BabelGlove



**Department of Electrical and Computer Engineering**

**University of Central Florida**

**EEL 4914**

**ECE Senior Design Spring/Summer 2020**

**Group 6**

Ann Dang Computer Engineering

Andrew Glisson Computer Engineering

Madison Manley Electrical Engineering

Shreya Mistry Electrical Engineering

**Table of Contents**

[1. EXECUTIVE SUMMARY 1](#_Toc46821866)

[2. PROJECT DESCRIPTION 2](#_Toc46821867)

[2.1 MOTIVATION 2](#_Toc46821868)

[2.2 GOALS AND OBJECTIVES 2](#_Toc46821869)

[2.3 REQUIREMENTS SPECIFICATIONS 3](#_Toc46821870)

[2.4 QUALITY HOUSE OF ANALYSIS 5](#_Toc46821871)

[3. RESEARCH 7](#_Toc46821872)

[3.1 EXISTING PROJECTS AND DIFFERENCES 7](#_Toc46821873)

[3.1.1 SLIG: SIGN LANGUAGE INTERPRETATION GLOVE 7](#_Toc46821874)

[3.1.2 SIGNALOUD 8](#_Toc46821875)

[3.1.3 ASLA: AMERICAN SIGN LANGUAGE ASSISTANT 8](#_Toc46821876)

[3.1.4 DIFFERENCES 8](#_Toc46821877)

[3.2 ETHICAL ISSUES 9](#_Toc46821878)

[3.3 AMERICAN SIGN LANGUAGE 9](#_Toc46821879)

[3.4 BLOCK DIAGRAM 10](#_Toc46821880)

[3.3 HARDWARE 12](#_Toc46821881)

[3.3.1 FLEX SENSOR 12](#_Toc46821882)

[3.3.1.1 INTRODUCTION 12](#_Toc46821883)

[3.3.1.2 CONSIDERATIONS 13](#_Toc46821884)

[3.3.1.3 BI-DIRECTIONAL FLEXIBLE BEND SENSORS 14](#_Toc46821885)

[3.3.1.4 SPECTRA SYMBOL 4.5” FLEX SENSOR 14](#_Toc46821886)

[3.3.1.5 ONE-DIRECTIONAL FLEXIBLE BEND SENSORS 14](#_Toc46821887)

[3.3.1.6 LITTLEBITS BEND SENSOR 15](#_Toc46821888)

[3.3.2 ACCELEROMETER AND GYROSCOPE 15](#_Toc46821889)

[3.3.2.1 INTRODUCTION 15](#_Toc46821890)

[3.3.2.2 SPARKFUN TRIPLE AXIS ACCELEROMETER BREAKOUT - ADXL335 18](#_Toc46821891)

[3.3.2.3 SPARKFUN TRIPLE AXIS ACCELEROMETER BREAKOUT - ADXL337 19](#_Toc46821892)

[3.3.2.4 SPARKFUN 6 DEGREES OF FREEDOM BREAKOUT - LSM6DS3 20](#_Toc46821893)

[3.3.2.5 SPARKFUN TRIPLE AXIS ACCELEROMETER BREAKOUT - MMA8452Q 20](#_Toc46821894)

[3.3.3 FORCE SENSOR 21](#_Toc46821895)

[3.3.3.1 INTRODUCTION 21](#_Toc46821896)

[3.3.3.2 FORCE SENSITIVE RESISTOR - SMALL 22](#_Toc46821897)

[3.3.3.3 FORCE SENSITIVE RESISTOR 0.5” 23](#_Toc46821898)

[3.3.3.4 FORCE SENSITIVE RESISTOR - SQUARE 23](#_Toc46821899)

[3.3.3.5 PHIDGETS TOUCH SENSOR 24](#_Toc46821900)

[3.3.4 MICROCONTROLLER UNIT (MCU) 25](#_Toc46821901)

[3.3.4.1 INTRODUCTION 25](#_Toc46821902)

[3.3.4.2 CONSIDERATIONS 25](#_Toc46821903)

[3.3.4.3 MSP430 26](#_Toc46821904)

[3.3.4.4 ATMEGA32U4 26](#_Toc46821905)

[3.3.4.5 ATSAMD21G 27](#_Toc46821906)

