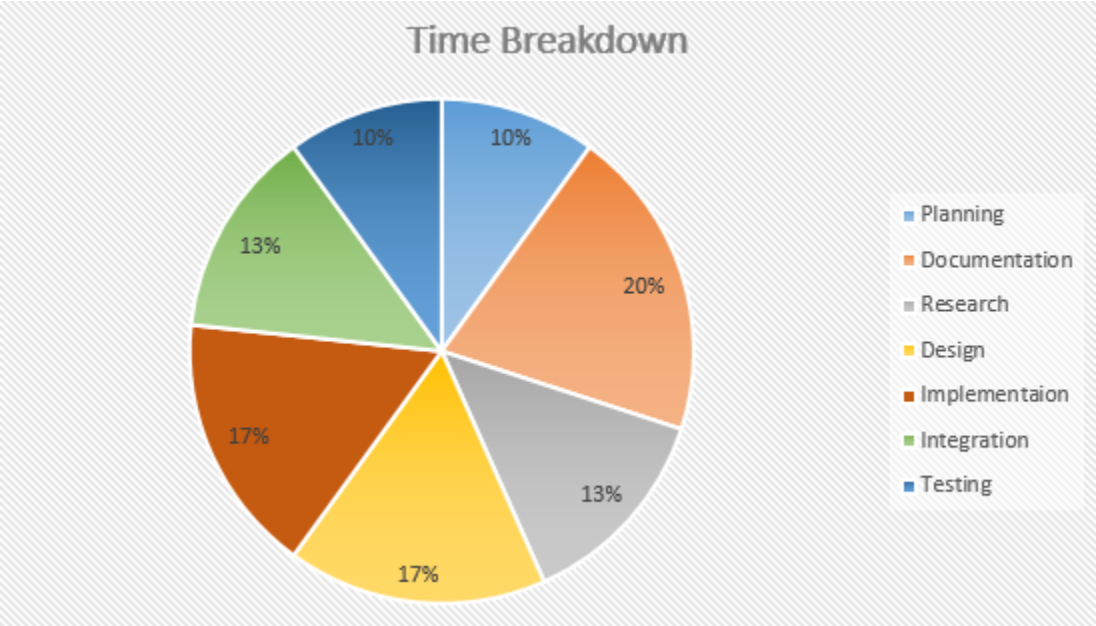
Milestones (Detailed) 3.5. Integrate with Mechanical Team and Test

Final Presentation/ Evaluation	Week 15	**12/1/16**
We will have a fully working and presentable demo. We will also have all of our documentation in order. We will make a final presentation to end the course. <u>Entire Team</u> ❑ 11/28 Make amendments to design paper based on final design ❑ 11/29 Have presentation rehearsed at least 3 times ❑ 11/29 Have demo run through at least 3 times ❑ 12/1 Give demos ❑ 12/1 Give presentation		

Milestones (Detailed) 3.6. Final Presentation/ Evaluation

3.3: Milestones Breakdown

When the project began, we knew that we would need to plan everything out very carefully if we were to be successful. That is why the milestones were very important to us. They allowed us to make an organized plan for the future, and forced us to think about problems that may arise in the future and how to solve them quickly. For this reason, we put effort into not only the selection of the milestones, but also the timing. A breakdown of this timing can be seen in diagram *Milestones Breakdown 1*.



Milestones Breakdown 1: This is a time breakdown of how we organized senior design I & II. The most time is giving to documentation, with design and implementation tying for second.

4: RESEARCH

4.1: Input Research

4.1.1: Understanding Movement

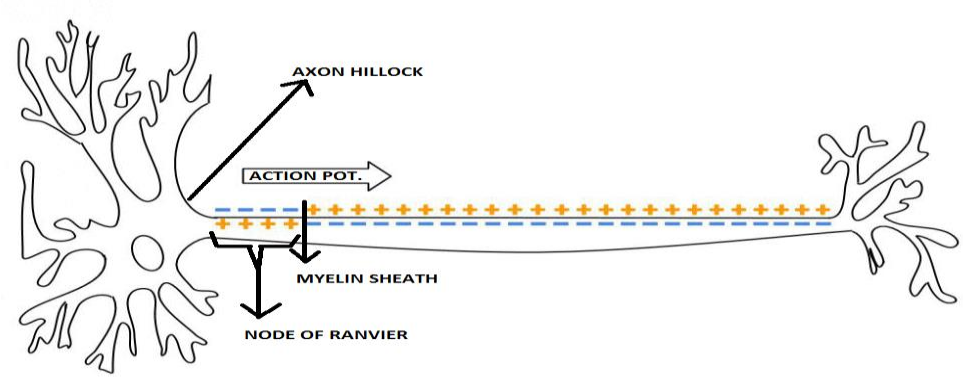
Biomedical applications on humans contain increasing degrees of difficulty. This is due to the fact the human body is a complex system and has many factors to account for that are not clearly understood at the moment. In order to design an exoskeleton that replicates the motion that is executed by a biological system, as would be an upper limb in the scope of this senior design project, it is imperative to know to some extent how the human body works to be able to design an exoskeleton arm that will recreate the motions of a biological arm; more specifically, it is imperative to know what causes an arm to move.

4.1.1.1: Resting Potential

Muscle fibers are connected to efferent neurons, or motor neurons, which are responsible for transmitting movement signals. All neurons contain Na^+ and K^+ channels and Na^+/K^+ pumps embedded in their cellular walls, which transport these ions in and out of the cell. Varied concentrations of positive and negative ions produce a difference in potential, or voltage, across the cell's plasma membrane. At rest, a neuron is at a resting potential of -70mV [1].

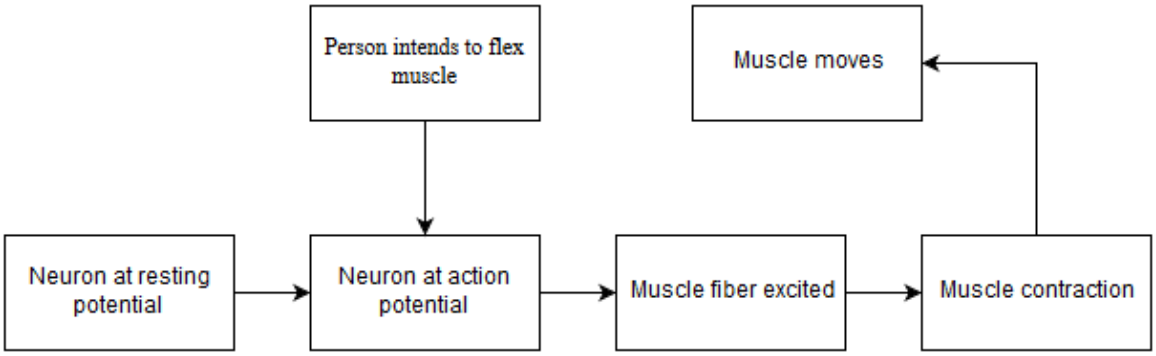
4.1.1.2: Action Potential

When a person attempts to move a muscle, the neurons in the brain receive a stimulus caused by a variation in ion concentrations; thus, the resting potential of the neuron changes to approximately +30 mV as a result of a depolarization of the cell [2]. This change in voltage occurs at the axon hillock, located at the cell body. Since the axon is made of different current-isolated sections called Nodes of Ranvier, the depolarized section of the neuron at the axon hillock causes the potential the adjacent node to become depolarized as well. As successive nodes become depolarized, the positive voltage becomes a signal that travels down the axon, while the sections that were depolarized polarize back into resting potential due to another change in ion concentrations [3]. For this reason, the signal will only travel in one direction. This potential is known as action potential and it does not drop due to the fact that it is not being propagated through current but rather through the potential itself. The *Understanding Movement 1* sketch retrieved from Wikipedia best shows how the neuron is structured as well as how the action potential works.



Understanding Movement 1: A sketch that shows neuron structure and how action potential works
Image from Wikipedia: Action Potential. User: Laurentaylorj [I-1]

The action potential that occurs at the excited neuron causes a signal to be transmitted electro-chemically to a synapse that connects the excited neuron to another neuron. The process continues to transmit signals across neurons in the same manner until it reaches the motor neuron. The motor neuron then sends the signal to the muscle fiber through a synapse called a neuromuscular junction. When the muscle fiber, or myofiber, receives the signal being sent by the motor neuron, it becomes stimulated and contracts. The more fibers that are stimulated, the stronger the signal transmitted to the muscle and the more muscle fibers that will contract, causing a stronger muscle contraction. Due to signals being sent electro-chemically through synapses, the potential degrades as the signal makes its way to the postsynaptic cell. For this reason, the muscle fiber does not possess the action potential of the motor neuron in its entirety. The *Understanding Movement 2* diagram below summarizes the process being done when moving a muscle.



Understanding Movement 2: A block diagram that shows the process behind muscle movement

4.1.2: Problem Statement

There are certain conditions that may affect the interaction between neurons and result in the signal not reaching the muscle fibers for excitation, or the signal that reaches the muscle cells being too weak to allow functional contraction of the muscles; these behaviors of limited mobility may be caused by death or inexistence of neurons at the synapses between neurons or motor cells. The scope of this project will be based on a patient that still receives motor signals but are not strong enough to provide functional mobility.

The patient suffers from AMC. Under this condition, the patient has decreased flexibility of the joints and a weaker muscle contraction [4], thus resulting in a weaker signal being sent to the muscle cells. Additionally, the patient's muscles have degenerated from reduced use. To determine the input that will be fed into the microcontroller for motor actuation, the team has created a series of electromyography tests, which are stated in this document and will be run on the patient before the implementation stage begins to obtain data related to the electrical signals generated on the muscles, such as adequate placement of the electrodes and signal acquisition.

4.1.3: Signal Acquisition

To provide motion for the artificial actuators, electrical signals produced in the body will be read as an input and sent over to the exoskeleton's logical system for mechanical actuation. There are several techniques used for signal acquisition as described below; however, many of these techniques require invasive procedures on the patient in order for the signal to be acquired. Any electrophysiology method implementation that requires electrodes being inserted into the brain or the muscles, while more accurate, are in direct violation of end-user requirement 1. Additionally, surgical procedures would require more overhead and complexity, and are well beyond the scope of this project.

EEG is a monitoring method that can be non-invasive and records electrical activity directly from the brain. Electrodes are typically placed along the scalp, and detect electrical signals in the brain as impulses are generated [5]. Since the electrical impulses that are generated in the brain will naturally be stronger, EEG is capable of acquiring the motor signals more accurately. On the other hand, an EEG requires more electrodes to be placed in order to accurately read all the electrical signals produced in the brain to determine which combination of signals map to a specific muscle input; this could result in a higher demand from the power requirements and the micro processing unit. Proper testing of signal acquisition would also be more complex, since the team would have to evaluate and determine accurately which signals constitute a motor impulse. Additionally, the patient would have to wear a cap containing all the EEG electrodes every time the exoskeleton arm is used.

EMG stands out as a simpler, more efficient approach to non-invasive signal acquisition. In a high level explanation, surface electrodes are applied to the muscle of interest and record any electrical activity that occurs [6]. Although electrodes are typically implanted with needles through the skin, surface electrodes could be placed on the arm instead for a

non-invasive procedure. The EMG approach would require less sensors to be placed on the patient, as most of the electrical activity in the muscle pertains to motor stimulation. Furthermore, the sensors could be easily concealed in the exoskeleton arm's framework, providing a more aesthetic design for the patient to use. Lastly, this method of reading muscle inputs would allow a more direct mapping between the inputs and the microcontroller, which will result in a simpler system. EMGs are typically less accurate, however, as the muscle cells are shielded by more layers of fat and skin, and the signal is usually weaker by the time it reaches the muscle fibers. Overall, the use of EMG sensors shows a better promise for the scope of this project, as shown by *Signal Acquisition 1* decision matrix below.

		EMG		**EEG**	
Criteria	Importance Weight %	Rating (0-5)	Weighted Rating	Rating (0-5)	Weighted Rating
Power Efficiency	15%	5	0.75	4	0.6
Signal Quality / Bandwidth	25%	4	1	5	1.25
Maintenance	10%	3	0.3	3	0.3
Low Cost	10%	5	0.5	3	0.3
Programming / Signal Processing	30%	5	1.5	3	0.9
Aesthetics	10%	4	0.4	3	0.3
Total	100%	4.45		3.65	

Signal Acquisition 1: Decision Matrix for EMG/EEG choice

4.1.4: EMG Implications

Due to the complexity of the human body in terms of expectations and behaviors, EMG calibration cannot be adjusted to fit any user's needs toward the exoskeleton arm; assessment must be made to fit each patient individually. Many factors must be taken into consideration when selecting the best products for this particular design. Moreover, the analogical nature of the project makes assessing the patient's condition in high detail a necessity in order to create a functional system. If the patient were to use an uncalibrated design that did not require previous evaluation of the user's capabilities, safety hazards would most likely be produced. For this reason, sound testing procedures must be taken. The most important benefit from measure testing is that it will allow the team to properly calibrate design to fit the patient's needs and capabilities. Another key advantage of assessment tests is that they will help determine what needs to be measured in order to properly constrain the system.

When creating test scenarios for data collection from the patient, it is also important to know what factors might affect the quality of testing. Adequate research assists the designers in providing accurate scenarios where significant data will be obtained to calibrate the system. Consequently, the team has worked with the mechanical/electrical interface lead for the mechanical team to provide a viable test bed that will be used to

obtain valuable information from the patient. A research paper prepared by Dr. Scott Day [7] explains in a great deal of detail what these factors are, along with several ways of eliminating them from the equation. Some of these factors that need to be addressed in order to create the test bed are explained below.

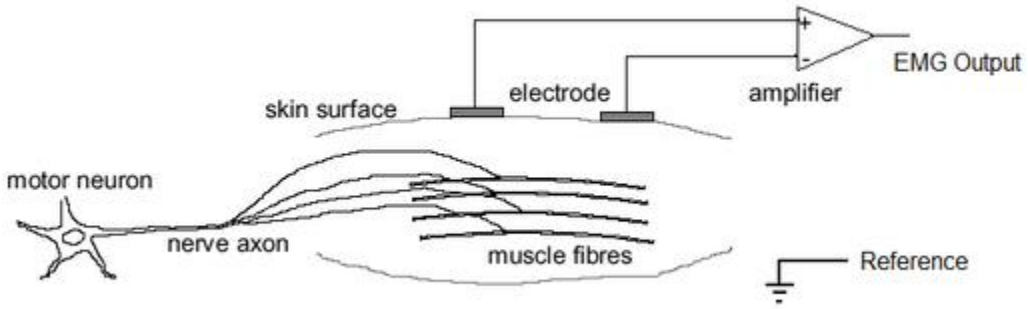
4.1.4.1: Noise

Noise is one, if not the most important factor to consider when performing testing. Dr. Scott Day's document [7] differentiates between two types of noises that are typically present in EMG readings: ambient noise and transducer noise

Ambient noise is produced by the presence of electromagnetic devices [7]. These devices range from computers to basically any device that connects to the wall. As expected, these noises occur at the frequency of 50 Hz, which corresponds to the U.S.'s standard A/C frequency. This type of noise is important to consider, especially since most of the data will be collected by a computer; however, it might not become an issue, especially since communication between the EMG and the computer is planned to be done remotely.

In short, a transducer noise is produced at the electrode-skin junction [7]. It consists of a D/C potential noise and an A/C potential noise. The D/C noise is caused by a constant difference in impedance between the skin and the EMG sensor, while the A/C noise is a consequence of fluctuations in impedances between the conductive material and the skin.

To maximize the SNR from the EMG reading, bipolar electrodes are desired [7]. This technique consists of using two electrodes at nearby locations within the same muscle; one is placed right on the belly of the muscle, while the other electrode is located on a close area along the muscle where the readings are coming from. They are connected to a differential amplifier that attempts to eliminate common signals, such as those produced by noise, and then amplify the difference. The concept behind the use of bipolar electrodes is that, since they are in close proximity and are reading from a common set of muscle fibers, they are likely to be burdened by the same type of noises. *EMG Implications 1* below was retrieved from NRSign Inc. and shows how the bipolar electrodes are used on the muscle.



EMG Implications 1: Diagram that explains how bipolar electrodes work
Image from NRSign.com: Monopolar vs. Bipolar EMG Readings [I-2]

4.1.4.2: Skin-Electrode Impedance

When performing tests using bipolar electrodes, it is important to keep in mind that there are still inherent impedances at the skin-electrode interface. Although these impedances are rather trivial when bipolar electrodes are used due to noise cancelation, they are always expected to be at roughly balanced levels. To illustrate, if there was a lot more impedance at one of the electrodes, the signal strength would be affected as well and some of the noise might then produce a signal that is not common to both electrodes and thus not be filtered out by the differential amplifier. What is more, the noise that is not cancelled out will be amplified, producing undesired results. Therefore, having similar impedances at the skin-electrode junctions results in more reliable data. From Dr. Day's research paper, two types of skin-electrode impedance factors were determined: variable impedance factor and static impedance factor.

Variable impedance factors relate to changes in signal quality with regards to a relationship between frequency components and impedance. From Dr. Day's research [7], it has been shown that an impedance lower than 10 kΩ resulted in a higher signal energy for frequency components under 100 Hz, when comparing with a higher skin-electrode impedance. In contrast, an impedance that was higher than 100 kΩ showed a better signal acquisition for frequency components in between 100 Hz and 150 Hz.

Static impedance factors rely on changes in signal quality due to impedances that might be present at the time of measurement and are not strongly dependent upon how these measurements are performed sequentially. These factors include the presence of dead skin material and oil secretions and constitute an important concern in the development of a D/C potential noise. To solvent this issue, it is always recommended to prepare the skin where the electrodes will be placed. Examples of skin preparation include shaving the skin area where the electrodes will be placed, washing the area with antibacterial soap and water for the elimination of dead cells and other impedance materials, and rubbing the area with alcohol to prevent the skin from excreting oil.

4.1.4.2: Cross Talk

This is one of the most important factors that will be taken into account when performing tests on the patient, especially given the fact that the patient for whom the exoskeleton arm is being designed has underdeveloped muscles due to low motor activity of his arms. When an electrode is placed on a group of muscle fibers that is in close proximity to another muscle, the possibility of receiving unwanted signals from that muscle arises. This might provide an undesired noise that might not be cancelled out by the differential amplifier when readings are performed.

Several solutions are proposed to help circumvent this issue. The first solution would be to use smaller electrodes that would be more specific to a particular muscle set; however, due to their small area, the electrodes may also lack enough conductive area and not be able catch the same strength of signal provided by more muscle fibers. Another solution is to choose an optimal distance between the two electrodes. This section will be explained in depth in this paper's section for optimal electrode selection.

4.1.4.3: EMG Normalization

For input-output mapping design purposes, EMG normalization is the most important factor to consider. Since there is a large number of factors that can affect the patient's EMG signal that is received from his muscles over a period of time, it is hard to collect exact data that allows for direct mapping of physiological inputs to artificial outputs, as would be the motors in this case. According to Dr. Scott Day's document, an absolute amplitude for the signal is unreachable due to these factors [7]. For this reason, it is significant to collect successive pieces of information for the same test over several testing periods; so that the team can normalize results and come up with a better understanding of the patient's physiology in order to provide an appropriate input-to-output mapping for the actuation system of the arm.

4.1.4.4: Electrode Selection

Apart from the fact that knowing which electrode to select for testing also aids in the factors to consider for design, it is ideal for the test performers to know some information regarding the electrodes to perform a precision testing. This section is divided between types of electrodes, electrode shape, electrode size, inter-electrode distance, electrode construction, and electrode placement.

4.1.4.4.1: Types of electrodes

Choosing the right type of electrodes will play an important part, not only in the test bed for the patient, but also for the design. There are currently two types of electrodes that can be used for this purpose: gelled electrodes and dry electrodes. The design section of this paper will specify which electrode will be used for the scope of this project.

Gelled electrodes are fundamentally electrodes that employ a conductive gel material to act as an interface between the skin and the metallic material. Due to the AgCl layer that connects the electrode and the conductive gel, they allow current to pass more freely across this junction, thus increasing the SNR. They are typically light in weight, measuring less than 1 gram, and can be both disposable and reusable [7]. Although disposable electrodes will be likely used for testing purposes due to their lightweight and easy application, reusable electrodes offer a better choice for the final design, since they will not be required to be changed.

Dry electrodes do not possess a conductive gel to act as a better conductor for signals from the muscle impulses to the electrodes themselves; as a result, they tend to produce a lower-quality signal. Additionally, they come with the differential pre-amplifier housed within the electrode structure, adding as much weight as 20 grams. This variant of electrodes is also more susceptible to noise produced by the electron being heavier and not fixed into place.

4.1.4.4.2: Electrode Shape

The SENIAM has recommended several standards to be used in the proper selection of electrodes for testing; one of the standards relates to the electrode shape. When choosing electrodes, the team must ensure that all electrodes used on the patient are of the same shape; different shapes among different electrodes could be subject to different input impedances.

4.1.4.4.3: Electrode Size

Size is a very useful consideration when selecting which diodes to use for testing or design. An increase in electrode size along a muscle is generally good for signal acquisition; however, one must always keep in mind that external factors might counter-influence the ideal size for the electrode. In other words, an electrode should be big enough to capture as many muscle fiber signals as possible, but at the same time small enough to avoid cross-talk muscles from interfering. According to SENIAM's recommendation, a suggested electrode size would have to be less than 10 mm [7].

4.1.4.4.4: Inter-Electrode Distance

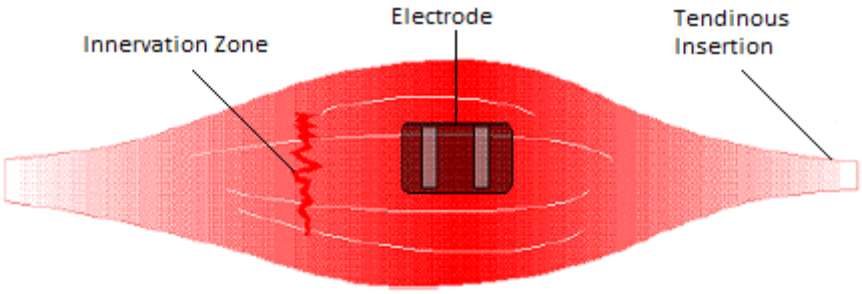
Inter-electrode distance is defined as the distance between the center of each electrode in a bipolar configuration. According to SENIAM, an ideal inter-electrode distance would be somewhere around 20 mm apart; however, for small muscles, the inter-electrode distance should not exceed ¼ of the muscle length in order to avoid unstable readings.

4.1.4.4.5: Electrode Construction

When elaborating a test bed for the patient, this is one of the factors that must be considered in order to avoid unexpected behavior such as that occurred by movement artifact. The best alternative to preventing test material on the patient from creating this kind of interference is by attempting to fix the electrodes in place. Disposable electrodes are very useful for preventing movement artifact due to their adhesive material, which sticks to the skin. Double-sided tape can also be used to ensure that the inter-electrode distance stays the same. For any cables that are connected to the electrodes, tape can be used to fix them in place as well, so as to avoid dangling of the wires producing artifact movement.

4.1.4.4.6: Electrode Placement

Clever placement of electrode will not only ensure a proper testing of inputs from the patient, but it will also help the testers determine which muscle groups are optimal for use. Consistency in the placement of electrodes on the patient over several tests will guarantee a more normalized testing sample. Additionally, the testers should always attempt to focus on a muscle of interest in an attempt to map the muscle input to the exoskeleton's actuation system as directly as possible, without having to use a less direct muscle for actuation. Lastly, when reading EMG signals from the muscles, it is required to avoid innervation zones. As shown in *EMG Implications 2* after this paragraph, innervation zones are located close to the end of the muscle. This image's credits go to Delsys' D. Luca, J. Carlo.



EMG Implications 2: Electrode placement on the muscle belly and innervation zones
Image from Delsys.com [I-3]

4.2: Software Research

4.2.1: Microcontroller Selection

4.2.1.1: Factors

The selection of a microcontroller is something that had to be taken very seriously. It will be responsible for being the brain of the entire system and thus should be able to handle certain needs. There are a few metrics that we used to give quantitative values to some properties. A few of these include ease of use, low cost, number of I/O pins, processing speed, availability of external components, power requirements, and support community.

Ease of use is the first metric. This is important because we only have a limited amount of time to complete the project. We need to use a controller that can be learned and utilized within this time. Also, there are many variables in our project coming from outside forces, such as patient needs or last minute changes in the mechanical team's design. We need to be able to make quick code changes on the fly without having to restructure our entire code base. This can be difficult in certain situations like when the code implementation tries to take the programmer too low in level and does not abstract enough details.

Another factor in controller selection was cost. Though this project is sponsored by Limbitless Solutions, there is still a budget to follow. The ideal controller is not going to cost too much and blow through the budget. One important thing to keep in mind, however, is that while we are looking for a great price, we do not want to sacrifice quality in the process. We only looked at genuine parts from the manufacturer and not the cheap clones that are sold across the internet.

The next metric is number of I/O pins. The implementation of our design requires many inputs and outputs. The EMG sensors alone should take up quite a few. In addition to this, we want to have a few left over in case we need to add more components or are given extra requirements from the mechanical team. Originally, we had planned to use an SPI design with a master and slave controller. Having two controllers would alleviate

some of the pressure of supporting enough pins. However, when we decided a single controller would be better, pin count became a serious concern once again.

Processing speed is also a factor. Because our system is going to be attached to and used directly by a person, the response time needs to be very fast. We need a controller that can provide an adequate clock speed. It needs to be able to sample input at a reasonable rate and run calculations fast enough to process it to output before the user can experience noticeable lag.

Next, we examined availability of external components. We wanted something that would be easy to attach things to, especially for testing purposes. Some of the issues faced were, whether or not the screen would be able to display information. Also, if there were adapters and relays. Another design consideration was whether or not Wi-Fi would be a good fit. Then there was the question of whether there are libraries supporting the code for this project. Some microcontrollers had better information regarding these questions than others.

The power requirements of the controller is important because we need to limit the drain on the battery. This device needs to be available for use hours at a time. We didn't want something that would consume too much battery life and make it harder to reach this objective. We looked in controllers that supported things like "Low-Power-Mode" or that drew minimal amounts of energy in standard usage.

The last factor we looked at was community support. This factor could easily be forgotten or overlooked, but we found it to be very important for a few reasons. In order to keep down the price of this system and stay within budget, we aren't able to purchase extensive documentation (data-books, manuals, standards, etc.). Also, the time it could take to read through everything and pull out the useful information could be very lengthy. Lack of support would also mean that simple things could become very difficult when we have to use specific bit codes and implement things at a low level. A large support community means that things have been done before. If we hit an obstacle, we want to be able to find solutions relatively quickly. There is also expected to be a large number of support libraries to assist with implementing common things, such as reading to an output screen.

4.2.1.2: Comparison

With these factors in mind, we were able to start examining our options. Both the mechanical team and Limbitless Solutions had done some research on the topic before we joined the effort, so we were able to pull some things from them. In addition, Limbitless already had certain controllers in hand and a code base we could use for reference. Based on what was available, as well as their research and some research from our team, we were able to narrow our selection down to two micro-controller families: The Adafruit Arduino & the Texas Instruments Launchpad.

To start the comparison, we chose the most popular micro-controller board in each family and examined features from a high level, knowing they could differ between models. These were the Arduino Uno and the TI MSP-EXP430G2. In the first category, ease of use, Arduino was the clear winner. According to our research, people generally have an

easier time programming and working with the Arduino than the TI Launchpad. We also did a comparison of the user-friendliness for the IDEs of each and looked at some sample code to see which was more comprehendible. The one thing that did favor the Launchpad in this category was that some of the members on our team were already familiar with the syntax. However, we decided that this did not negate the flexibility of the Arduino and the ease we would have implementing new things outside the bounds of our pre-existing knowledge.

Examining cost was something else that we had to do. The Uno retails at about $30-35, while the MSP-EXP430G2 is just $10-15. In this category, the Uno could not compete. Texas Instruments, being a much larger company than Adafruit, is just able to make these things much cheaper. Initially, when we had planned to use multiple controllers, this was a huge concern. Once we decided to scale it down to one, the $20 difference, while still notable, did not seem as substantial.

A comparison between the numbers of I/O pins was also done. The Uno has 20, while the MSP-EXP430G2 has just 16. This is a 20% difference. We knew that with the scope of our system, every pin we can get/spare is important. This is why we gave this category a higher weight than the others.

The review of processing speeds yielded no useful results. Both controllers are capable of reaching a 16 MHz clock rate (though this is impressive for the Launchpad, given its price). The Launchpad uses the MSP430 family of chips for processing, while Arduino uses Atmega. Both, were found capable of doing what we needed them to do for the implementation of the system.

The Arduino was better when it came to availability of and compatibility with external components. During our research, we did find some components that were good for TI controllers, but overall, much more where available for Arduino. Arduino is stackable and has components called “shields.” What this means is that, certain components can be attached right on top of the Arduino and can be “stacked” vertically into one piece. The availability of these components is astounding, with many third-party manufacturers making these items for low-prices under $20.

In the area of power needs having a micro-controller that could draw less power would be optimal, the Launchpad won this contest. The reason for this is that the Launchpad has what are called “Low Power Modes.” When the processor is not doing anything, it can go into one of these modes, where it will turn off certain things and save power. When it needs to run at its full capacity again, an interrupt can be triggered which will disable low power and allow the processor to do what it needs to do. This is a very useful feature and one that should not be undervalued. With this being said, the Arduino still is not terrible. However, it still has a disadvantage in the power consumption department.

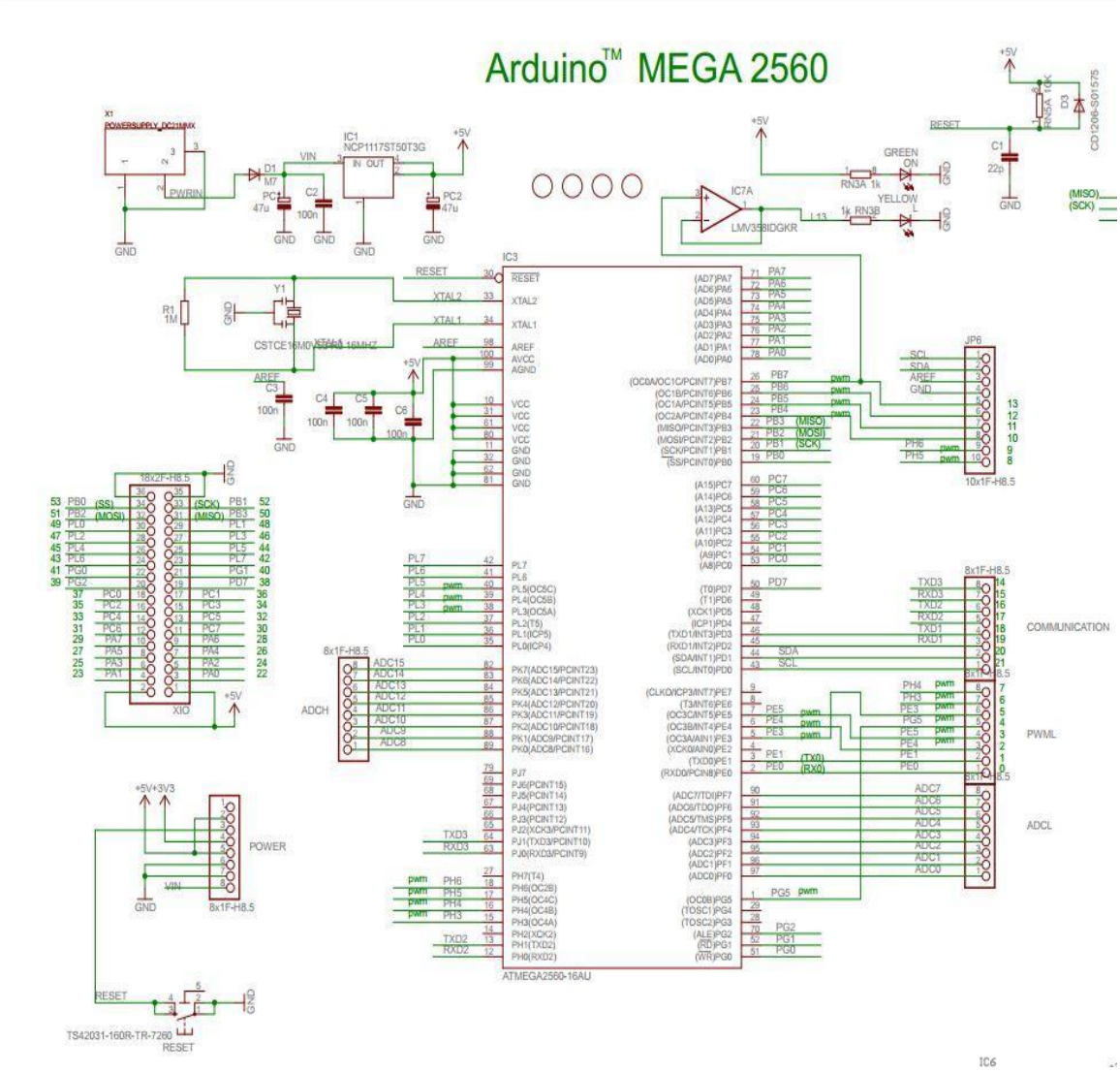
The category where there is the largest difference is community support. The Uno clearly wins. Arduino has information and examples everywhere. It seems as if most people that get into embedded development have done at least a few thing with Arduino and there are many who love to talk about it. Online forums are filled with people answering questions, giving advice, and posting code samples. In addition to this, there are many open-source libraries available for use that can make development much easier and more

straightforward. The Launchpad, however, is the opposite in ways of support. There are a very few individuals and resources out there, but most of the information that one will require to complete a project will need to be located directly from manuals or data-books (which TI does provide plenty of). This can become very frustrating if 20 pages must be read through in order to figure out how to do one function.

With the results of these comparisons, we were able to form a decision matrix and tally up scores for each controller. The matrix is shown in Figure *Microcontroller Selection 1*. From this matrix we see that the winner, by 1.15 points, is the Arduino. Once we decided this, we were able to decide which model we need specifically based on our system specifications. With everything that is to be included, we ending up opting for power and a large number of inputs with the Arduino Mega2560. This board houses an Atmega2560 chip and has 54 I/O pins. It is perfect for our system. A schematic from Arduino can be seen in figure *Microcontroller Selection 2*.

		Arduino		TI Launchpad	
Criteria	Importance Weight	Rating	Weighted Rating	Rating	Weighted Rating
Ease of Use	20%	5	1.00	2	0.40
Low Cost	20%	3	0.60	5	1.00
# of I/O pins	25%	5	1.25	3	0.75
Processing Speed	15%	4	0.60	4	0.60
Availability of External Components	5%	5	0.25	3	0.15
Power Needs	5%	4	0.20	5	0.25
Large Community	10%	5	0.50	1	0.10
Total	100%	4.4		3.25	

Microcontroller Selection 1. This is the decision matrix between the Arduino and the TI Launchpad as a microcontroller for the system



Microcontroller Selection 2. This is a schematic of the Arduino Mega2560.
Image from Arduino.cc [I-4]

4.2.2: Code Research

4.2.2.1: Basic Code

4.2.2.1.1: Initial Code Research

Once we decided on a microcontroller, it was time to learn how to program it. When we chose to work with an Arduino, we knew that no one on our team had any previous experience with it. We made a decision that we would do what it takes to learn and would put in the effort. From there, it was crucial that we make a plan of action and make strides to follow that plan.

The plan was developed at the same time as the milestones, so they reflect the thought process pretty closely. The milestones start off with basic things such as installing the IDE and then graduate to things like being able to use the Arduino to operate an LED.

Eventually, we should be able to write complex code and have a full understanding of how everything works. Only after this do we start developing pseudo code for the system. We want to make sure that we are designing code that makes sense and follows the convention of the micro-controller hardware.

The first step that we made was to download an IDE. We decide to use “Arduino” as it is the standard IDE for the Arduino microcontroller and would offer a lot of compatibility. We obtained the software at no charge directly from Arduino’s website and installed it on all computers that would be used for code development. It is a surprisingly light program and runs on even the slowest computers. The next step was to choose the language. This was easy as there are already standards in place. We decided that the code base will be written in C/C++, using specific libraries for the Arduino platform. We thought this made the most sense as we did not wish to deviate too much from the accepted way of doing things.

The next step was to start watching some videos. While exploring the web for research, we found a series of very valuable videos on YouTube. They are by user, Martin Lorton who goes by the user name mjlorton [8]. He has a wide assortment of playlists on his channel. There is one in particular called Arduino Tutorials and Projects [9] in which we goes over some basic Arduino information. He uses the UNO to model, but since the UNO and MEGA2560 models are somewhat similar, we were able to adapt what he was saying to our controller. The difference between this series and others like it are that he doesn’t focus on the code itself. Examples can be found all over the internet so it really isn’t that helpful to put it in a video. Instead, he goes over things that aren’t that obvious and explains issues that one might run into while working with an Arduino. These topics ranged from the computer not recognizing the device during debug to having issues integrating with an external LCD screen.

His videos also go over initial setup of the board, downloading libraries, proper terminology, and basics of what the code structure should look like. They even included some different ways that you can hook it up to hardware for some practicality. His most useful sections, in our opinion, were the ones on troubleshooting. Overall, this was a good series to watch and only took around 4 hours in total. It was a good place to get our feet wet with Arduino.

4.2.2.1.2: Breadboard Code Testing

After this initial research it was time to do some real life testing. The first thing we did was purchase an Arduino for testing. For testing purposes we get an UNO because it came with an entire electronics kit that could be used. This kit contained a breadboard, wires, resistors, capacitors, diodes, transistors, LEDs, and anything else that an electrical or computer engineer might want to begin creating models. Again, because the UNO is so similar to the MEGA2560, we did not worry about differences at this point of the project.

When we were ready, we took out the Arduino and the breadboard and got to work. The first thing we did was test that we could connect the board to the computer. We plugged it in via USB connection and the computer recognized it. We opened the IDE and encountered our first problem when we tried to run some example code. The IDE was looking for the wrong COM Port. We adjusted this and made sure that we could

communicate with the Arduino. This test passed. We were now ready to test some code and get some practice.

We created a new sketch and started coding based off of what we had seen in the videos as well as our prior C/C++ and embedded programming experience. We thought the first thing to do was to test if we could print to the serial monitor. We created a simple loop that simulates 3 lights; green, yellow, and red. Each light flashed in order, with a one second pause in between. To simulate this, we printed the color light that would be flashing. This was pretty simple with the use of the Serial.println function.

Once that was working, we tested controlling outputs and decided to actually implement the light pattern previously discussed. We pulled out a breadboard, some resistors, LEDs, and wires and connected everything up in a circuit. We attached the LEDs to pins 8, 9, and 10 on the UNO. Next we added to the structure of the code using digital writes. We had to keep in mind that the LEDs turned on from a high signal. Soon the light pattern was no longer just a simulation, but a working prototype. After this we decided to connect a power relay, as one would need to be added to the actual system and we wanted to get a sense of how to use it. For the relay, the activation happens on low signal, so we had to program it the opposite way of the LEDS. We wired it to pins 11, 12, and 13, as well as the 5 V and ground pin of the UNO. Code was added that made the different parts turn on in correlation to the lights on the board.

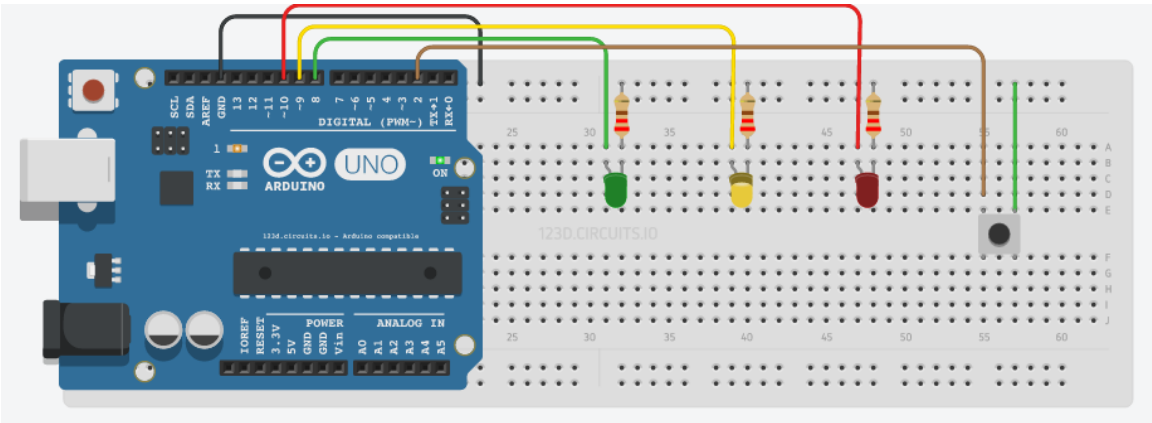
Following this, we wanted to see how we could take input. We wired a push button to the breadboard and initially connected it to pin 7 on the Arduino. The button needs a low signal to be considered “pushed.” We coded it this way and soon noticed that it was only semi-responsive. We realized this was because it had to be pushed in the exact same moment that the piece of code checking for it was running in order for it to really work. The last thing we tested during this session was the use of interrupts.