[3.3.4.6 MSP430 VS ATMEGA32U4 VS ATSAMD21G 27](#_Toc46821907)

[3.3.5 WIRELESS COMMUNICATION 28](#_Toc46821908)

[3.3.5.1 NEAR FIELD COMMUNICATION 29](#_Toc46821909)

[3.3.5.2 ZIGBEE 29](#_Toc46821910)

[3.3.5.3 WI-FI 30](#_Toc46821911)

[3.3.5.4 BLUETOOTH 32](#_Toc46821912)

[3.3.6 POWER SUPPLY 35](#_Toc46821913)

[3.3.6.1 BATTERIES 35](#_Toc46821914)

[3.3.6.1.1 LEAD ACID 36](#_Toc46821915)

[3.3.6.1.2 NICKEL CADMIUM 37](#_Toc46821916)

[3.3.6.1.3 NICKEL METAL HYDRIDE 38](#_Toc46821917)

[3.3.6.1.4 LITHIUM ION 39](#_Toc46821918)

[3.3.6.2 VOLTAGE REGULATORS 42](#_Toc46821919)

[3.3.6.2.1 SERIES VOLTAGE REGULATORS 42](#_Toc46821920)

[3.3.6.2.2 SHUNT VOLTAGE REGULATORS 42](#_Toc46821921)

[3.3.6.2.3 SWITCHING VOLTAGE REGULATORS 43](#_Toc46821922)

[3.3.7 FPGA 43](#_Toc46821923)

[3.3.8 PCB 43](#_Toc46821924)

[3.3.8.1 EXPRESS PCB 44](#_Toc46821925)

[3.3.8.2 BAY AREA CIRCUITS 44](#_Toc46821926)

[3.3.8.3 PCB ZONE 44](#_Toc46821927)

[3.3.9 SERIAL COMMUNICATION 45](#_Toc46821928)

[3.3.9.1 ANALOG TO DIGITAL CONVERTERS 45](#_Toc46821929)

[3.3.10 GLOVE 45](#_Toc46821930)

[3.3.10.1 INTRODUCTION 45](#_Toc46821931)

[3.3.10.2 CONSIDERATIONS 45](#_Toc46821932)

[3.3.10.3 NIKE ALL WEATHER 46](#_Toc46821933)

[3.3.10.4 NIKE HYPERWARM ACADEMY 46](#_Toc46821934)

[3.3.10.5 UA RADAR SOFTBALL GLOVES 46](#_Toc46821935)

[3.3.10.6 ULINE ULTRA-LITE POLYURETHANE COATED GLOVES 47](#_Toc46821936)

[3.3.10.7 TACTILE RUNNING GLOVES 47](#_Toc46821937)

[3.3.11 BLUETOOTH LOW ENERGY MODULE 48](#_Toc46821938)

[3.3.11.1 TEXAS INSTRUMENTS CC2541 48](#_Toc46821939)

[3.3.11.2 NORDIC SEMICONDUCTOR nRF51822 49](#_Toc46821940)

[3.3.11.3 MICROCHIP RN4020 49](#_Toc46821941)

[3.4 SOFTWARE 50](#_Toc46821943)

[3.4.1 GESTURE RECOGNITION 50](#_Toc46821944)

[3.4.2 PROGRAMMING LANGUAGES 51](#_Toc46821945)

[3.4.2.1 PROGRAMMING ENVIRONMENTS 51](#_Toc46821946)

[3.4.2.2 VERSION CONTROL SYSTEMS 53](#_Toc46821947)

[3.4.3 MACHINE LEARNING 54](#_Toc46821948)

[3.4.3.1 ARTIFICIAL INTELLIGENCE 54](#_Toc46821949)

[3.4.3.2 MACHINE LEARNING 54](#_Toc46821950)

[3.4.3.3 FINE-TUNING 55](#_Toc46821951)

[3.4.4 USER INTERFACE 56](#_Toc46821952)

[3.4.4.1 MOBILE PLATFORMS 57](#_Toc46821953)

[3.4.4.1.1 ANDROID 57](#_Toc46821954)

[3.4.4.1.2 iOS 58](#_Toc46821955)

[3.4.4.2 DESKTOP APPLICATION 58](#_Toc46821956)

[4. CONSTRAINTS 60](#_Toc46821957)

[4.1 DESIGN CONSTRAINTS 60](#_Toc46821958)

[4.1.1 ETHICAL AND SOCIAL 60](#_Toc46821959)

[4.1.2 MANUFACTURABILITY AND SUSTAINABILITY 60](#_Toc46821960)

[4.1.3 ECONOMICS AND TIME 62](#_Toc46821961)

[4.1.4 ENVIRONMENTAL AND HEALTH 62](#_Toc46821962)

[4.2 STANDARDS 63](#_Toc46821963)

[4.2.1 BLUETOOTH 63](#_Toc46821964)

[4.2.2 LITHIUM-ION BATTERIES 63](#_Toc46821965)

[4.2.3 SOFTWARE STANDARDS 64](#_Toc46821966)

[5. DESIGN 65](#_Toc46821967)

[5.1 HARDWARE DESIGN 65](#_Toc46821968)

[5.1.1 FLEX SENSORS 65](#_Toc46821969)

[5.1.1.1 SELECTION AND INTEGRATION 65](#_Toc46821970)

[5.1.2 ACCELEROMETER AND GYROSCOPE 66](#_Toc46821971)

[5.1.2.1 SELECTION AND INTEGRATION 66](#_Toc46821972)

[5.1.3 FORCE SENSOR 67](#_Toc46821973)

[5.1.3.1 SELECTION AND INTEGRATION 67](#_Toc46821974)

[5.1.5 PCB 69](#_Toc46821975)

[5.1.5.1 DESIGN 69](#_Toc46821976)

[5.1.6 MICROCONTROLLER UNIT 69](#_Toc46821977)

[5.1.6.1 SELECTION AND INTEGRATION 69](#_Toc46821978)

[5.1.7 POWER SOURCE 69](#_Toc46821979)

[5.1.7.1 BATTERY 69](#_Toc46821980)

[5.1.7.2 CHARGER 70](#_Toc46821981)

[5.1.8 BLUETOOTH MODULE 71](#_Toc46821982)

[5.1.8.1 BLUETOOTH GENERIC ACCESS PROFILE 72](#_Toc46821983)

[5.1.8.2 BLUETOOTH GENERIC ATTRIBUTE PROFILE 73](#_Toc46821984)

[5.1.9 VOLTAGE REGULATOR 74](#_Toc46821985)

[5.1.9.1 SELECTION 74](#_Toc46821986)

[5.1.10 GLOVE 75](#_Toc46821987)

[5.1.10.1 SELECTION 75](#_Toc46821988)

[5.1.10.2 INTEGRATION 75](#_Toc46821989)

[5.2 BILL OF MATERIALS 76](#_Toc46821990)

[5.3 SOFTWARE DESIGN 77](#_Toc46821992)

[5.3.1 SOFTWARE PROTOTYPE 78](#_Toc46821993)

[5.3.2 PROGRAMMING LANGUAGE 79](#_Toc46821994)

[5.3.3 CODING ENVIRONMENT 79](#_Toc46821995)

[5.3.4 MACHINE LEARNING 80](#_Toc46821996)

[5.3.4.1 SUPPORT VECTOR MACHINE 82](#_Toc46821997)

[5.3.4.2 NEURAL NETWORKS 84](#_Toc46821998)

[5.3.5 GRAPHICAL USER INTERFACE 85](#_Toc46821999)

[5.4 HAND GESTURE RECOGNITION 86](#_Toc46822000)

[5.4.1 INTERPRETING DATA FROM FLEX SENSOR 86](#_Toc46822001)

[5.4.3 CALIBRATION 88](#_Toc46822002)

[5.4.4 DIFFERENTIATION BETWEEN SIMILAR HAND GESTURES 89](#_Toc46822003)

[6. CONSTRUCTION 91](#_Toc46822004)

[6.1 TESTING 91](#_Toc46822005)

[6.1.1 HARDWARE TESTING 91](#_Toc46822006)

[6.1.1.1 FLEX SENSORS 91](#_Toc46822007)

[6.1.1.2 ACCELEROMETER AND GYROSCOPE 92](#_Toc46822008)