4.2.2.1.3: Interrupts

The interrupts were the most difficult thing to do, because we had to understand how they work for the Arduino as far as the syntax and difference commands that need to be in the code. There is a function, “attachInterrupt” that is used to set the interrupt service routine. This was set and the function was written. An unexpected setback we had was not knowing that not every pin was capable of triggering an interrupt on an Arduino. We had to move the button from pin 7 to pin 2. Once everything was reconfigured, we tried it again. We set the code so that a boolean variable could be set and unset from the button push. If the boolean was true, then the program would run in a closed loop of nothing, essentially freezing the program, otherwise the program would continue. The problem with this is that once the code entered that loop, the boolean could not be changed by the interrupt to break out. The solution to this was the make the boolean volatile. Once this happened, everything went according to plan. If the button was hit, the lights would stop in their current position. When the button was hit again, the lights resumed. It was a good day as far as initial testing of functionality went.

A diagram of the structure for the test code written can be seen in figure *Code Research 1*. A diagram of the breadboard setup (excluding the power relay) can be seen in figure *Code Research 2*. The code was written using the Arduino IDE and the schematic was made using Autodesk.

Code Research 1. This is a diagram of the code written to test the basic functions of the Arduino. These functions include input, output, and interrupts.



Code Research 2. This is the diagram of the breadboard setup that was used to do the testing. This diagram does not include the power relay or the serial monitor.
Image created using Autodesk [I-5]

4.2.2.2: Analog Input and Pulse Width Modulation

The first steps in learning how to program the Arduino were complete. Now we could focus on some other details and try to do some more advanced things. Our system isn't going to be made up of LEDs and push buttons so we would need to learn how to do things that were a little more practical. We thought about how the system was going to work and determined a few areas where our initial code research and testing would fail us. These would be the next topics of focus. The problem we found was this: The system has to know how much signal it is receiving from the user and then must do the appropriate amount of work (which is impossible from a strictly binary/digital signal. The solution came in the forms of analog input and pulse width modulation.

4.2.2.2.1: Analog Input

The first part of the dilemma was that we need to be able to know how much signal the user is giving the system. In the breadboard setup we did previously, the pins on the Arduino either received signal or didn't. This did not allow us to know the levels of each one received. We solve this with Analog pins. The Arduino Mega2560 has 16 analog input pins and these pins have the ability of not only returning if a signal is received, but also the voltage from 0 to 5 V. This is very useful.

The idea is that the user will be hooked up to various EMGs. These EMGs will send electrical signals whenever a muscle is flexed. The Arduino will receive these signals in its analog pins and know the voltage of each one. We can then process them differently than if they had come through a digital pin and lost the information.

The function to read an analog pin is analogRead(). There is an analog to digital converter inside of the microcontroller that the signal goes through and a result will be returned based on the voltage of the signal. The signal should be 0 to 5 V and the returned value will be between 0 and 1023, proportional to the voltage. 5 divided by 1024

is .00488. This means that the voltage is going to be (roughly) the returned value multiplied by 4.9 mV.

With this, we know how much voltage is being given from the input, and therefore how hard the muscles are flexing. This opens up new things that we can do, such as set thresholds for what is considered valid input and the move the arm at variable speeds for how hard the muscle is flexing. The next challenge was determining how to keep this signal as analog when delivering to the system outputs instead of converting it to digital. Our first thought was to use some of the analog pins as outputs and for all intents and purposes keep the signal analog all the way through. The issue with that technique is that we would run into the problem of possibly running out of pins. Between all of the EMGs and the feedback sensors, we just wouldn't have room considering the Mega2560 only has 16 analog pins. To solve this problem, we discovered pulse width modulation.

4.2.2.2.2: Pulse Width Modulation

Pulse width modulation (PWM) is a means of simulating analog signals by digital means. The digital signal is output in a square waves, which can mimic voltages between full on (5 V) and full off (0 V). This is done by changing the proportion of time that the signal is on vs. off. The duration of time on is called the "pulse width." You can modulate this width and as a result get a signal that acts as a steady voltage between full on and full off.

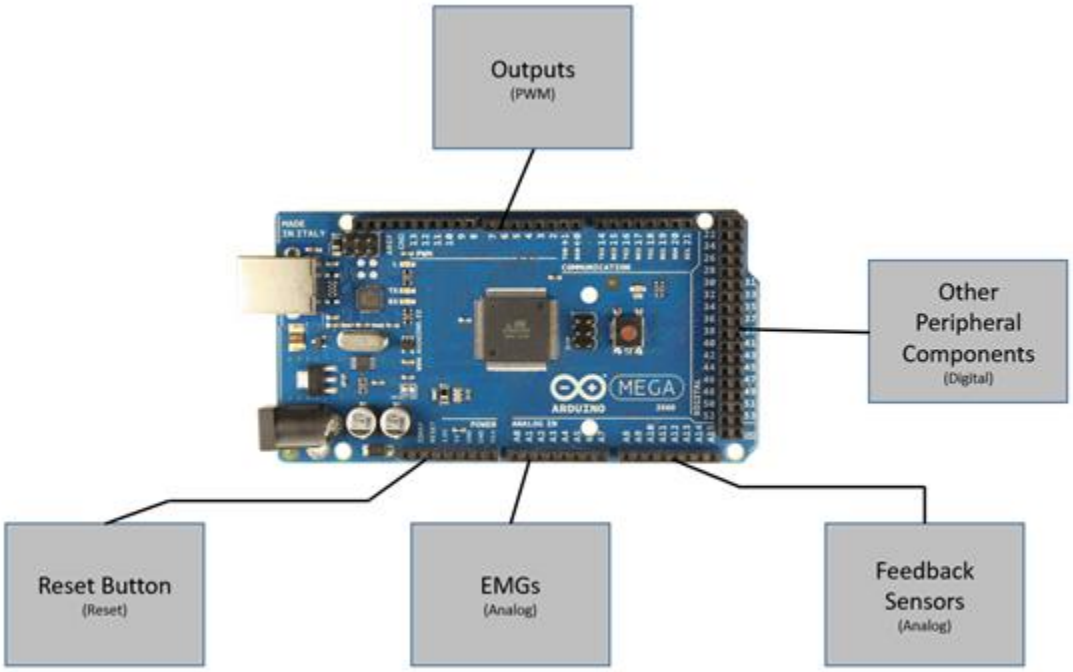
This is exactly what we needed. With PWM we can take the input from the analog pins, process it and apply the logic and then output it through these pins with fine-tuning capabilities on the voltages. This allows us to run components like motors at different speeds and create a more responsive system. We started to research the specific ways that Arduino handles PWM so we could apply these principles to our project.

The Arduino Mega2560 has 54 digital I/O pins. 15 of these pins are capable of PWM. This is more than enough for us (a design of how the board will be set up can be seen in diagram *Code Research 3*). They are used similar to how the analog pins are used. We call the function analogWrite() and this function will take as a parameter a value between 0 and 255. 255 will send the output as a 100% duty cycle (on all the time) and any value under that will send a signal at a proportional rate.

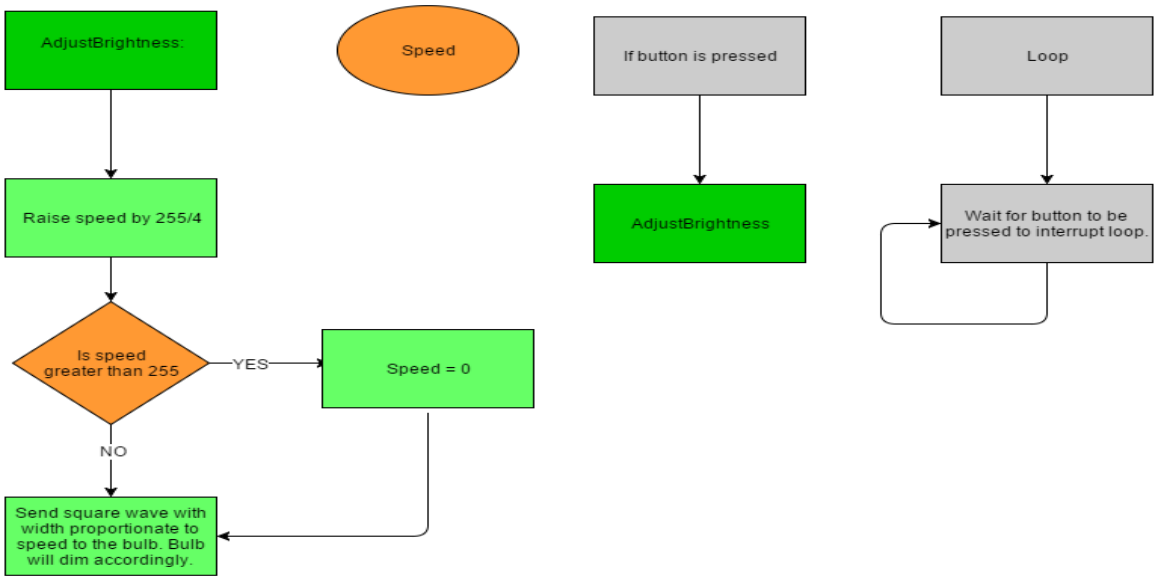
While studying how to use PWM in a practical sense, we wanted to build a circuit and write some code. To fully understand how to use it, we needed to have something test. This time we skipped the physical breadboard and went right to Autodesk to simulate the circuit. We tested with an Arduino Uno as it was available and the code would be similar to the Mega2560. It was placed on a breadboard and connected to a push button as well as a light bulb. The expected behavior of the circuit was that as you push the button, the light bulb cycles to the setting in brightness. The cycle should go like this; 25%, 50%, 75%, 100%, and back to off.

After the board was properly built and ready to go, we began to code. The code was written to follow the behavior described above. The code was pretty simple to write as it just required the analogWrite(). We used a variable called "speed" to control the speed of the modulation and we updated it each time the button was pressed and the interrupt

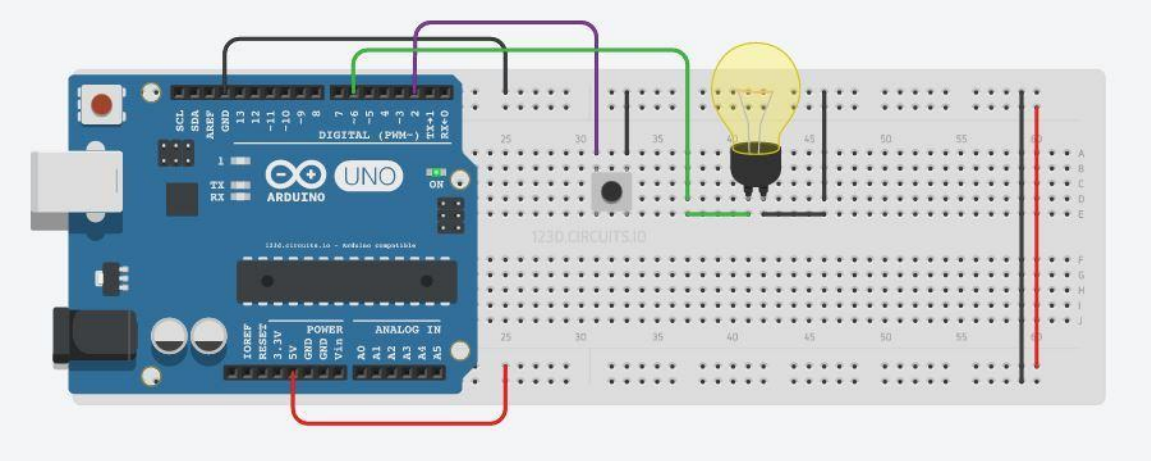
service routine was triggered. The code for the circuit can is represented in diagram *Code Research 4* and the virtual circuit build can be seen in diagram *Code Research 5*.



Code Research 3. This diagram shows an idea of what pins will be used for different components of the system. Inputs will be taken in as analog signals and sent out using pulse width modulation. Also, the reset button has its own pin. Any extra peripherals we need can be connected using digital pins.
Photo from Arduino.cc [I-4]



Code Research 4: This is a flowchart of the code to control the light bulb circuit. Each time the button is pressed the duty cycle on the lightbulb goes to the next setting. The settings are 25%, 50%, 75%, 100%, and then back to off.



Code Research 5: It is an Arduino connected to a breadboard with a push button and a light bulb. (5 V wire and additional connection to ground across the bottom row were for additional components we were testing with and are unused in shown circuit)
Image made with Autodesk [I-5]

4.2.2.3: WIFI

After we had gotten the basics of the Arduino down and had started with some more complex topics, we started to research some other things, like peripherals. One of the things that we need to do for the implementation of the system is live input testing on the patient. This is where we hook up the EMG sensors to the patient and have him do some tasks. We should be able to get outputs of the voltage levels read from his arm. This is done in order to establish a baseline and thresholds for what we should consider as valid input and what should be considered as noise.

We will need to assist the mechanical team with setting this up and creating a test environment for the patient. One of the major components of this test system is the Arduino, which will act as the brain of the system and handle all of the processing. While discussing these tests and deciding of a plan of action, a good point came up. There was something that we hadn't thought of, but would be a pain if we overlooked; movement artifact.

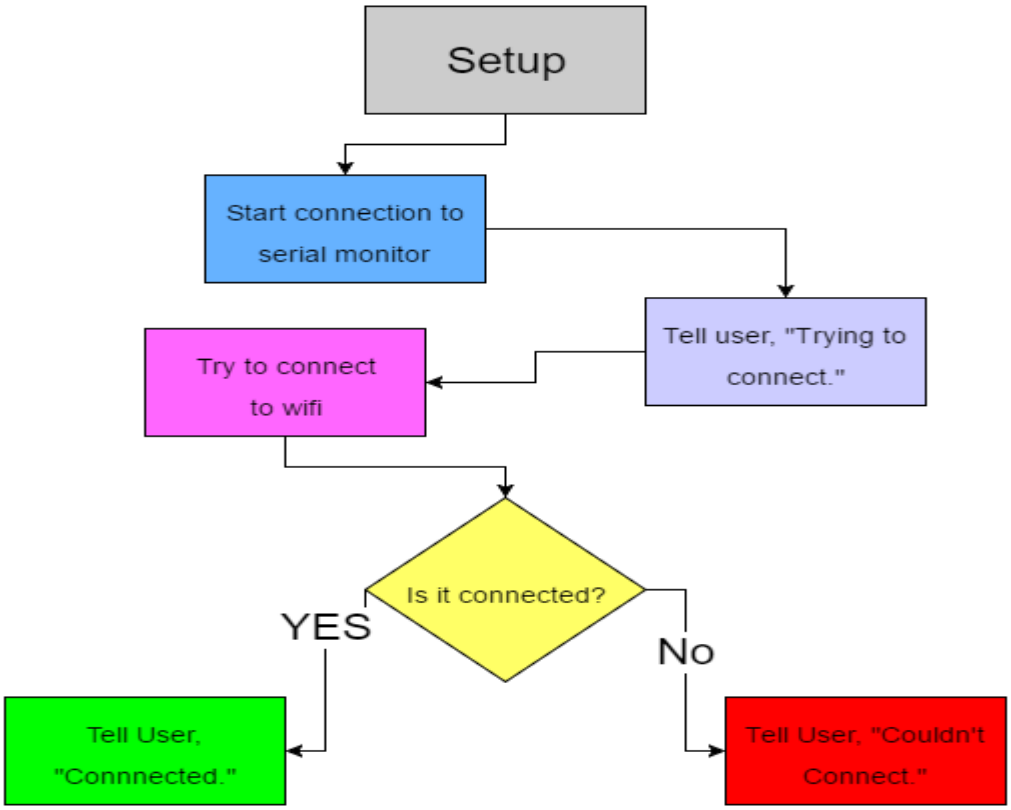
Movement artifact is baseline noise in the EMG reading that occurs due to movement. We determined that in order to get a clear signal, we would have to minimize this artifact as much as possible. Certain movements, such as the movement of the muscle beneath the skin cannot be avoided, but we need to mitigate where we can. One of these areas is the movement of connected wires. Having a long wire that connects the controller to the computer to get reading can be bad for this as well as uncomfortable for the patient. This is where we started looking into going wireless.

After going through a few options, where were able to find a possible solution. On Arduino's website, we found the Arduino WiFi Shield 101. This is a component that can be attached to an Arduino and will be able to give it access to the internet as either a

server or a client. It operates on either 3.3 V or 5 V and can be plugged directly into the host, creating a stack.

We were interested in this as opposed to other communication related components mostly because of the ease of integration that it will foster. It is specially made to be compatible with the Arduino and comes with a large library of functions that can be used to assist in the development of WiFi communication features. We wanted something that would definitely work and didn't want to spend weeks on compatibility resolutions. This is important as we will need to complete the patient testing before we build the system and time is critical.

After we chose the shield, we started searching for and examining example code for it. The examples are not as numerous and extensive as some other Arduino features, but there are there. Some can be found on the Arduino website. These include demonstrations on how to connect to an open network, connect to a network encrypted with WEP or WPA2 Personal, display all networks in range, set up a chat server, connect to a remote server etc. A flowchart can be seen in figure *Code Research 6* demonstrating how we can program the Arduino to connect to an open network according to an example on Arduino.cc. It mostly shows how to test the process. The line of code to try the connection would be status = WiFi.begin(networkName). The status would then be an indication of if it worked or not. It was a bit more challenging than the last pieces of code we did, and none of us had programmed anything web-related in an embedded environment before, but we were up for a challenge.

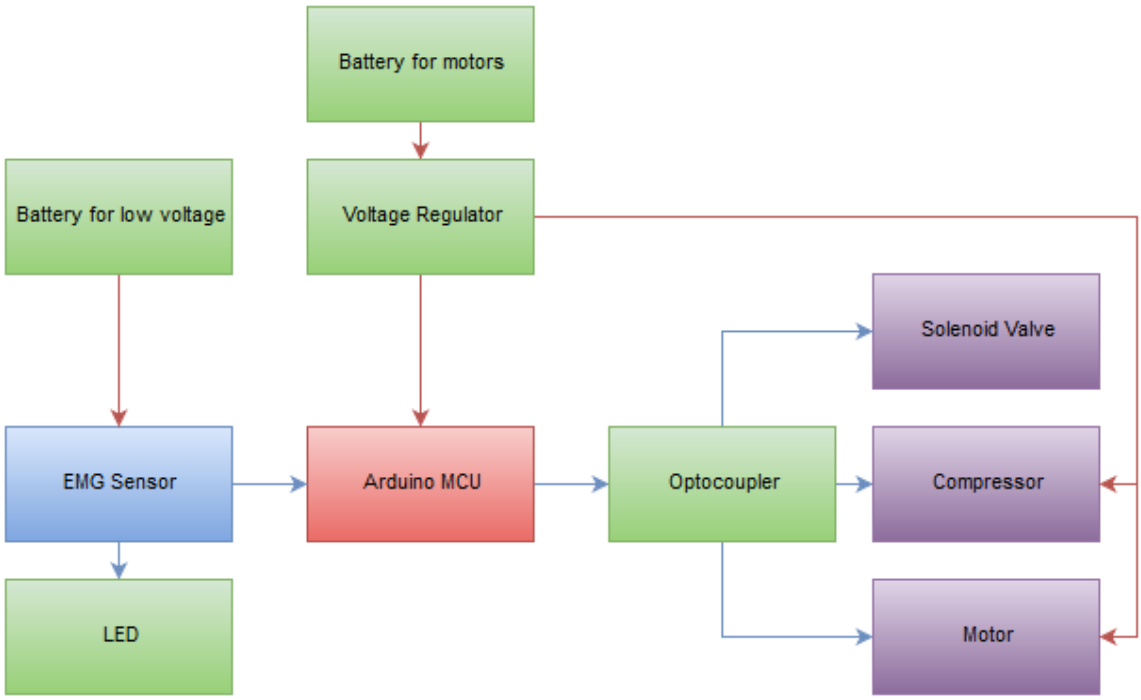


Code Research 6: This a flowchart for the logic of connecting an Arduino to an open network and testing that network.

4.3: Hardware Research

4.3.1: Optocoupler

An opto-isolator is a component that transfers electrical signals between two isolated circuits by using light. The optocoupler prevent high voltages from affecting the system receiving the signal. A common type of isolator and the one that we will be implementing into our circuit is the phototransistor type. The way that they operate is by usually transferring digital on-off signals, but some techniques such as PWM allow them to be used with analog signals. The opto-isolator contains a light, a near infrared light emitting diode, which converts electrical input signal into light. The sensor could be a photoresistor, a photodiode, a phototransistor, a silicon-controlled rectifier or a triac, because LED can sense light in addition to emitting it. Another advantage of the opto-isolator is that they are effective in breaking ground loops caused by high or noise return current in ground wires [10]. That is really important for the design, since it has components like the EMG sensors, which do not play well with noise. In the following flow chart, the relay was replaced by the optocoupler in order to isolate component that required higher voltage/current with those that do not. A different power supply has also been added so that it will take care of the components that do not require as much voltage. The reason the team decided to go down this route is that it could be thought that having only a power supply should be enough, but this idea might help with portability and space. Also, if the battery from the motors needs to be changed by a fully charged one, there would still be power supplied to the Arduino. The microcontroller is the most critical system and the one in charge of sending and receiving signals. If for some reason the microcontroller loses its voltage, nothing else will work. Figure *Optocoupler 1* shows the flowchart of the original hardware design.



Optocoupler 1: Block diagram for interaction between optocoupler and system

4.3.2: Types of Batteries

This section attempts to define different types of batteries so as to provide a more supported choice for powering the system. Pros and cons are defined for each type of battery, as well as some vital information that should be considered when choosing the battery.

4.3.2.1: Alkaline Batteries

Alkaline batteries can store a big amount of capacity, and they are very cheap comparing to others and are widely available. Some of the disadvantages of the alkaline batteries are that they are not rechargeable, and if they are, they are highly inefficient; also, they are not efficient in high-drain devices because of the inherent weakness of supplying the power in a fast manner. Digital cameras are examples of high-drain devices. This system, since it possesses high power usage, is also considered a high-drain device [13]. Their voltage drop is linear with time which means they do not maintain their voltage like some other batteries that will be discussed in the following sections. To summarize, alkaline batteries have a good capacity, but they will not last long to supply power for the device. This battery does not have the basic requirements to be used in the system, therefore it will not be considered.

Lead Acid Batteries: These types of batteries are usually big and heavy, but there are small sizes that can fit the device. Their cost is very low comparing to how much power they can supply. They are also robust, which means there is no need for high maintenance [11]. *Types of Batteries 1* below will summarize their advantages and disadvantages.

Advantages	Disadvantages
Low cost and can be found everywhere and at all sizes.	They are very heavy.
Can deliver a very high current.	Danger of overheating during charging.
Solid or robust and can resist for hard shocks. They are recycled products.	They are toxic because of the sulfuric acid solution they contain, which means they are not good environmentally.
Easy technology to understand which means easy to maintain.	Number of life cycle are limited (300-500), and low efficiency (70%).
They can be charged immediately after use, and they last longer for periodic discharge.	In order to keep the chemical active, these batteries need to be maintained charged.

Types of Batteries 1: Advantages and Disadvantages Lead Acid Batteries
Table from Battery University [I-6]

4.3.2.2: Nickel Cadmium Batteries

Unlike the other discussed batteries, NiCd is the type of the battery that prefers to be recharged periodically because if it stays charged for a long period of time without discharge, crystals will cover its pole cells and deteriorate it. The table *Types of Batteries* 2 below will summarize the pros and cons of the (NiCd) .

Advantages	Disadvantages
They can provide up to 1000 cycle comparing them to NiMH batteries.	Memory problems: they need to be in use periodically to prevent any crystal build in their cells.
Even after of the crystal on their cells that was caused by the long period of storage, this type of batteries can be charged fast with no complexity.	They contain toxic metals which makes them environmentally unfriendly and potentially unsafe for biomedical applications.
Temperature free: they do not get very hot when charging, they can be charged at different temperatures, and temperature will not cause any effect on its cycle life.	They need to be recharged after storage for a long period of time which means they have a self-discharge problem.
A very good deal about the price. They are very cheap and at abundance. Most of the sizes are cylindrical.	
They can be charged and discharged at any number of times, and NiCd can be discharged to a very low percent.	

Types of Batteries 2: Advantages and Disadvantages of Nickel Cadmium (NiCd) Batteries
Table from Battery University [I-6]

4.3.2.3: Lithium Polymer Batteries

Here is another very light battery because of the lithium characteristic. Lithium ion polymer battery will be a very good choice for this project when ignoring the high cost of it. This lithium ion polymer battery is similar to the lithium ion shown in section 4.3.2.4 with some differences that will be shown in *Types of Batteries 3* below.

Advantages	Disadvantages
Its size is very small: they come as a very thin rectangular which makes very easy to implement it in a device like this.	High cost to manufacture.
The weight is very light.	Sizes come out standard.
It is very resistant to overcharges and electrolyte leakage is minimized.	High cost for energy ratio comparing to that of lithium ion.
	Comparing to Li-Ion, this one has lower energy density and decreased cycle count.
	Unsafe in hand of customers because of the chemical lithium.

Types of Batteries 3: Advantages and Disadvantages of Lithium Polymer Batteries
Table from Battery University [I-6]

4.3.2.4: Lithium Ion Batteries

From its name this type of battery is very light because of the lightness of the lithium material. It is used especially for mobile devices such as cell phone. This type of battery sounds very useful for the project. For more information, look at *Type of Batteries 4*.

Advantages	Disadvantages
Very low maintenance needed. Unlike alkaline batteries, lithium ion do not need a partial recharging.	Conditions storage: battery must be at a minimum of 40% charge to be stored. Storage in cold places causes aging problem of the battery because of the chemistry of it. Even at recommended temperatures of 15 °C, aging is a serious problem for this kind of batteries.
When first charged, it does not need a prolonged charge and they can stay connected to the charger even after they are fully charged.	Expensive in price (high cost for high stored energy). They are a subject of change every six months because of new chemical combinations that are made of.
High energy density for its weight. Potentially high in capacity.	Because it is very fragile, It needs some extra circuits built within the device to maintain the current and voltage within the safe limits.
Very lightweight and small shape. They fit perfect to mobile devices.	Precautions needed to be taken in consideration while charging this battery.
Relatively low self-discharge.	Changes in temperature can hurt battery life.

Types of Batteries 4: Advantages and Disadvantages of Lithium Ion Battery
Table from Battery University [I-6]

Size, weight, capacity, and power density are the primary selection considerations for batteries in externally powered applications such as this exoskeleton. The most popular types of rechargeable batteries in use today are nickel-cadmium, nickel-metal-hydride, and lithium-ion. Li-ion is fast becoming the chemistry of choice because of its high capacity-to-size ratio and low self-discharge characteristics. The amount of time it takes to discharge a battery depends upon the battery capacity expressed in milliamp hours, or mAh. A really important measure for this type of application is the discharge rate. The discharge rate measures the rate at which energy can be delivered to the load. The discharge rate is the maximum allowable load or discharge current. The higher the discharge rate, the fasted a battery can meet the demand for energy. On the other hand, when a battery loses/drains its charge with no load, this is known as the self-discharge rate. Li-ion batteries are two times better than NiCd or NiMH in terms of discharge rate. The number of charge and discharge cycles is the average number of times a battery can be discharged and then recharged; that is how service life is measured [12].

For this design the team will depend on an externally power source that can operate and store electricity for multiple hours. The power source selected will have specific ranges which will determine what actuators, motors and components will be used. The goal is to

have the exoskeleton run for as long as possible without the need of replacing or recharging the batteries. The biggest contributor that will need to be addressed is the weight and space of these portable devices, and of course the power source, which will be battery. Battery technology, specifically rechargeable batteries technology, is vital to portable electronic equipment and is driven by the billions of dollars spent by the laptop computer and cellular phone industries. In an electrically powered exoskeleton, the main current draw comes from the motors used to actuate the device. To illustrate, in a DC motor, the output is torque and is directly proportional to the amount of current drawn. The motor that will be used in this exoskeleton is not continuous but it is intermittent.; that is, it will quickly fluctuate between states. Consequently, it is important not only to know how much energy a battery can provide but also how fast the battery can provide it.

The biggest factor when choosing a method to power the project is how long it must be powered for and how reliable and stable this source of power will be. This system is comprised of multiple components that require power. Among these components are the microcontroller, LED's, the stepper motor, EMG sensors, and the compressor which is a critical part for the pneumatics to operate. For an exoskeleton, this becomes even more critical, knowing that a human being will depend on it. The team has the responsibility to make sure the exoskeleton arm is electrically safe and if for any reason it stops working that it does not put the user in any form of danger. The microcontroller that will be used in this system is an Arduino Mega. This microcontroller accepts voltages from 6 V up to 20 V, but it has a built-in regulator that takes an unregulated input voltage that could be changing over time and gets out a perfectly regulated 5 V signal. However; the higher the input voltage the more heat will be dissipated therefore raising the temperature. The team decided that at least 2 V over the wanted output, which is 5 V in this case should be more than enough. Therefore the team will be looking on batteries that are around 7 V to 7.5 V for the logic device to reduce the amount of power wasted, or dissipated at the regulator.

4.3.3: Battery Factors

Before choosing a battery for this application, there are multiple factors that must be taken into consideration. One of the first considerations is how much the expected usage of the system will be. Since this application is expected to be used daily, a rechargeable battery had to be chosen. From the beginning of the power's research, non-rechargeable batteries were eliminated as a possibility to be the source of power for the exoskeleton. It simply would not make sense to use them, because in the long run it would be many times over more costly than just buying the most expensive rechargeable battery from the start.

4.3.3.1: Battery Safety

This project, if designed, integrated, and prototyped correctly, will be used by a human being. With that being said, safety is the most important consideration of them all. For that reason, the team started investigating at the different chemistries for each batteries and how safe they were. Also, concerns were raised on how the battery will affect the environment. In the table *Battery Safety 1*, the comparison of multiple types of batteries

and their chemistry is shown. As can be seen, the last two chemistries from the table are unsafe without the proper electronics, even though they appear to be the ones that can provide the most power; specifically, electronics can also fail and the team does not want to take any risks when it comes to the well-being of the patient. Through research, the team has found that Li-ion batteries, if not charged, or if discharged too quickly, they could get caught on fire. At that point, this type of chemistry was decided to be discarded from the biomedical system that is being built.

Chemistry	Voltage	Energy Density	Working Temp	Cycle Life	Safety	Environmental
$LiFePO_4$	3.2 V	> 120 wh/kg	-20-60 °C	>2000	Safe	Good
Lead acid	2.0 V	> 35 wh/kg	-20-40 °C	>200	Safe	Not good
NiCd	1.2 V	> 40 wh/kg	-20-50 °C	>1000	Safe	Bad
NiMH	1.2 V	> 80 wh/kg	-20-50 °C	>500	Safe	Good
$LiMnXNiYCO_2O_2$	3.7 V	> 160 wh/kg	-20-40 °C	>500	Unsafe w/o PCM	OK
$LiCoO_2$	3.7 V	> 200 wh/kg	-20-60 °C	>500	Unsafe w/o PCM	OK

Battery Safety 1: Chemistries of different batteries with their basic specifications
Image from Battery Space [I-7]

4.3.3.2: LifePO4

The next type of chemistry that could provide the most power while being safer than the one previously mentioned is the LiFePO4, or Lithium Iron Phosphate chemistry. This battery sets itself apart from the others and it is because of its safety and stability. This type of chemistry is considered to be so stable that when subjected to dangerous events, such as collision or short-circuits, they would not explode or catch fire. This is what was being sought out by the team, since it significantly reduces the possibilities of causing harm [13].

4.3.3.2.1: Performance

When it comes to performance, LiFePO4 batteries are also exceptional, especially in their life span. This battery normally exceeds other lithium formulations by a long margin. The only downside is that energy density is typically lower than others, such as cobalt and nickel oxide. However; another advantage of LiFePO4 cell is that after a year or so it

typically has a similar energy density as a LiCO02 lithium ion cell. Another perk to add to this incredible chemistry is that battery charging time is also reduced, which is very convenient for the project's application [13].

4.3.3.2.2: Environmental Impact

For the project the team is not only researching for the strongest safest battery to use for our project. Concerns about the environment have been taken as a factor as well, and since batteries do not last forever, the team wants to ensure that a battery will be used such that, when it is time to get rid of it, it will also be safe for the environment. LiFePO4 are non-toxic, non-contaminating and contain no rare earth metals, making them the perfect choice for any environmentally conscious concern. The team was concerned about batteries like lead acid, because their chemicals degrade the structure overtime and can eventually leak. A decision was made by the team to the determine that the battery that will have the least impact in the environment is the LiFePO4 [13].

4.3.3.2.3: Space Efficiency

As of right now, the working model of the system's prototype will be established on a table, so the space efficiency is not an issue; however; it is worth mentioning that most of LiFePO4 at one third of the weight of most lead-acid batteries and almost half the weight of popular manganese oxide, LiFePO4 provides an effective way to get as much power as possible while reducing space and keeping the weight down. By the time the final design is ready, there will be less preoccupations about finding a better one, since the team had already decided to choose this type of small, environmental safe and powerful battery [13].

4.3.4: Battery Decision

For the battery decision, the team first was taking into consideration the components that actually draw more of the power; in this case, the components would be the motor, which has max current of about 4 A and the compressor, which is at approximately 2.5 A. In the beginning of the research, it was thought that battery specification shown in the table *Battery Decision 1* below showed a LiFEPO4 that would be enough to power our components for a longer period of time, but after motor and compressor choices were finalized, the team came to the conclusion that the battery was not even going to be able to power the system for half an hour. This battery ended up being eliminated from the decision list.

Battery	Custom LiFePO4 18650: (25.6 V, 3000 mAh, 76.8 Wh, 7 A rate, 2Rx8) UN38.3 tested
Voltage	25.6 V (Working), 30.4 V (Peak), 21.2 V (Cut-off)
Capacity	3000 mAh, (76.8 Wh)
Cycle Life	>1000 cycles (80% of final capacity @ 0.2 rate, IEC standard)
Protection	PCB (16 A Limited), Over-Charged (>30.4 V), Over-Discharged (<21. 2 V), Over-Drain (>16 A), Short Circuits, One 7 A polyswitch to limit max discharging current and to protect wrong polarity.
Charging rate	1.5 Amps Max.
Discharge rate	7 A Max. limited by 1 pc 7 A polyswitch
Dimensions	(L x W x H) 155 mm (6.1”) x 50 mm (2.0”) x 72 mm (2.8”)
Weight	1.0 lb 11.4 Oz (780 g)
Price	From $81.95 to $90.44

Battery Decision 1: Light weight and small but not strong enough for our system
Table from Battery Space [I-7]

The second battery that was on the list is shown below in the table *Battery Decision 2* below. This battery has the ability to power the system for a longer period of time than the previous one since its capacity is 4500mAh and 115Wh. The stepper motor uses a power of approximately 96Wh and the compressor around 30Wh; those components by itself in max condition have surpassed the maximum power supplied by this source. This battery is a good option but it is not the best, so it will be kept as a backup in case some use might come for it.

Battery	Custom LiFePO4 18650: (25.6 V, 4500 mAh, 115 Wh, 14 A rate, 3Rx8) Not UN38.3 tested
Voltage	25.6 V (Working), 30.4 V (Peak), 21.2 V (Cut-off)
Capacity	4500 mAh, (115 Wh)
Cycle Life	>1000 cycles (80% of final capacity @ 0.2 rate, IEC standard)
Protection	PCB (16 A Limited), Over-Charged (>30.4 V), Over-Discharged (<21.2 V), Over-Drain (>16 Amp), Short Circuits, Two pcs 7 A polyswitch to limit max discharging current and to protect wrong polarity.
Charging rate	1.5 Amps Max.
Discharge rate	14 A Max. limited by 2 pcs 7 A polyswitch
Dimensions	(L x W x H) 155 mm (6.1”) x 72 mm (2.8”) x 72 mm (2.8”)
Weight	2.0 lb 11.4 Oz (1130 g)
Price	$125.00

Battery Decision 2: This battery is the second option for our design
Image from Battery Space [I-7]

The last battery in the list is the LiFePO4 26650. This battery seems to have all the specifications that are required for the system. The stepper motor and the compressor combine for a total of 126 Wh from this battery, which provides well over that amount. The other components that complement the system doe not use as much power, so it is estimated that it will be a total of around 40 Wh that goes up to 166 Wh for the maximum power delivered by the battery. This means that if every component were working at its full capacity, non-stop for a whole hour, this battery should be able to handle it. The engineering requirement states a minimum duration of one hour under full power in a single charge. This battery, from all the ones that were researched, seems to have the capabilities of accomplishing the requirement. Also, this battery can provide a maximum discharge rate of 16 A, which the system is far away from in terms of power draw, but it is good to be on the safe side because there are still doubts on how much the inrush current will be for the stepper motor and the compressor. This battery comes with a built-in PCB to prevent over-charged, over-discharged and over-drain and therefore is safer.

Battery	LiFePO4 26650: (25.6 V, 6.6 Ah, 168 Wh, 16 A rate); not UN38.3 tested
Voltage	25.6 V (Working), 28.8 V (Peak), 16.0 V (Cut-off)
Capacity	6600 mAh, (168 Wh)
Cycle Life	>1000 cycles (80% of final capacity @ 0.2 rate, IEC standard) 2 times more than NiMH and 10 times more than SLA
Protection	PCB (16 A Limited), Over-Charged (>31.2 V), Over-Discharged (<16.0 V), Over-Drain (>16 A), Short Circuits.
Charging rate	7 Amps Max.
Discharge rate	16 A Max. limited by PCB
Dimensions	(L x W x H) 184 mm (7.2”) x 125 mm (4.9”) x 67 mm (2.6”)
Weight	3.9 lbs (1769 g)
Price	$209.00

Battery Decision 3: This are the specification for the battery that will be used for this project
Image from Battery Space [I-7]

4.3.5: Battery Charging

After choosing a battery, the team needs to understand its characteristics in order to charge it correctly. Battery capacity eventually drains out some voltage, so after the battery gets drained it will need to be recharged. The ways to charge the battery depend on its type of battery. The chemical material and the characteristics of each battery are different; that is why different methods apply to charge them. Temperature plays a big role when deciding which charger is better for each battery. For example, Lithium-Ion cannot be charged when it is cold. Another aspect that needs to be taken into consideration is how fast a battery can be charged. Some batteries do better when they are charged at a slower rate, while others do at a higher rate. Therefore, the chosen battery charger has to satisfy the specifications of the chosen battery for this particular design. Some of the different chargers will be discussed in the following sections and a final decision will be made as to which one will work better for the system.

4.3.5.1: Simple charging

These chargers are also called constant voltage chargers. This type of charger is economic and employs a really simple technique for charging. The way it works is a simple step-down transformer that takes an AC voltage from the outlet and rectifies to the desirable DC voltage to charge the battery. Lithium batteries can use this type of charger. In addition a safety circuit is added to protect the batteries and protect the user. Another downside of this charger is that it takes too long to charge the battery. Also, constant

voltage chargers become hot, which could affect the longevity of the battery. A Li-ion battery does not have that problem because it comes with a protection circuit.

4.3.5.2: Time-Controller Charger

The only difference of this type of charger is that a controller to manage the time of charging is added. It is really cheap and heat-free. The only downside is that it charges slowly, so removing the battery constantly could affect the battery life. The good part is that the timer can be set to greater or less time depending on the specifications of the battery. So if a NiMH battery is used, this would be a great addition to the project.

4.3.5.3: Smart Charger

This type of charger is really expensive, but it controls the charging process by a timer that shows how much time is left for the battery to be fully charged. This concept prevents overcharge and therefore it extends the life of the battery. This charger also determines if a NiMH battery is old, and in some instances refuses to take any other battery if is not NiMH type. If the NiMH battery was to be chosen this would be a great option to charge it with.