[6.1.1.3 FORCE SENSOR 92](#_Toc46822009)

[6.1.2 SOFTWARE TESTING 92](#_Toc46822010)

[6.1.2.1 MACHINE LEARNING 93](#_Toc46822011)

[6.1.2.3 USER INTERFACE 94](#_Toc46822012)

[6.1.3 PASS/FAIL CRITERIA 94](#_Toc46822013)

[6.2 FACILITIES 94](#_Toc46822014)

[6.3 SUPPLIERS 95](#_Toc46822015)

[7. ADMINISTRATIVE CONTENT 97](#_Toc46822016)

[7.1 PROJECT MILESTONES 97](#_Toc46822017)

[7.2 BUDGET 98](#_Toc46822018)

[Adafruit 328 Battery, Lithium Ion Polymer, 3.7V, 2500mAh 99](#_Toc46822019)

[7.3 DIVISION OF LABOR 100](#_Toc46822020)

[7.4 PROFILES 100](#_Toc46822021)

[APPENDIX A: COPYRIGHT PERMISSION 102](#_Toc46822022)

[APPENDIX B: REFERENCES 112](#_Toc46822024)

[APPENDIX C: DATASHEETS 116](#_Toc46822025)

# 1. EXECUTIVE SUMMARY

American Sign Language (ASL) has become one of the most popular language classes in North America and is claimed to be the third most commonly used language in the United States. Around 9 out of 10 children who are born deaf are offspring to parents who can hear and have had limited to no experience in sign language and communicating with deaf people.

There are many online resources to learn ASL that typically use recorded lessons and videos. Although these methods have proven to be successful, one challenge in learning sign language online is not having immediate interactive feedback to immerse oneself in the language inside and outside of the “classroom.” This can greatly limit how fast or slowly one can gain fluency. Therefore, our project aims to create a sign language tool that can break down the language barrier between the deaf community and people who do not sign while also being a learning tool that is more interactive for non-signers and aspiring learners compared to today’s conventional online methods. By doing this, our glove makes learning sign language more accessible to people not in the deaf or hard-of-hearing community.

While there have been gloves that have been created in the past to translate Sign Language to speech or text for the deaf or hard-hearing community, our project aims to create a glove that can be used by both signers and non-signers alike. Not only can signers use this glove to communicate with others, it can also be used as an interactive learning tool. Non-signers or people who are new to signing, will be able to get feedback on their gestures. This provides an immersive experience where users can potentially practice anywhere in areas where they don’t have the opportunity to practice with other signers. It is also more engaging compared to online learning resources that rely on videos. Some criticisms in the past for previous gloves have been that sign language gloves are insensitive to the deaf or hard-of-hearing community. Our glove is made for people who are hard-of-hearing, but to people who want to learn sign language to bridge the gap between the two communities.

The goal of the project is to design a glove embedded with sensors that can translate the hand gestures of the wearer to a written and computer voice, using a computer program. A fun, interactive program will be designed for beginning learners to create a more immersive experience. The components that will be needed are a microcontroller to communicate between the computer and the sensors, software that will take the data from the sensors and produce an output, and a glove that utilizes sensors to collect data to send back to the microcontroller. The glove will be lightweight and an average hand size. The entire system will be designed to be user friendly for all learning levels.

# 2. PROJECT DESCRIPTION

The BabelGlove is a glove, embedded with sensors, that can translate the hand gestures of the wearer to a written and computer voice, using a computer program. It will be lightweight and comfortable for the user to wear. A fun, interactive program will be designed for beginning learners to create a more immersive experience. The glove will be fitted with flex sensors and an accelerometer. A microcontroller and a Bluetooth module will be used to communicate between the sensors and the computer and software will take the data and produce an output on a desktop application. The entire system will be designed to be user friendly for all learning levels.

The objective of the project is to help people learn sign language. It can also be used to help people who don’t know ASL to communicate with sign language speakers. The data collected from the different sensors will be converted into computer generated text and voice for non ASL speakers to hear and read.

## 2.1 MOTIVATION

Our group’s motivation for choosing to create a glove that translates sign language hand gestures into text was that we wanted a project that people could use. While the idea of a glove that translates gestures into text is not new or novel, our glove is geared more towards people who are looking to learn ASL rather than people who have speech impediments. We wanted to create a glove that would be beneficial to society. Another motivation for choosing this project was that it would allow us the opportunity to learn more about areas and components, such as sensors, microcontrollers, wireless communications, and machine learning. We thought that this project had a good mix of learning areas that would equip us with valuable skills outside of college.

## 2.2 GOALS AND OBJECTIVES

The simplified version of our group’s goal is to create a system to translate American Sign Language to English Text and Speech. As we develop and improve on this system, we will implement a learning module, where users can become students and study American Sign Language through our BabelGlove. There will be specialized software to guide and train inexperienced users in the complete and natural language of ASL. The BabelGlove system will give our users a unique experience and training service other ASL training processes cannot achieve. As we finalize and enhance this system, we hope to help many deaf and deaf allies’ users as they learn American Sign Language. We also hope that a company or corporation will fund this system to produce more products with more powerful and capable parts, to continue helping people learn the subtle hand movement in American Sign Language.

## 2.3 REQUIREMENTS SPECIFICATIONS

The Requirement Specifications can be split into two categories of requirements, the General Requirements as shown in Figure 2.3-1 and the Functional Requirements and Specifications that are shown in Figure 2.3.2. In this section the two categories and their requirements are explained thoroughly.

|  |
| --- |
| **General Requirements** |
| The glove must be flexible for the user to complete the ASL gestures. |
| Each finger in the glove must have a flex sensor. |
| On the glove, there must be an accelerometer, a MCU, a power supply, and a bluetooth module. |
| The glove must collect the data from the various sensors and send them to a computer to process. |
| The computer connected to the glove must process the sensor data and produce output through a user interface. |

**Figure 2.3-1: Table of General Requirements**

When starting any major project, it is important to find and understand the General Requirements. With these requirements we can gain more insight into the basics of our BabelGlove system and we can build upon these to create more specific functional requirements. Figure 2.3-1 lays out our general requirements in an easy to read table. One of our most important requirements is that the glove must be flexible of the user to complete their ASL gestures. Without the user being able to move their hands in the subtle movement of ASL, our glove system would not be able to find the letters and phrases to translate them properly. In order to allow the glove to be flexible, it must be lightweight. With a heavy glove, it will limit the hand movements of the user and will be uncomfortable for the user to wear. For the glove to sense each finger movement when the user is signing ASL, there needs to be a specialized flex sensor running up the glove in each finger hole. These requirements add to the accuracy of the glove and let us achieve pin-point accuracy of the location and position of the user’s fingers. The accelerometer and gyroscope will help track the movement of the hand. In order to power the glove system there must be a central control unit or MCU to connect the sensors, and power supply together. The PCB designed for the glove will need to be small and lightweight. This cannot interrupt the hand movement and will need to be placed on the glove in a location where it will not be damaged and be easily connected to the appropriate equipment. This can also be located off the glove if needed.

We must have a sustainable power supply also to power the system. There also must be a strong and reliable Bluetooth module to connect the glove to the computer to send the sensor data to read and translate the user’s ASL. Using the Bluetooth module, the MCU and glove must send the data from the sensors to the user’s computer to translate, which is one of the most important parts of the system. The faster and more reliable the connection and data rate is, the better the experience for the user and the more time the computer must translate the sensor’s data. Then once the data is sent to the computer, there must be efficient software to translate the sensor data to English text and English speech. The faster and more effective the translation is the better experience the user will have using the BabelGlove.

There are many limitations and constraints to consider for this project. The glove needs to be lightweight and portable. This means that the parts selected should be small and should not weigh much. Also, a lot of data will be collected for the sensors. The software used will need to be able to understand the data inputted and display the correct response. It will be difficult to make sure that the software is able to distinguish each letter or phrase since many are similar to each other and differ in minor ways. It will also be difficult since not everyone does the same exact gestures. Some people may not bend the fingers as much or may move the hand more. The software needs to be able to accommodate these differences.