4.3.5.4 Trickle Charger

Trickle charging is the process of charging a fully charged battery under no load at rate equal to its self discharge rate, therefore enabling the battery to stay fully charged. This method it is usually used for self-discharge batteries, but does not work for batteries like the NiMH. This charger has different charging rates, which work by supplying constant current to the battery being charged.

4.3.6: Charger Decision

4.3.6.1: Charger 1

This 1.2 A charger was selected to charge battery 1. This is a smart charger designed to charge a 25.6 V LiFePO4 pack with capacity 1200 mAh. This smart charger has a maximum input power of 36 W. This smart charger is CE certified and comes with LED indicator for when it is charging and for when it is fully charged. It also has over-voltage, short-circuit and output reverse protection. This charger has been eliminated as a candidate because it is not compatible with the battery chosen for the design. For more specifications on this charger see the table below *Charger Decision 1*.

Charger	Smart Charger (1.2 A) for 25.6 V LifePO4 Battery Pack
Input Voltage	100-240 VAC, 50-60 Hz
Input Power	36 W Max
Output Voltage	29.2 VDC
Charging	1.2 A
Battery	>= 1200 mAh Capacity to use
Protection	Over Voltage Protection, Short Circuit protection, Output reverse protection
LED indicator	RED = Charging, Green = Fully Charged
Size (LxWxH)	120 mm (4.7”) x 60 mm (2.4”) x 38 mm (1.5”)
Certification	CE listed
Weight	14.8 oz (420 g)
Price	$32.95

Charger Decision 1: This charger was intended to use with Battery 1
Image from Battery Space [I-7]

4.3.6.2: Charger 2

This smart charger was chosen because it is able to charge Battery 3, which is the one that was chosen for the project. Additionally, it would also work with Battery 2. It charges at a constant rate of 6 A. It is designed to charge battery packs that are 25.6 V and contain 8 cells. This charger was designed with a built in cooling fan to ensure proper operating temperature and to prolong its life. This intelligent charger has a built-in integrated circuit that will cut off power automatically when the battery is fully charged. By shutting itself off the charger will prolong the battery pack life while preventing any damages for overcharging. To know more about the characteristics of this charger please see the table below *Charger Decision 2*.

Charger	Smart Charger (6.0A) for 25.6 V LifePO4 Battery Pack
Input Voltage	100-240 VAC, 50-60 Hz, USA AC plug
Output Voltage	29.4 VDC
Charging	6.0 A, Charging time = (1.5 x Ah rate of the pack) / 6.0 A charge current
Battery	>= 6 Ah Capacity to use
Protection	Over Voltage Protection, Short Circuit protection, Output reverse protection
LED indicator	RED = Charging, Green = Fully Charged
Size (LxWxH)	190 mm (7.5”) x 92 mm (3.6”) x 52 mm (2.0”)
Certification	CE listed
Weight	2.0 lbs 7.4 oz (1120 g)
Price	$79.00

Charger Decision 2: This charger will work with battery 2 and battery 3
Image from Battery Space [I-7]

4.3.7: Charging Methods

Once the battery and the charger are picked out it is good to know which method to charge the battery is the most efficient, safer and would also extend the battery’s longevity. There are several techniques used to charge a battery, but the team needs to take into consideration the characteristics of the battery first. Some of the methods available to charge the batteries will be discussed below. Gaining a better understanding on how the battery behaves when charging is important to extend performance.

4.3.7.1: Pulse Charge

To eliminate bad chemical reactions and the creation of crystal on the electrode surface, the pulse method is a great option. This in fact is the best method to charge NiCd batteries. This charging rate it is also controlled by the width of the pulses which is very valuable for battery life.

4.3.7.2: Taper Current

This charging technique is great when used with a smart charger that has a built-in timer. When overcharging any battery that is built with cell, the cell voltage of the battery builds up, which increases the current of charging. When this happens, it causes the cell of the batteries to get damaged. To avoid this phenomenon and avoid damage or personal injury it is a must to have a smart charger.

4.3.7.3: Constant voltage

Basic step down transformer with a rectifier that transfers and rectifies a DC power supply to a lower voltage to our device. Lithium-ion cell batteries are suitable to be used with this technique, but an additional circuitry for protection and safety would have to be implemented in addition.

4.3.7.4: Constant Current

Keeping the current flow constant is achieved by varying the voltage applied from the battery. When the voltage gets to its max, the charger will turn off. This technique is better for NiCd and the NiMH batteries.

4.3.8: Power Distribution

In this section, the team attempts to tackle different ways in which the system's power could be distributed. Since many components in the system have a much lower operating voltage than that which is provided by the battery, regulating techniques shall be applied in order to step-down the voltage to desired ranges.

4.3.8.1: Individual Power Source

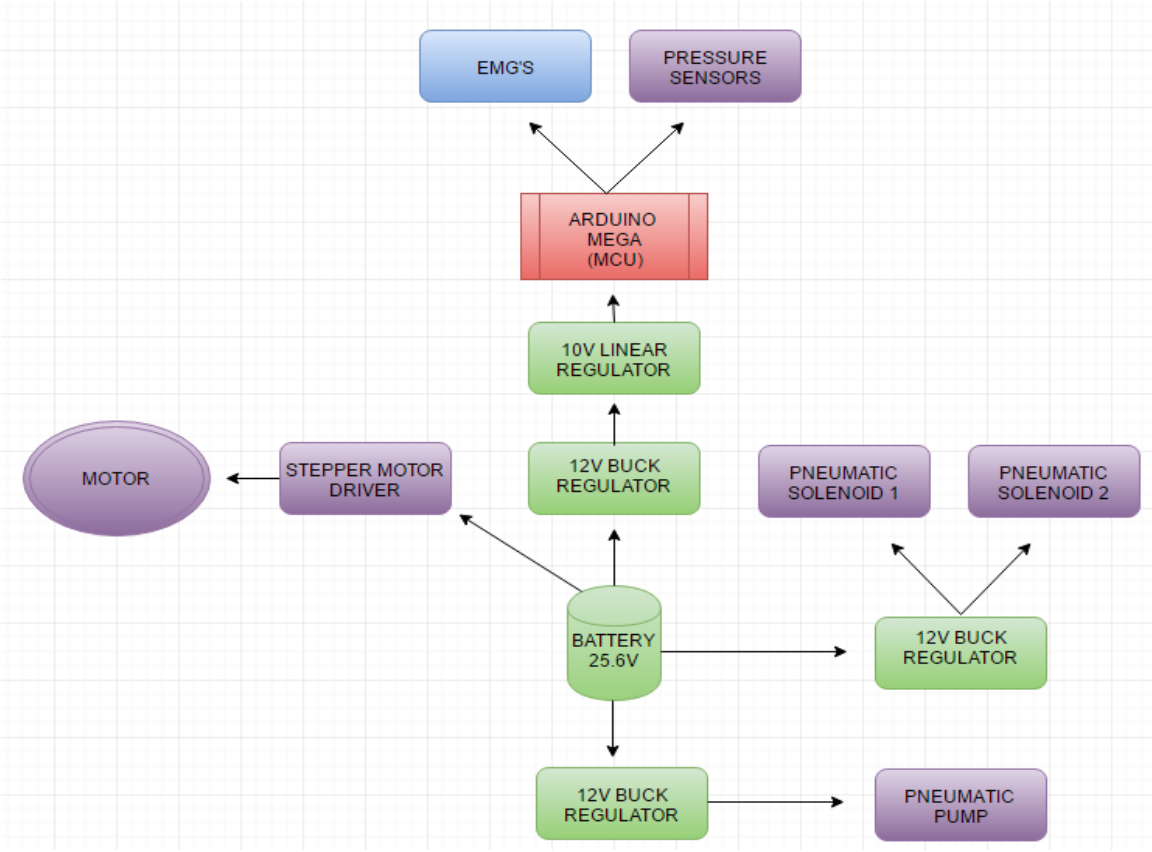
By doing this method each sub system would have its own power supply. This method simplifies the design and it would be a lot easier to diagnose, since all the sub system can be turn on or off individually. This technique for testing purposes it would definitely be the best one, since most of the circuits are isolated and you could shut down each system separately to find the affected system when diagnosing. The bad news is that by doing this type of power source it would take more space and probably more money. An example of this design will be having a battery for the motor, one for the pneumatics, one for the controller etc.

4.3.8.2: Integrated Power Source

To have an integrated power system is a lot more involved and complicated. This will make the system a lot tougher to diagnose since circuits and subsystems cannot be turned off independently. Since everything is integrated into one circuit, it will be easier to get confused and make mistakes. A lot of thinking, measurements and calculations will be made to ensure the system is working correctly. Unlike the individual system, this system will only require a power source to feed the whole system. Even though this system is complex, the team decided that this is going to be the most beneficial for the project.

4.3.8.2.1: Integrated Power Source 1

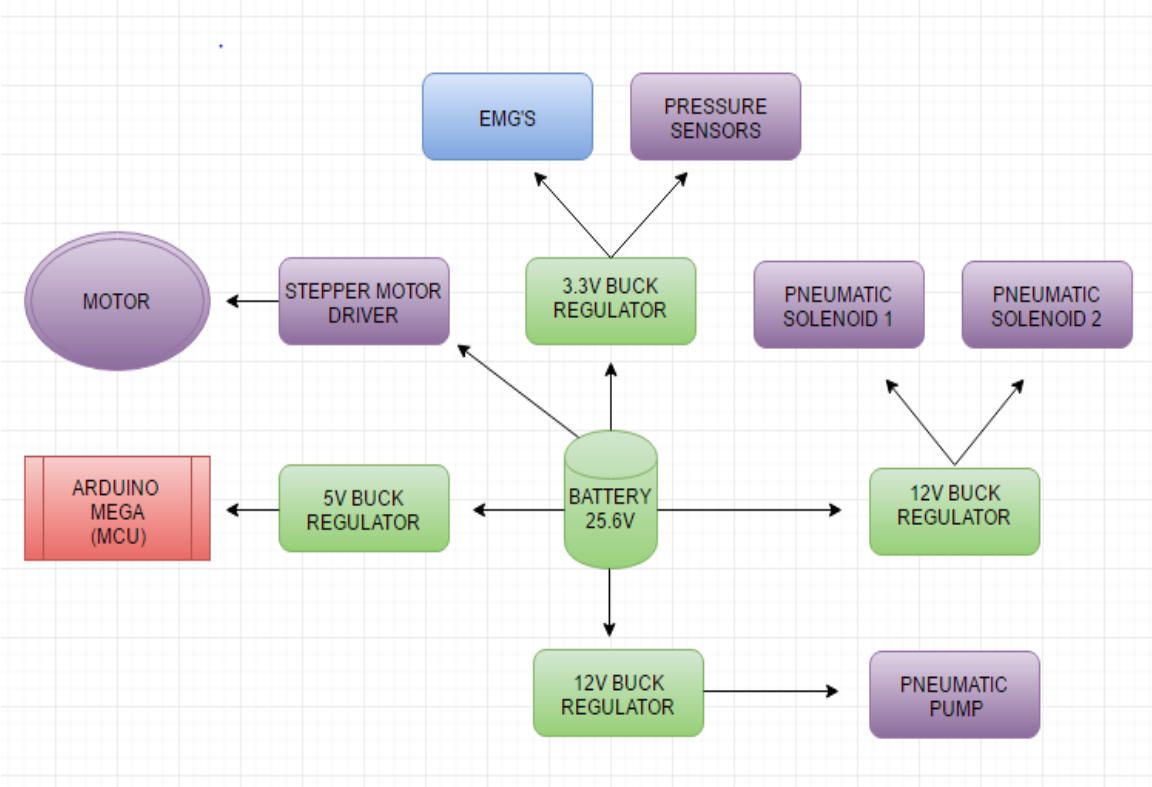
As a team effort, the team decided to go for the integrated power distribution over the individual power distribution system. Multiple designs were drawn out, and from those designs only two of them will be considered for the final design idea. The team has decided to build a simpler version for functional purposes, but once a better understanding is achieved, adding or removing parts will be easier in order to have a more efficient design. The version in section 4.3.8.2.2 will be implemented for the final design that will be built on the printed circuit board. The reason the first design is simpler is because the Arduino breakout board will be used. The breakout board already comes with built-in voltage regulators that could be use to power small components like temperature sensors and the EMG sensors. Below in figure *Power Distribution 1*, there is a flowchart indicating how the power will be distributed between the components.



Power Distribution 1: Implementation of the system using the Arduino Development Board.

4.3.8.2.2: Integrated Power Source 2

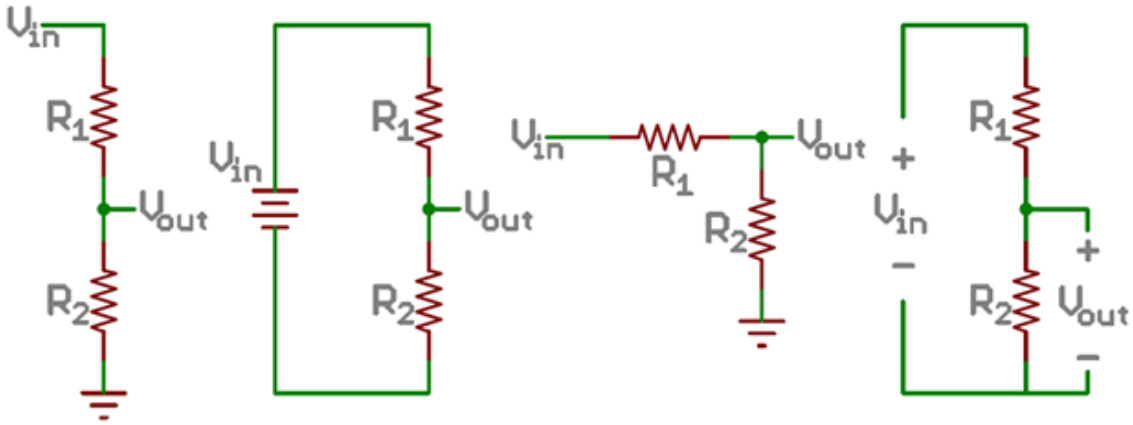
The second design is shown in figure *Power Distribution 2*. It can be seen that it is really similar to the previous design. The most noticeable difference is that previously, a 10 V linear voltage regulator was being used to step down to the development board operating range, while on this one it was decided that powering the MCU with a buck converter will give out a higher power efficiency. In this design more voltage regulators will be added since control and regulation will be needed for the voltage that will be fed into the temperature sensor and EMGs. The temperature sensor and the EMG sensors were previously powered by the built-in regulators in the Arduino development board. As of right now this is the chosen draft design, but as the prototype gets built, the main goal is to reduce the amount of components. By reducing the amount of parts, cost will also be reduced and more importantly, a smaller PCB could be used.



Power Distribution 2: Implementation of the system without the Arduino Development Board.

4.3.9: Voltage Dividers

A Voltage divider is also known as a potential divider. Voltage dividers are among the most common electrical circuits used by engineers. They take in a larger voltage and make its output a fraction of the input voltage. The result is accomplished by distributing the input voltage between the components of the divider. A voltage divider is one of the most fundamental circuits in electronics and to build one is fairly simple. By using only two resistors in series and an input source, a voltage divider is produced. Figure *Voltage Dividers 1* illustrates multiple ways on how a voltage divider could look like.



Voltage Dividers 1: Examples of voltage dividers; they might look different but they are essentially the same circuit.
Image from SparkFun [I-8]

There are two important parts to the potential divider, those being the circuit and the equation. The voltage divider involves applying a voltage source across both resistors. The resistor that is closest to the source is called R_1 and the one closest to ground R_2. The voltage drop across R_2 will give out an output voltage which will be a fraction of the total input voltage of the circuit.

4.3.9.1: Voltage Divider Equation

The equation of the voltage divider states that the output voltage is directly proportional to the input and the ratio of R_1 and R_2. Shown in figure *Voltage Dividers 2* and followed by *Voltage Dividers 3*, characteristics of the voltage dividers are shown.

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

Voltage Dividers 2: Voltage Divider equation.

Keep in mind that…

If...	Then...	Which equals...
$R_2 = R_1$	$V_{out} = V_{in}*R_2/2R_2$	$V_{in}/2$
$R_2 >> R_1$	$V_{out} \approx V_{in}* R_2/R_2$	V_{in}
$R_2 << R_1$	$V_{out} \approx V_{in}*0/R_1$	0

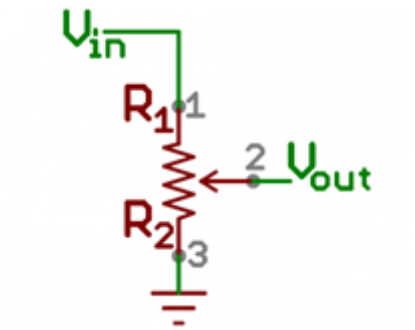
Voltage Dividers 3: Voltage Divider Specification

The first line of the table states that if R_1 and R_2 are equal then the output voltage will be half of the input. This will be true since it does not matter what the values of the resistors are as long as they are the same. The second line states that is the value of R_2 is much

greater than R_1 then the output voltage will be very close to the input because the voltage drop at R_1 will be very small. The third line states that if R_1 is much greater than R_2 then the output voltage will be approximately zero because most of the voltage will be drop in R_1 leaving little to no voltage at the output.

4.3.9.2: Applications

One of the components used in the field to create a voltage divider is a potentiometer. A potentiometer is a variable resistor which can be used to create an adjustable voltage divider. Inside the potentiometers there is a single resistor and a wiper, which cuts the resistor into two and moves to adjust the ratio between both halves. They have three pins; two connect to the resistor, while the third connect to the potentiometer's wiper. When the outside pins are used to connect a voltage source (one to ground and the other one to V_{in}), the output pin will mimic a voltage divider. When turn all the way in one direction, the voltage will be zero and when turn all the way into the other direction the output voltage will equal the input voltage. Potentiometers have many applications and come in a variety of packages. They could be use to create a reference voltage, adjust volumes, measure the position on a joystick, plus thousands of applications that need a variable input voltage. To have a better understanding on how a potentiometer will operate, a valid reference is shown in figure *Voltage Dividers 4*.



Voltage Dividers 4: Voltage Divider using a potentiometer.
Image from SparkFun [I-8]

4.3.9.3: Reading Resistive Sensors

In the real world many sensors are simple resistive devices. Unlike the potentiometer, there are variable resistors such as photoresistors, which produce resistance proportional to the amount of light it senses like a photocell. Other devices like flex sensors, force sensitive resistors, and thermistors are also variable resistors. Microcontrollers cannot measure resistance but, by adding another resistor to the pull-down resistor, a voltage divider can be created. When the output voltage is known, the resistance of the sensor can then be calculated.

4.3.9.4: Level Shifting

There are more complicated sensors that might transmit their readings using heavier serial interfaces, like UART, SPI, or I2C. The majority of those sensors operate at really low voltage, in order to have better power efficiency. Commonly, those sensors are

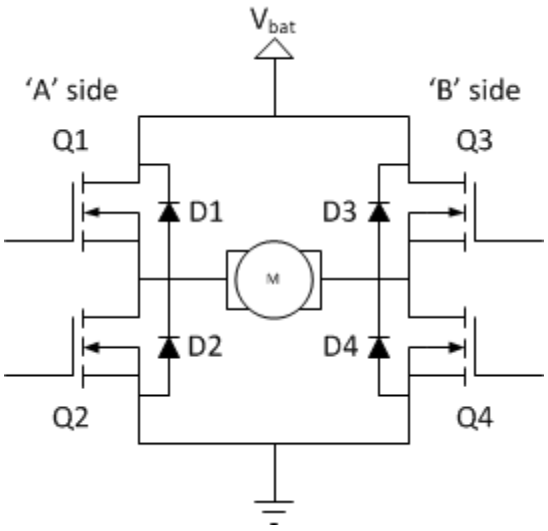
interfacing with a microcontroller operating at a higher voltage. This could cause a problem of level shifting. For example, if one of the EMGs that will be used in the project allows for a maximum input voltage of 3.3 V, and it interfaces with the Arduino Mega operating at 5 V, some modifications will need to be done in order to step that voltage down to 3.3 V. Voltage dividers with resistors in the range of 1k to 10k are usually used for this type of application. It must be kept in mind that voltage dividers only work in one direction and that is stepping down the voltage; they would never be able to step the voltage back up with a divider.

4.3.9.5: How NOT to use a voltage divider

A voltage divider should never be used to supply power to a load. The current that the load requires is also going to flow through R_1. The voltage and current across the resistor produces power, which will be dissipated in the form of heat. If that power exceeds the rating of the resistor, which is usually from 1/8 W to 1 W, the generated heat will become a major problem and potentially melt the resistor. This does not even mention how unpractical a voltage divider power supply would be. For a rule of thumb, voltage dividers must not be used as voltage supply for anything that requires even a modest amount of power. If a voltage needs to be stepped-down to be used to control a power supply, a voltage regulator or switching supplies will be needed [14].

4.3.10: H-Bridge

There are multiple ways to determine which would be the best way to drive a motor, but before that the team needs to look at their data sheet and see how much current this motor can handle. As known through research, current flowing in the winding generates heat, which is usually the limiting factor on the component itself. A person will be wearing this exoskeleton and the team needs to make sure that it does not get too hot and burn the patient, so keeping the system as cool as possible is a must. The same applies to the components used to drive and switch the current used that controls the motor. In order to drive the motor in both directions, an H-bridge will be used. The H-bridge is generally made up to four power MOSFET's that are employed by a controller (in this case the Arduino Mega) [15]. The higher the current, the larger the MOSFET's will need to be; however, a smaller one can be chosen by adding a heat sink to the smaller MOSFET's to help dissipate the heat. *H-Bridge 1* below shows a basic H bridge MOSFET's configuration that will later be implemented in the team's design.



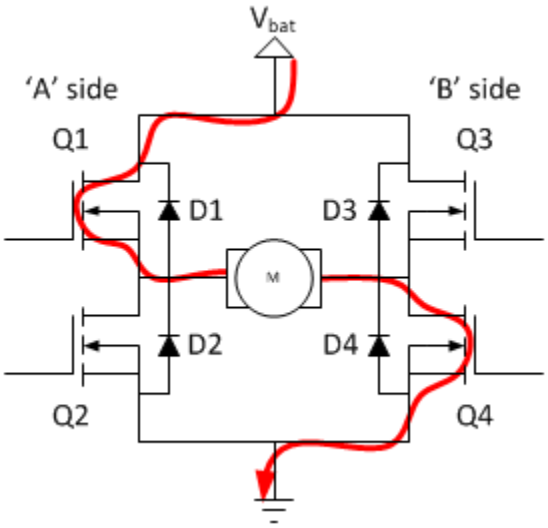
H-Bridge 1: H-bridge MOSFET's configuration
Image from ModularCircuits.com [I-9]

4.3.10.1: Static Operation

H-Bridge is a simple circuit that contains four switching elements with a load (e.g. motor) in the center. The parts of the circuits that are active create the letter "H," thus giving the component its characteristic name. There are different H-Bridges designs, and the actual design will depend on the number of components like transistors, the layout, the number of control lines, the voltage of the bridge and some other factors. This circuit is normally used with brushed DC or bipolar stepper motors. If using stepper motors, two H-Bridges per motor will be needed. These four switching elements can be switched OFF and switched ON independently, but there are some restrictions on why it should not be done. The switching elements could be BJTs, FETs, MOSFETs or in some high voltage applications IGBTs. The diodes used in this application are called catch diodes and are usually of a Schottky type. The top of the H-bridge is connected to a power source while the other half at the bottom of the H-bridge is connected to ground.

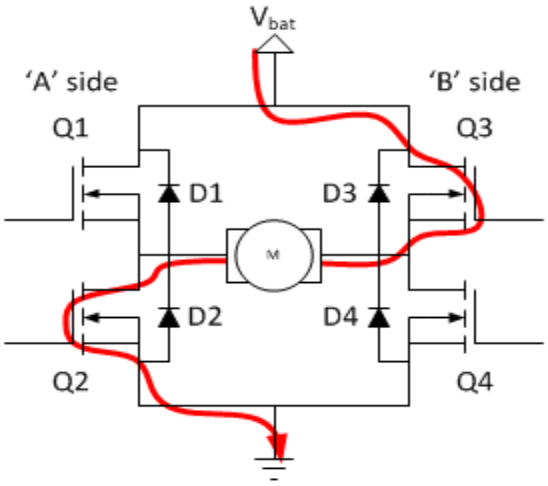
The operation of an H-Bridge is not complicated; in fact, it is fairly simple. For example; when looking at side 'A' in the figure below there is a current basically waiting at the drain, because the gate is open. This does not allow drain and source to be connected and the diode D1 does not allow the current to flow in that direction. In order to get the current from V_{bat} to flow from the drain side to the source side of the transistor, a signal/current from the microcontroller must be at the gate. At this point the current has made it to the left lead of the motor because D2 does not allow the current to flow in that path either. Now that there is current in one of the sides of the motor, there is a need to find a way to create a ground path to the other side of the motor. The only logical option will be to send a signal/current from the microcontroller to the gate of Q4 to allow its source and drain to be connected and therefore allowing ground to reach the right side of the motor at that point the motor should be spinning let say clockwise. Note that the signal/current coming from the Arduino at Q1 gate and Q4 gate must stay constant for the motor to stay energized, if for some reason this signal is removed, then the motor will not

be energized and therefore stop. Figure *H-Bridge 2*, shows the H-bridge working to operate the motor in a clockwise direction.



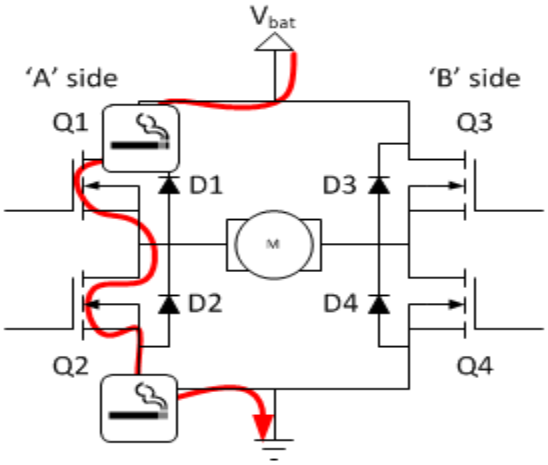
H-Bridge 2: H-Bridge operating the motor in the clockwise direction
Image from ModularCircuits.com [I-9]

The previous diagram showed how the motor would operate in one direction. In order to get the motor operating in the other direction the polarities will need to be reversed. This means that if we it were positive in the left lead of the motor, there will be negative and where there was the negative which was on the right lead of the motor, there will have to be a positive. Now, in the 'B' side, there needs to be a signal/current from the controller at the gate of transistor Q3 that would allow for the positive coming from V_{bat} to reach the right side of the motor, and another signal from the controller to the gate of Q2 that will allow to connect the left lead to the ground and therefore reversing the direction of the motor as shown in figure *H-Bridge 3*.



H-Bridge 3: H-Bridge operating the motor in the counterclockwise direction
Image from ModularCircuits.com [I-9]

In the previous paragraphs there were basic definitions of an H-bridge, how to operate the motor and also how to reverse the polarity of the motor to change its direction. The following will explain what should never be done while using and H-bridge. For example; if side 'A' transistors Q1 and Q2 are closed at some time, this will create a low resistance path between power and ground; this is basically a short to ground also known as "shoot-through'. When this happens a way to burn the bridge or something else in the circuit is guaranteed. The image below *H-Bridge 4* represents how the current will flow from the battery source straight to ground.



H-Bridge 4: H-Bridge shorting to ground
Image from ModularCircuits.com [I-9]

Because of this restriction from the four possible states the side-A switches could be in only three that make sense. The following tables in *H-Bridge 5* and *H-Bridge 6*, tells the logical combinations without causing the a short that was seen in *H-Bridge 4*.

Q1	Q2
Open	Open
Close	Open
Open	Close

H-Bridge 5: H-Bridge table with logical combinations for Side A

Similarly for side B:

Q3	Q4
Open	Open
Close	Open
Open	Close

H-Bridge 6: H-Bridge table with logical combinations for Side B

4.3.10.2: Designing an H-Bridge

Before choosing parts to build the H-Bridge, there needs to be an understanding on how the loads behave in order to choose the proper parts that will withstand the heat and current flowing through it. For example; to get a motor to start under load it requires at least three times the running current, but also a motor would take 3 times the running current when it is loaded. That is the reason why and H-bridge must be able to handle and deliver high current. To deliver high current, transistors must be fully saturated.

In order to saturate a transistor, the base current must be around 1/10 the collector-emitter current. A majority of the transistors has a gain on 100, but this only applies when the transistor is passing 10 mA. Based on that previous statement to pass high current, the base current must be increased to about 10 mA for every 100 mA collector emitter current. Transistors are nonlinear devices, and this is something that is rarely mentioned. When researching data sheets, it can be seen that they highlight the advantages of the transistors, but fail to mention the problems that will be encountered.

Consequently, this is why the output of the transistors in an H-Bridge will be operated by a driver transistor. This driver for transistor will have a gain of around 50, so the input current to the H-Bridge will be only a few milliamps. The other factor that needs to be taken into account is the voltage drops across each leg of the bridge.

A transistor when it is saturated will drop around 0.3 V whereas a Darlington transistor will drop around 1 V and a MOSFET will drop only to about 0.05 V while and emitter follower will drop about 0.7 V. Before committing to any design it is good practice to build a test circuit and measure the voltage drops on each legs to understand voltage available after the transistor is on [16].

4.3.11: Voltage Regulators

Electronic circuits are designed to operate of some sort of power supply voltage. This supplied voltage it is assumed to be constant at all times when in reality it is not. The supply voltage is changing over time. To determine how it is changing over time and at what rate, a special diagnostic tool called an oscilloscope will be needed. Also, a multi-meter could be used, but real time readings would not be obtained. In fact, the multi meter takes multiples samples and then displays the mean of the measured values over a period of time. The way the oscillation affects the circuits depends on the design, the operating voltage of the circuit, and the supplied voltage. For example; if the power supply is supposed to be pushing out 12 V, but it is oscillating an output signal from 8 V to 12 V, but the circuit operates at 5 V most likely very little difference to none will be seen in terms of operation. However, if the circuit were operating at 10 V then the system would work intermittently or not work at all. The proper analogy of this instance is flipping on and flipping off a light switch; that is a similar representation of how the circuit will behave.

To prevent the fluctuating signal that was explained in the previous paragraph, voltage regulators were used [17]. Voltage regulators are designed to maintain a constant voltage

level for AC or DC signals. Even though it is called a voltage regulator, it also controls current [18]. The reason for the control of current is that voltage and current are related to each other; this is easily proven by Ohm's Law. Ohm's Law states that the current multiplied by the resistance will equal the total voltage (V = I x R). The concept can be visualized by inputting different values to see how the variables behave. If voltage is held constant and resistance is increased, the outcome will be lower current, but if resistance is lower the current will increase. The reason this is mentioned is because the design involves sensible components like the Arduino microcontroller and the EMG sensor.

4.3.11.1: Types of Voltage Regulators

4.3.11.1.1: Buck Converter

The Buck converter is a power converter that takes DC as an input and outputs a lower DC voltage. Basically what it does is stepping down the voltage while stepping up the current from its input to its output. It is classified as a switched mode power supply that usually contains at least two types of semiconductors, a diode and a transistor. Newer buck converters replace the diode by another transistor which is used for synchronous rectification. Also, they contain at least one storage element this being an inductor, a capacitor or a combination of both. In order to reduce voltage ripple, filters made of capacitors are normally added to this converter's output. These DC to DC converters are recognized by their power efficiency, which most of the time is higher than 90%. Buck converters are better than linear regulators, which are simpler circuits that lower voltages by dissipating the power as heat, which does not step up the output current [19].

4.3.11.1.2: Electronic Voltage Regulator

This simple voltage/current regulator can be easily made with a resistor followed by a diode in series or a series of diodes. If precise voltage, control and efficiency are really important for this design, this type of regulator it is not recommended. This regulator operates by using a fixed reference voltage and comparing it to the actual output feedback voltage. Any residual is amplified and used to control the regulation element in such a way that reduces voltage error. This created a negative feedback loop, but if the open loop is increased this tends to increase accuracy but reduces stability. If the output is too low, the regulation element is commanded to produce higher output voltage [18].

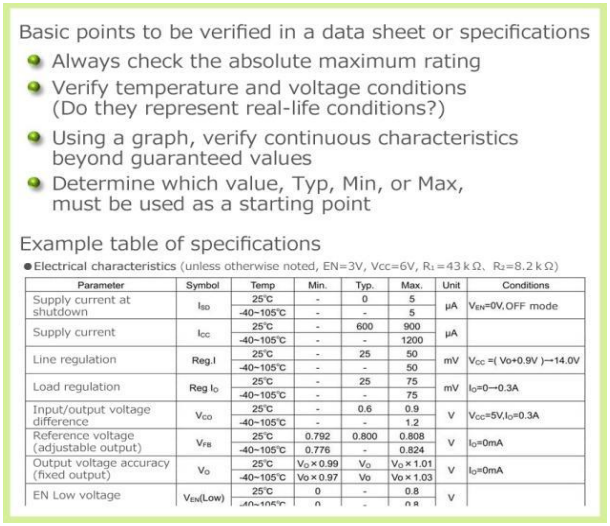
4.3.11.1.3: Electromechanical Regulator

Regulation is completed by cooling the sensing wire to make an electromagnet. As voltage increases, so does the current, making the field stronger. The magnets are physically connected to a mechanical switch, which opens as the magnets moves into the field produced by the coil. Low voltage fluctuations make this regulator really sensitive. The motion of the solenoid core can be used to move a switch across a range of resistance or transformers winding to gradually step the output voltage up or down, or to rotate the position of a moving-coil AC regulator [19].

4.3.11.1.4: Linear Regulator

Active regulators employ at least one active (amplifying) component such as a transistor or operational amplifier. A linear regulator is an example of one of them. Every regulator has its standard limits and criteria. These limits and criteria are specifications that will be found in the data sheet for each particular component. The data sheet of a linear regulator provides a table of limit values, including such parameters as output voltages and accuracy [20]. Also, it contains necessary information including but not limited to; maximum rating, guaranteed operating conditions, parameters, characteristics and their graphs.

When looking at a data sheet, a maximum rating is a value that must not be exceeded at any given time. Some specifications can also be stated as a minimum value and they can be often overlooked. In order to avoid complications the team has decided to consider using a linear regulator with a maximum rating that can tolerate and operate under multiple conditions. Below in *Voltage Regulators 1* a table is shown, which gives an example of the basic points that must be verified in the data sheet or specifications.



Basic points to be verified in a data sheet or specifications

- Always check the absolute maximum rating
- Verify temperature and voltage conditions (Do they represent real-life conditions?)
- Using a graph, verify continuous characteristics beyond guaranteed values
- Determine which value, Typ, Min, or Max, must be used as a starting point

Example table of specifications

● Electrical characteristics (unless otherwise noted, EN=3V, Vcc=6V, R1=43 kΩ, R2=8.2 kΩ)

Parameter	Symbol	Temp	Min.	Typ.	Max.	Unit	Conditions
Supply current at shutdown	I_{SD}	25°C	-	0	5	μA	V_{EN}=0V, OFF mode
		-40~105°C	-	-	5		
Supply current	I_{CC}	25°C	-	600	900	μA	
		-40~105°C	-	-	1200		
Line regulation	Reg.I	25°C	-	25	50	mV	V_{CC} =(Vo+0.9V)~14.0V
		-40~105°C	-	-	50		
Load regulation	Reg I_O	25°C	-	25	75	mV	I_O=0~0.3A
		-40~105°C	-	-	75		
Input/output voltage difference	V_{CO}	25°C	-	0.6	0.9	V	V_{CC}=5V,I_O=0.3A
		-40~105°C	-	-	1.2		
Reference voltage (adjustable output)	V_{FB}	25°C	0.792	0.800	0.808	V	I_O=0mA
		-40~105°C	0.776	-	0.824		
Output voltage accuracy (fixed output)	V_O	25°C	V_O×0.99	V_O	V_O×1.01	V	I_O=0mA
		-40~105°C	Vo×0.97	Vo	Vo×1.03		
EN Low voltage	$V_{EN(Low)}$	25°C	0	-	0.8	V	
		-40~105°C	0	-	0.8		

Voltage Regulators 1: Table shows basic points to look for on a data sheet.
Image from Micro.ROhm.com [I-10]

The linear voltage regulator is nearly used in every power supply used in electronics. These regulators will be needed to control how much flow of electrons will be allowed to the electronics components Linear regulators are based on devices that operate in their linear region. The advantage of a linear regulator is the very clean output with minimal noise introduced into their DC output, but most often are much less efficient and unable to step up or invert the input voltages like switches supplies. All the linear regulators require a higher input than the output than the output. If the input voltage approaches the desired output voltage, the regulator will “drop out”; the input to output voltage differential at which this occurs is known as the regulator's dropout voltage. These low-dropout regulators allow an input voltage that can be much lower. They also waste less energy than the standard linear regulators.

Advantages:

- Low Output noise.
- Best when a fast response to input and output disturbances is required.
- For low power levels, they are cheaper and occupy less PCB space.
- More efficient in cases such a 5 V microprocessor in sleep.
- Low dropout voltages cost more money but are worth it if the input voltage comes really close to the actual output voltage.

Disadvantages:

- Are inefficient, the higher the input voltage the lower the efficiency.
- Have a drop out voltage.
- Power wasted example on linear regulators are: Power = $(V_{in} - V_{out}) * I_{out}$ if there is a 16 V unregulated with a 5 V regulator and pulling 300 mA, that equals to 3.3 W, which is a lot for a small component and it will likely need a heat sink. However if an 8.4 V with a 5 V regulator = 1.02 W is used, then a heat sink would not be required.

4.3.11.1.5: Switching Regulators

Switching regulators rapidly switch a series device on and off. This is controlled by a similar feedback mechanism as in a linear regulator. Because the series element is either fully conducting, or switched off, it dissipates almost no power; this is what gives the switching design its efficiency. Unlike linear regulators, these types of regulator are able to generate output voltages that are higher than the input, or opposite polarity. This could not be done with a linear design.

Advantages:

- Are best for power efficiency such as laptops.
- When only power supply is DC and a higher output voltage is required.
- At power levels above a few watts, switching regulator are cheaper.

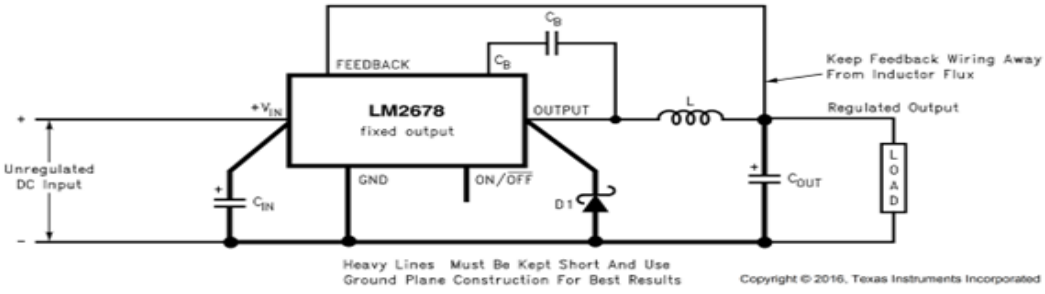
4.3.11.2: Voltage Regulator Selection

The team has been doing research and has come up with many interesting components. In this case in particular our focus was into voltage regulators. A good amount of time was spent on research, just to make sure to find which components will perform better with the current design. From a huge selection of voltage regulators that are sold by multiple companies, one of the team's choices was the LM2678. The LM2678 is a step down voltage regulator, which has many of the characteristics needed for the project, and for that reason is one of the strongest picks. Also, this component is manufactured by Texas Instruments, which is known for its wide selection of electrical components.

Some of the reasons why this voltage regulator has been considered by the team is because the LM2678 voltage regulator can reach up to an efficiency of 92%, is simple and really easy to design. Its design easiness comes from the fact that the components needed to complement the design are easily accessible. This voltage regulator has many advantages that others cannot replicate. This step-down voltage regulator has a real wide input voltage which gives us some buffer in case we have to increase or decrease the power source that we will be using, in this case our 25.6 V battery. This voltage regulator has all the active functions of a switching regulator which is capable of driving loads that are up to 5 A.

This buck converter could be set up to be used to get multiple output voltages. Therefore, this component can be used to control multiples circuits in the design. By just replacing some of the components values, the output can change to be higher or lower. When changing these components the team could design a fixed output of 3.3 V, 5 V, 12 V and even an adjustable output version. In *Voltage Regulators 2* shown below, the typical circuit for a fixed output in an LM2678 model is shown. *Voltage Regulators 3* below shows a table for buck converter LM2678, will specify which components will be needed to obtain those specific outputs. This same circuit with different inductor and capacitor values will be used to regulate different components in the system.