|  |
| --- |
| **Functional Requirements and Specifications** |
| The glove system must recognize all 26 letters in the ASL Alphabet, including 3 or 4 simple phrases. |
| The glove must be able to fit the average hand, approximately 7.3 inches and weigh under 3 pounds. |
| The glove’s microcontroller must be able to sample data every 1 millisecond. |
| The Flex Sensors must have a flat resistance of 25k Ohms, a resistance tolerance of +/- 30% and a bend resistance range of 45k to 125k Ohms. |
| The power rating of the flex sensor must be continuous at 0.5 Watts with a peak of 1 Watt. |
| The glove must have a battery life of at least 2 hours from full utilization. |
| The machine learning neural network must have at least 85% accuracy. |

**Figure 2.3-2: Table of Functional Requirements and Specifications**

Our Functional Requirements and Specification, shown in Figure 2.3-2, specify exact requirements for the BabelGlove. These requirements are built from the general requirements and are what the BabelGlove must achieve as far as performance and efficiency. The glove system must recognize all the 26 letters in the ASL Alphabet (shown in figure 3.3-1) including three or four simple phrases chosen by our group, such as “How was your day?” or “Where is the bathroom?” The gestures for the alphabet and phrases will be stored in the database library. Once the library is created, the system should be able to quickly and efficiently detect the hand gestures stored in the library. One of our leading objectives in designing and building the glove is ease of use, the glove weight and hand size are major factors in making sure the user can easily sign in. We will keep the glove under three pounds and have a glove size of approximately 7.3 inches to accommodate the average hand size. We need the glove system to be fast to keep with the user signing, therefore we are requiring the glove’s microcontroller to be able to sample data every 1 millisecond. We believe the parts we choose and implement into our design will have a great balance of price to performance.

The Flex Sensors we add to the glove must have the following technical requirements: have a flat resistance of 10k Ohms, a resistance tolerance of +/- 30% and a bend resistance range of 45k to 125k Ohms. This ensures the flex sensors can gather the correct data we need to translate the ASL from the user. A major concern in our system is power, we need to save power consumption wherever we can. We have limited the power rating of the flex sensor to be continuous at 0.5 Watts with a peak of 1 Watt. Also, the glove must have a battery life of at least two hours from full operation from the user. We as a group decided to give the machine learning neural network a base accuracy of 85%. The machine learning process and algorithm and get implemented in the glove system, must reach 85 percent accuracy to be good enough for the users to operate. It is possible that some functional requirements can be added as needed.

## 2.4 QUALITY HOUSE OF ANALYSIS

Shown below in Figure 2.4-1 is our House of Quality chart which identifies our targeted demographic’s needs and correlates it to an engineering requirement. This chart helps us prioritize certain features of our project. In relation to the house of cards figure below:

↑ = Positive Correlation

↑↑ = Strong Positive Correlation

↓ = Negative Correlation

↓↓ = Strong Negative Correlation

+ = Increases the Requirements

- = Decreases the Requirements



**Figure 2.4-1: House of Quality**

# 

# 3. RESEARCH

## 3.1 EXISTING PROJECTS AND DIFFERENCES

In our research, we have found that there were a lot of different projects with a similar idea of creating a glove that translates gestures into text. We went through a couple of projects and their documentation to see points on which we can improve upon and issues that we may be able to avoid based on past group failures. While our project has a slightly different purpose than other similar glove projects, we will have to make similar choices for the different technologies we will be using in our project. Researching existing projects will help us be able to differentiate our project from past projects and help us avoid mistakes that will increase our budget. This section will discuss previous projects and will include differentiating factors along with points on what our group can do to improve our project during the design and development process.

### 3.1.1 SLIG: SIGN LANGUAGE INTERPRETATION GLOVE

SLIG was a University of Central Florida ECE senior design project created in Spring/Summer 2016. SLIG was a lightweight, thin glove that translates sign language hand gestures into text. The glove that SLIG created was intended to be used for people who have speech impediments. SLIG used a combination of flex sensors, an accelerometer, and contact sensors to capture the hand position and orientation. The sensors were connected to an MCU which communicated to a mobile application through Bluetooth.

The algorithm that they used included conditional statements that mapped gestures to their respective letters. In their project documentation, they noted that they used conditional statements instead of machine learning, another technology that could be used to create the language database, because it would take too much time to train the glove and that the MCU did not have enough processing power or storage for those tasks. Our group plans to use machine learning in order to make our project scalable if we wanted to add more phrases. We will be doing the algorithm training on the computer, so SLIG’s issues of not having enough processing power or memory on the MCU should not be a problem.