To illustrate, the team will regulate the voltage that will be feeding the MCU to a max of 5 V. For the air pump, the team will be setting the circuit to allow an output of 12 V and for the EMG sensors, voltage will be stepped down to a value of 3.3 V. By doing this, the team is making sure that every component is getting a constant input signal to avoid any fluctuations when in operation.



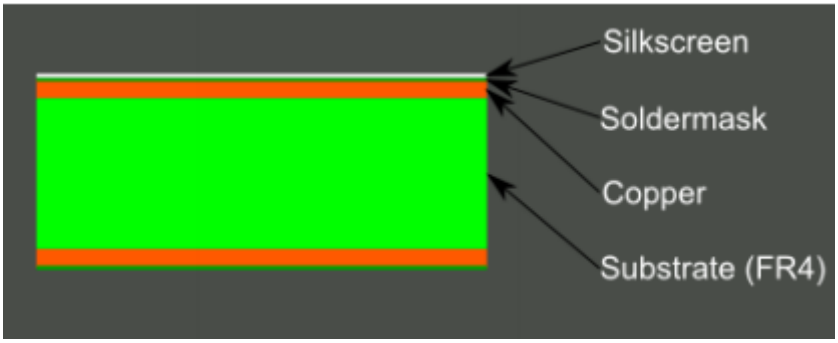
Voltage Regulators 2: Diagram for a LM2678 buck converter
Image from Texas Instruments [I-11]

Input Voltage	Output Voltage	I_{out}	Inductance	C_{in}	C_{out}	C_B
8 V to 40 V	3. 3 V	4 A	15 µH	2 x 560 µF/50	3900 µF/10	10.0 nF
8 V to 40 V	5 V	5 A	22 µH	3 x 15 µF/50	2 x 180 µF/16	0.01 µF
8 V to 40 V	12 V	4.5 A	27 µH	3 x 4.7µF/50V	100 µF/16	10.0 nF

Voltage Regulators 3: Table LM2678:
Table from Texas Instruments [I-11]

4.3.12: Printed Circuit Board

A PCB mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. PCBs can have one copper of layer which is called a (single sided), two copper layers (double sided) or multiple outer and inner layers for a (multi-layer PCB) [21]. Manufacturing PCBs is cheaper and faster than using older methods, the only effort required is put into the design of the layout. See *Printed Circuit Board 1* below to see the basic construction of a double sided PCB.



Printed Circuit Board 1: Simple construction of a double sided PCB.
Image from SparkFun [I-8]

Before considering to start a PCB for this project, all the components must be evaluated and tested on a breadboard. Having a prototype is essential because once the schematic is sent out for fabrication, there is little to no room for additional changes on the PCB. Additional to the breadboard, there are other tools that can be used to test the design before fabrication. Multisim and LTSpice are great simulation software that could provide a good feedback on how the circuit behaves. By using those tools before fabrication, there is a greater probability that the design will work on the printed circuit board.

Once initial testing phase is finished, the team can then move to design the circuits in CAD software for PCB design. There are multiple programs that allows for building a PCB; but EAGLE will be used. This is a free software that has many tutorials and a big community online on how to use it. This software has many libraries with parts that will be needed in order to create the wiring schematics for our circuits. For those parts that are not in the library of EAGLE, this program has the capabilities of creating a library in which components can be added by designing them. Since this program is free, its license is limited as it only allows building a double sided PCB. A double-sided PCB should be more than enough to create the design.

Once the schematics are done, EAGLE will create a file for both the schematic and the board layout. The team will have to make sure to change the format before it is sent to the manufacturer. The files wanted by the supplier are the Gerber files. The Gerber files have specific information about the board layouts, which include traces' width, copper wires, where holes must be drilled, etc. These files contain the necessary information that companies need in order to build the boards in a language that their machine can

understand. It is crucial that these files are revised by the team members before they are send out to the manufacturer.

When receiving the PCB, attention must be paid not to install components or other parts onto the board right away. First, the team must verify and compare to current design prints. Second, the team shall analyze the board to make sure everything is where it is supposed to be. Third, the team is to physically test point to point connections by the use of continuity with a multimeter. The university offers various locations where circuits can be built and tested. These facilities have to be used to the team's advantage accordingly. If any issues are found with the board, the team will have to quickly figure out what is wrong with it. For that reason, the prototype that was made on the breadboard should stay there for as long as possible. It will be the team's best reference.

4.4: Electro-Mechanical Research

4.4.1: Electro-Mechanical Hand Hardware

4.4.1.1: Compressor

For the compressor there are a few requirements which have to be considered. The first is that the system is expected to run quietly so a noisy compressor will not do. The second is that the compressor must be compact and wearable. This indirectly leads to the requirement that the compressor must not generate excessive heat. Some heat generation is inevitable considering the laws of thermodynamics but this can be mitigated by using an efficient compressor and designing the tanks such that there is a proper reserve of air pressure. This reserve should allow sufficient time for the compressor to cool down in between running. If heat generation does become an issue, then a cooling system will have to be devised. The other implied requirement is that the compressor must be lightweight and portable considering it will be worn for extended periods of time by an individual who is already at a strength disadvantage. The third and final requirement is that it must be sufficient to run the system. The configuration of the tanks may be able to offset the power requirement of the compressor but only to a certain extent.

4.4.1.2: Tanks

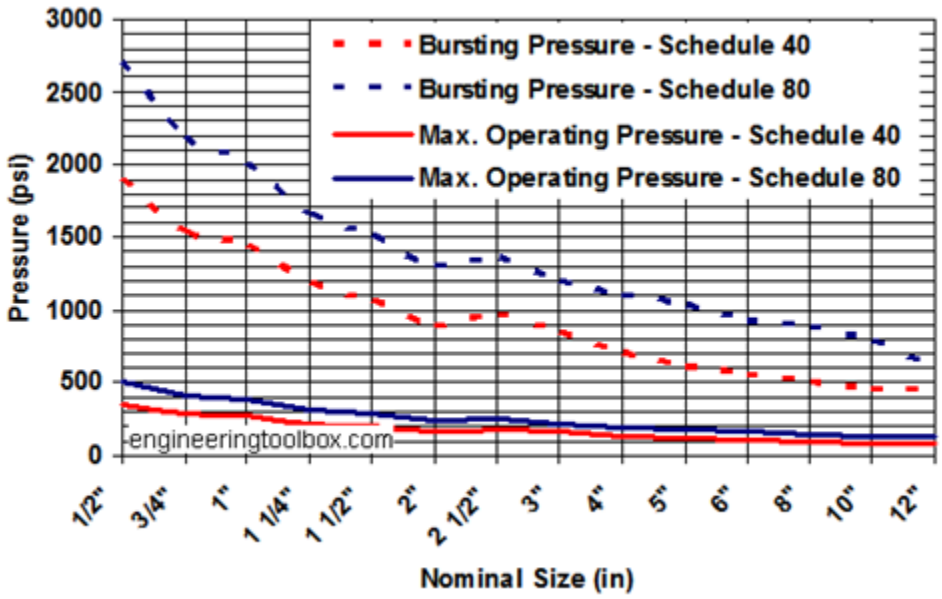
There will be two tanks to power the pneumatic hand actuators. One tank will be for holding air at pressure below that required of the pneumatic actuators and the other will be for holding air at pressure higher than that required by the pneumatic actuators. This will allow the high pressure side to inflate the hand when needed and then the low pressure side will be able to suck the air out to relax the hand. The precise volume and pressure of these tanks will depend on the final design of the pneumatic hand itself which at this time is still under development by the mechanical team. The idea is that each tank will be able to inflate or deflate the hand multiple times without the need for the compressor to kick on.

There are numerous different styles of compressed air tanks which could be used for this design but the main design consideration of these components will be to minimize size

and weight. Compressed air tanks are normally constructed from one of three main materials either steel, aluminum, or carbon wrapped aluminum. Steel is a great choice for high pressure tanks when size is a consideration. Steel being stronger than aluminum results in a more compact cylinder for the same pressure and volume. Aluminum on the other hand has a leg up on steel in the weight department being that aluminum has a superior strength to weight ratio. When it comes down to there is one choice which stands above the rest and that is the carbon wrapped aluminum cylinders. Carbon wrapped aluminum cylinders are used by firemen when they need an air supply to enter a burning building, the reason behind the choice is weight. The carbon wrapped cylinders are significantly lighter than both steel and aluminum cylinders of similar ratings. The only downside to these carbon wrapped cylinders is that they are more fragile. Surface scratches or dings are much more likely compromise the integrity of a carbon wrapped tank. One other downside is that high pressure carbon wrapped tanks require hydrostatic testing every three years to be certified safe to fill as oppose to the five years for steel and aluminum tanks. The tanks for this design may be utilizing low enough pressure (around 50 psi) such that they will not be subject to the hydrostatic certification tests but that will remain unknown until the design by the mechanical team is finalized. Below is *Tanks 1*, which shows the layered composition of a typical fiber wrapped cylinder.

Tanks 1: Fiber Wrapped Cylinder Construction
Image from alibaba.com

Current the idea is for the tanks to hold around 50 psi of pressure. However, if the pneumatic hand calls for it then that number might have to be increased to keep the tanks compact while maintaining the volume needed for actuation. For the test rig, which will be constructed to test the control of the hardware by the electronics, the tanks will be constructed out of PVC. PVC is idea for the prototype and testing phase due to its low cost and high accessibility. In addition, PVC is ideal because it can be configured and reconfigured quickly and easily to troubleshoot any issues which may arise. The maximum operating pressure rating for PVC (shown in the figure below) is significantly higher than the intended working pressure thus in turn should provide a safe analog to the fiber wrapped cylinders which will be used in the final design. *Tanks 2* shows pressure rating values for PVC material.

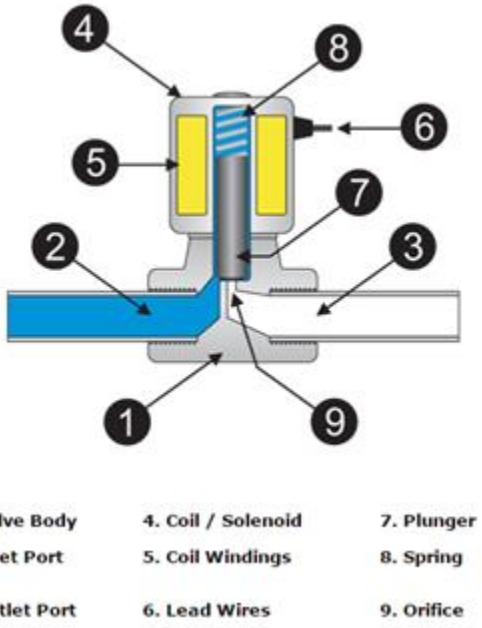


Tanks 2: PVC Pressure Rating
Image from engineeringtoolbox.com

4.4.1.3: Solenoid Valves

The solenoid valves will be used to control the flow of the air either into or out of the pneumatic hand. There will be one valve which will open to let the air from the high pressure tank into the pneumatic hand and another valve to let the air into the low pressure tank from the pneumatic hand. There are two main types of solenoids; ones which remain open until signaled to close and ones which remain closed until signaled to open. For this design it would obviously be advantageous to use the valves which remain closed until told to open. Utilizing this type of solenoid would make the system more robust in case of power loss such that if the power were cut the system would remain in the same state it was in at the point of power loss. The only possible issue which could arise would be if the hand were closed during power loss and would not allow the user to remove the device. To mitigate this there will need to be a manual override valve to depressurize the actuator in case of a power failure in the closed hand mode.

Solenoid valves basically works as an electro magnet, by passing a current through the coil a magnetic field is generated which moves the valve plunger. A spring is used to keep the plunger in the opened or closed position until the coil is energized at this point the magnetic field generated by the current passing through the coil causes the plunger to move against the force of the spring switching the state of the valve. Solenoids are commonly used to control the movement of fluids in a system. The biggest advantage to the use of solenoid valves is their speed. The valves can fully open in an instant and in a repeatable manner which can be easily controlled by the system. *Solenoid Valves 1* below is a diagram of the inner workings of a solenoid valve.

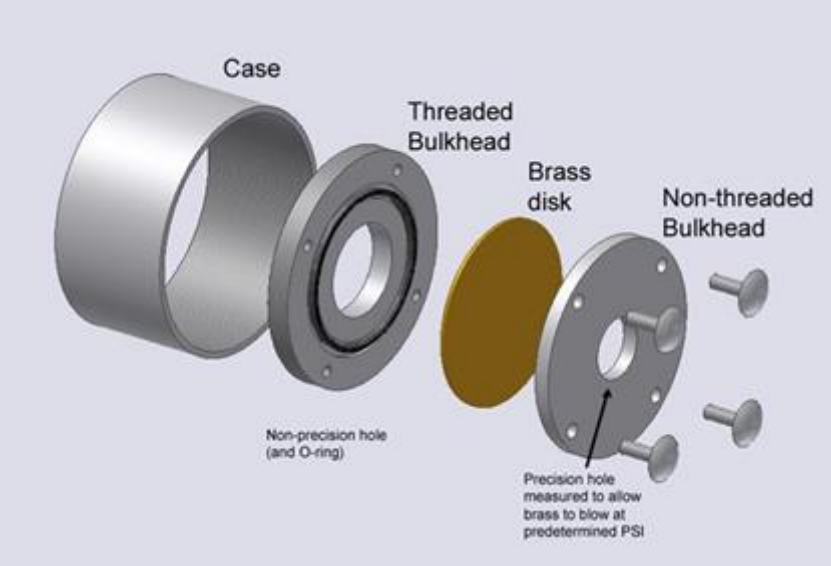


Solenoid Valves 1: Solenoid Valve Internal View
Image from solenoid-valve-info.com

4.4.1.4: Overpressure Valves

Overpressure valves are critical to the safe design of any pressurized system. The proper use of overpressure valves will not only ensure that this design does not turn into a bomb but it will also ensure that an over pressurizing incident does not compromise the integrity of the system's components. There are a couple of different types of over pressure valves the main two are the kind which use a spring to vent if the pressure becomes too high and the other utilizes a disk which ruptures above a certain pressure threshold. Both types of valves have their place, the spring type is good when the system is likely to exceed the intended pressure causing it to vent and allowing the system to continue working. On the other hand, with the burst disk valve once the threshold is met the valve completely lets go and the disk will need to be replaced before the system can be repressurized. There are upsides to both designs, the spring valves are easier to work with and do not require any changing of parts if pressure is exceeded. Though if safety is of paramount concern, such as in high pressure cylinders, then burst disks have the advantage in that they are not prone to malfunction.

For this design burst disk overpressure valves will be utilized due to their robust function and simplistic design. Using the burst disk overpressure valves will save on weight and space of the arm. For this intended use the chances of the pressure reaching the point at which the bust disk will blow is relatively low and in the event which it does it would likely be during testing where the disk could be easily replaced for a new one. The *Overpressure Valves 1* diagram below shows the basic components of a burst disk valve.

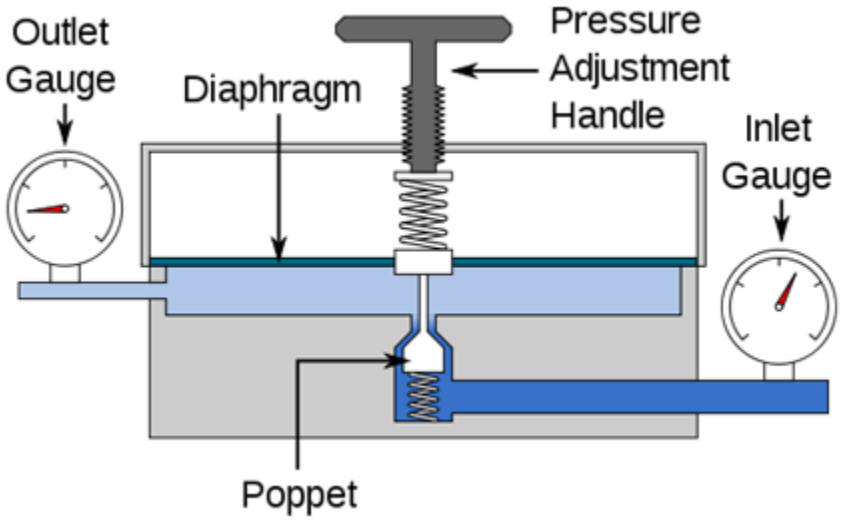


Overpressure Valves 1: Burst Disk Valve
Image from spiegl.org

4.4.1.5: Pressure Regulator

Pressure regulators a very similar to the spring type over pressure valves, they only allow for the passage of fluid until a certain pressure is reached. A pressure regulator works by using a spring, poppet, and diaphragm to sense the pressure in the system and either open or close once a desired pressure is reached. Basically these valves work to make sure that the pressures on the inlet and the outlet of the valve reach an equilibrium. That equilibrium can be altered by adding force to the diaphragm through the use of an adjustment spring.

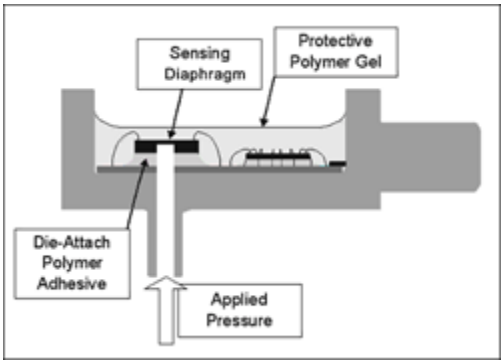
The way the pressure regulator would be used in this design would be by reducing the pressure coming out of the high pressure tank to the lower pressure which the pneumatic hand would operate at whilst in the closed position. By using this pressure regulator as opposed to just the two solenoid valves to inflate and deflate the hand the design becomes more streamlined and reduces the complexity of the feedback loop controlling the pressure in the hand. This regulator will allow the hand to inflate without overinflating which could result in some jerky movement as the hand tries to adjust to the desired pressure. The system could function without the pressure regulator, but it would leave more room for error and require a greater complexity to ensure the stability of the feedback loop. Below is the *Pressure Regulator 1* figure showing the inner workings of a typical single stage regulator.



Pressure Regulator 1: Pressure Regulator
Image from wikipedia.org

4.4.1.6: Pressure Sensors

Pressure sensors are devices which allow for an analog input of pressure to be converted to an electrical signal which can then be read by a computer. Pressure sensors work by measuring the force over a predetermined area then by a number of different means converting that force into an electrical signal. There are numerous different types of these pressure sensors which all have their merits when it comes to a particular design. For this particular design the pressure sensor will have to be relatively sensitive to provide the information needed. The likely candidates for the pressure sensors used will be either piezoelectric or capacitive sensing types. These two types of pressure sensors are fairly common and robust allowing for a wide selection to choose from. Both piezoelectric capacitive sensors use the movement of the diaphragm to generate a signal. The advantage to the piezoelectric is that a voltage is generated by the movement of the piezoelectric without the need for added power. While on the other hand the capacitive sensors require an input voltage and the movement results in the change in capacitance of the sensor leading to a measurable change in current. Below is the *Pressure Sensors 1* figure showing the general construction of a pressure sensor.



Pressure Sensors 1: Pressure Sensor
Image from maximintegrated.com

For the pneumatic hand system at least two pressure sensors will be required. One pressure sensor will need to be located in the pneumatic hand to ensure that the pressure will be at the desired level when the hand is opened and closed. The other sensor will need to be on the high pressure tank to allow the compressor to know when the pressure reaches the point at which the compressor will need to turn on and off. Another pressure sensor could be placed on the low pressure tank however it would not be critical to a minimalist design. Knowing the pressure in the low pressure tank may be useful in troubleshooting any possible problems which may arise but once again it would not be necessary for the feedback of the system.

4.4.1.7: Hoses and Fittings

There are many different types of hoses and fittings for pressurized fluids, but for this design the name of the game is lightweight. The system will not be at an overtly high pressure such that overbuilt plumbing would be necessary. The main options for hose are steel braided, rubber with fiber weave, and all plastic. Both the steel braided and the rubber with fiber weave are relatively heavy due to their ability to withstand very high working pressures around 3000 psi and greater. The all plastic hose sometimes referred to as macroline is lighter and better suited for lower pressures. Another advantage of this macroline is that the fittings are just push to fit and the pressurization of the system is what works to seal the lines. These lines are flexible and can be easily cut to desired length to add to ease of prototyping. The downside of the use of the macroline is that it requires pressure to properly seal. If the pressure is insufficient to create a seal, then another option may need to be used. Also the low pressure side of the system the issue of proper seal may be exacerbated by the lower pressure. More development will have to go into determining just which hoses will and will not work with the pressures that will be going through the lines.

4.4.2: Electro-Mechanical Elbow Hardware

4.4.2.1: Motor

The elbow motor for the arm will be tasked with mobilizing the exoskeletal arm at the elbow. At the elbow there will need to be sufficient torque to move the arm as well as any additional weight which the user intends to lift. Now the burden of the torque of the elbow does not solely rely on the torque of the motor, but a gearbox would provide additional torque without an increase in power required for the motor. The drawback is that as the burden of the torque requirement shifts to the gearbox the speed of the end shaft decreases. As it is intended that the motor should be able to rotate the arm at a timely pace, this being the case the motor will need to provide a relatively large amount of torque.

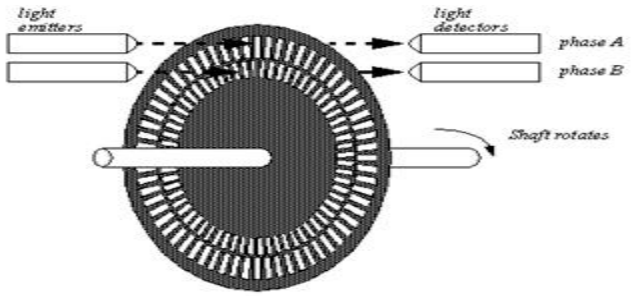
When it comes down to DC motors there are three main types; stepper motors, servos, and brushed DC motors. Stepper motors, such as the ones found in 3D printers and other precision machines, are made up of multiple windings of coils which need to be energized in sequence to produce coherent movement. The steppers motors typically have two windings and for those two windings the motor has four wires. The wires coming off the stepper motor are two for each set of windings Due to the complex nature of these motors a motor controller is needed to successfully alternate current between these coils. One advantage to using stepper motors is that they are precise, typically they can be controlled within one step which can equate to an increment around 0.1 degree of angle (depending on the gearbox). The stepper motors also have an edge in efficiency over that of other motor types.

Servo motors are also a good choice for position control situations which is why they are used in applications with model aircrafts to precisely actuate flaps and rudders. Servos typically have an encoder built in because without this the motors ability to accurately reach a position degrades. That is why the feedback for a servo is usually closed loop. DC brushed motors are another choice however they are better suited for movement for things such as wheels which do not require precision control. They can be fitted with an encoder however backlash of the DC motors tends to be an issue resulting in complications to the tuning of the feedback loop.

For the actuation of exoskeletal arms elbow a stepper motor will be used due to its high torque and precision movement characteristics. Even though it would be possible to use other types of motors to control the elbow the stepper gives the most leeway in design allowing for operation in either open or closed loop setups. This particular setup will be using the closed loop design so an optical encoder will be added to the motor to ensure the desired position can be achieved. In the event that that another setup would be used a servo could replace the stepper motor in that they work in a very similar fashion. Utilizing the stepper just provides a way that the encoder is not a critical aspect for the motor section of the elbow control system. This allows the encoder to be placed on the joint itself as opposed to the motor if the design calls for it.

4.4.2.2: Motor Encoder

The motor encoder is a device which allows for a position of a shaft to be calibrated and then tracked. One of the way which this is accomplished is through the use of an optical encoder. An optical encoder achieves this position tracking by using a set of light emitters and light defectors coupled with a disk to detect the rotation of the disk. The disk has a series of slots or dark spots which create an interruption in the light detected as the wheel rotates. By using two sets of light detectors and emitters the direction of rotation can be determined by which detector notices the interruption of the emitted light first. Below is the *Motor Encoder 1* figure outlining the basic construction and working of an optical encoder.



Motor Encoder 1: Optical Encoder
Image from codeforfree.weebly.com

There are two subtypes of encoders, they can be either absolute encoders or incremental encoders. Absolutes encoders not only measure rotation but they also store the position data even when the device is powered off. Incremental encoders on the other hand only retain position data since the device was powered on. For this application an absolute encoder would be desirable, this would allow the position to calibrated once. It would be possible for the exoskeletal arm to use an incremental encoder it would just mean that the position would need to be calibrated each time the system was powered up. This could be accomplished through an orientation protocol which would run at the beginning of each use of the exoskeletal arm. The upside to this is that there would be less of a chance for the system to lose calibration over continued use, but running a setup program each time the device is powered on could cause irritation to the user and reduce ease of accessibility. A calibration protocol will still need to be installed to allow for initial calibration and this will allow for recalibration in case a loss of calibration occurs.

The actual location of the encoder on the elbow assembly is yet to be determined and will remain that way until a design from the mechanical has been finalized. It would be ideal if in the final design the encoder were to be placed at the rotating joint of the elbow. This would allow for direct measurement of the elbow rotation and would reduce the error associated with correlating the rotation of the motor shaft to that of the joint. However, for testing purposed the encoder will be attached directly to the motor so that the control of the system can be refined and modified as necessary.

4.2.2.3: Motor Controller

The motor controller will be needed to control the speed and direction of the stepper motor. Once again stepper motors require the alternation of energizing coils in sequence to produce coherent movement and the best way to accomplish this is through the use of a motor controller chip. The motor controller will be tailored to the motor and will depend on the number of phases, the voltage required, and the maximum current draw. The motor controller boards consist of a set of inputs which take in both analog and digital inputs which consist of things such as speed, direction, and break then interpret them into the schema which is needed to produce the desired movement. Additionally, the motor controller can act as a gate between the high voltage battery and motor itself only requiring a low voltage signal from the MCU.

For this design the motor controller will likely be for a bipolar stepper motor and will be relatively simple to obtain. However, the motor may change requiring a change in motor controller if the differences are significant. It is the intent of the electrical team to at least get a set of all components together to refine the control system even if the components must be changed later down the road. The components for such a test bed would need to be low cost but have a similar design to that which may be used in the final design. The idea is that if the system must be adapted to conform to different components that the modifications to the system can be done quickly without requiring an extensive amount of time.

5: STANDARDS

5.1: Software Standards

5.1.1: Arduino Standards

On the Arduino website, there is a style guide for the recommended way to write and format Arduino code. These guidelines focus on readability and create clearer code that can be interpreted by more people. They also include sections are making tutorials and building circuits. It is stated that they are not strict rules, but rather suggestions that an Arduino programmer should think about and consider.

5.1.2: Tutorial Standards

When writing a tutorial, there are a few things that are good practice for the programmer to do. The style guide provides consistency in writing techniques and reduces disjoint between separate tutorials. It is good to write in the active voice, and to write clearly and conversationally. Do not write as if it were a textbook, but as if the person was in the room with you. It is also good to write short, declarative sentences and put instructions in the second person. Directions should be in definite terms and pictures should be included with the schematics.

The style guide also makes things easier from readers that may not have much experience. The writer is to check all assumptions, explain things conceptually from a high level before going step by step, and be consistent with the terms he or she uses. It is also against standards to use abbreviation without spelling them out first and trying to make an example do multiple things and combine functions (unless it is a tutorial about combining functions).

5.1.3: Code Standards

The coding style standards for Arduino are designed to make the code easy to follow. This can be very good if there are multiple programmers or if the project is going to be maintained for an extended period of time. Some of the standards are in relation to writing examples and suggest things such as going with readability over efficiency. This does not truly relate to our project, as we value efficacy and are all qualified to read code. For this, reason we will choose to forgo certain standards listed in the guide and focus on others.

One important style technique is to put the *setup* and *loop* functions in the beginning of the program. They are the most important and should not be difficult to locate. Another standard is to use heavy commenting. Every variable/constant, code block, and loop should be commented and the title block of the program should contain a full explanation of the code. If statements should be verbose and in block format. Also pointers and *#defines* should be avoided.

Variable naming conventions are another big topic. Single letter variable names and non-descriptive names such as *val* or *pin* should be avoided. Variables that won't change should be declared as constants. Variable types used should include boolean, char, byte, unsigned int, float, long, etc. They should not contain types such as uint8_t. Number schemes that are confusing, such as pin2 = 3 should not be used and if pins need to be renumbered an array should be considered.

5.1.4: Example Circuits Standards

Arduino has a few standards on building test and examples circuits. There are not a lot, but the ones listed are somewhat useful in maintaining readability and comprehension. The two big ones are that digital input switches should use a pull-down resistor so the logic of the switches interaction makes sense and circuits should be kept simple and avoid unneeded components.

5.2: Hardware Standards

Many standards have to be taken into account when designing a reliable system that interfaces with humans, so as to prevent fatal errors. Many of the standards stated in the sections below were found from UL's and OSHA's websites. For organizational purposes, the hardware standards section has been divided into three different categories; rough electrical standards, interface standards, and component standards.

5.2.1: Rough Electrical Standards

This standards' section deals with the components that make up the PCB. They essentially deal with the proper use of electronic components to guarantee safety and prevent incompatibilities between different parts of the system. Since the PCB is not intended for use in direct interface with the patient, many of the standards that are stated in this section attempt to define how electrical components must be arranged and safe practices for creating a PCB design that can reliably handle the load it was intended for usage.

5.2.1.1: Power Distribution Standards

From the OSHA website for standards, reliable power standards were derived from standard 1910.269, which deals with power generation. Although the section defines various sorts of power generation such as mechanical power, the team will be mainly focused on electrical power generation and distribution. This is important, since the team is dealing with a powered system which might be hazardous for the engineers that are assembling the electrical components with the battery that will be used. The following sections were obtained from standard 1910.269 and are believed to be applicable to the project.

- 1910.269(a)(1)(i)(C) specifies the use of a specialized testing site for electrical measurements. This is vital to the team, since breadboard measurements must be taken on components that will be in direct contact with current. Fortunately, the

sponsor is planning on opening a laboratory for the team to perform tests on, where the appropriate tools will be provided, and safety will be in place. Alternatively, the team will make use of testing sites provided by the university in order to obtain design measurements. In-home measurements should only be taken as a last resort while still under the right safety conditions. These conditions include the use of correct tools and safe practice of power distribution to a component or a subsystem.

- 1910.269(a)(2) defines the level of training and electrical knowledge that is required from the team members being in direct contact with live components. Since all designers have an electrical background due to coursework taken, it is expected that right judgment will be applied when dealing with energized equipment by all team members. In any case, any members that have doubts regarding whether or not a particular implementation might endanger the system or even their lives should ask a fellow designer for advice. Given the case that no valuable input can be generated from the team, expert advice shall be sought from a knowledgeable professor or a professional in the field.

- 1910.269(c) explains a category named job briefing. For the scope of the project, the job briefing standard applies to the extent of the team being fully aware of as many safety implications as possible when dealing with an energized system, whether when taking measurements alone or in a group. Power regulation and impedance shall be set in place when necessary to avoid overcurrent and the destruction of components and even the system.

There are many other power-related standards cited under section 1910.269 from OSHA that the team will have to comply with in order to ensure a safe work-space for creating a prototype; however, most of these standards are derived from common sense and will be easily fulfilled.

5.2.1.2: Wiring Standards

Standard 1910.304 from OSHA defines wiring design and protection. This is especially important in electrical design applications, since all components that are electrically wired on the PCB as well as outside of the PCB with jumpers constitute a safety hazard for users using interfacing equipment as well as a design hazard for the creation of the prototype. Although most of its sections pertain to general electrical requirements, some standards have been assimilated differently for the team to consider.

- 1910.304(a)(2) defines the polarity of connections. This must be taken into consideration for the team in order to prevent scenarios such as reversing voltage toward the MCU and damaging the board. The idea that the team has planned out after extensive research is to isolate the sensitive, logic elements from the power components so as to prevent current spikes or back EMF from the motor from damaging the equipment.

- 1910.303(b)(5) relates to the current protection for the connections being done in a system. This is listed under a different standard; however, it still derives from the need of standards for wiring of electronics. In order to conform to these

standards, the team is responsible for developing a system that can handle current protection and that guarantees the integrity of the system as a whole when a subsection of said system becomes damaged. The regulation components selected for the system will have to work reliably to ensure easiness of recovery.

Many standards found under UL and IEEE pertaining proper wiring connections are currently unavailable for acquisition given the cost per standard and the scarce resources available for obtaining them; nevertheless, the team will consistently work to make sure that the wiring of the system will be constrained by safety requirements and hazards will be avoided at all costs once the prototype has been created. These standards include the maximum temperature that is considered safe by the electrical circuit and how the current is handled in the system. To determine the maximum temperature considered safe, individual components shall be gauged accordingly and compensated by the use of heat sinks.

5.2.2: Interface Standards

This section pertains to standards that enter in direct contact with human interaction. UL 61010-1 contains a vast amount of information on what requirements are necessary for the safe interaction between the user, who in this case would be the patient, and the electronic device. Standards regarding body temperature are found in ASTM C1055 [22], and from the information found, the maximum acceptable temperature to human touch is roughly 48°C [22]. For values above the given maximum, insulation should be provided for the patient to be safe from burning hazards. Additionally, for the use of EMG sensors, temperature regulation must be considered for the instance in which the electrodes or cables attached to the patient become hot.

For motors' temperature, insulation might also need to be provided in order to prevent the patient from feeling heat discomfort given by the temperature generated by the energized motor. The motor will be explored upon testing and decisions will be made regarding which insulation will work best for the scope of the project, depending upon the current motor's rating.

5.2.3: Component Standards

The components selected must adhere to the general standards set forth by IEEE, OSHA, ASME, and the ASTM International. All of the components should be safe to assemble and work with through the use of proper methods. None of the selected components will pose a hazard to the engineers constructing the system, the end-user, or any other individual who may come in contact with the system.

All of the selected electrical components will fall within the standards of IEEE and will be configured in accordance with those same standards. All of the materials used will be approved by the ASME standards. The inputs and outputs of the components will adhere to the standards set by the different engineering governing bodies ensuring that the end product will be safe above all else.

6: DESIGN

6.1: Input Design

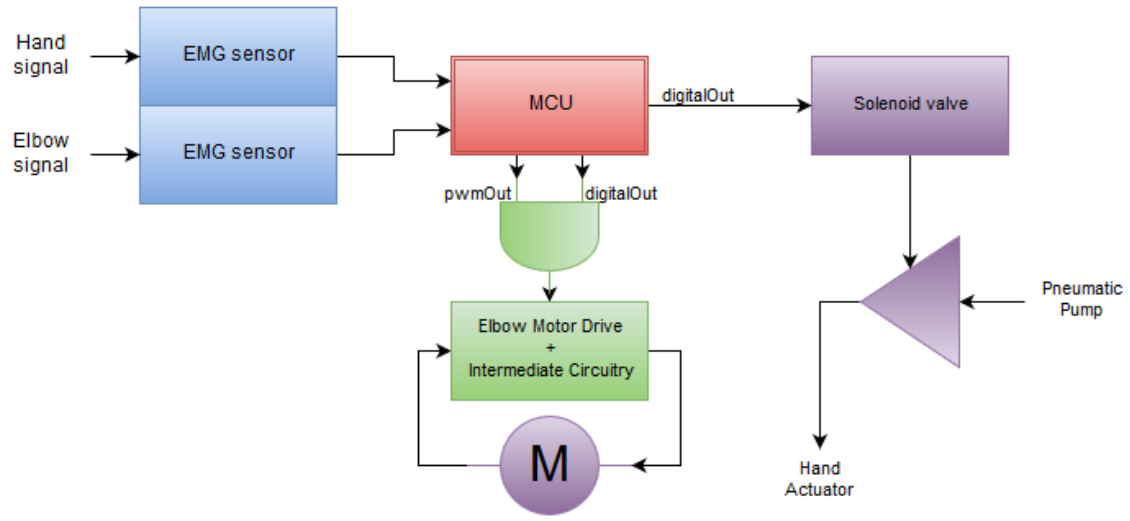
For the scope of this design, the system is required to be as responsive as possible, so as to allow the patient to have fluidic motion without the robotic feel that is given by many of the current implementations of an exoskeleton. This constraint is provided by end-user requirement EU.2 and is likely to be one of the requirements that require most research and design efforts to get right, in conjunction with most of safety requirements.

Arguably, with enough information provided by the EMG data collection test bed, the team supports the statement that this collected data could be used to tell the system some information about the patient that will be used in order to design an exoskeleton that best fits the patient's needs. In short, the electrical signals' produced by the patient muscles will be used, with help of a reference, to tell the system how much force must be applied in order for the exoskeleton to replicate intended range of motion.

6.1.1: High Level Design for EMG Inputs

The team will perform a test bed as indicated in the section 7.1.1 as soon as the patient comes down to the lab for some measurement testing. Proper EMGs as chosen in the section 6.1.3.1 will be used to measure electrical activity and record patterns that can map electrical readings to desired mechanical speed for the motor system of the exoskeleton. Since the hand has been designed to only perform a digital event of opening/closing for the scope of this project, it is expected to use only any significant measurement of the patient's electrical signals provided for the hand in order to use the pneumatic systems to open and close the hand. Although the hand is expected to have two solenoid valves for flexion and extension, only one EMG sensor will be used to receive digital events from, the actual hand. Feedback sensors are planned to be used to control the gripping pressure that the hand is exerting upon an object.

After obtaining information from the patient's test bed, the team should be able to have an idea on how to map given EMG signal inputs to speed outputs for the motors. This mapping will be programmed on the Arduino board and signals will be sent to both motor encoders and solenoid valves. In the next two sections are the steps that will be taken from a muscle fiber signal reading to transform into a speed signal for the elbow motor. For further clarification, *High Level Design for EMG Inputs 1* represents how the group plans to achieve analog movement on the arm, as well as digital movement for the hand.



High Level Design for EMG Inputs 1: Shows relationship between inputs and outputs

6.1.1.1: I/O Mapping for Elbow

1. The patient flexes the muscle of actuation for the elbow.
2. A signal is sent down the nervous system into the muscle fibers that make up the muscle.
3. The signal is received by the EMG sensors and sent to the MCU's analog pins.
4. The MCU reads the analog signal sent by the EMG sensor and turns it into a digital signal by means of an ADC.
5. The MCU processes the mapping of the signal and sends the information out through both a PWM and a digital pin.
6. The digital works as a gate, opening the way for the PWM signal to travel to its destined location.
7. The PWM signal passes through some additional hardware logic in order to send information to the motor encoder as to how much speed must be applied.

6.1.1.2: I/O Mapping for Hand

1. The patient flexes the muscle of actuation for the hand.
2. A signal is sent down the nervous system into the muscle fibers that make up the muscle.
3. The signal is received by the EMG sensors and sent to the MCU's digital pins.
4. The MCU reads the input received by the digital pin that connects to the hand actuation's EMG and sends a signal to the solenoid valves for opening.

5. As the pneumatic pressure can build up on the hand actuators, the hand starts closing.

6. Pressure sensors sense when enough pressure is applied on an object for gripping and send a signal to the MCU.

7. The MCU receives the signal sent by the pressure sensor and sends a signal back to the solenoid valve that controls the hand in order to stop air flow and stop the hand from gripping down the object.

6.1.2: PWM Input to Elbow Actuator

After the EMG sends the muscle input to the MCU, the board will be in charge of producing a suitable analog output for the muscles to read. The Arduino board that has been chosen can only produce an analog output by means of PWM. Since PWM is represented as a square wave signal with different times of being in an "on" and "off" state, it is said to be in switching form; thus, the motor cannot accurately provide a constant result based on the patient's input. For that reason, two choices have been considered by the team. *PWM Input to Elbow Actuator 1* shows which of the choices rank highest and what factors were considered.

The first choice would be to convert the PWM signal into a suitable DC signal via a small IC unit that can be easily used to convert the input into an ideal DC output and control the motor speed by sending signals of different voltages depending on the different inputs provided by the patient's muscle. This might include more complexity in the system, as more steps must be taken in order to convert the PWM signal into a speed output. As the system becomes more complex, more unit testing is required and the system becomes less reliable.

The second choice, which is considered to be the most suitable given the current design, is to equip the motor with a drive that can automatically convert an input PWM signal into an output speed for the motor. This is done by a DAC that is built-in on the drive. Section 9.4.1.2 explains in depth how this is achieved.