An issue that the SLIG group encountered was choosing a mobile platform for their application to display the feedback. They ended up choosing to create an Android application since their team had access to Android devices. Our team does not already have Android devices so that is something that our group will have to consider when deciding on a platform to use for the application for the user interface. A large reason why they didn’t choose to develop an app for iOS was because they stated that Apple only supported their IDE on Mac OS, which is something else our group must consider.

In their documentation, SLIG’s initial project budget accounted for nearly $550 worth in expenses but ended up being around $825 at the end of the project because of unaccounted for materials and components. We will make sure to work these miscellaneous costs into our budget so that the costs won’t surprise us.

### 3.1.2 SIGNALOUD

In 2016, University of Washington students created the SignAloud glove. The glove captured ASL gestures with sensors similar to the ones in the SLIG project. The signals from the sensor data are sent via Bluetooth to a central computer and translated using neural networks. In their project, they emphasized that most of the sign language translation devices already out there were not practical for everyday use. An important note that came from their project was that they said most sign language gloves don’t take into account the natural syntax of ASL. Similar to how different spoken languages have different sentence structuring rules, the same can be said for ASL.

### 3.1.3 ASLA: AMERICAN SIGN LANGUAGE ASSISTANT

ASLA is another UCF ECE Senior Design project that was created in Spring/Fall 2016. The group had a lot of similarities to SLIG, regarding the sensors that they used and how the hardware interfaced with the software, but a key difference between ASLA and SLIG was that ASLA used machine learning instead of conditional statements to recognize gestures being made with the glove.

Their glove was able to be trained and the phrase data was saved to a computer in a CSV file. The user had the ability to expand the library of phrases beyond the original twenty-six letters of the alphabet that does not require movement.

The biggest differences between this project and the previous projects was that they implemented a text to speech feature that used an API to read aloud the translated gestures. Because our project will be more focused on educating non-signers, we won’t be including text-to-speech in our project.

Another feature that they implemented that they would be using two gloves instead of one glove so that sentences can be formed. We will also be using two gloves so that sentences and simple phrases can be formed. ASLA’s glove ended up costing around $600.

### 3.1.4 DIFFERENCES

There were many other projects like SLIG, SignAloud, and ours. To differentiate our glove to the other gloves that have been created in the past is that we are creating the glove with the intention that the glove will be used by non-signers to learn ASL. Along with that, for the gesture recognition, many projects used algorithms that mapped to a dictionary of known letters or words. Using these types of hard-coded algorithms aren’t very scalable, so we will be using machine learning for the purpose of scalability.

Both of the UCF previous projects cost over $550, while our glove aims to be less than $450 to reach a lower price point in the market.