		PWM-to-DC Converter		Motor Drive	
Criteria	Importance Weight %	Rating (0-5)	Weighted Rating	Rating (0-5)	Weighted Rating
System Simplicity	15%	3	0.45	3	0.45
Controllability	20%	4	0.8	5	1
Multi-purpose	20%	2	0.4	5	1
Low Cost	10%	5	0.5	2	0.2
Isolation	15%	3	0.45	5	0.75
Input/Output Mapping Easiness	20%	3	0.6	5	1
Total	100%	3.20		4.40	

PWM Input to Elbow Actuator 1: Decision matrix between converter component and motor drive

6.1.3: System Materials for Input

For the subsystem that handles the input obtained from the patient's muscle signals, materials are wisely chosen in order to guarantee the cleanest signals. This is a necessity due to the fact that signals obtained at the skin of the patient often tend to be very noisy compared to their needled counterparts. For the input design, the main focus shall rest upon proper choosing of EMG sensors and electrodes.

6.1.3.1: EMG Sensor

Several choices were defined for EMG sensor selections. Originally, the MyoWare Sensor was to be used for signal retrieval of the exoskeleton's control system. It consists of a small PCB sensor that contains two types of outputs. One output is the raw signal, while the other output is the smooth rectified signal. Since the raw data is only needed for the patient's EMG testing purposes to determine valid ranges for actuation, this MyoWare EMG sensor is a great choice for the patient's EMG testing.

The team is constrained by the sponsor to use their designated EMGs for system integration. This is not a negative factor against the team, however, as their EMGs have proven much better results than their MyoWare counterparts in terms of signal retrieval. The idea for EMG integration will be to use a schematic for the design that the organization employed for its known prosthetic limbs with servo actuation, and choose suitable components that will then be planned out on a single PCB. Since the exoskeleton will originally be planned out as a wheelchair-bound model, the EMG units do not necessarily need to be placed on the patient and can thus be somewhat bulkier than what the team had in mind at first. Additionally, the cost of using Limbitless's EMG sensors would make a big difference with respect to MyoWare's due to the fact that they only output smooth rectified signals and are made up of simpler components, which will be placed on a single PCB. *EMG Sensor 1* below shows some of the aspects in which Limbitless' EMG sensor proves to be a better choice.

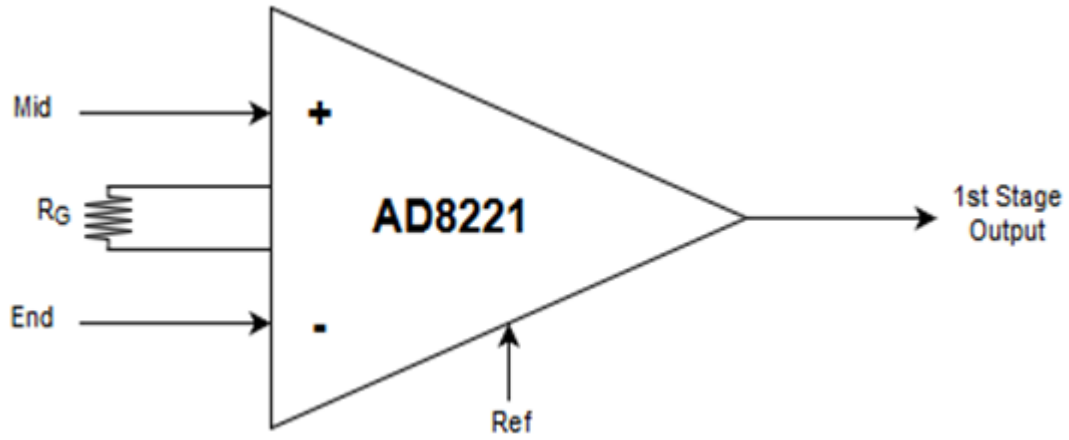
		MyoWare		**Limbitless**	
Criteria	Weight %	Rating	Weighted Rating	Rating	W. Rating
Power Efficiency	20%	3	0.6	3	0.6
Signal Quality / Bandwidth	25%	4	1	5	1.25
Maintenance	15%	3	0.45	4	0.6
Low Cost	15%	3	0.45	5	0.75
Availability	15%	3	0.45	5	0.75
Portability	10%	4	0.4	3	0.3
Total	100%	3.35		4.25	

EMG Sensor 1: Why Limbitless EMG's prove to be a better option for integration

After determining which EMG sensors are to be used, the team must now understand how the EMG is structured in order to assemble it on a PCB. This is especially important since the retrieved signals will need to be as precise as possible in order for the MCU to handle the proper interrupts.

6.1.3.1.1: EMG Construction First Stage: Pre-Amplifying

The first factor that is considered upon creating an EMG sensor lies at the interface between the sensor itself and the electrodes. Since bipolar electrodes are being used, two electrodes are placed at the muscle at an optimal inter-electrode distance. This distance will be obtained from the patient's input testing. An instrumentation amplifier shall be used in order to pre-amplify the signals prior to differentiating them to eliminate noises that are common between the two electrodes. The reference electrode is to be placed at a bony area of the body, which would equate as a voltage with respect to ground at the output. A bias is then placed on the instrumental amplifier in order to readout the signals given by the muscle. For an AD8221 instrumentation amplifier as the ones used by Limbitless Solutions, the operating voltage for the supply bias ranges from 2.8 V to 18 V. A gain resistor R_G is used to amplify the signals that enter the amplifier. This gain ranges from 1 to 1000 and the equation for the gain is given by the Analog Devices' AD8221 datasheet as G = 1 + , on page 4. Below is *EMG Construction 1*, which represents the first stage of an EMG readout that was created by the team, based on both an AD8221 datasheet and some schematics that were provided by Limbitless. Keep in mind that the signal that would be read from the patient's muscle at the electrode would be in microvolts, as there is high impedance due to the layer of skin between the muscle and the electrode.



EMG Construction 1: Stage 1 Instrumentation Amplifier

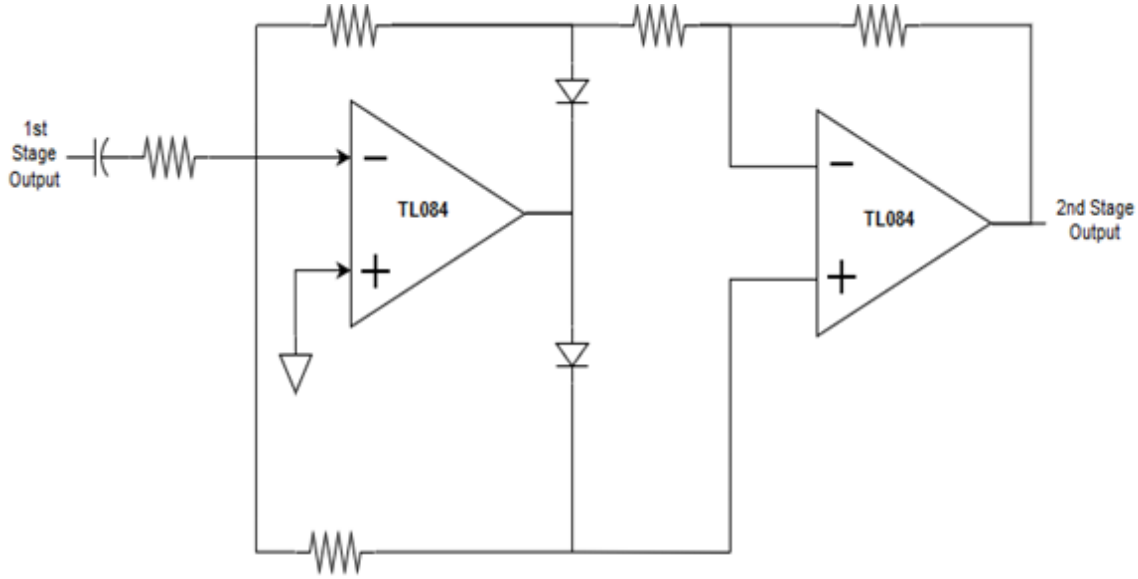
6.1.3.1.2: EMG Construction Second Stage: Filtering and Rectifying

Past the first stage in the creation of an EMG, the signal is already considered raw; the common noises have mostly been canceled out and the signal has been amplified. For digitalized purposes, the signal coming inside of the Arduino should not be negative. This is because the MCU only reads inputs from 0 to 5 voltages in the positive peak. Not only might this damage the board, but it will cause instability at the programming of the MCU. For this reason, the signal must be rectified, so that its processing is much easier. As seen

in *EMG Construction 2* below, a bridge of diodes is used in combination with TL084 amplifiers and some resistors to convert the negative components of the signal into positive components. This is still useful since a negative signal might still be meaningful enough for the controller to process as an actuation signal.

The way the rectifier works in this circuit is simple; positive signals pass down the circuit toward the positive terminal of the amplifier and output the same signal. Negative signals take the next route of the circuit and enter the negative terminal of the amplifier, thus converting its signal back into positive. This way, the signal never reaches a negative state.

Additionally, the signal must be made as DC as possible; for that to happen, some high frequencies must be eliminated by use of high-pass filters. Active high-pass filters, as shown in *EMG Construction 2* below at the beginning of the diagram, attempt to reduce the noise produced by higher-frequency components of the muscle fibers firing up signals that are read by electrodes. Active low-pass filters can be seen used on the second amplifier by means of resistors incoming from the negative side. Past this stage, the signal has been rectified and many unwanted frequency components have been eliminated by selective filtering.



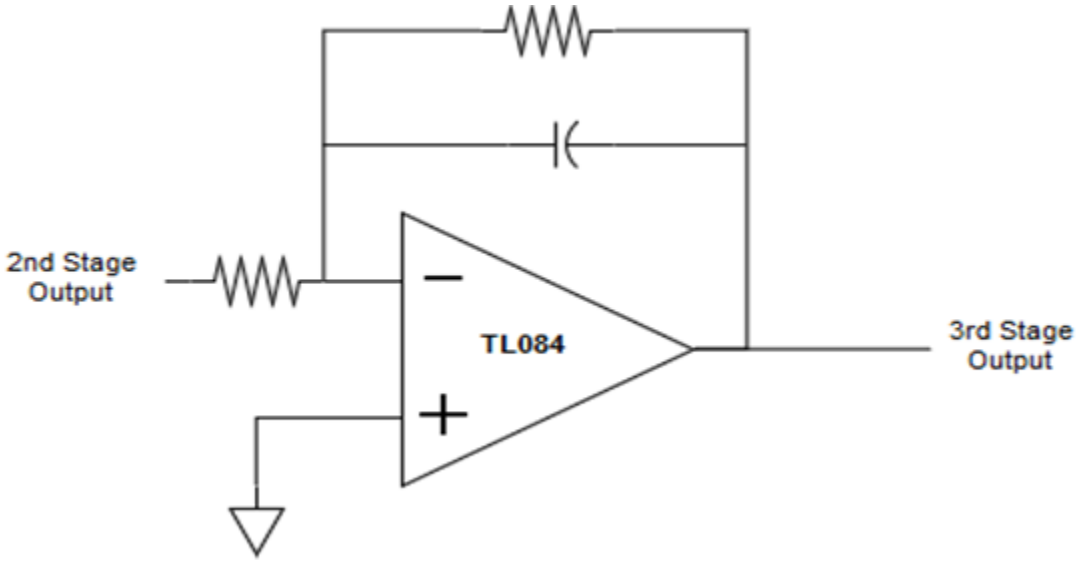
EMG Construction 2: Stage 2 Filtering and Rectifying

6.1.3.1.3: EMG Construction Third Stage: Smoothing

Once the signal has been rectified and filtered by the circuit, it must be smoothened. The reason behind smoothing a signal is simple; hysteresis. A signal that tends to stay at a particular peak may be thought of as an intentional contraction of the muscle. If a signal were to change a lot over time when in reality it is not meant to in terms of output, the MCU would be constantly sending erroneous inputs to the motor drive. An algorithm could be developed such that the normalized values after successive readouts are the only ones that get sent to the motor actuator; however, this would potentially take longer for the prototype to be ready, as more research would have to be developed in order to find

such algorithm. Additionally, running the algorithm would decrease the responsiveness of the system, as the MCU would be busy more often trying to make sense of the input, which keeps fluctuating between different values.

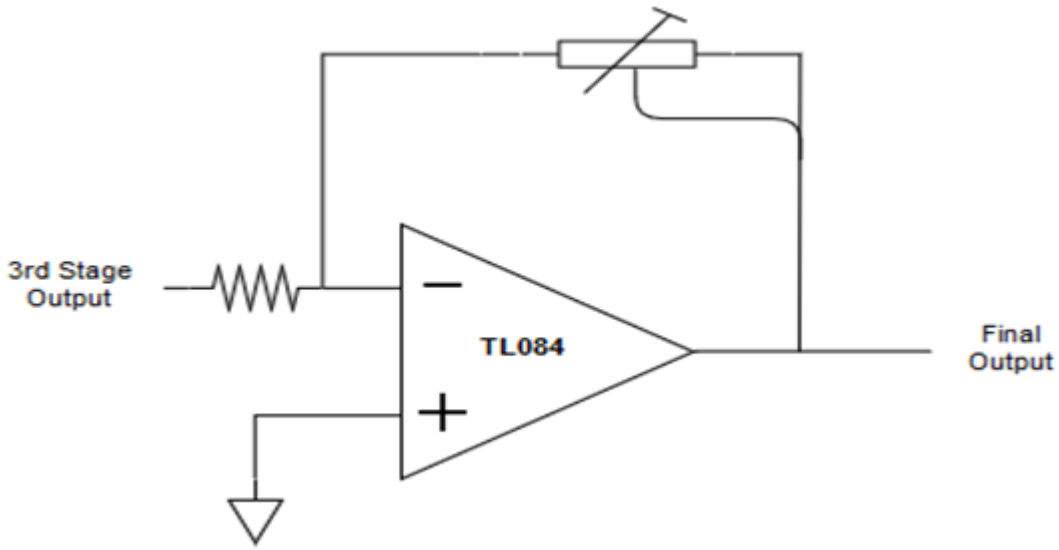
For the purposes of easiness, the envelop of the peaks at the signal will be roughly obtained by use of an envelope detector. Since the signal had been previously rectified, this envelope detector will be in charge of making the signal as smooth and representative as possible, by trying to gap the peaks that the signal reaches as the arm is flexed. The circuit for the envelope detector is shown on *EMG Construction 3* below.



EMG Construction 3: Stage 3 Smoothening

6.1.3.1.4: EMG Construction Last Stage: Selective Amplification

After the signal has passed through the third stage of the circuit, it should be ready to be sent. The last stage that the signal must go through is selective amplifying. Even though the signal has been a little amplified from all the amplifiers it has gone through, it had not been particularly amplified significantly since the pre-amplifier, which was already reduced by the differentiation. In this last stage, the gain of the signal is adjustable by a trimming potentiometer, which essentially adjusts the gain by adjusting the resistance on one side of the equation. The diagram for the trimpot circuit is shown on *EMG Construction 4* shown in the next paragraph. The basic difference between a trimpot and a regular potentiometer is that the trimpot is smaller and easier to place on a PCB, and is intended for use only when building the circuit, whereas the potentiometer can be used while the system is in use. Note that the negative sign will still enter the signal as positive due to all 4 amplifiers cancelling each other's negative out.



EMG Construction 4: Stage 4 Amplification and Sending of Signal

6.1.3.2: Electrodes

Since it is hard to determine where the electrodes will be located on the patient, different ways of placing electrodes must be taken into account, so that one will rule out the design once the input testing determines electrode placement. The initial assumption is that all the EMGs will be able to connect to the patient's arm. For this implementation, a conductive sleeve shall be used for the patient to wear. Snap kit electrodes will be sewn in on the sleeve. The EMG sensors will have to run two or three cables per EMG down the patient's arm until they reach the electrode location. The sleeve will assist the cable in staying in place to prevent parasitic capacitance produced by the moving of cables. Below is *Electrodes 1*, an image retrieved from the mechanical team's senior design slides for their sleeve ideas and the materials that will be needed to produce a conductive sleeve. Note that the rectangular patches are conductive tissues that will replace the gel used on disposable electrodes.

Electrodes 1: Conductive sleeve concept put together by the mechanical team

Alternatively, if the patient requires alternative muscle locations for motor actuation, disposable electrodes shall be placed on those locations. The patient will then need to change these electrodes every time he takes off the device to prevent contamination given by skin impedances such as oil excretions. Ideally, since the patient suffers from muscular degeneration, the electrodes might have to be as small as possible to prevent cross-talk interference. The mechanical team for the exoskeleton will have to determine a way to make the cabling system stay in place so as to prevent interference.

There have been some ideas for the types of disposable electrodes that could be used for the alternative way. So far, the team has used 3M electrodes, which are cheap in price but very big and impractical for our patient. Additionally, it is hard for bipolar electrodes to be placed at a closed distance due to the radius of the adhesive material. The proposed electrodes that would be used are the Noraxon dual EMG electrodes. They are cheap as well and also smaller, which will help to provide better communication between the patient's EMG signals and the interface of the electrodes. Variants of this brand of electrodes include dual electrodes, which combine two electrodes into one single piece in order to provide an even smaller inter-electrode distance.

6.2: Software Design

6.2.1: High Level Software Design

6.2.1.1: Original Software Design

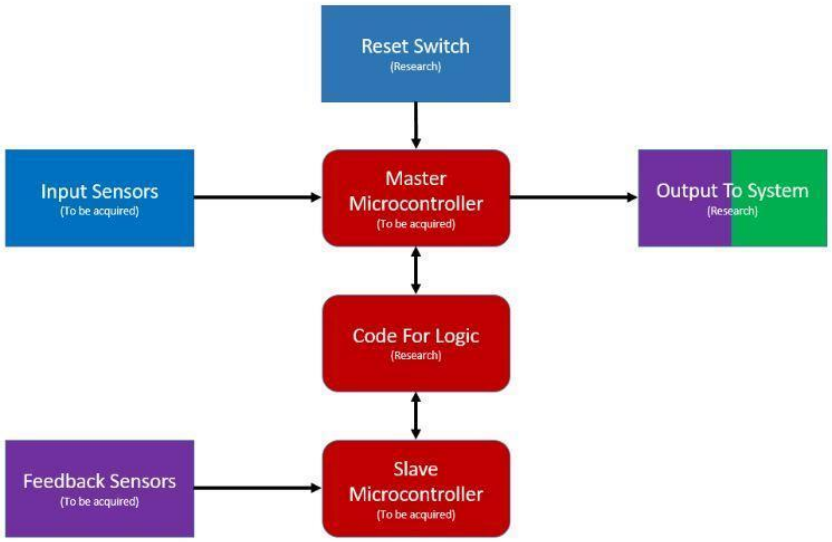
The high level software design was something that we looked at early on when designing our system. We didn't know specific components, but knew general groups. With this information, we began to develop a layout for what would be connected to what and how the system was going to operate. There were a few iterations.

The component groups that we have can be broken down as such: Input sensors, feedback sensors, the output system, a reset switch, and the microcontroller(s). The idea is that the input sensors would read input and there would be some output to the rest of the system. Feedback sensors would stop the output if they detect anything that deems it necessary and a reset switch could be hit to do a hard reset of the system if needed. A control unit would be needed to run the logic and make all of this happen.

One factor that influenced our design was the fact that we would need a large number of pins to attach all of our components to. Originally, there were to be at least eight input sensor, at least 13 outputs, and also a few feedback sensors. Another factor was that this processing is very time sensitive. There cannot be unreasonable lag on this system or the system becomes unusable and therefore worthless. The combination of these two factors lead us to consider an SPI design.

Our design was pretty straightforward. We were to have two Microcontrollers communicate with each other and co-process. This would solve the problem of having limited pins as now we would have the pins of two controllers to work with. This would also help with the processing speed as they could share the load. The idea was to have the master (potentially a more powerful controller) take input from the input sensors as well

as the reset switch and send corresponding signals to the output system. It would be coded to handle the logic for this. The slave would be responsible for just the feedback sensors. It would handle all of that logic and a send a signal up to the master. The master would read this and could factor the slave's response into its algorithm for deciding what to send to the output system. The diagram for this design is shown on *High Level Software Design 1*.

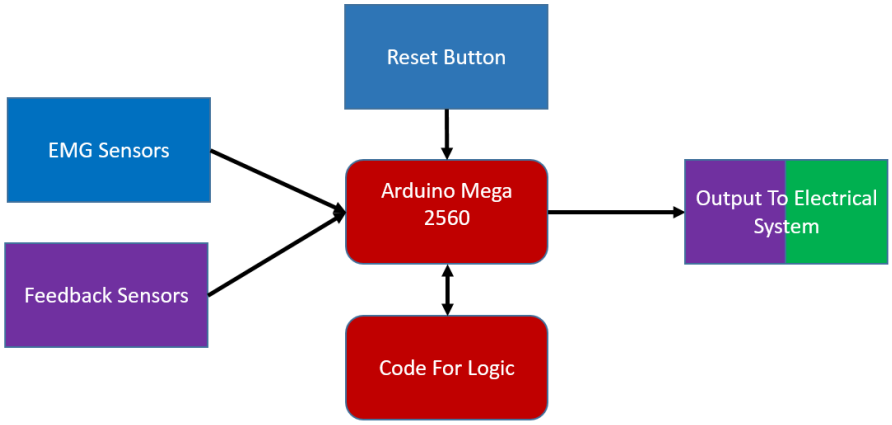


High Level Software Design 1. This is the block diagram for our original high level software design.

6.2.1.2: Final Software Design

After the creation of the initial design, some constraints were changed from external sources. The mechanical team had a meeting and adjusted some things. The main thing that changed as far as the software was concerned is that they reduced the number of outputs needed. This in turn, freed up some pins. We decided to adjust the software design to something more practical for the current needs.

The significant change that was made to the design was that we got rid of the second microcontroller and made it a single processor system. There were some thoughts behind this. The first was that we no longer needed the extra I/O pins because we no longer had so many outputs to deal with. Another thought was that because it did not have to do such heavy processing on deciding with output to activate, some of that processor power could be used for the for the feedback sensors. We figure that if we use a fast enough processor, we should be able to manage the delay time with needing an extra one. Lastly, using a single processor will reduce the cost of the system. This is always good as it allows us to allocate funds to other components and resources. The updated software design can be seen in the introduction, in figure *Initial Software Block Diagram 1*. Once we finalized the hardware design and components, we were able to update our software design to be a little more specific. This final design is displayed in figure *High Level Software Design 2*.



High Level Software Design 2. This is the block diagram for our final high level software design.

6.2.2: Software Design Testing

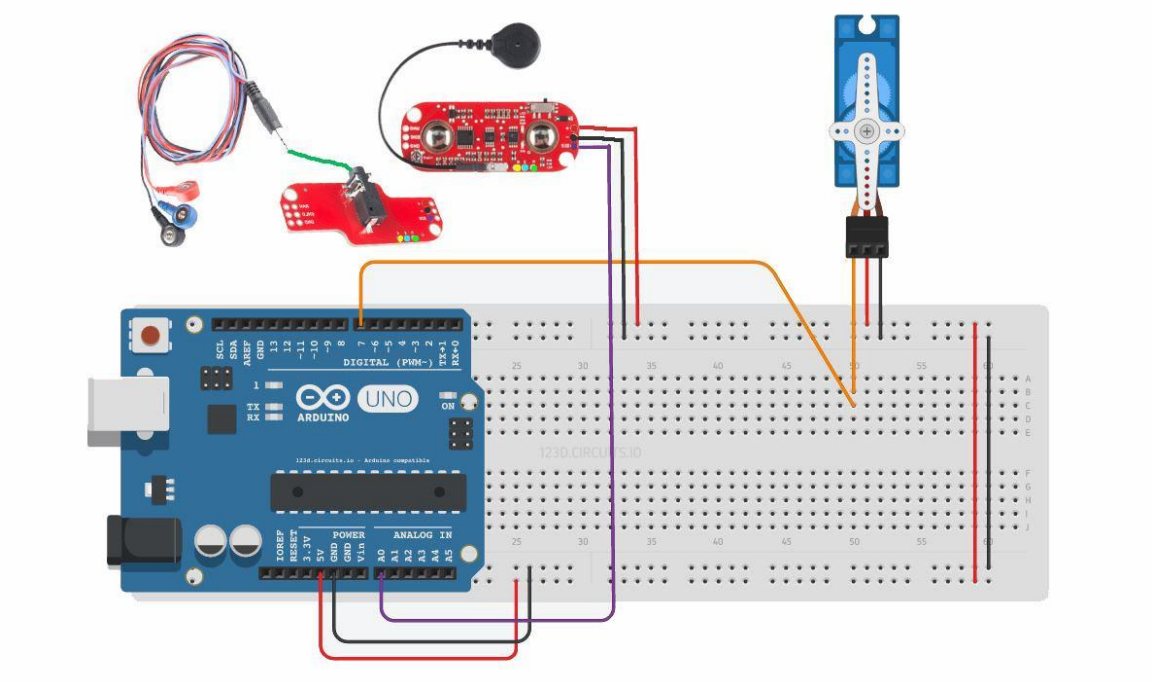
6.2.2.1: EMG Reading Software

As we got further into the design of the project and things started coming together, we decided that we need to get some preliminary tests done to make sure things would work as we imagined. On the software side of things, we decided to try out some test code and see if we could get the EMG signal read to the development board and produce some type of output with this. A day was spent on this task.

We had a MyoWare EMG sensor, a cable shield, and some electrodes. The first step was to connect them together into something that could go from a user's arm to the microcontroller. It went as follows: The electrodes get attached to the arm with sticky pads. The cable for these electrodes then gets attached to the cable shield by means of a 3.5mm jack. The cable shield had to be soldered to the EMG with all of the connections lining up and then the EMG could be wired to the controller.

For a means of output, we went through a few different stages. We set up a serial monitor to display the readings of the EMG as voltage numbers, but we decided that we wanted something a little nicer to look at. We set the monitor to display the information as a graph and we were pleased. Next we wanted some type of hardware to react to the input. We tried some new things. First we tried an LED. We wired it to a PWM pin so we could map the brightness to the strength of the input signal. The problem with this is that it was hard to tell with the naked eye, what the readings were. The differences in brightness were not always that great. This is when we landed on the idea of a servo.

We had seen servos being used as a means of output in other testing situations and thought it would not be a bad idea for us. We wrote some code to have the servo move whenever the EMG picked up some muscle activity and we tested our system. It worked. The wiring for the test circuit we built can be seen in figure *Software Design Testing 1*.



Software Design Testing 1. This figure shows the wiring connections for the circuit we used to test that we could process input from an EMG into a tangible output. EMG images were used courtesy of SparkFun [I-8]. The breadboard circuit was built using Autodesk [I-5].

6.2.3: Software Low Level Design

6.2.3.1: Low Level Software Implementation

The flow diagram displayed in *Software Low Level Design 1* shows how the code will work for the system. The functions that convert analog signals to voltage are for debugging purposes and do not generate to output in the production environment, but everything else in the diagram will play a vital role in the success of this system. The first thing to do when the program starts is to enable the motor and turn on the compressor. After this we go into our main loop.

In out loop we will begin by reading the values from all of the EMGs. We then check if the wrist value was high enough to warrant a response, if it is we go to the wrist function. In this function we check if the hand is open or closed and set all appropriate variables accordingly. This means setting the pressure sensor we want to be reading, the solenoid we want to control etc. Next we open the solenoid and continuously read the pressure. Once the pressure we want is reached, we close the value. This will give the effect of opening the hand. If the wrist value is too low, we ignore it and skip this step.

Next it will check if the biceps or triceps value is higher and we go to function of the leading muscle group. If they are equal, neither will be done. The biceps and triceps function are mostly identical, we check the EMG value. If the value is over the high threshold we set the direction of the motor and move it at speed 2. If value is not above the high threshold, but is above the low threshold, the direction of the motor is set and the speed is set to 1. If the value is not above the low threshold, we ignore it and do nothing.

After the biceps and triceps functions are complete, we do a small delay for stability and restart the loop. The program will continue to loop through and process input from the user.

6.2.3.2: Microcontroller Pin Mapping

6.2.3.2.1: Mapping for Elbow Control

The subsystem to control the elbow is comprised of two main component classes attached to the microcontroller.There will be two EMGs (one responsible from flexion and the other for extension) and we have a driver for the stepper motor. The stepper motor driver has 7 pins we need to account for. The first is Vcc, which is the voltage. The second is PUL, which is the pulse to control the motor. Third is DIR for the direction of the motor (since it is bipolar). The fourth is ENA, which can enable or disable motor A. The last three pins are ground for PUL, DIR, and ENA. We designed an optimal way for everything involved to be connected to the Arduino that would save pin space for other components we need.

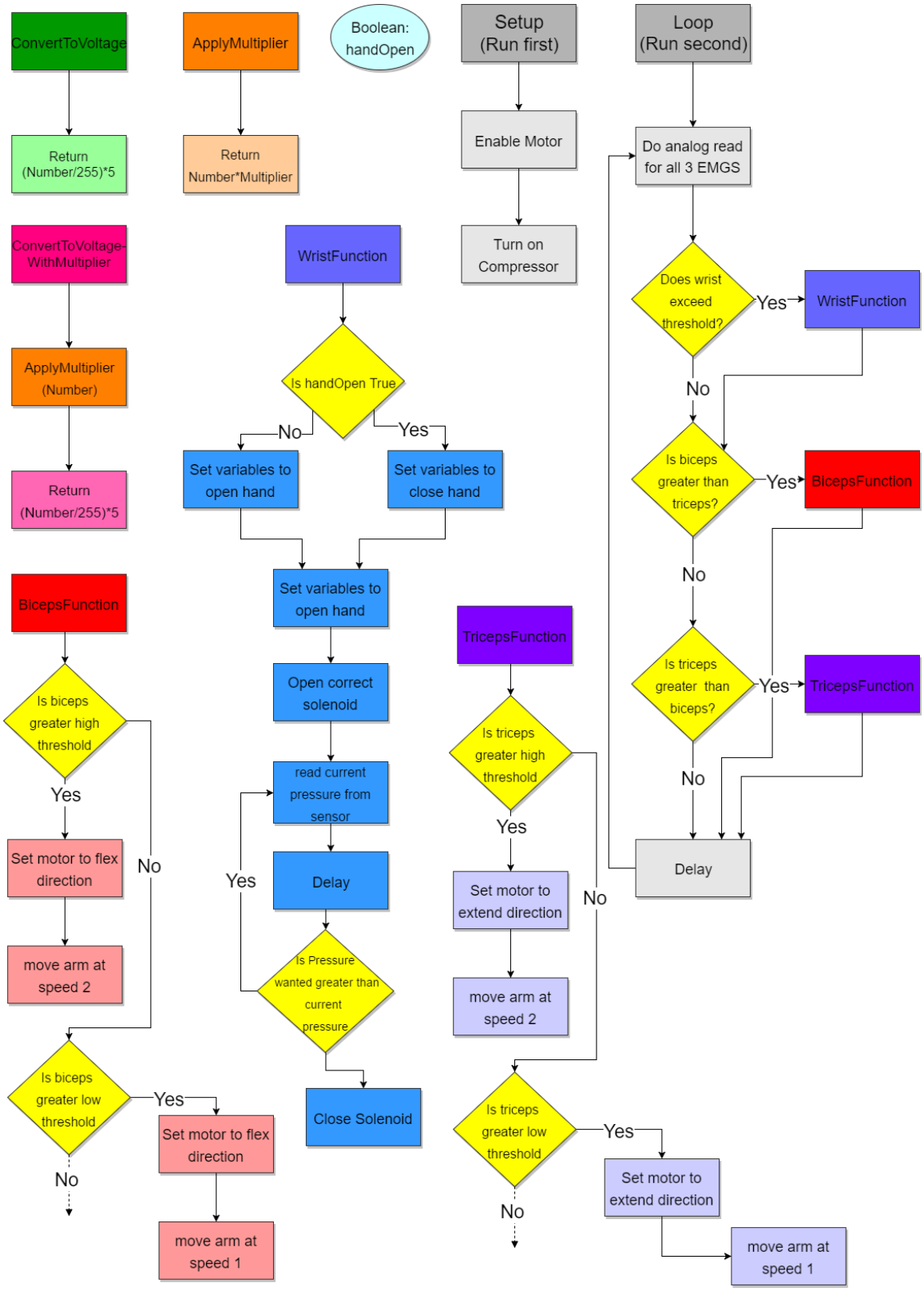
The way we mapped the pins is to put everything in the lowest priority pin it can function with. The priority goes (excluding power and communication pins) analog, PWM, and lastly digital. The EMGs needs to be read as analog signals so we must use analog pins, however, the PUL can be sent using square waves, which is doable with a PWM pin. Because DIR and ENA only need a high or low signal, we give them strictly digital pins, saving out PWM space for other things. The wiring diagram for elbow movement of the arm can be seen in figure *Software Low Level Design 2*.

6.2.3.2.2: Mapping for Hand Control

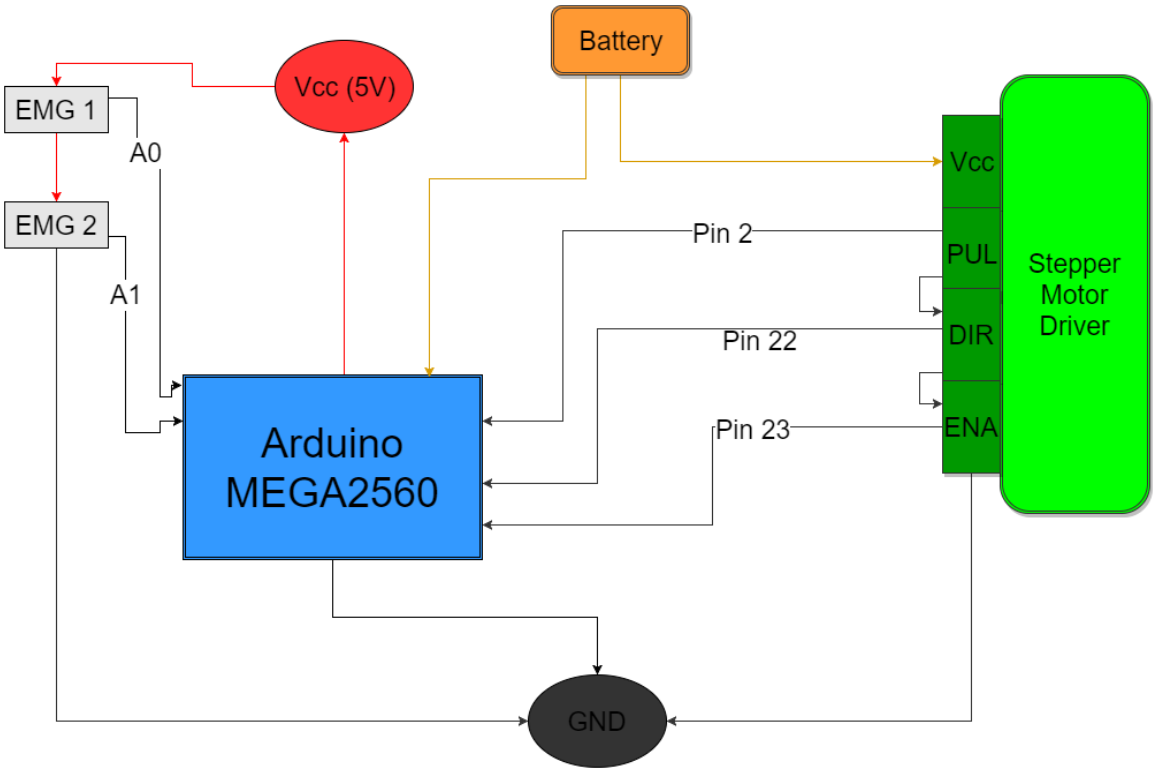
The hand control subsystem is mapped as seen in diagram *Software Low Level Design 3*. There are two pressure sensors that will be connected to the analog pins of the Arduino and output a signal between 0.5 V and 4.5 V. The solenoids and compressor will be connected to power relays that each need a digital input. For this reason, they will be connected to the digital pins of the Arduino.

6.2.3.2.3: Mapping for System Reset

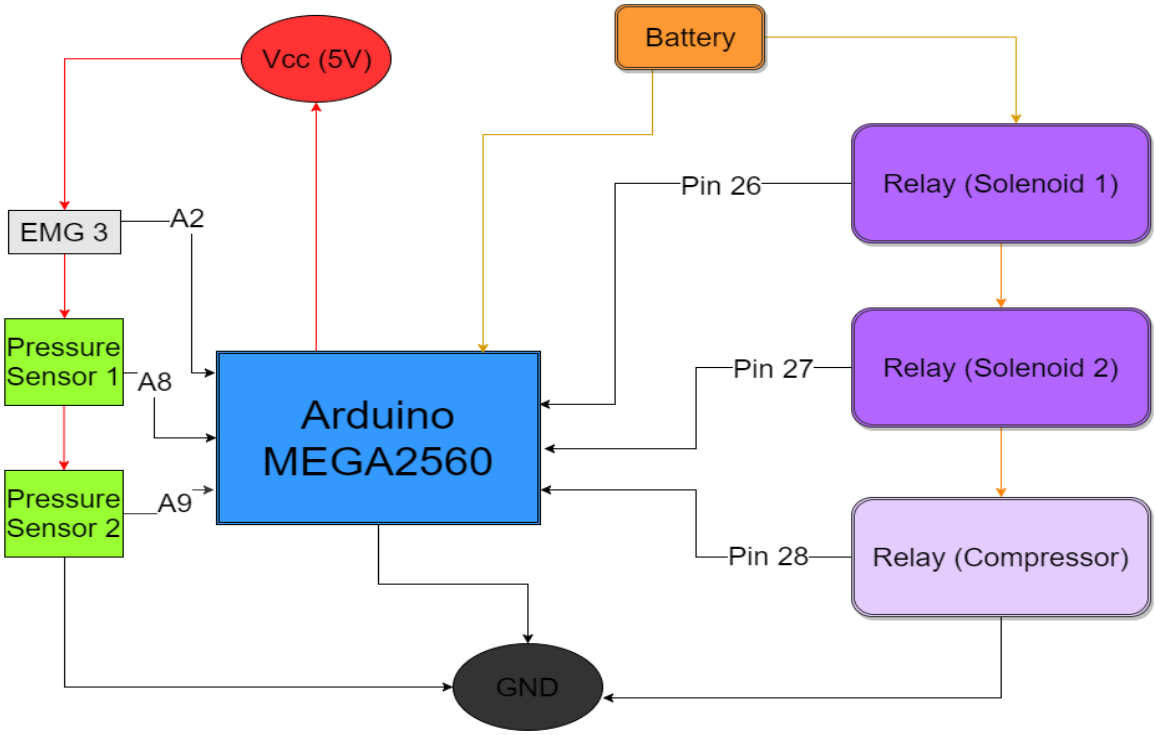
The pin mapping for the system reset is designed as simple as possible. The reset button will be wired directly to the reset pin on the Arduino. When the pin is hit, this will trigger a hard reset of the system. A wiring diagram describing this can be seen in figure *Software Low Level Design 4*.



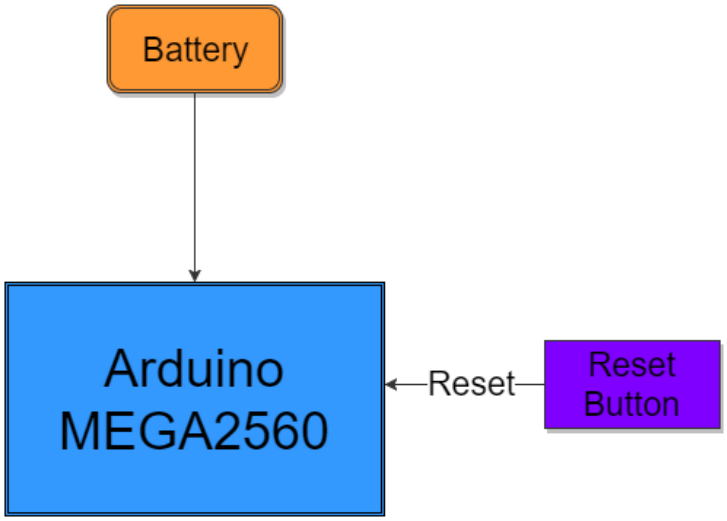
Software Low Level Design 1. This is a flow diagram of how the system will be coded.



Software Low Level Design 2. This is a wiring diagram of how the components needed for elbow movement with be controlled to the microcontroller.



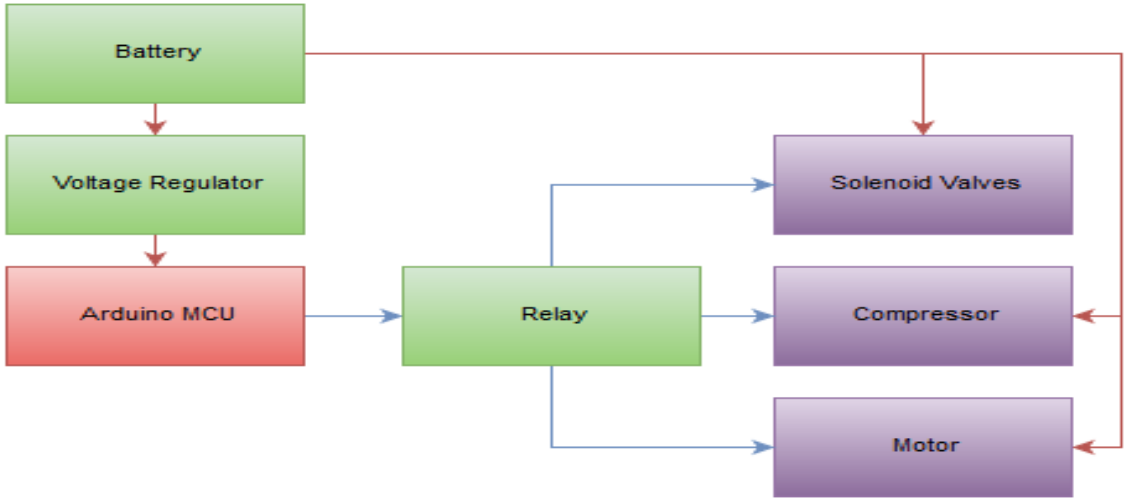
Software Low Level Design 3. This is the wiring diagram for the hand control subsystem with the microcontroller.



Software Low Level Design 4. This is the wiring diagram for the system reset button.

6.3: Hardware Design

Exoskeletons are promising assistive/rehabilitative devices that can help people with force deficits or allows the recovery of patients who have suffered from pathologies such as stroke. When thinking on how to start designing a circuit to operate the motor, this is the first design that came to mind. In *Hardware Design 1* shown below, the hardware team had planned to energize the relay that will be powering the motor with an output signal from the Arduino. This would work, but when the relay collapses some back EMF could find its way back and eventually burn one of the output pins of the MCU. A diode could be added at the coil as it will help reduce any positive voltage from returning back to the system. However, when the relay collapses the voltage could change polarities for an instance and at that point the team will encounter the same issue again and probably with a burned output from the microcontroller.



Hardware Design 1: Relay implementation for easy control of actuators

To encounter the possibility of getting some back EMG, the team had to find a way to get the same result without jeopardizing the micro controller or its outputs. After doing research, it was found that an opto-coupler with amazing characteristics would get the same results, while isolating the micro controller from any possibility of getting some back EMF.

6.4: Electro-Mechanical Design

6.4.1: Electro-Mechanical System Overview

The electro-mechanical system for the exoskeletal arm will be comprised of two subsystems; one for the actuation of the hand and another for the actuation of the elbow. These systems will be controlled by the MCU which will receive signals from the EMG sensors to then be interpreted into movement. For the hand actuator the pneumatic system will only perform two actions either being open or closed. The pneumatic hand system will be using a series of compressed air tanks, valves, and pressure regulators to control the soft robotics finger actuators which will comprise the hand. The actuation of the elbow will be more complex in the sense that the position will be variable based on the strength of the EMG signals received. The elbow will rotate through the same range of motion as a typical elbow would. However, there will be mechanical stops for the elbow such that overextension will not be a possibility. A stepper motor will be used to articulate the movement of the elbow and an encoder will be used to verify the position of the elbow. Below is the *Electro-Mechanical System Overview 1* figure of the overall design of the exoskeletal arm as a system.

This section of the exoskeletal arm will be where the electrical design and the mechanical design join together. Creating a device which meshes seamlessly together is one of the most noticeable, or unnoticeable if done correctly, attributes a product can have. Without all of the components for the device working properly together then the entire device will be a failure. In a way a product is much like an orchestra, each section has its own part to play and when combined a symphony comes alive. However, if that orchestra was missing the woodwinds section then it would be incomplete and your experience would be less than the full symphony. This is why a product must have all of its components working together in unison to product the full symphony. Many time we find devices which have excellent hardware but terribly design software and vice versa. Those are devices which typically fall behind in the markets. The products that tend to rise to the top are usually the same ones that have a coherent flow between the different sections as if each were designed with the other in mind. This is why the electro-mechanical subsystem of the exoskeletal arm is crucial to the success of the entire project. While is it easy to come up with ways to solve problems, the hard part is coming up with good ways to solve problems which mitigate risk and improve performance.

The design intent of this system is to make it simple while retaining function. This particular subsystem should be designed it the least bulky manner such that it does not cause user to avoid utilizing this tool simply because it is more of a hassle than a help. Even if the design is minimal but remains something which will get lots of use then that is an ideal design. The goal is to make something which will be used not something that

is extensively overbuilt and complex. The design intent is to make this system function in a failsafe manner, much like regulators used in SCUBA diving. The valves on these regulators are designed in a manner such that if the first stage regulator should fail, thus over pressurizing the system, then the second stage regulator, which is the part located in the diver's mouth, will free flow allowing the diver to still breath air from the system. SCUBA is normally viewed as a recreational activity, however the gear is life support equipment and is designed with multiple modes of failure in mind. For the exoskeletal arm similar modes of failure must be identified and mitigated to make this a viable product for the patient to utilize on a daily basis.

Electro-Mechanical System Overview 1: Exoskeletal Arm System Block Diagram

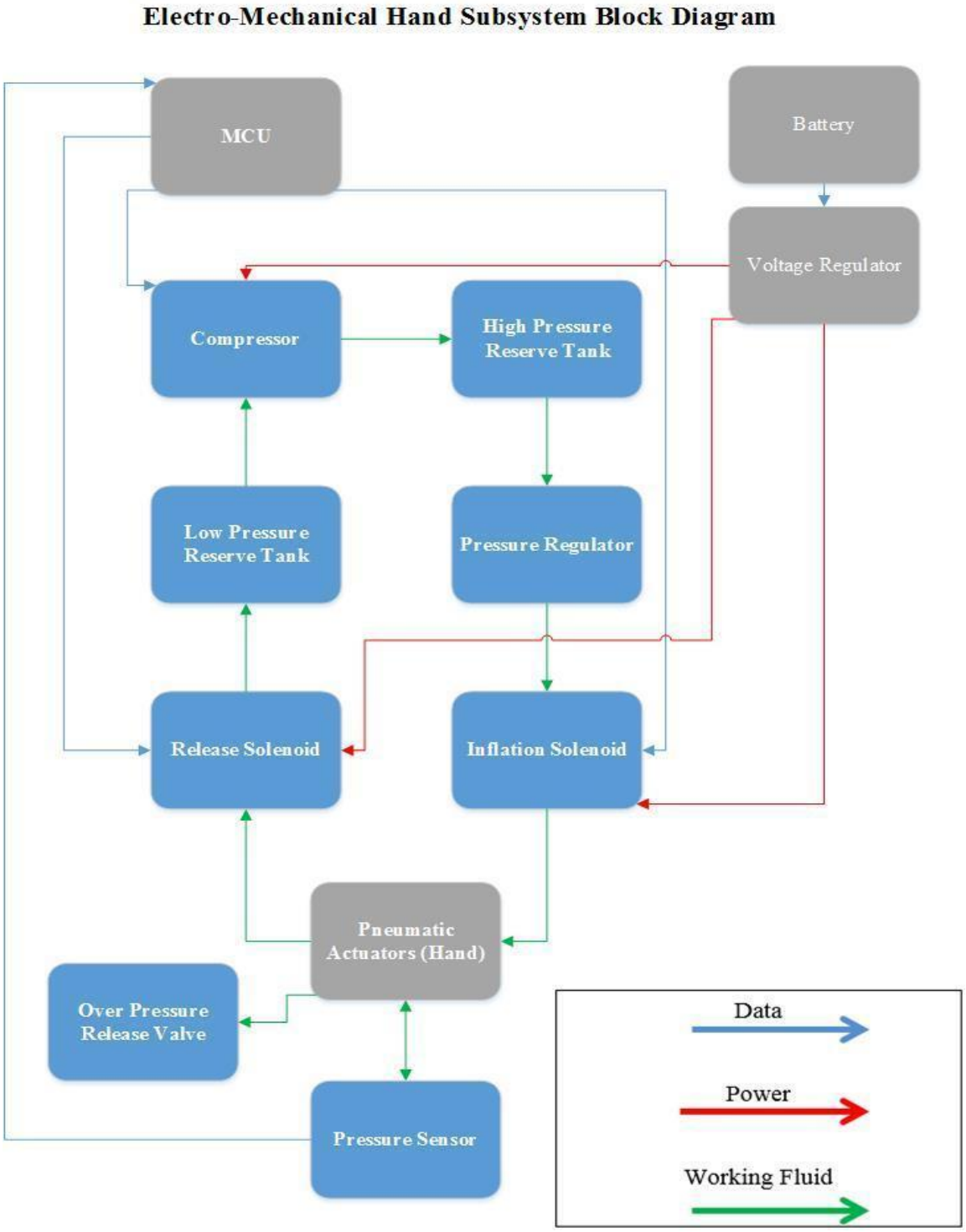
6.4.2: Hand Actuation Subsystem

The actuation of the hand will be accomplished through the use of a pneumatic system. This pneumatic system will be comprised of a compressor, two tanks, two solenoid valves, a pressure sensor, a pressure regulator, and three over pressure relief valves. The use of the two tank system will allow for the hand actuator to react quickly and not be limited by the power of the compressor to constantly be either inflating or deflating the hand actuator. The tanks for the hand actuator will be located on either side of the compressor such that one of the tanks will be for high pressure and the other tank will be for low pressure. The compressor will compress the air from the low pressure tank into the high pressure tank. By compressing the high pressure tank from the low pressure tank as opposed to directly from the atmosphere this will allow the system to remain a closed loop system. Additionally, keeping this system closed loop will allow more leeway in selection of the working fluid. The working fluid will likely be simple air direct from the surroundings and the system will be charged by an easily accessible device such as a bicycle pump. However, there will still remain the option to use other fluids such as filtered dry compressed air or nitrogen. Nitrogen would be an optimal choice because it would not contain the water vapor or oxygen found in simple air, this would allow the system to be less susceptible to changes in ambient temperature. Another advantage to nitrogen is that it does not permeate through the walls of the system to the extent which the oxygen in air does. The drawback to nitrogen is that it is expensive, not readily available for use, and the charging system hardware will need to be regulate the very high pressure (around the order of 3000 psi or more) of a nitrogen tank. Filtered compressed air would be more advantageous than simple atmospheric air in that it does not have the water vapor, which is the major factor in the pressure variation of simple air due to temperature. On the other side compressed air is costlier than simple air (less expensive than nitrogen), not as readily available for use as simple air, and the charging system hardware will need to be regulate the very high pressure (around the order of 3000 psi or more) of a compressed air tank.

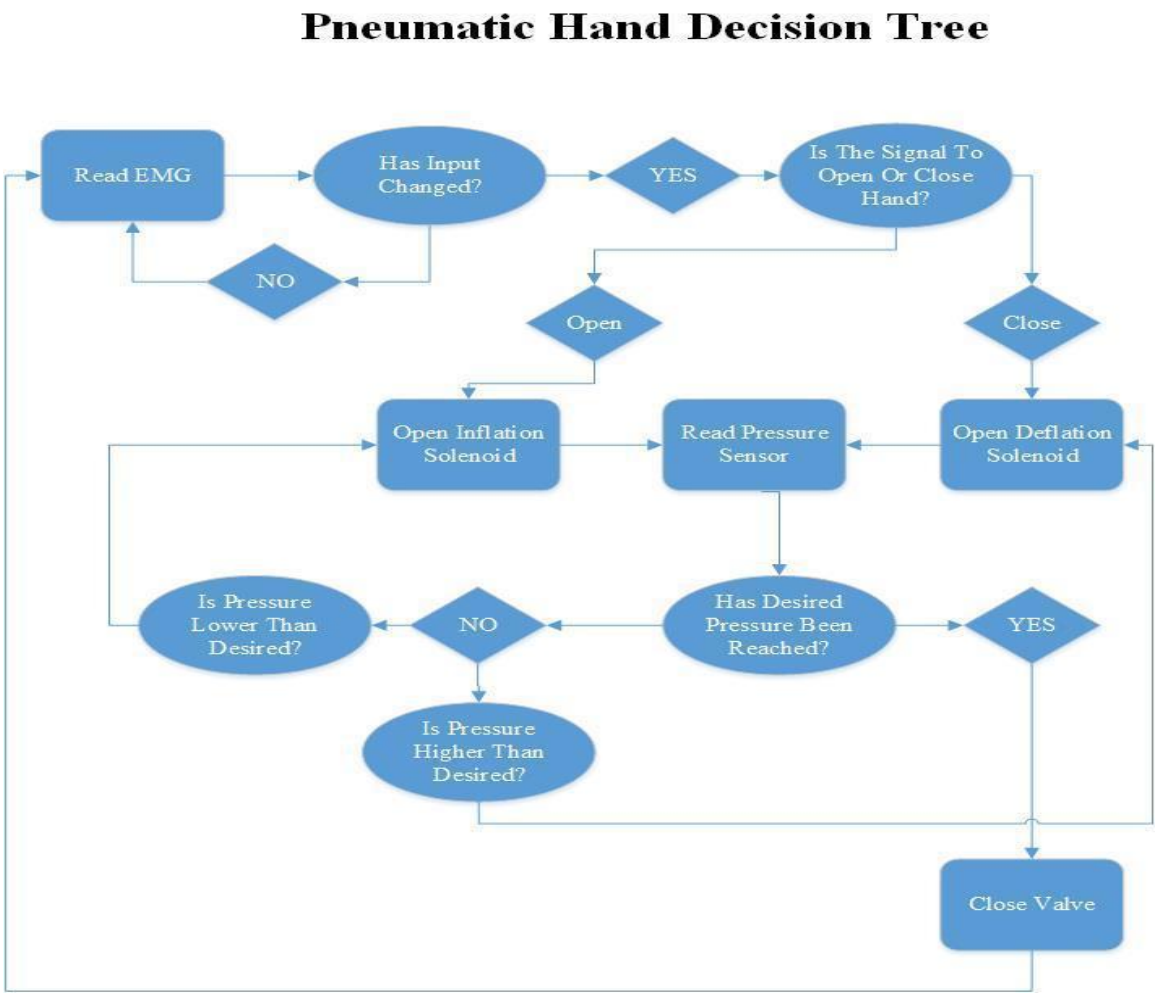
There will be two solenoids on each of the two tanks. The solenoid on the high pressure tank will be opened to inflate the pneumatic hand and to deflate the hand the solenoid on the low pressure tank will be opened to deflate the pneumatic hand. The pressure sensor will be placed such that it can read the pressure in the hand actuator. The sensitivity of the pressure sensor will need to be enough to detect a pressure to 0.1 psi or better. The pressure sensor will then be able to read the pressure and relay the information the MCU. Then the MCU can decide which valves need to be opened or closed. The solenoid valves themselves will be the type which remain closed until energized.

The pressure regulator will be used to ensure that that the high pressure tank can only output the working fluid to a pressure near that which is desired for the pneumatic actuators. There will be an overpressure relief valve located on both of the tanks as well as on the pneumatic actuators themselves. The overpressure valve will maintain the safety aspect of the design; this will ensure that no portion of the pneumatic system will reach a pressure that could become dangerous. Below are figures *Hand Actuation Subsystem 1* and *Hand Actuation Subsystem 2*, which define the actuation subsystem for

the pneumatic hand in terms of hardware and the more software-oriented decision-making process that will provide actuation for the patient.



Hand Actuation Subsystem 1: Electro-Mechanical Hand Subsystem Block Diagram

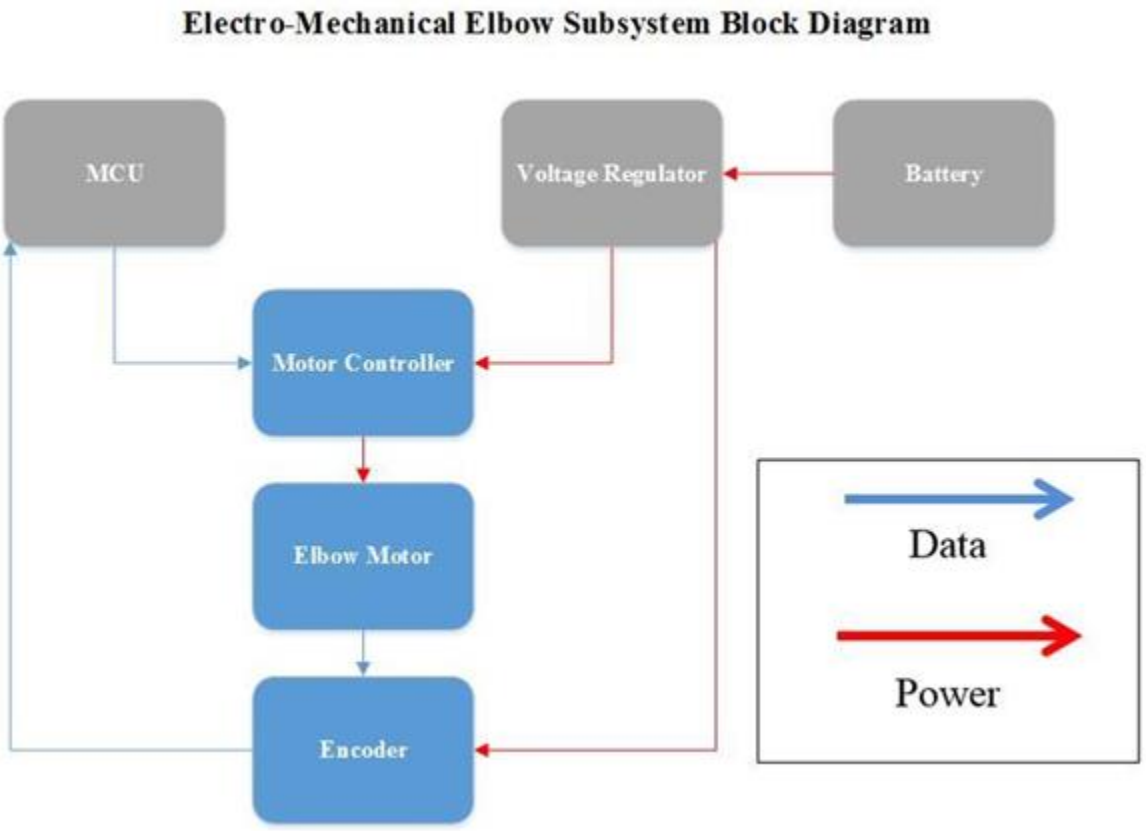


Hand Actuation Subsystem 2: Pneumatic Hand Decision Tree

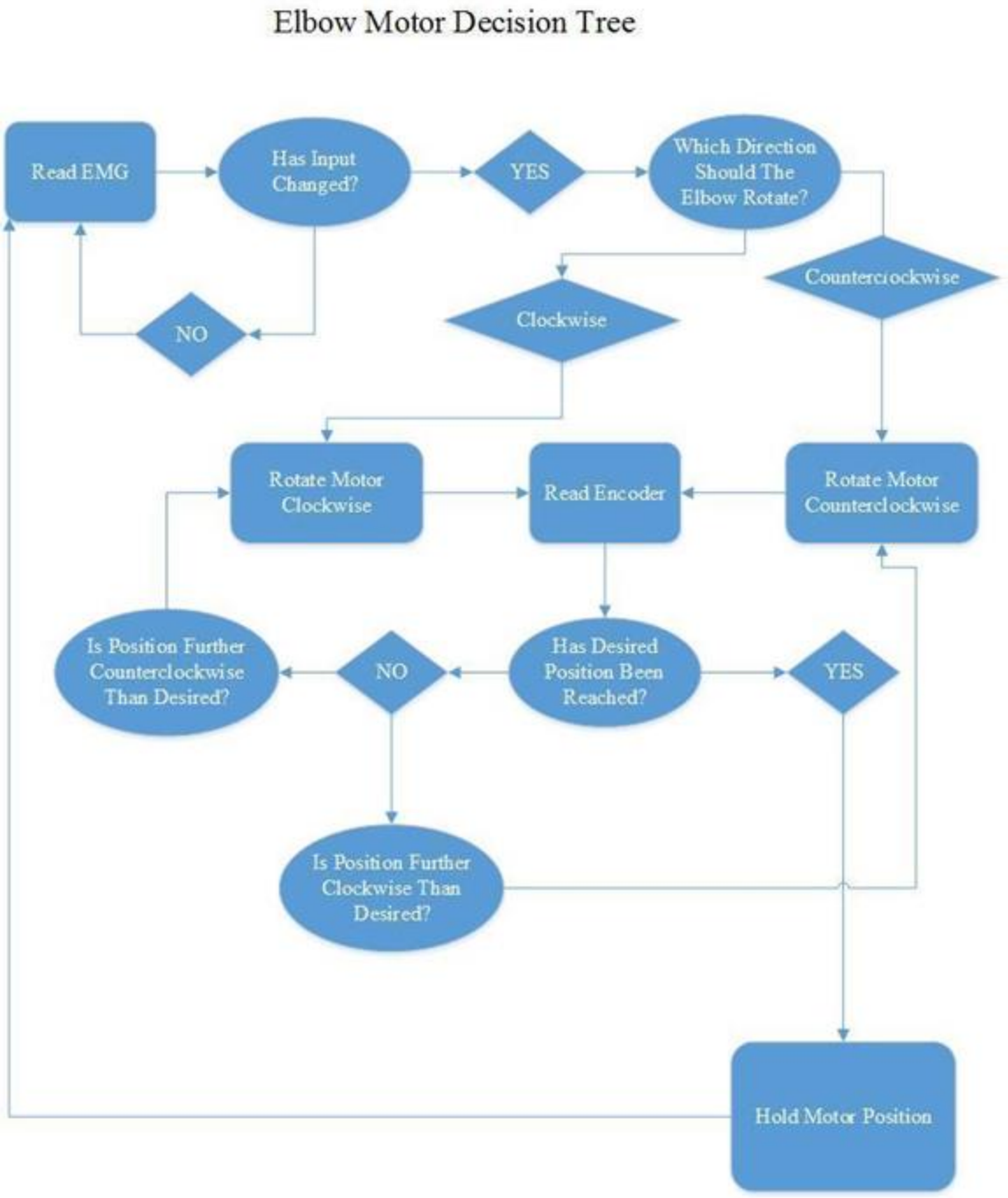
6.4.3: Elbow Actuation Subsystem

The actuation of the elbow will be accomplished utilizing a stepper motor and encoder system. This system will be comprised of three main components; the stepper motor, the motor controller, and the encoder. The system will read the inputs from the EMG sensors placed on the subject's arm, then the MCU will interpret the signals to control the motors position. A stepper motor will be used due to its ability for fine position control. Typically, the position of a stepper motor can be controlled within one degree of angle or less on a repeatable basis. The encoder will allow for feedback of the actual position of

the motor allowing for the MCU to know the current position and adjust accordingly if necessary. Below are figures *Elbow Actuation Subsystem 1* and *Elbow Actuation Subsystem 2*, which define the actuation subsystem for the electrical elbow in terms of hardware and the more software-oriented decision-making process that will provide actuation for the patient.



Elbow Actuation Subsystem 1: Electro-Mechanical Hand Subsystem Block Diagram



Elbow Actuation Subsystem 2: Elbow Motor Decision Tree

6.4.4: Hand Actuation Component Selection

The components for the hand actuation will have to meet certain requirements to be able to work for this design. The tanks will need to have sufficient volume to allow for sufficient fluid movement without the need for the compressor to drive the actuation. The tanks should be rated to hold a pressure of at least 50 psi. The tanks should be lightweight and compact to maintain user comfort. The solenoid valves should be the type which

remain closed unless energized. The solenoid valves should be rated for a maximum pressure of at least 50 psi. These solenoid valves should be relatively compact such that they do not add excessive weight or become a hindrance to the user. The pressure sensor should have a repeatable accuracy of 0.1 psi or better, while still remaining compact. The compressor should not be excessively large or noisy while still being able to run the system without delay. If the compressor does become a limiting factor, then the volume of the tanks may need to be adjusted to compensate. The over pressure valves will need to be compact and reliable. The over pressure valves will likely fall into one of two types; burst disk or spring regulator type. The burst disk is ideal for use when accidental over pressurization presents a serious hazard. The drawback to the burst disk design is that once it bursts the component depressurizes and the disk then will need to be replaced before the system can be re pressurized. The spring regulator type is ideal for systems that tend to need to blow off excess pressure regularly, they do not completely depressurize the system and will allow the system to work at some level. The disadvantage to the spring regulator relief valves is that they are more cumbersome and complex leading to a possible point of failure.

6.4.5: Elbow Actuation Component Selection

The main requirements for the elbow components is that they must allow for the elbow to have sufficient strength and for the position to be accurately controlled. The motor will have to have sufficient holding torque to hold up the arm, exoskeleton, and the amount of additional weight desired. The specific requirements will be determined during the testing which will be conducted with the end user. The motor will be a stepper type due to their ability for fine control. DC motors were another consideration however due to their inability to for fine precise control without additional hardware. The encoder will need to be able to accurately determine the angle of the elbow within one degree or better and be able to either interface directly with the selected motor or the joint on the exoskeletal arm. The motor controller for the stepper motor will need to be able interface with the Arduino as well as be able to handle the current draw from the motor. Stepper motors require a motor controller because unlike a typical motor they contain multiple winding coils which need to be energized in proper sequence to allow for coherent movement. Some of the advantages to this type of motor is that they run more efficiently than traditional DC motors and have the ability to hold position when energized. The stepper motor alone will not likely have sufficient torque by itself so it will need to be interfaced with a gearbox system that will then have to properly interface directly with the elbow joint designed by the mechanical team.

7: IMPLEMENTATION

7.1: Input Implementation

In order to understand what mapping is needed to be programmed into the Arduino board, tests must be run on the patient to obtain some valuable data in regards to the signals that are produced by certain muscle contractions. With the help from Lietsel Richardson, who is the lead for the mechanical team in charge of the pneumatic system, the electrical team has developed a testing framework, which will be used for the patient in order to acquire raw data.

7.1.1: Test Bed Purpose

The purpose of section 7.1.1 is to define a set of testing protocols, as well as requirements, for EMG placement for the purpose of designing a controller to act as an assistive device for subjects with muscular dystrophy. Prior to bringing in subjects for testing purposes, a protocol must be set in order to adequately prepare for testing sessions and clearly outline the knowledge that will be gained. With this understanding in mind, different tests and trials may be conducted for the purpose of obtaining specific information relevant to control design.

Since the controller is meant to assist with elbow flexion, extension, pronation, and supination, as well as finger grasping, very specific and targetable muscle groups must be tested. The major muscles involved in these movements are the biceps brachii, brachioradialis, posterior and anterior deltoid, triceps brachii, and wrist flexor and extensor. In the case that a few of these muscles are not feasible, even with amplification, then the upper trapezius may also be tested. In total, eight muscles will be analyzed as they exhibit their functionality in real time during testing.

7.1.2: Input Test Requirements

The controller in question must be calibrated prior to use in order to adjust the feedback system to function in synchronicity with the user. Thus, testing requirements must be explicitly defined in order to meet the demands of the feedback design. With these designs in mind, a concise protocol may be developed in order to obtain results that will be conducive to project development.

1. Information about the subject's muscle contractility shall be obtained by testing for maximum voluntary isometric contraction. EMG signals may differ with location and vary amongst individuals, therefore these exercises over a period of 60 seconds will allow for engineers to analyze the signals obtained in the post-processing phase of testing. The results of this test will be normalized in order to analyze the amplitude of generated impulses during a trial. Based on the analysis, the data will be sufficient to determine a.) if the EMG is strong enough in the respective muscle group, b) the contractile speed of the designated muscle, and c.) the rate of fatigability based on the subject's physiological condition.

2. The tests shall collect data from all potential muscle groups associated with joint movements that the controller will compensate for. In addition, neighboring muscles, such as deltoids, will be targeted in the event that motion-specific muscles do not generate a strong enough EMG for the purposes of the control system.

3. Subject's current ability must be qualitatively analyzed via task-oriented tests. These tests shall mimic everyday tasks in order to comprehend just how much the controller shall compensate for. This test should mirror a procedure that may be repeated for later device calibration.

4. The subject's maximum loading capacity shall be quantified via a weight increment test. The results will easily determine the subject's muscle strength and joint stiffness.

5. Testing shall include a visual display to encourage subject to achieve various levels of voluntary strength thresholds.

6. Optimal inter-electrode distance should be tested prior to beginning protocol. Optimal positioning typically is foolproof on the belly of the muscle; however, for the purposes of designing a controller, the actual inter-electrode distance may be bigger or smaller depending on where the signal is the strongest on the muscle.

7. Electrode position must be marked once the inter-electrode distance is selected. This can be done using a permanent marker with the consent of the subject undergoing testing.

8. Skin must be prepped with alcohol prior to placement of electrodes.

9. The microcontroller that will be in use for testing shall have a sampling frequency that is at least double or triple that of the input. Thus, the selected device shall have a high sampling frequency of at least 1000 Hz.

10. The microcontroller shall have enough pins for the EMG sensors in use.

11. The test sleeve shall be wireless and snug to fit to prevent movement artifact. Movement artifact occurs when the wires, electrodes, and/or other electrical components are shifted whilst the subject undergoes testing. This poses as a problem by adding a significant amount of noise to the signal, which in turn makes it difficult to filter during post-processing. This holds true due to the fact that it is not differentiable between noise generated from muscle crosstalk and other forms of interference.

12. The test device (sleeve, sensors, and other components) shall transmit data in real-time during each trial.

13. The system shall have a power source that will be sufficient for test durations of 2-3 hours.

14. Data obtained must be saved automatically for post-processing and analysis.

15. The test shall not impose on subject's safety and comfort.
16. The full test system must be tested prior to following protocol on intended subject to ensure that systems run without error in order to avoid delays during protocol. This is also to verify that the system is safe for a human user.
17. The system shall not generate heat or cause irritation to subject's skin.

7.1.3: Input Test Equipment

A designated lab space would be the ideal environment to perform the tests outlined in this protocol. Testing area should be clean, spacious, and enclosed for the privacy of the subjects involved and to provide a focused environment. This space shall have internet access, various electrical outlets, a desktop computer, chairs, tables, and good lighting.

The following list of equipment comprises of hardware and software required for testing purposes:

Software:

- MatLab
- LabVIEW
- Arduino Code

Hardware:

- DAQ
- Oscilloscope
- EMG sensors (8)
- Electrodes
- Jumper wires
- Sleeve
- Arduino MCU
- Arduino Wifi Shield 101
- Batteries
- Computer
- Cup
- Weights
- Tape measure

- Alcohol/cotton pads
- Permanent Marker
- Timer

These items should not be counted against the set budget for the team, since they will be mostly borrowed and provided by different parties for testing purposes only. It is expected that only the Arduino MCU and EMG sensors will be recycled for the design.

7.1.4: Procedures

Once the system has been set up and is up and running, the subject may be prepared for procedural testing. Prior to beginning, however, the physical measurements of the subject's upper limbs should be taken and recorded for future reference. Following these logistical steps, the subject's skin shall be cleaned using alcohol and cotton pads. Location of electrodes should be marked and photographed to save time in the preparation stage of each test.

In general, muscles experience fatigue after a series of exercises, but this may be experienced much sooner with subjects whose muscles are atrophied or weak. With this in mind, it would be best to spread these tests over a period of two or three days with adequate rest between each trial to allow time for recovery.

7.1.4.1: Maximum Voluntary Contraction Test (MVCT)

Three trials per motion should be repeated for both the left and right arms. *Procedures 1* explains how this test will be undertaken.

Motion	Test Description
Flexion, no resistance, 90º	Subject to flex arm from neutral to 90º and hold for 3s, then release. Repeat for a duration of 60s.
Extension, no resistance, 90º	Subject to extend arm from flexed position to extended position and hold for 3s, then back to flex. Repeat for a duration of 60s.
Flexion, resistance, 90º	Subject to flex arm with maximum strength while researcher applies resistance to top of forearm. Hold for 3s and repeat for a duration of 60s.
Extension, resistance, 90º	Subject to extend arm from flexed position to extended position while researcher applies resistance to bottom of forearm. Hold for 3s and repeat for a duration of 60s.
Pronation, no resistance	Subject to rotate elbow inward and hold for 3s, then release. Repeat for a duration of 60s.
Supination, no resistance	Subject to rotate elbow outward and hold for 3s, then release. Repeat for a duration of 60s.
Pronation, resistance	Subject to extend arm out and face palm inward while researcher places palm (facing outward) in contact with subject's palm. Subject rotates elbow inward and hold for 3s, then release. Repeat for a duration of 60s.
Supination, resistance	Subject to extend arm out and face palm inward while researcher places palm in contact with back of subject's hand. Subject rotates elbow inward and hold for 3s, then release. Repeat for a duration of 60s.
Grasping	Subject to attempt grasping motion with maximum available strength

Procedures 1: Testing patient must go through for data collection for maximum voluntary contraction

7.1.4.2: Weight Increment Test

This test should be conducted at least 23 hours after the MVCT. A break of at least 15 minutes should be taken between the arm extended and arm down trials. *Procedures 2* explains how this test will be undertaken.

Configuration	Test Description
Arm extended, 2lbs	Subject to extend arm to chest level. Researcher to add 2lb weight to hand (alternatively, wrist weight can be used). Hold for 20s and release.
Arm extended, 4lbs	Subject to extend arm to chest level. Researcher to add 4lb weight to hand (alternatively, wrist weight can be used). Hold for 20s and release.
Arm extended, 6lbs	Subject to extend arm to chest level. Researcher to add 6lb weight to hand (alternatively, wrist weight can be used). Hold for 20s and release.
Arm extended, 8lbs	Subject to extend arm to chest level. Researcher to add 8lb weight to hand (alternatively, wrist weight can be used). Hold for 20s and release.
Arm down, 2lbs	Subject to relax arm along side of body in relaxed position. Researcher to add 2lb weight. Hold for 20s and release.
Arm down, 4lbs	Subject to relax arm along side of body in relaxed position. Researcher to add 4lb weight. Hold for 20s and release.
Arm down, 6lbs	Subject to relax arm along side of body in relaxed position. Researcher to add 6lb weight. Hold for 20s and release.
Arm down, 8 lbs	Subject to relax arm along side of body in relaxed position. Researcher to add 8lb weight. Hold for 20s and release.

Procedures 2: Testing patient must go through for data collection for weight increment

7.1.4.3: Task-Oriented Test

Since this test provides qualitative information, it is not necessary to conduct post-processing procedures on data collected. Researchers should observe what happens to the signals in each muscle involved in motions required to complete everyday tasks that the controller will compensate for. This will give a general idea of the muscles put into action to complete a task which is reflected upon patient-specific muscle health. Results may vary. Everyday tasks shall include picking up an item, bringing an item to the face, opening a door, closing a door, holding items, removing an article of clothing, etc. This test should also be timed in order to determine how long it currently takes subject to

complete certain tasks. These recorded times should be compared to a healthy subject's times (healthy subject will be tested prior to or following tests outlined in this paper).

7.2: Input Implementation Software

7.2.1: Data Collection

Once patient undergoes testing, data collected shall be sent to a computer that will be running a Simulink program that specializes on data collection from EMG signals. In order to prevent movement artifact caused by the moving of wires as the patient flexes different muscles, information will have to be sent wirelessly. The team is looking for different ways of doing so, but so far the biggest promise is held by the use of the Arduino board's capabilities, which include the use of a hardware component that allows for wireless communication, along with libraries that can be found online for this purpose. The use of PWM output will likely be used, as this is the only way for the MCU to send analog information out. As a consequence, the computer will need to recognize the PWM pattern and properly convert the signal to data that can be easily read by the Simulink program.

7.2.2: Input Software Algorithm

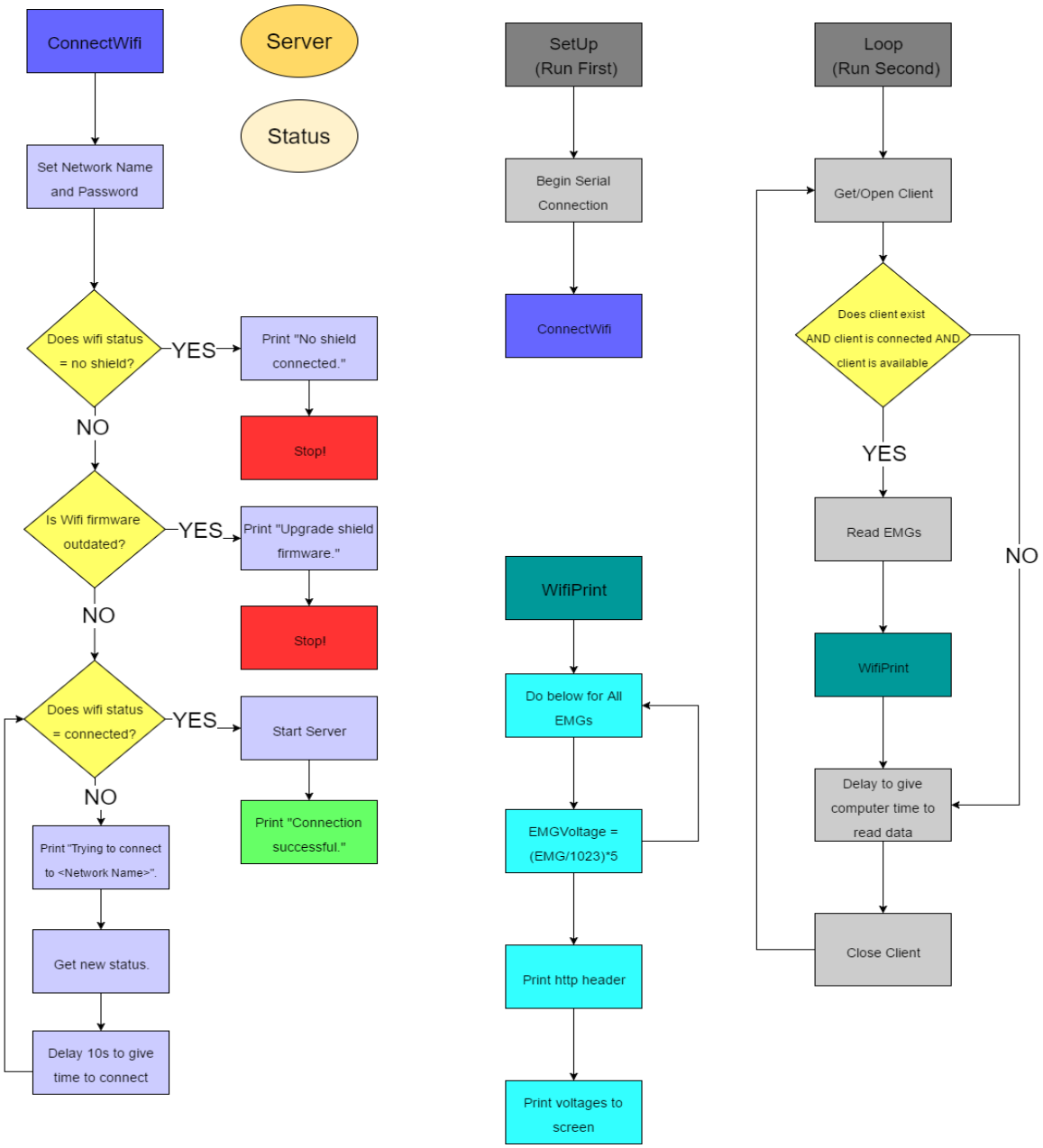
To determine how the EMG signals are going to look/behave and what the thresholds for input from the patient should be, we have to do input testing. The idea from a software point of view is that we read what typical signals from the patient should look like. We can then configure the software specific to the user.

Say for instance if the patient gives off a voltage from the EMG placed at the biceps when he does some activity. This voltage can be 4 V in this scenario. Let us also say that occasionally, the EMG reads 1 V from very minimal movement. We know that if the EMG only gives off 1 V, then this is probably unintentional. We also know that the standard movement gives off about 4 V (we can go a little lower to factor in fatigue over time). With this information, we are able to code a solution that ignores values of 1 V and uses values of 4 V to make the arm do something.

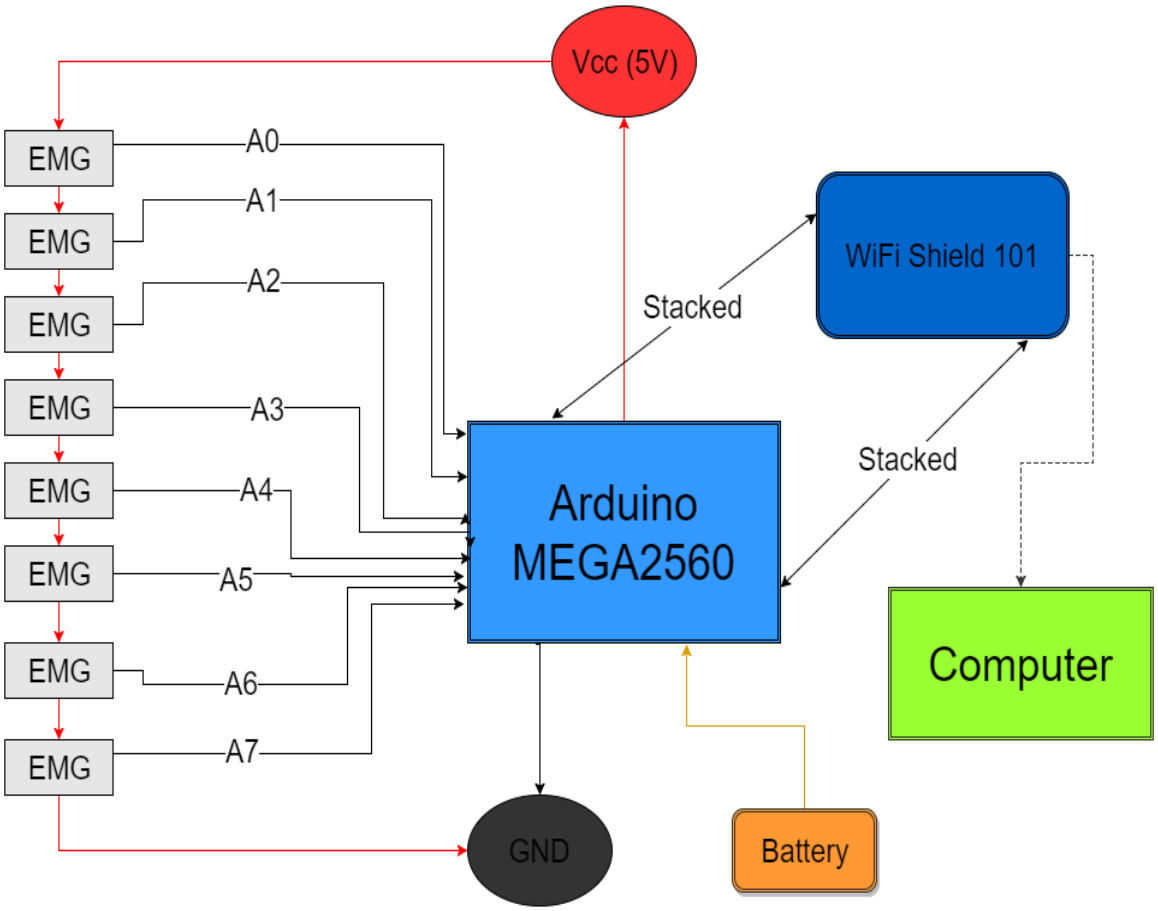
The first thing to do was design and develop pseudo code for the test system. This took some work to factor in all of the components we would need (software-wise). It requires an Arduino MEGA2560, a WIFI Shield 101, 8 EMGs, a computer, and a board to build a circuit with. The design is as follows.

First we set up a connection to the serial monitor so we can print out statuses for the WiFi. Next we attempt to actually connect to the WiFi. To do this, we set the network name and the password, then we check the status of the WiFi. We have a few conditions that if evaluated true will print out an error message and stop the application. These conditions are if the WiFi shield is not connected or if the firmware is outdated. After that, we check if the WiFi is connected. If it is not, we print to the user that we are trying to connect, get the new status and wait 10 seconds before we try again. If it is connected, we print that it is connected and move on to the next step.

The next step is to first, open the client (the computer). After that, we check if the client exists, is available, and is connected. If these conditions are false, we wait, close the client, and try again. If they are true, we read the EMG values. The EMGs come in a value between 0 and 1023 in proportion to the voltage. So we have to divide the values by 1023 and multiple by 5 for each one. We then print the http header as well as the actual values to the client. Lastly, we wait to make sure the client received everything and then we close the client and loop through again. A flow diagram of what this logic looks like can be seen in figure *Input Software Algorithm 1*. A diagram of what the connections will look like is in *Input Software Algorithm 2*.



Input Software Algorithm 1. This is a flow diagram of how the logic works for the input testing to be done on the patient.



Input Software Algorithm 2. This is a diagram of how the connections will be set up for the user input testing.

7.3: Electro-Mechanical System Integration

7.3.1: Component Calibration

Calibrating the pressure sensors would be done by first fully examining the data sheet to understand the relation between pressure change and the change in signal. Then since the testing site will be located at approximately sea level the atmospheric pressure would be used as a reference to get a value for zero psig (atmospheric pressure at sea level is about 14.7 psi; converting it to gauge pressure would make it more relevant to the more typical convention). Next the pressure sensor would be hooked up to a pressure vessel which would then be pressurized to 50 psi, which would be determined using an accurate pressure gauge. Then using these two points a schema can be developed to interpolate for other pressure readings.

To calibrate the pressure regulator, the calibrated pressure sensor could be used to check that the output of the pressure regulator came to the desired level. The Pressure regulator input would be connected to the high pressure tank and then the outlet would be connected to the mockup of the pneumatic hand actuator. The pressure sensors would be located in both the high pressure tank and the pneumatic hand actuator. Then when the

system is pressurized the pressure sensors could verify that the pressure regulator cuts off the flow once the desired level is reached.

For the calibration of the encoder a protractor would be used to insure that the perceived rotation correctly correlated the measured rotation by the encoder. The motor would be rigidly fastened to a surface and a needle would be attached to the shaft of the motor. A full circle protractor would be attached to the surface to ensure that movement would not become an issue. The zero degree point would be a reference then rotation to the 90 degree point would be tested for accuracy.

7.3.2: Modes of Failure

Although it is the desire that every design should work fully as intended, more often than not problems do arise. The best way to solve a problem is to anticipate it. Unfortunately, not all problems can be anticipated which is what prototypes are for. A test bed will be constructed prior to integrating the electrical components with the actual arm. This will allow the electrical team to test and troubleshoot the system in a more efficient manner.

The main concern for both the hand and the elbow actuator is that they can be controlled correctly to move in a coherent manner. There is a very likely chance that the first round of controls will fail and require modification. Other possible issue may arise with individual components not working properly or below the desired level. Another possible concern would be the heat generation by the system in which case an additional cooling system would need to be designed.

For the pneumatic hand the potential problem which tops the list would definitely be the possibility of leaks. In such a case leaks tend to occur in and around fittings so the design of those may need improvement or it could be as simple as adding some Teflon tape to the threads of the fittings. The easiest way to find leaks is to listen for them. However, if that does not lead to the root of the cause then soapy water can be sprayed onto the system to find where it bubbles (assuming the soapy water would not have a substantial risk to create a short circuit around the application site). Another way to test for leaks is to look at the pressure sensors to narrow down the location of any leaks. The likely cure for any leaks found will be to disassemble, clean, and replace any o-rings or seals which may be compromised then reassemble and recheck the system.

Another possible problem for the pneumatic hand system is overpressurization. While there will be burst disks to ensure that any overpressurization does not reach a critical level that will not solve the root of the problem. The pressure will need to be monitored by the sections to determine where exactly the overpressurization occurs. Likely causes for this would be the pressure regulator improperly working or the compressor not shutting off. Another concern is that being a closed loop system if there is a leak and there is enough pressure loss such that the compressor cannot fill the high pressure tank to its shutoff pressure then what will occur? A schema will need to be devised such that if the compressor runs for too long that is will shut off before it burns up. Also an emergency kill switch will be a good investment in case things decide to get wonky.

For the elbow subsystem any problems will likely be due to the controls or a bad connection. There the first step would be to carefully inspect all connections looking for any unintentional shorts and if nothing is found use the voltmeter to ensure that power is going where it needs to and the motor controller is functioning as desired. If the movement is jerky, then the first step would be to reexamine the feedback controls. If the problem persists then the encoder would need to be checked by manually rotating the shaft and seeing if the movement is read as intended.

8: TESTING

8.1: EMG Testing

For reliability purposes, the EMG sensors shall be tested on a breadboard design. The outcome of this testing essentially provides the team with information regarding how EMG works and what signals are to be received from the EMG sensor. The team will have to become familiar with the pattern that the signals follow when certain forces exerted by the muscle are applied.

Additionally, EMG sensor testing might develop ideas for the designers to optimize the current design set forth by the sponsor. Sensors that are placed on a single PCB might require less components to be used to obtain the same quality of the signals. As the team becomes more and more familiar with the EMG technology, improvements might need to be made to provide a better SNR.

8.2: Software Testing

Software testing will not require hardware components at the early stages of integration, aside from the EMG sensors that will be used. The initial objective will be to obtain a signal that can be processed for an accurate result. As flaws are found in the code, they will be properly addressed and re-tested. The exit criteria for the early stages of software testing shall be met once the MCU is able to obtain inputs from the EMG and provide some data output. Actuation will be looked at at the later stages of integration. Input EMG ranges shall be adjusted accordingly as the input testing is performed on the user, and some of the original design might change due to information found from the patient's EMG signals that can further constrain the high level design for the software. Elbow motor control through its drive should be tested at the early stages of integration, however, so that a better understanding can be obtained on its functionality. This will provide a better head-start toward designing a robust software for its control.

The later stages of integration requires the software to be more reliant on the rest of the hardware in order to ensure a reliable interface between the controls and their respective actuators. Testing will undergo a bottom-up implementation, where the hand and the elbow actuators will be tested separately for functionality. Additional hardware support might be required to ensure a full interface with the software. Once both actuators prove to work individually with the controller, they will be tested in combination to ensure no synchronization issues occur that could affect the stability of the system. At this stage, the software will be less subject to high-level design changes as the implementation has solidified.

8.3: Hardware Testing

Proper testing of hardware must be thoroughful prior to assembling all components into a single PCB. A component checklist will be generated for organizational purposes, for which all components must be tested individually in order to check for reliability and

make sure that they are constrained under the specs provided by their respective datasheets. In the event that a component fails to adhere to the specifications set by their datasheet, an equal component shall be tested to check for defects. Alternatively, the team will need to order a similar part from a different vendor if there is a pattern that affects the operation of the first component chosen. These components include regulators, capacitors, resistors, amplifiers and other integrated circuits. As the electrical components are put together to form a larger, more complete system, they will be tested and properly measured for functionality. The end goal is to integrate all the different parts that make up the design until the expected functionality goal is reached.

Once all the assembly has been put together, single PCB testing would be necessary in order to ensure reliability of the system. The main goal of testing hardware equipment is to guarantee functionality. Factors such as excessive load, power efficiency and temperature control will be sought out in order to ensure that the design is functioning under its proper constraints. The team will allow the system to be optimized once it works properly if enough time is provided for changes for the first iteration. Past any optimization, whether or not it could be done by time constraints, the hardware will not seek to provide more changes to the design.

8.4: Electro-Mechanical Testing

8.4.1: Hand Actuation Test Criteria

The testing for the hand actuator will need to be defined such that it does not pose a danger to the user or others. The pneumatic hand actuator should be able to mimic a function similar to that of a typical hand. The pneumatic actuator should allow the user to pick up and hold items securely without crushing them. The movement of the pneumatic actuator should not be so fast to the point which it causes the user discomfort, but at the same time the movement should not be so slow that it causes a significant delay in the desired reaction. In all the hand should open and close in a manner similar to that which a typical hand would perform the same action.

A test bed will be created to determine the specific amount of force which the pneumatic hand actuator will produce as well as to calibrate the system ensuring that the actuator performs as desired. These tests will be conducted before the exoskeletal arm is fitted to the end user. The testing rig will be set up in way to be able to numerically quantify the actual force data generated by the pneumatic hand actuator.

The most accurate way to measure the force exerted by the grip of the pneumatic hand actuator would be by measuring the pressure generated in a flexible container being gripped. The flexible container would need to be a sort of vessel which is easily compressible but at the same time be resistant to expansion. A simple example of a vessel which would meet this criterion would be an empty water bottle or something to similar effect.

This vessel will need to be fitted with pressure sensor which has a wide range and a be sensitive enough to detect pressure changes around the order of one psi. Utilizing a

simple plastic bottle will likely be the easiest way to get a rough estimate of the force exerted by the hand. Whether or not a more sophisticated testing rig will be necessary will depend on our findings. Assuming that the exoskeletal arm will only be used for basic tasks it is likely that the bottle will provide sufficient data. The actual force of the grip can then be calculated by finding the contact area of the hand and relating it to the pressure increase produced in the bottle.

The optimal grip forced will be adapted for the specific needs of the individual patient. Thus this is not a one size fits all design. The patient's fingers may be curled in a certain way and there would be a certain amount of force to overcome this condition. The best design for this this exoskeletal arm is one in which the user is most likely to use and benefit from it. Therefore, if the design is overly cumbersome, too strong, or too weak then the patient is not likely to use and benefit from the device. The measure of success for this design will be determined by the end user and just how much they will actually use the device to help them throughout the day.

Testing to determine the extent of the grip strength required will be a process in which the patient's feedback will be crucial. The level of desired grip strength is a requirement which must be established by the user, especially since this system will have only two modes either open or closed. Thus the end user will have to provide feedback on what extent of grip is expected.

8.4.2: Elbow Actuation Test Criteria

Elbow actuator testing will have to be conducted to ensure that the elbow works in a safe controllable manner. The way that we will test the maximum torque of the elbow would be through a dead lift. This would require the attachment of a hanger which certified weights could be stacked to determine produce a force. The arm then would need to be parallel to the ground to allow for an accurate measurement. The distance from the center point of rotation for the elbow to the point of contact for the weight hanger will be multiplied by the force of the weights to find the total torque. Since torque is essentially a moment doing this will allow calculation of the moment. By increasing the weight on the arm the point at which it does not move, or stalls out, then the maximum torque for the system can be found.

This reason for determining the maximum torque which the elbow can produce is that it is the limiting parameter of the system. If the torque on the elbow becomes too great, then it may pose a serious threat to the user. On the other hand, if the torque is insufficient then this makes the elbow portion of the exoskeletal arm essentially useless. This is why the design must fit into the desired range of elbow torque being both safe and function because if that criterion is not met then the design become a failure.

9: CONCLUSION

In a high level, the system is complex to accurately design, since it has never been built before. None of the exoskeleton arms that the team has found in the market implements a hybrid design between a motor actuation and the use of pneumatics. The team feels that the current scope of work for the design suffices the need to present a valid first iteration that will be improved upon by the next generation of designers that will further optimize the system by choosing smaller parts and working closer with the mechanical team. As a working prototype is delivered, some electrical subsystems that make up the larger system could constitute senior design projects in their own right. Later iterations could even attempt to make the design more portable for patients that are not wheelchair-bound.

Part of its complexity is also derived from working with mechanical components, which pose a larger number of interfacing requirements between the electrical and mechanical fields of expertise to be taken into consideration for the creation of a patient-working model. For the purposes of demonstration, the arm's functionality presented on a table model is a viable option to simulate the patient's anatomy and how the actuators would allow for arm movement.

9.1: Input Testing Conclusion

The reuse of the EMG sensors that the sponsor provided appears as a good idea in terms of use of resources. By reusing its components, not only does the team have more access to the equipment but it can also find weak points in the design. These attributes, if found, can be improved to provide better data collection for the exoskeleton arm as well as any other application that the sponsor currently has for the EMG sensors.

As far as the patient's input testing goes, it is suitable to collect valuable data that can be used to calibrate the design. After the input testing has been completed, two potential optimizations are going to come out of it: design optimization for both mechanical and electrical systems, and optimization of the input testing procedure itself. Since this input test has been established for the first time at the onset of this project, it is expected that it will be used for subsequent tests on different patients for device calibration. The only factor that plays against any time constraint is the paperwork required in order to collect data from the patient, and as the organization that sponsored the team develops, the needed time for authorization paperwork will be reduced as a result of more available resources.

9.2: Software Conclusion

9.2.1: Microcontroller Conclusion

The microcontroller selected was a good choice for this project. Given its specifications, which include a high clock rate and many pins, its community support, and its shallow learning curve, we could not have come up with a better controller. It took about 6 weeks to become fully acquainted with it and it has not let us down in terms of performance.

The first steps of the process included selecting the controller and learning to use it. Once familiar with it, then began the practice of writing example programs for it. Soon we were able to develop a design for the system and come up with how the system would work in regards to our controller. When everything was done, the final part before coding was to map the pins to each hardware component. Now that this is complete and hardware and software support are on the same page all that's left to take our design and put together the system; all based around this Arduino MEGA2560.

9.2.2: Code Approach Conclusion

The process for our code approach through many iterations and pages of pseudocode and diagrams were created. Through each step of the way we kept in mind how we wanted the system to function when we were finished. We want something that is responsive and easy to configure to the user. With these goals were able to come up with something reasonable.

Ultimately the final design for the system is fairly straightforward. The basic idea is that input will be taken from the EMGs. These signals will be processed in the microcontroller. We look at the amplitudes of the signals and do some calculations. These signals are then relayed as outputs to the pneumatic system of the hand or the stepper motor of the elbow. The code contains constant thresholds and multipliers that can be configured to the strength of the user and thus makes for a user-fitted system. When the system is complete we should have a functioning electrical system to an exoskeletal arm that reacts well to the muscle activity of most given users.

9.3: Hardware Conclusion

9.3.1: Selected Batteries

The battery is an essential part of the project, and our goal was to choose the smallest battery that could output the highest amount of power. Also, we needed a battery that was safe enough to be used around the patient. The strongest candidates were the lithium polymer because of its power and size, and also the lithium iron phosphate (LiFePO4) which has similar characteristics.

In the other hand, lithium iron phosphate has a lower energy density when compared with other chemistries like LiCoO2, but offer a longer lifetime, better density and are inherently safer. They also offer a longer cycle life than other lithium batteries. To add to the advantages of the lithium iron phosphate, this battery it is non-toxic which makes it safe to the environment. The design in these batteries allows them to have a very constant voltage during discharge and its voltage stays really close to its 3.2V until the cell is exhausted. By keeping its voltage as close as possible to 3.2V it prevents fluctuations in voltage that could damage components. These batteries are so stable than in cases could eliminate the need to regulate its output. For the reasons stated above, these were the reason why the core team has decided to move forward with the LiFePO4 batteries.

9.3.2: Selected Voltage Regulator

Voltage regulators receive an unregulated input voltage, and manipulate the signal to get a higher or lower constant output voltage. In the beginning of the project we thought that we could regulate an input voltage by using a voltage divider which, in real life scenarios, it is not a good idea to regulate sources with a voltage divider. Also, voltage dividers only allow stepping down the voltage and would never step up the voltage. The reason why, is because components change due to temperature, and loads do not draw the same power at all times. By trying to power a load by using a voltage divider would be extremely inefficient because the resistor on it would not be able to dissipate a lot of heat. A resistor usually does not handle much power; therefore, the voltage divider has been ruled out as circuits that will regulate power in this project.

Therefore, since voltage dividers were eliminated as an option to regulate voltage, we had to focus into linear and switching regulators. Linear regulators and switching regulators both have their advantages and disadvantages. In the table below, *Selected Voltage Regulators 1* we can see and have a better understanding on when it would be suitable to use each one of them.

	Linear	**Switching**
Function	Only steps down, so input voltage must be greater than output voltage	Step up (boost), step down (buck), inverts polarity
Efficiency	Low to medium, but actual battery life depends on load current and battery voltage over time. Efficiency will be high is difference between input and output voltage is small	High, except at very low load currents (μA), where switch-mode quiescent current (I_Q) is usually higher
Waste Heat	High, if average load and/or input to output voltage difference is high	Low, as components usually run cool for power levels below 10 W
Complexity	Low, usually requiring only the regulator and low value bypass capacitors	Medium to high, usually requiring inductor, diode, and filter caps in addition to the IC; for high-power circuits, external FETs are needed
Size	Small to medium in portable designs, but may be larger if heatsinking is needed	Larger than linear at low power, but smaller at power levels for which linear requires a heat sink
Total Cost	Low	Medium to high, largely due to external components
Ripple/Noise	Low, no ripple, low noise, better noise rejection	Medium to high, due to ripple at switching rate

Selected voltage regulators 1: Comparison of the characteristics of switching and linear regulators

For example, our battery is 25.6 V, if we want to power our motor that will be running at 24 V, the linear regulator will be the better option. The linear voltage regulator in this application will be cheaper, easier to implement and the efficiency will be higher, probably even better that if we were using a step down buck converter. Using a buck converter for the motor would require multiple external parts which complicates the circuit and raise the cost. Plus motors would produce noise, which a linear regulator would also reject better than the buck converter.
On the other hand, to power our development board a buck converter will be the right selection. For example, The Arduino development board is recommended to have an input voltage range of 7 V to 12 V. By using a linear voltage regulator in this application a lot of power would be wasted. In fact since we would be stepping down from 24 V to around 10 V we would need to add a big heatsink to be able to dissipate the heat. As stated on the table *Selected Voltage Regulators 1,* using a buck converter is better since the components usually run cool for power levels below 10 W. Since we have multiple applications in our design, our team has decided that it would be beneficial to use both types of regulators in out project. By using both types of regulators, we could control, space, price and efficiency.

9.4: Electro-Mechanical Conclusion

9.4.1: Selected Elbow Components

9.4.1.1: Stepper Motor

The motor selected was a Nema 23 stepper motor with a 47:1 gear ratio gearbox attached to it. The gearbox is a planetary type, allowing for the output shaft to remain centered to the motor shaft. The specific gear ratio of the gearbox is actually 46.656:1, the 47:1 comes more as an easy marketing number. The 47:1 gearbox allows the holding torque of the combination to be greatly increases from 1.89 Nm coming out of the motor directly to 40Nm on the other end of the gearbox. 40 Nm equates to right around 29.5 ft lbs. of torque. This number seems large and so does the size of the motor however, 30 ft lbs. of torque at the elbow only equates to a maximum of about 20 ft-lbs. at the hand (assuming the hand is located approximately 1.5 feet from the center of rotation for the elbow). That 20 ft lbs. is still neglecting the weight of any added hardware as well as any weight or inherit force in the patient's arm itself. Thus in realistic terms this torque would equate to the patient being able to lift and move around about 10 lbs worth of equipment.

Now the 10 lbs lift expectancy is only an estimate and will still depend on the patient (who has yet to be fully sized up for the system) and the hardware developed by the other teams working on the exoskeletal arm. One upside to using this motor is that it still allows wiggle room in the design for the mechanical team. While the motor on its own should have sufficient torque for the needs of the patient a gear system could still be added to the design. An additional gearing system to increase the torque could easily increase the output four fold. While not anticipated it is good to have leeway in case something such as the patient's arm being strongly fixed in its position may require significantly more torque to move.

On the other hand, this motor also allows for the design to go in the other direction. For example, if the mechanical team decides to go in a different direction to increases the motor's torque then the gearbox could be removed thus reducing the weight of the combination by half. Even if the mechanical team decided that a different motor combination would suit the needs of the exoskeletal arm better then, a smaller motor could easily be accommodated. The hardware design of our system can easily be adapted to just about any stepper motor as long as it does not require a voltage greater than that of our battery, remains operable at a minimum of 12 V, and has a maximum current draw below 4.5 A. As long as the voltage remained the same between motors then the only thing which would need to be modified would be the code. The code would need to be adjusted to the new motor to ensure that it moved at a speed similar to that desired. The reason the speed would change would likely be due to a change in the motors gearing. This situation can be easily remedied with a few lines of code. The only other possible source for change would be if the new motor was only a 12 V then, an additional 12 V voltage regulator would need to be added to the system such that the additional current of the motor wouldn't overload the 12 V voltage regulator already in place.

The current required by this particular motor is 2.8Amps per phase. This particular motor is a bipolar stepper meaning that it has two phases. Though it has two phases the maximum current draw is not double the current per phase. The maximum current of a bipolar stepper motor occurs at the point which the poles of the rotor are right in between the coils of the motor. An image of this is shown in the figure *Stepper Motor 1* below.

Stepper Motor 1: Maximum Current Rotor Position
Image from youtube.com

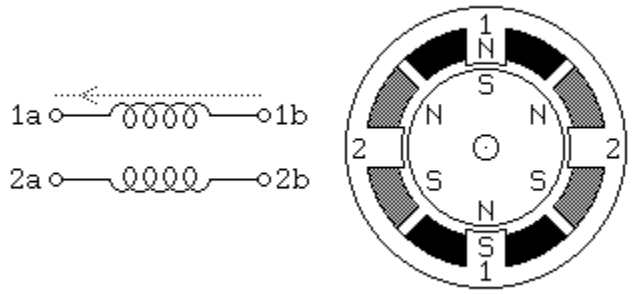
At this point shown in the figure above each of the two pairs of coils draw about 70% of their maximum current per phase. This correlates to a total current draw of about 1.4 multiplied by the current per phase. For this particular motor with a current per phase of

2.8 A that correlates to a maximum current draw of about 4 A. This level of current draw is relatively high for the motors of this scale which limited the options of stepper motor drivers which were rated for this high of current and easily interface with the Arduino. This will be discussed further in the stepper motor driver section.

The torque of this motor is relatively high, however it does come a price and that is size. Unfortunately to find a motor with sufficient torque at a reasonable price the only options were large. Though the motors size is directly related to the power of the motor, this Nema 23 stepper combination is not maximized for size. There are other sources for larger more optimized motor combinations from manufactures such as Moog and the Maxon motor company, however their prices are easily three times higher and tend to use proprietary components. Another consideration for the large size of the motor is that it still needs to be able to interface with the required load. Having a thick keyed shaft allows for a robust attachment to the joint at which the motor will be mounted. For the time being the Nema 23 stepper motor will provide a sufficient analog to the end model that will be used in terms of torque and power draw.

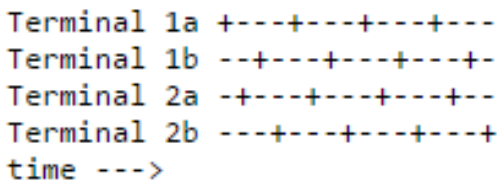
Though the gearbox of the stepper motor significantly increases the output torque it also decreases the speed of the motor proportionally to the gearbox ratio. A gearbox ratio of 47:1 equates to a speed reduction of 47:1 meaning that it will require 47 revolution of the motor's shaft (on the shaft before it enters the gearbox) there will only one revolution of the output shaft coming out of the gearbox. The Nema 23 stepper motor selected has a maximum speed of about 3000 rpm which would be about 50 revolutions per second. That means that on the output side of the gearbox shaft the maximum rotational speed would only be around one revolution per second which is still much faster than the elbow of the exoskeletal arm would need to move. A speed of around 0.25 revolutions per second would approximately be the ideal speed of the elbows movement. This would allow controlled fluid movement without excessive jerk on the patient.

A bipolar stepper motor such as the Nema 23 selected for this application has four wires coming off of it. There are two sets of coils or phases in a bipolar stepper motor and each of these coils has two wires associated with it as shown in the figure *Stepper Motor 2* below.



Stepper Motor 2: Stepper Motor Windings
Image from mechatronics.mech.northwestern.edu

Each of these coils have two terminals and each terminal is a wire coming off of the stepper motor. Now these coils need to be energized in sequence to provide movement in the desired direction and the figure *Stepper Motor 3* below shows what that sequence is. Here the + would equate to the positive terminal of the battery and the – would be the negative terminal of the battery.



Stepper Motor 3: Stepper Motor Sequence
Image from mechatronics.mech.northwestern.edu

Now each column of this sequence is a single step only moving the motors shaft about two degrees each time. So for this motor to rotate at a speed 3,000rpm there would need to be over 540,000 steps per minute or 9,000 steps per second. This means that the polarity of these wires would need to be changed over 9,000 times every second to produce a motor shaft speed of 3,000rpm. This is also taking into consideration the fact that these are full steps and microstepping is not involved, but that will be discussed further in the stepper motor driver section. So, for the motor's shaft to rotate at the necessary speeds a stepper motor driver is necessary to properly achieve the switching necessary. The speed is determined by the code commanding the motor to take the desired number of steps over a given interval. Although the motor is idealized as having only two section of windings in actuality the real makeup of the coils is more complex such as in the figure *Stepper Motor 4* below. As can be seen there are actually eight wraps of coil as opposed to the four which is normally shown in generalized diagrams. This allows for less of a jump between poles increasing the level of torque available.

Stepper Motor 4: Stepper Motor Cross Section
Image from Pololu.com

9.4.1.2: Stepper Motor Driver

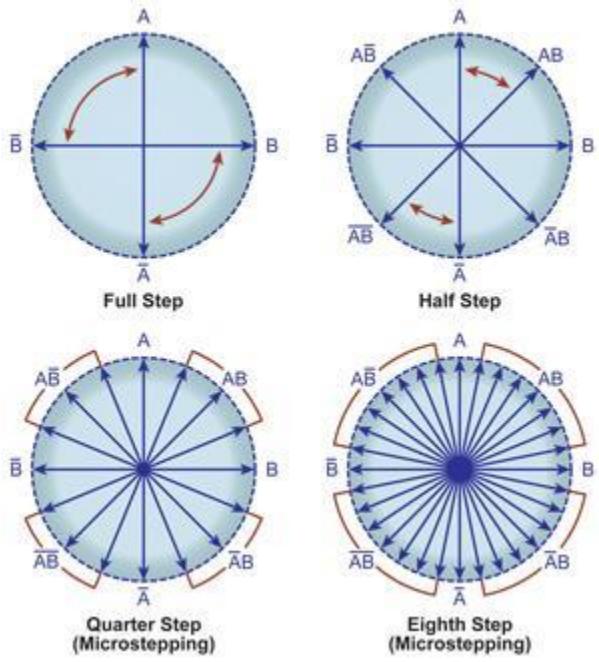
The stepper motor driver was selected based on the Nema 23 stepper motor which will it will be controlling. The stepper motor driver selected is the TB6600 stepper motor driver manufactured by SainSmart. The driving force behind the selection of a stepper motor driver itself came from two sources; the first was the motor itself and the second was the Arduino. A motor of this scale is typically connected to a wall mounted power supply and usually controlled by something like a desktop computer. To connect the motor to a battery and control it through the Arduino came with a couple of complications. First, is that the maximum current draw from the motor is just under 4 A. This eliminated the majority of hobby grade stepper motor drivers right off the bat. The next issue which was encountered was that the stepper motor driver had to easily interface with the Arduino itself. There were a few potential stepper motor drivers which had micro USB or a SPI interface, both of which would not be easily integrated with the Arduino. Then the SainSmart TB6600 was found and it met all of the requirements for both the motor and the Arduino. The only downside to this stepper motor driver is that is a bit larger than the other options. Part of the added size is due to the housing and the heat sink, which may come in handy due to the high current draw of the motor.

The TB6600 is an ideal stepper motor driver for prototyping it offers eight over-current protection setting between 0.6 A and 4.5 A. This is perfect seeing how the selected motor is rated to draw a maximum of about 4 A. This also allows for wiggle room if a different motor is to be used that would draw a little more or a significant amount less of current. The input voltage range for the driver falls between 12 and 40 V. This voltage range works perfectly for the selected motor which will likely be running off a 25.6 V battery. Another upside to this wide input voltage range is that it allows for the use of a motor that can operate a 12 V should the design later call for a different motor and if the size of the

motor should need to be increased then the drive is still able to accommodate a voltage up to 40 V.

There are six signal input terminal for this particular stepper motor driver, all of which range from 0 to 5 V. The first set of inputs is two pulse inputs, there is a positive and negative terminal and the voltage is varied between 0 and 5 V to control the number of pulses or the speed of the motor. Then next set of inputs is the direction for which there is a positive and negative terminal and they act in a binary fashion of either high 5 V or low 0 V which controls the direction which the motor rotates. The final set of inputs controls the enable which also works in a binary fashion with the high being enable and allowing the motor to run or the low being disable the motor.

This particular stepper motor driver accommodates microstepping and has six microstepping settings ranging from full step to 1/16 step. Microstepping allows for the motor to move in smaller increments, taking more steps per revolution. The figure *Stepper Motor Driver 1* below shows how microstepping affects the movement of the motor.



Stepper Motor Driver 1: Microstepping
Image from allegromicro.com

Microstepping moves the motor in smaller increments, but it also requires more increments or steps to the complete a full revolution. For example, on full step mode say one revolution requires four steps to complete and then the driver is switched to quarter stepping mode. In that case then it would now require 16 steps to complete a revolution and this inevitably affects the motor's maximum speed because the limiting factor is how many steps or times the polarity of the windings can be changes per second. So, the

motor may move more precisely with less vibration in microstepping mode but it will lose torque and have a decrease in the maximum output speed.

Although microstepping allows the motor to move more precisely it can cause a reduction in torque and since this design need all the torque it can get the stepper motor driver will likely be running in full step mode. However, should it be decided that the torque needs to be decreased then microstepping can be used to accomplish this goal to some degree.

9.4.2: Selected Hand Components

9.4.2.1: Compressor Pump

The vacuum pump selected is capable of creating a positive pressure of 6 Bar or about 87 psi. This pump is also capable of creating a negative pressure of 85 kPa or about 12.7 psi which will be utilized on the low pressure side of the system. This particular pump will run on 12 V and have a maximum current draw of 3.75 A. The pump is basically a DC motor which turns an impeller creating suction from the low pressure port and directs the flow to the high pressure port thus pressurizing the high pressure tank in the design. The final compressor pump selected was the model 7A12D60R52 from the Shenzhen Justar Electronic Technology Co., this company's product was mainly selected due to the low cost compared to similar pumps produced by domestic manufacturers.

There were four main issues in selecting the compressor pump. The first issue which arose was that the compressor needed both an inlet and an outlet. This was due to the closed loop nature of the design. The pump had to be able to not only pressurize the high pressure tank but also be able to accomplish this by depressurizing the low pressure tank. The next issue was finding a suitable compressor which allowed for a slight vacuum to be created on the low pressure side of the system. This vacuum will aid in sucking the air out of the hand actuator without the need to run the pump continuously to accomplish the same goal. The other issue encountered was finding a compressor pump which would allow for a relatively high output pressure of at least 50 psi. Since the high pressure tank will need to be maintained at about 50 psi the compressor pump would need some degree of compression power beyond that to ensure that it would not be constantly running at maximum capacity. The last hurdle was finding a pump which was designed specifically for moving air at the pressures needed. While there were many pumps that fit the criteria most of them were only rated to be used with fluids such as water.

9.4.2.2: Solenoid Valves

The solenoid valves which were selected came from two different sources and should allow for interchangeability between the two. The selected solenoid valves are both 12 volts and one draws 0.54 amps while the other draws 0.4 amps. Both of the valves operate in the normally closed position meaning that they will only draw current and be open while powered on. Both of the solenoid valves use standard 1/4" NPT threading on the inlet and the outlets to allow for easy acquisition of fittings. The valves should operate with negative pressure on low pressure side but more testing will need to be done

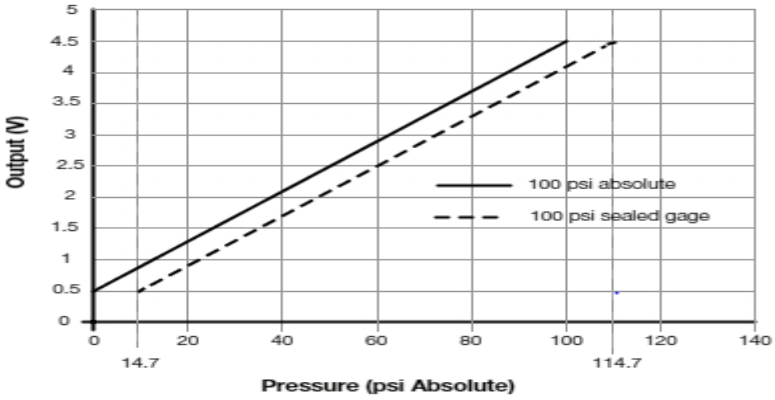
with the valves to determine this for certain. Which is the reason behind selection of two different valves where only one of which will experience negative pressures.

The Arduino will not be opening the solenoid valves directly, rather the solenoid valves will be attached to relays and the Arduino will send the command to open and close the valves to signal pins on the relay board. Since the valves operate in only two modes, either open or closed, controlling them souls remain a relatively simple task.

9.4.2.3: Pressure Sensors

The pressure sensors selected belong to the model PX3AN1BH100PSAAX manufactured by Honeywell. This pressure sensor is rated for pressure ranges from 0 psi to 100 psi with an output voltage range from 0.5 V to 4.5 V. The pressure sensor requires a supply voltage between 4.75 V and 5.25 V. The sensor draws a maximum current of 3.5 mA; this means that the power can be supplied by the Arduino itself. This pressure sensor has a wide range of operating temperatures ranging from -40 °F to 257 °F, the extremes of which will never be reached for this particular system. The treading on the sensor is a standard 1/4-18 NPT which is the same as the threading used on the solenoid valves retaining continuity throughout the system. There are three wires coming off the pressure sensor itself which consist of a voltage supply terminal, a ground terminal, and a signal voltage terminal.

Ideally pressure sensors should have a linear response and for this particular sensor the response follows a very linear trend. Figure *Pressure Sensors 1* below shows the correlation of the pressure sensed and the output voltage. As can be seen, the correlation is a very linear slope, which is ideal to the function of an accurate pressure sensor.



Pressure Sensors 1: Pressure and Output Voltage Curve
Image from arrow.com

This pressure sensor will provide the perfect addition to the exoskeletal arms hand actuation system with its accuracy of +/-1% over the temperature range of -4 °F to 185 °F. For all intensive purposes the user should never be in any conditions that would lie outside of this temperature range. In addition, the pressure range of the system is perfect considering that the maximum pressure reading is above that at which the compressor pump can even supply.

REFERENCES

Text References

[1] https://faculty.washington.edu/chudler/ap.html
[2] http://hyperphysics.phy-astr.gsu.edu/hbase/biology/actpot.html
[3] http://webspace.ship.edu/cgboer/actionpot.html
[4] http://www.aanem.org/Patients/Disorders/Arthrogryposis-Multiplex-Congenita
[5] http://www.britannica.com/science/electroencephalography
[6] http://kidshealth.org/en/parents/emg.html
[7] http://www.bortec.ca/Images/pdf/EMG%20measurement%20and%20recording.pdf
[8] https://www.youtube.com/user/mjlorton
[9] https://www.youtube.com/playlist?list=PLF86F263013F106C0
[10] https://en.wikipedia.org/wiki/Opto-isolator
[11] http://www.BatteryUniversity.com
[12] http://smpp.northwestern.edu/downloads/Design%20Of%20Artificial%20Arms%20And%20Hands%20For%20Prosthetic%20Applications.pdf
[13] http://www.batteryspace.com/
[14] https://learn.sparkfun.com/tutorials/voltage-dividers
[15] http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/
[16] http://www.talkingelectronics.com/projects/H-Bridge/H-Bridge-1.html
[17] http://www.furmansound.com/pdf/catalog/flyer_volt_regulation.pdf
[18] https://en.wikipedia.org/wiki/Voltage_regulator
[19] https://en.wikipedia.org/wiki/Buck_converter
[20] http://micro.rohm.com/en/techweb/knowledge/dcdc/s-dcdc/01-s-dcdc/80
[21] https://en.wikipedia.org/wiki/Printed_circuit_board
[22] http://www.iig-llc.com/blog/2015/02/too-hot-to-handle/

Image References

[I-1] https://en.wikipedia.org/wiki/Action_potential#/media/File:Action_Potential.gif
[I-2] http://www.nrsign.com/monopolar-vs-bipolar-emg-readings/
[I-3] J. Carlo, De Luca, 2002 Surface Electromyography: Detection and Recording", Delsys Incorporated
[I-4] http://www.arduino.cc
[I-5] http://www.autodesk.com
[I-6] http://www.BatteryUniversity.com
[I-7] http://www.batteryspace.com/
[I-8] https://www.sparkfun.com
[I-9] http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/
[I-10]http://micro.rohm.com/en/techweb/knowledge/dcdc/s-dcdc/01-s-dcdc/80
[I-11] http://www.ti.com/lit/ds/symlink/lm2678.pdf
[I-12] https://www.alibaba.com/product-detail/composite-gas-cylinder-PCP-cylinder_60026099315.html

[I-13]http://www.engineeringtoolbox.com/pvc-cpvc-pipes-pressures-d_796.html
[I-14] http://www.solenoid-valve-info.com/solenoid-valve-basics.html
[I-15] http://www.spiegl.org/rocket/Burst/burst.html
[I-16] https://en.wikipedia.org/wiki/Pressure_regulator
[I-17] https://www.maximintegrated.com/en/app-notes/index.mvp/id/762
[I-18] http://codeforfree.weebly.com/optical-encoders.html
[I-19] https://www.youtube.com/watch?v=Bs1zIZV1uSY
[I-20] http://mechatronics.mech.northwestern.edu/design_ref/actuators/stepper_intro.html
[I-21] https://www.pololu.com/product/2267
[I-22] http://www.allegromicro.com/en/Products/Motor-Driver-And-Interface-ICs/Bipolar-Stepper-Motor-Drivers.aspx
[I-23] https://www.arrow.com/en/products/px3an1bh100psaax/honeywell

Appendix

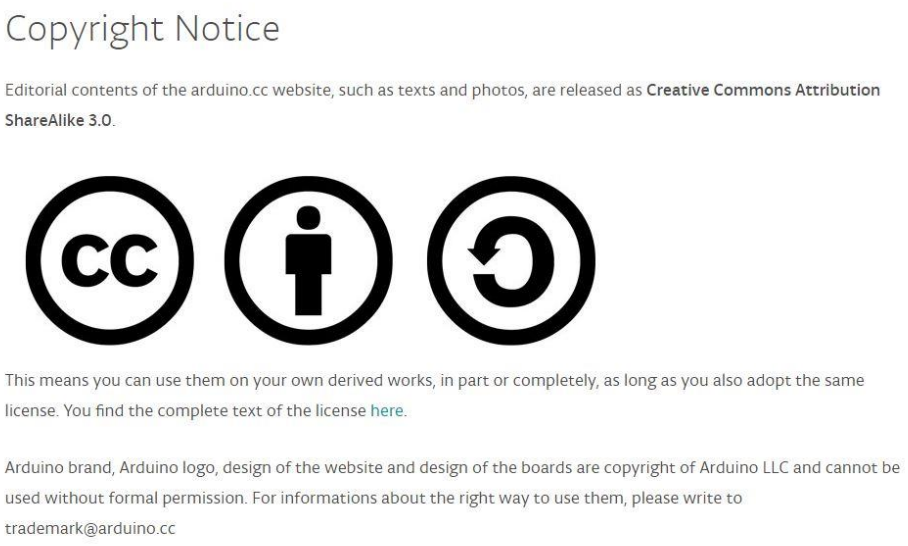
Abbreviations and Acronyms

A	= Amps
AMC	= Arthrogryposis Multiplex Congenita
ASME	= American Society of Mechanical Engineers
ASTM	= American Society for Testing and Materials
BJT	= Bipolar Junction Transistor
DC	= Direct Current
EAGLE	= Easily Applicable Graphical Layout Editor
EEG	= ElectroEncephaloGraphy
EMF	= Electro-Magnetic Field
EMG	= ElectroMyoGraphy
Ft-lbs	= Foot-pound force
FET	= Field Effect Transistor
g	= grams
IDE	= Integrated Development Environment
IEEE	= Institute of Electrical and Electronics Engineers
IGBT	= Insulated-Gate Bipolar Transistor
I/O	= Input/Output
K^+	= Potassium ions
kPa	= kiloPascal
LED	= Light-Emitting Diode
Na^+	= Sodium ions
mA	= Milliamps
mAh	= Milliamps-hours
MCU	= Microcontroller Unit
MOSFET	= Metal Oxide Semiconductor Field Effect Transistor
Nm	= Newton*meter
OSH	= Occupational Health and Safety Administration
Oz	= Ounces
PSI	= Pounds per Square Inch
PWM	= Pulse Width Modulation
SENIAM	= Surface ElectroMyoGraphy for the Non-Invasive Assessment of Muscle
SNR	= Signal-to-Noise Ratio
SPI	= Serial Peripheral Interface
Trimpot	= Trimming Potentiometer
UL	= Underwriters' Laboratory
USB	= Universal Serial Bus
V	= Volts
W	= Watts
Wh	=Watt-hours

Permission To Use

Arduino

Figure *Permission To Use 1* shows the copyright notice for Arduino. It is under a creative Commons license, so we can use it for this document.



Copyright Notice

Editorial contents of the arduino.cc website, such as texts and photos, are released as **Creative Commons Attribution ShareAlike 3.0**.

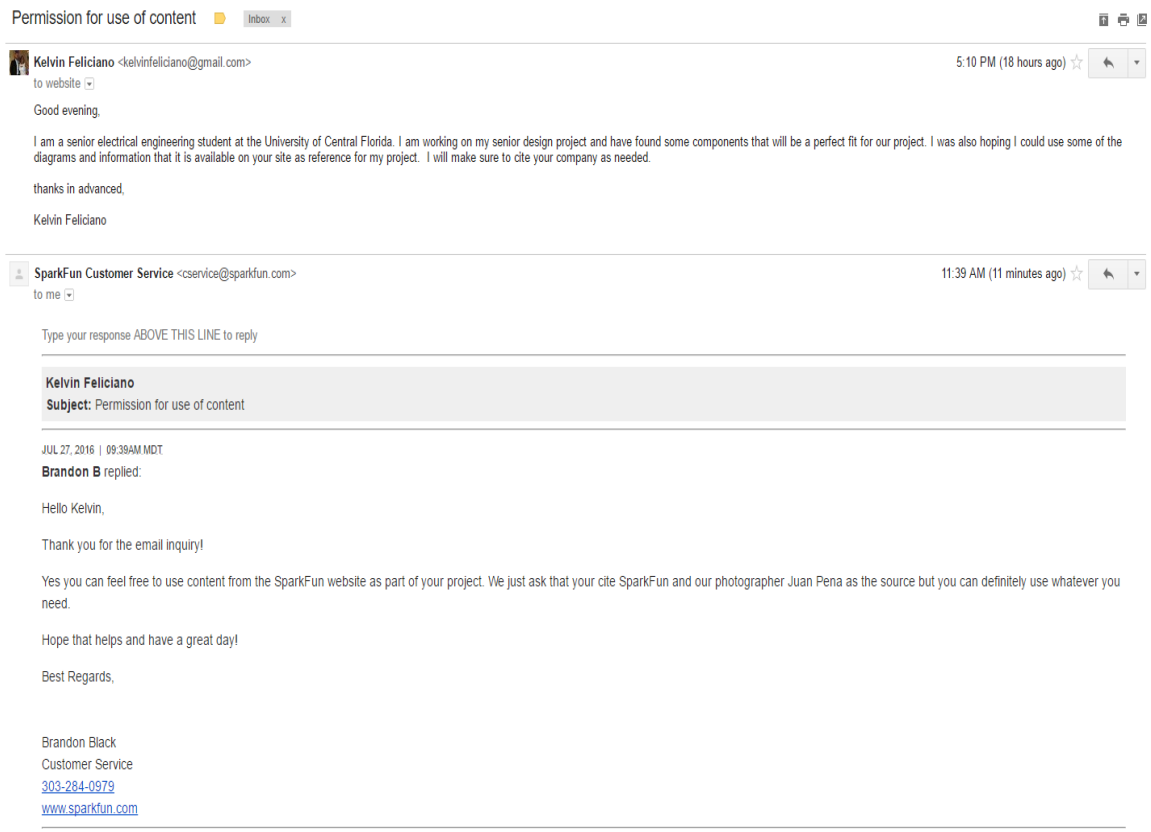
This means you can use them on your own derived works, in part or completely, as long as you also adopt the same license. You find the complete text of the license here.

Arduino brand, Arduino logo, design of the website and design of the boards are copyright of Arduino LLC and cannot be used without formal permission. For informations about the right way to use them, please write to trademark@arduino.cc

Permission To Use 1. Arduino is under a Creative Commons license. This means that we can use it for the purposes of this document.

Sparkfun

Permission To Use 2 is an email that we have sent to Sparkfun asking to use their resources. They responded back yes.


Permission for use of content Inbox x

Kelvin Feliciano <kelvinfeliciano@gmail.com> 5:10 PM (18 hours ago)
to website

Good evening,

I am a senior electrical engineering student at the University of Central Florida. I am working on my senior design project and have found some components that will be a perfect fit for our project. I was also hoping I could use some of the diagrams and information that it is available on your site as reference for my project. I will make sure to cite your company as needed.

thanks in advanced,

Kelvin Feliciano

SparkFun Customer Service <cservice@sparkfun.com> 11:39 AM (11 minutes ago)
to me

Type your response ABOVE THIS LINE to reply

Kelvin Feliciano
Subject: Permission for use of content

JUL 27, 2016 | 09:39AM MDT
Brandon B replied:

Hello Kelvin,

Thank you for the email inquiry!

Yes you can feel free to use content from the SparkFun website as part of your project. We just ask that your cite SparkFun and our photographer Juan Pena as the source but you can definitely use whatever you need.

Hope that helps and have a great day!

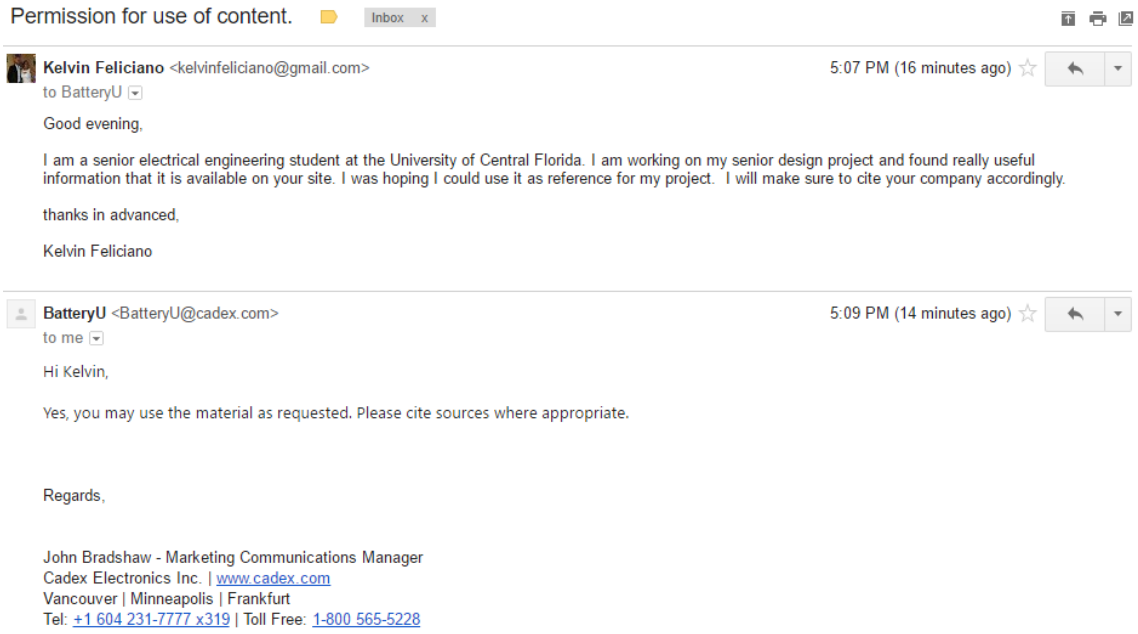
Best Regards,

Brandon Black
Customer Service
303-284-0979
www.sparkfun.com

Permission To Use 2: Sparkfun has given us permission to use their resources.

Battery University

Figure *Permission To Use 3* shows the shows that battery university has accepted our request to use their information, as long as we cite where appropriate.


Permission for use of content. Inbox x

Kelvin Feliciano <kelvinfeliciano@gmail.com> 5:07 PM (16 minutes ago)
to BatteryU

Good evening,

I am a senior electrical engineering student at the University of Central Florida. I am working on my senior design project and found really useful information that it is available on your site. I was hoping I could use it as reference for my project. I will make sure to cite your company accordingly.

thanks in advanced,

Kelvin Feliciano

BatteryU <BatteryU@cadex.com> 5:09 PM (14 minutes ago)
to me

Hi Kelvin,

Yes, you may use the material as requested. Please cite sources where appropriate.

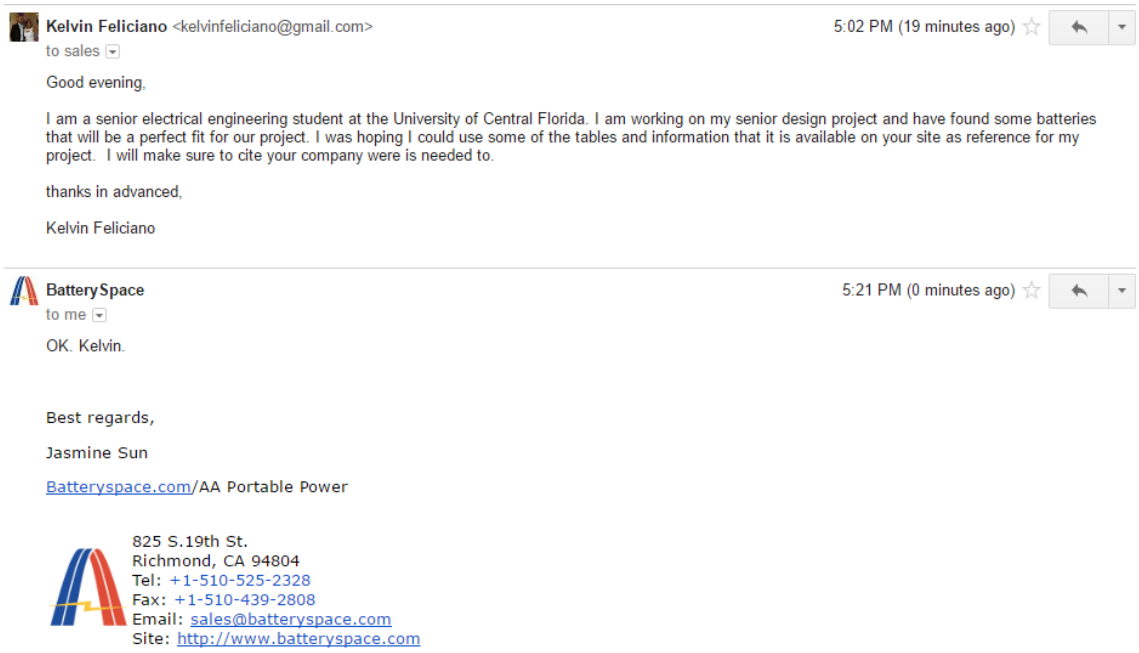
Regards,

John Bradshaw - Marketing Communications Manager
Cadex Electronics Inc. | www.cadex.com
Vancouver | Minneapolis | Frankfurt
Tel: +1 604 231-7777 x319 | Toll Free: 1-800 565-5228

Permission to use 3: Battery University has given us permission to use the information from their site.

Battery Space

Figure *Permission To Use 4* shows that battery battery has given us the ok to use their information.



Kelvin Feliciano <kelvinfeliciano@gmail.com> 5:02 PM (19 minutes ago)

to sales

Good evening,

I am a senior electrical engineering student at the University of Central Florida. I am working on my senior design project and have found some batteries that will be a perfect fit for our project. I was hoping I could use some of the tables and information that it is available on your site as reference for my project. I will make sure to cite your company were is needed to.

thanks in advanced,

Kelvin Feliciano

BatterySpace 5:21 PM (0 minutes ago)

to me

OK. Kelvin.

Best regards,

Jasmine Sun

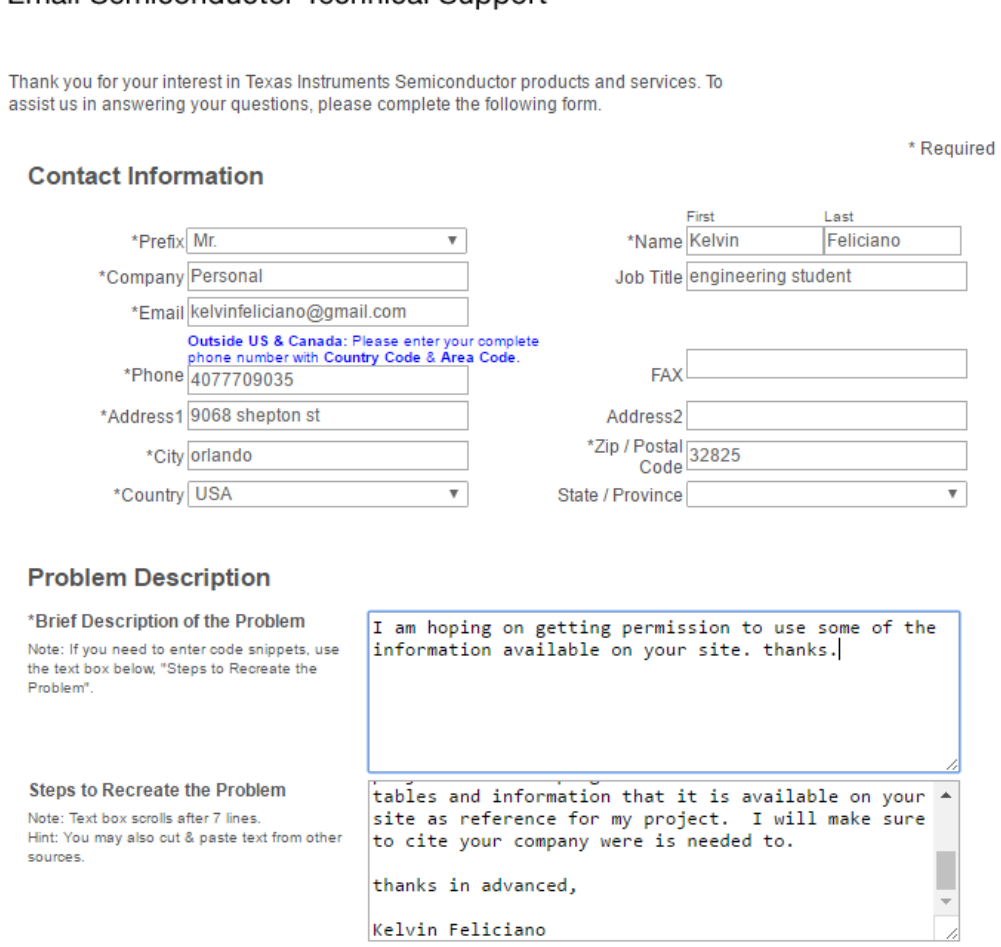
Batteryspace.com/AA Portable Power

825 S.19th St.
Richmond, CA 94804
Tel: +1-510-525-2328
Fax: +1-510-439-2808
Email: sales@batteryspace.com
Site: http://www.batteryspace.com

Permission To Use 4: Battery Space has given us permission to use the information from their site.

Texas Instruments

Figure *Permission To Use 5* shows an email that we have sent to Texas Instruments asking to use their resources. This is currently pending as they have not yet responded.


Email Semiconductor Technical Support

Thank you for your interest in Texas Instruments Semiconductor products and services. To assist us in answering your questions, please complete the following form.

* Required

Contact Information

*Prefix	Mr.	*Name (First / Last)	Kelvin / Feliciano
*Company	Personal	Job Title	engineering student
*Email	kelvinfeliciano@gmail.com		
*Phone	Outside US & Canada: Please enter your complete phone number with Country Code & Area Code. 4077709035	FAX	
*Address1	9068 shepton st	Address2	
*City	orlando	*Zip / Postal Code	32825
*Country	USA	State / Province	

Problem Description

*Brief Description of the Problem

Note: If you need to enter code snippets, use the text box below, "Steps to Recreate the Problem".

I am hoping on getting permission to use some of the information available on your site. thanks.

Steps to Recreate the Problem

Note: Text box scrolls after 7 lines.
Hint: You may also cut & paste text from other sources.

tables and information that it is available on your site as reference for my project. I will make sure to cite your company were is needed to.

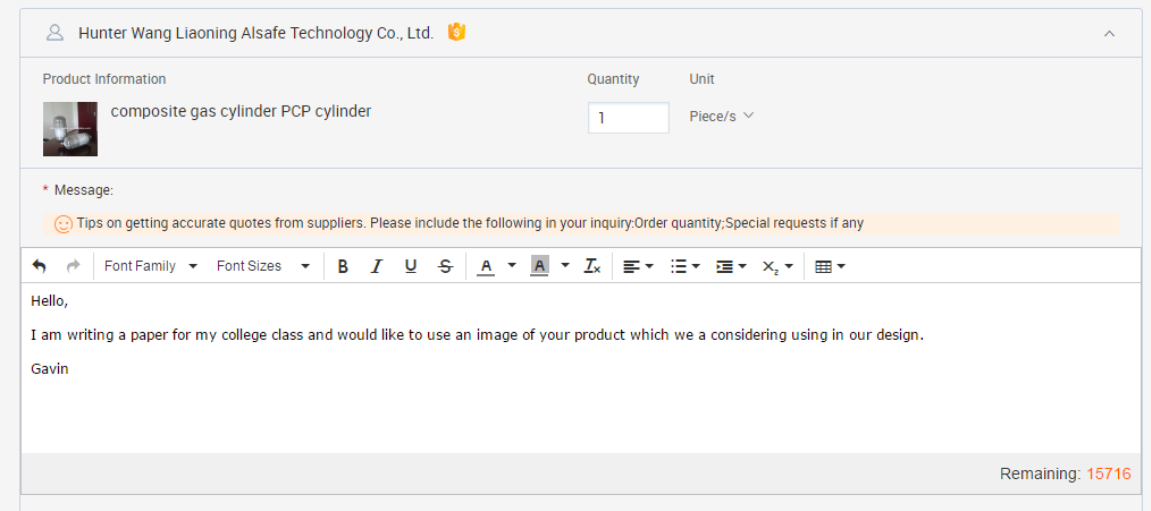
thanks in advanced,

Kelvin Feliciano

Permission To Use 5: Awaiting approval from Texas Instruments

Alsafe Technology Co.

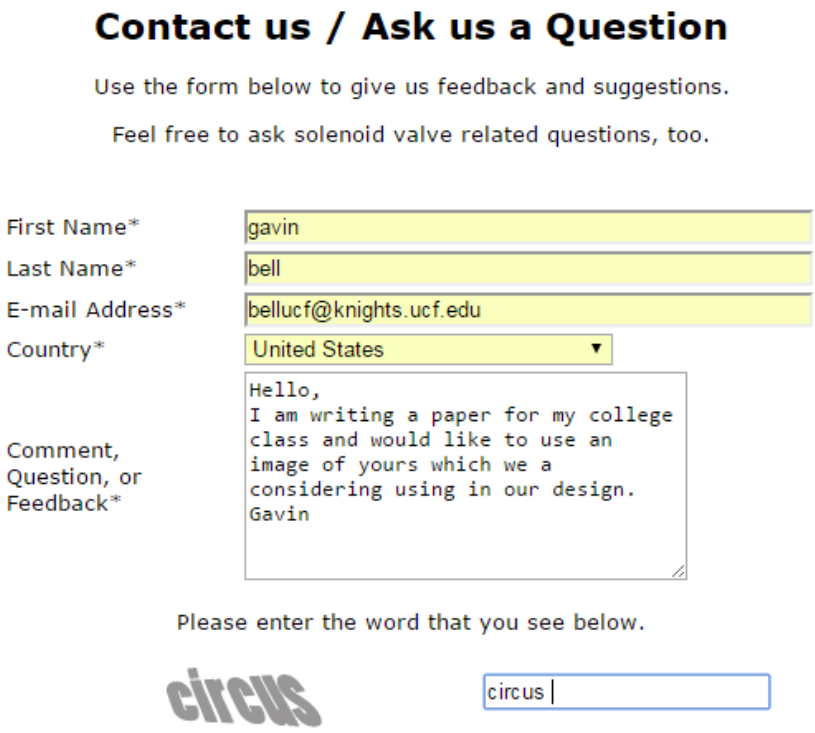
Figure *Permission To Use 6* shows the email sent requesting to use the image of the Fiber wrapped cylinder composition.



Permission To Use 6: Email sent requesting to use the image of the Fiber wrapped cylinder composition.

Solenoid-Valve-Info

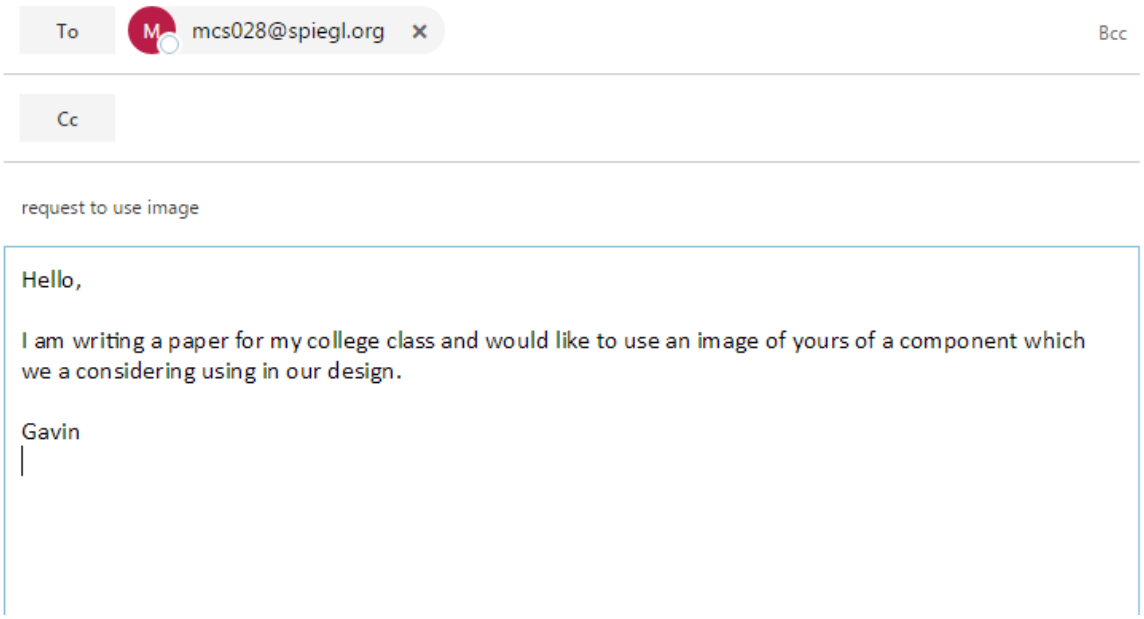
Figure *Permission To Use 7* shows the email sent requesting to use the image of the solenoid valve.



Permission To Use 7: Email sent requesting to use the image of the solenoid valve.

Spiegl

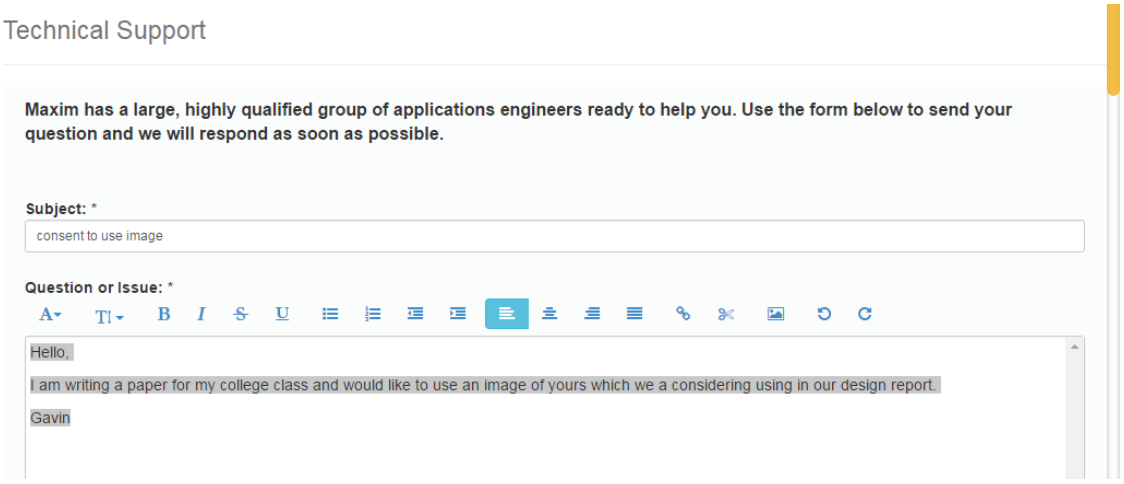
Figure *Permission To Use 8* shows the email sent requesting to use the image of the burst disk.



Permission To Use 8: Email sent requesting to use the image of the burst disk.

Maxim Integrated

Figure *Permission To Use 9* shows the email sent requesting to use the image of the pressure sensor.



Permission To Use 9: Email sent requesting to use the image of the pressure sensor.

Pololu Corporation

Figure *Permission To Use 10* shows the email approving use the image of the stepper motor cross section.

Hello, Gavin.

Thank you for your inquiry. You are welcome to use images from our website with proper attribution.

Please let me know if you have any additional questions.

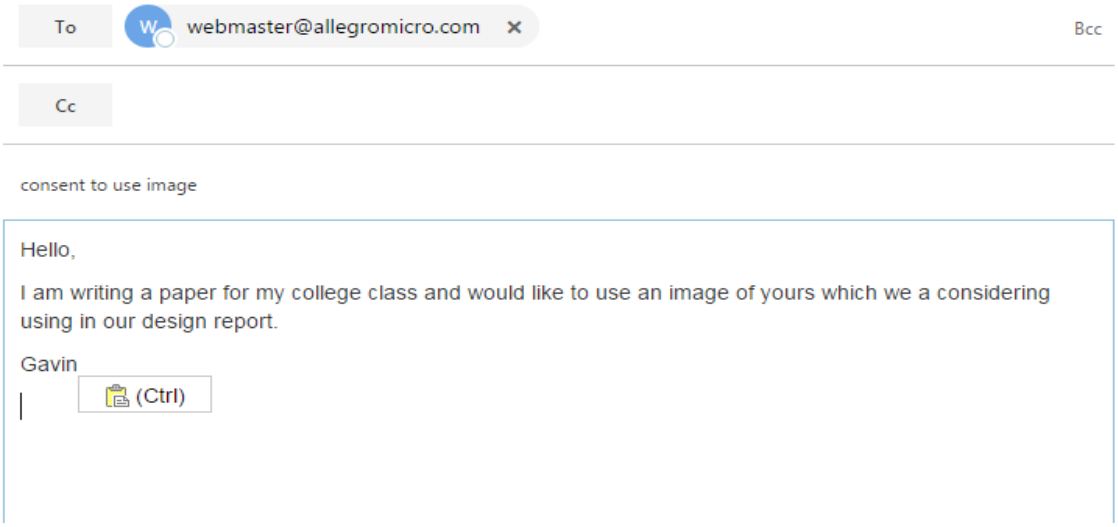
Sincerely,
Derrill Hartley
(702) 262-6648
www.pololu.com

Pololu Corporation
920 Pilot Rd.
Las Vegas, NV 89119
USA

Permission To Use 10: Email approving the use the image of the stepper motor cross section.

Mechatronics at Northwestern University

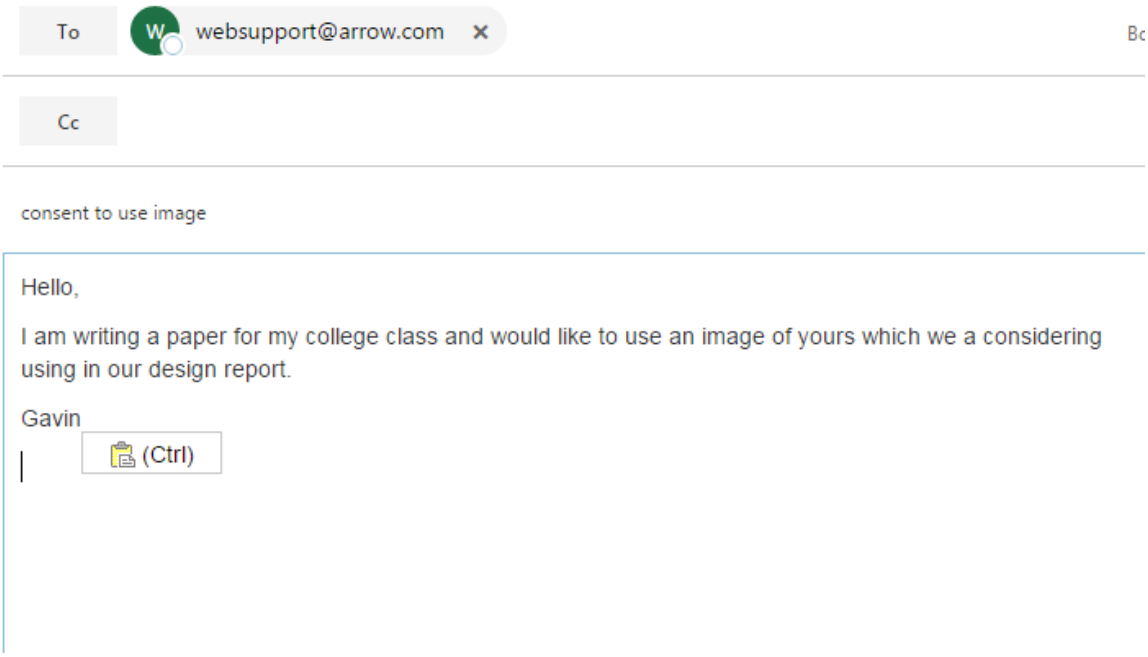
Figure *Permission To Use 11* shows the email sent requesting to use the image of microstepping.



Permission To Use 11: Email sent requesting to use the image of the stepper microstepping.

Arrow

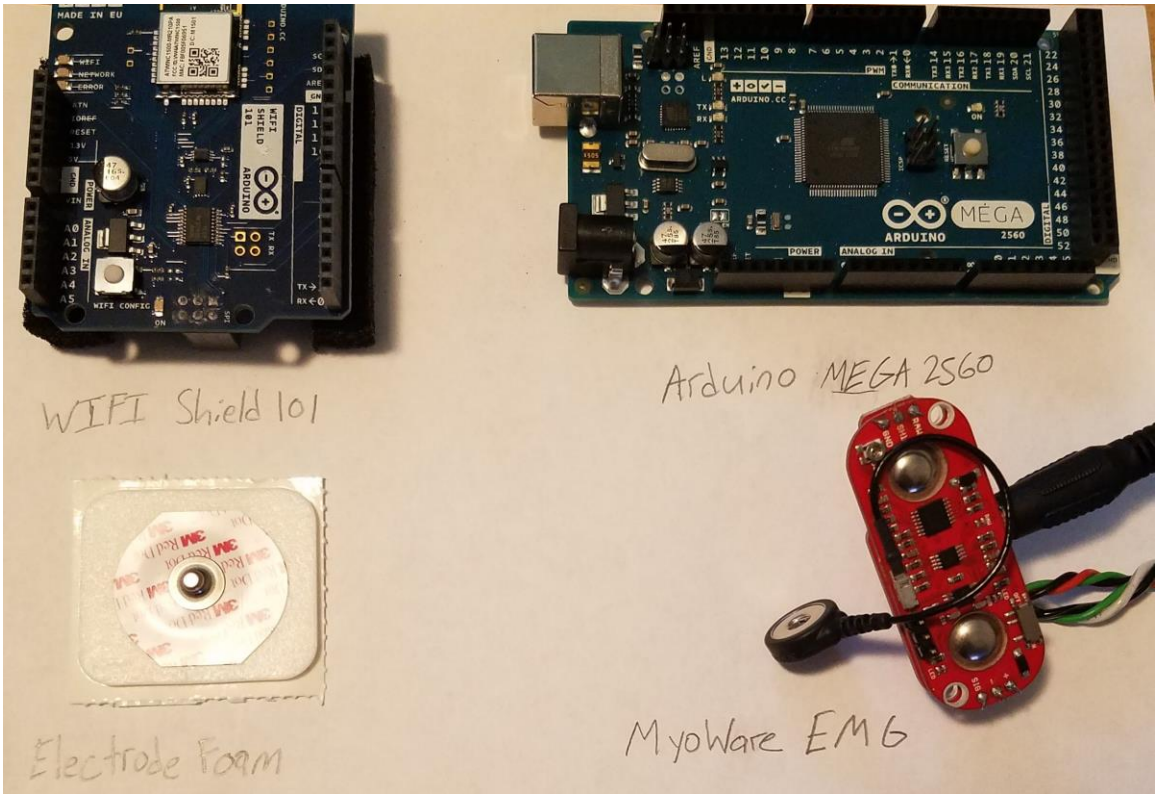
Figure Permission To Use 12 shows the email sent requesting to use the image of the pressure and output voltage curve.



Permission To Use 12: Email sent requesting to use the image of pressure and voltage curve.

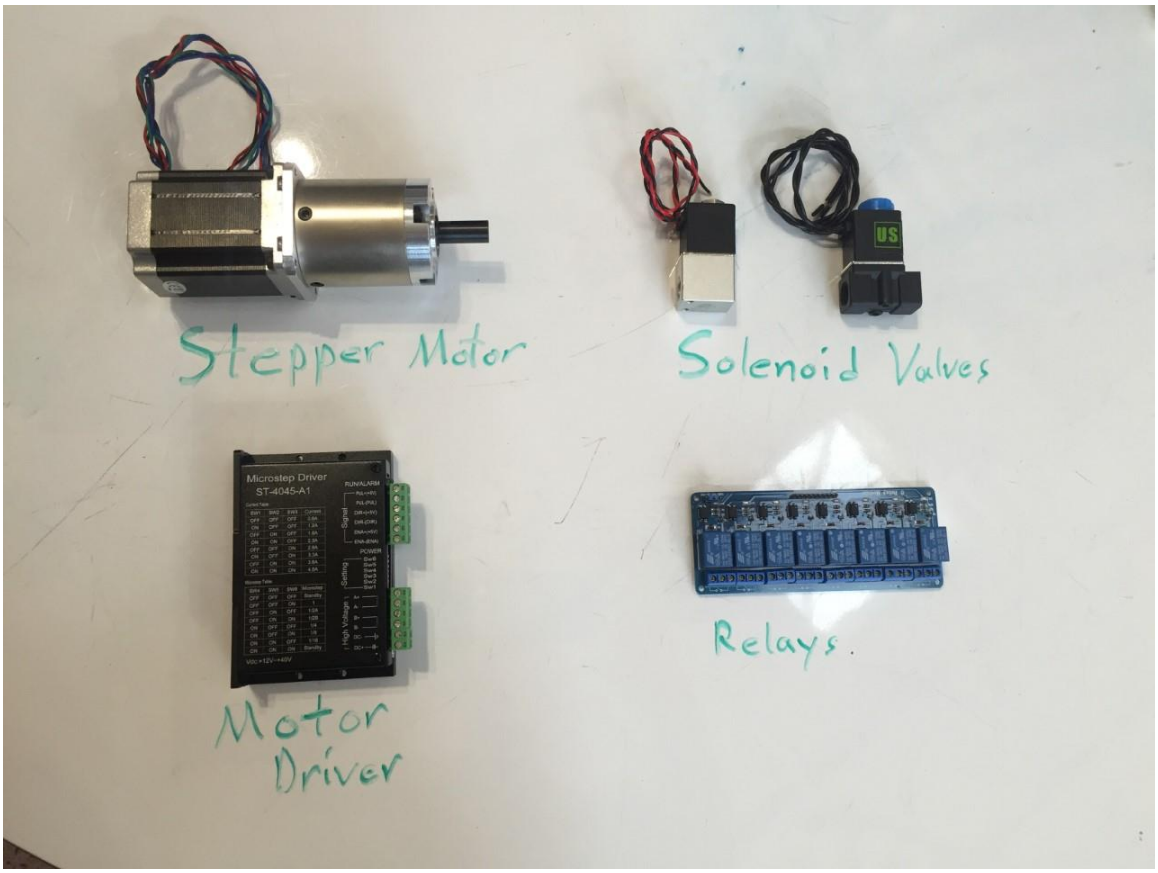
Components Available

This semester we were able to get a jump on ordering some of the components required for the system. *Components Available 1* shows the WiFi shield, Arduino MEGA, Test EMG, and a piece of electrode foam.



Components Available 1. This figure shows a Wifi Shield 101, an Arduino MEGA 2560, a MyoWare EMG, and some electrode foam.

For motor actuation, there were parts obtained as well. *Components Available 2* shows the Stepper Motor, Solenoid Valves, Motor Drivers, and Relays.



Components Available 2. This figure shows a Stepper Motor, Solenoid Valves, Motor Driver, and Relays.