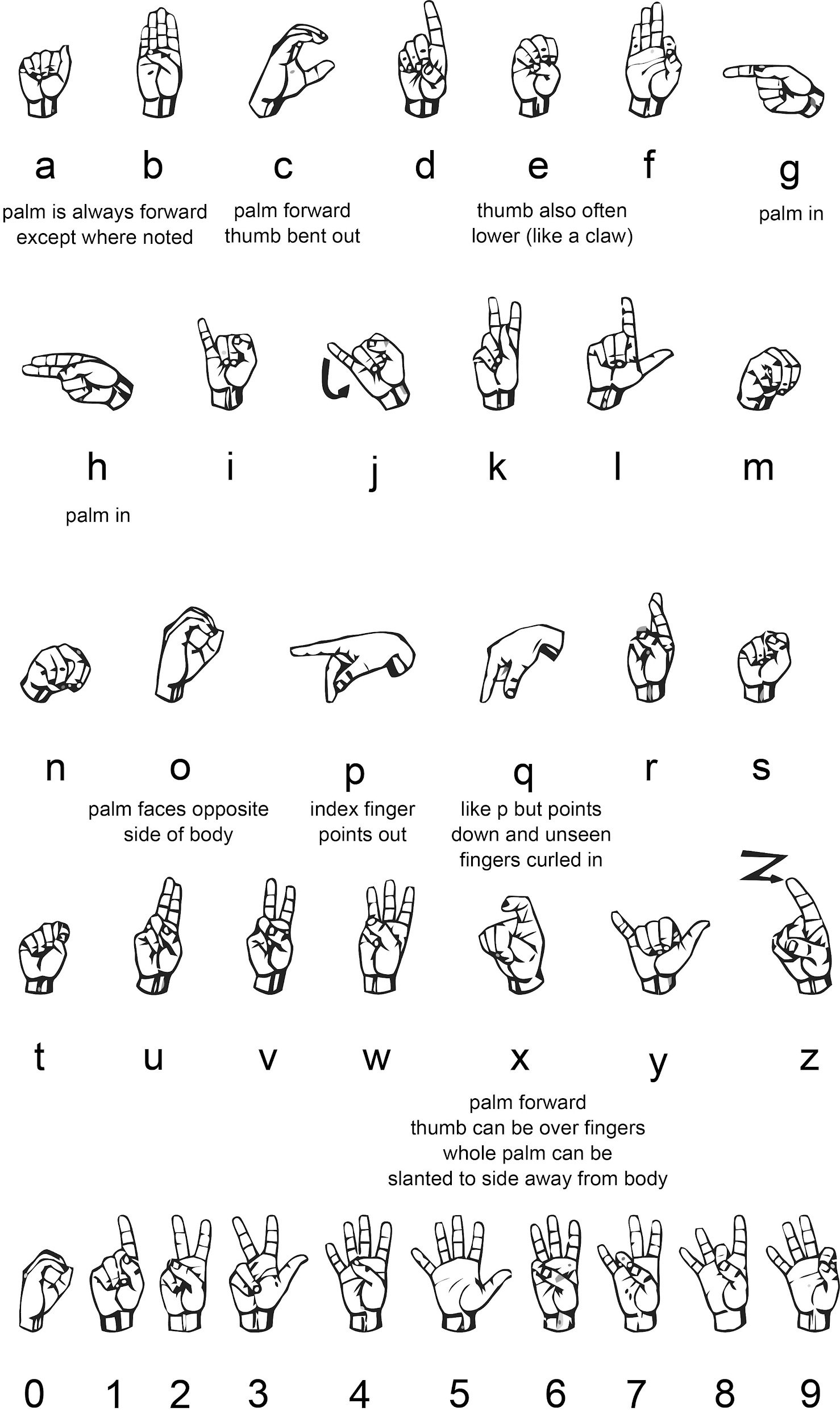
## 3.2 ETHICAL ISSUES

During our research, we came across an article that was published in The Atlantic titled “Why Sign Language Gloves Don’t Help Deaf People”. In the article, a major concern about these types of projects that involve a glove translating gestures into sign language were that they didn’t make it any easier for deaf people, only easier for people who aren’t in the community. The main idea of the article was that ASL gloves were mainly created to benefit hearing people.

Because of these concerns from the community, we aim to make our glove sensitive to the concerns of the deaf community and made our glove with the intention of making learning American Sign Language more accessible to people outside of the deaf community. Our hope is that eventually, the glove will get a learning hearing person to abandon the glove and be able to speak a few phrases or use the alphabet to communicate with deaf or hard of hearing people.

## 3.3 AMERICAN SIGN LANGUAGE

American Sign Language is a visual-manual language that around 500,000 deaf people in the US and Canada communicate with. This is used as the primary form of communication mostly by people who are hard of hearing or completely deaf in the United States and Canada. ASL consists of individual letters in the English alphabet and words and phrases used by English speakers. There is no singular universal sign language, so we have chosen to base our project off of American Sign Language. ASL has its own structure which isn’t the same as American English. For example, commonly used in ASL is topicalization, which is the process of establishing a topic at the beginning of a sentence. For example, in English, if we were to say, “He sells food,” a person who signs might translate it as “Food, he sells.” Since our project’s goal is to be able to translate the ASL alphabet and 1-5 phrases, the sentence structuring is not as important, but it is good to take note of when scaling our project for more practical uses. Figure 3.3-1 below shows the complete ASL Alphabet and numbers 0 thru 9, showing the complexity of some of the hand movements. There are some similarities between some of the alphabet letters with similar finger positions, so we will consider that when working on increasing the accuracy of the glove.

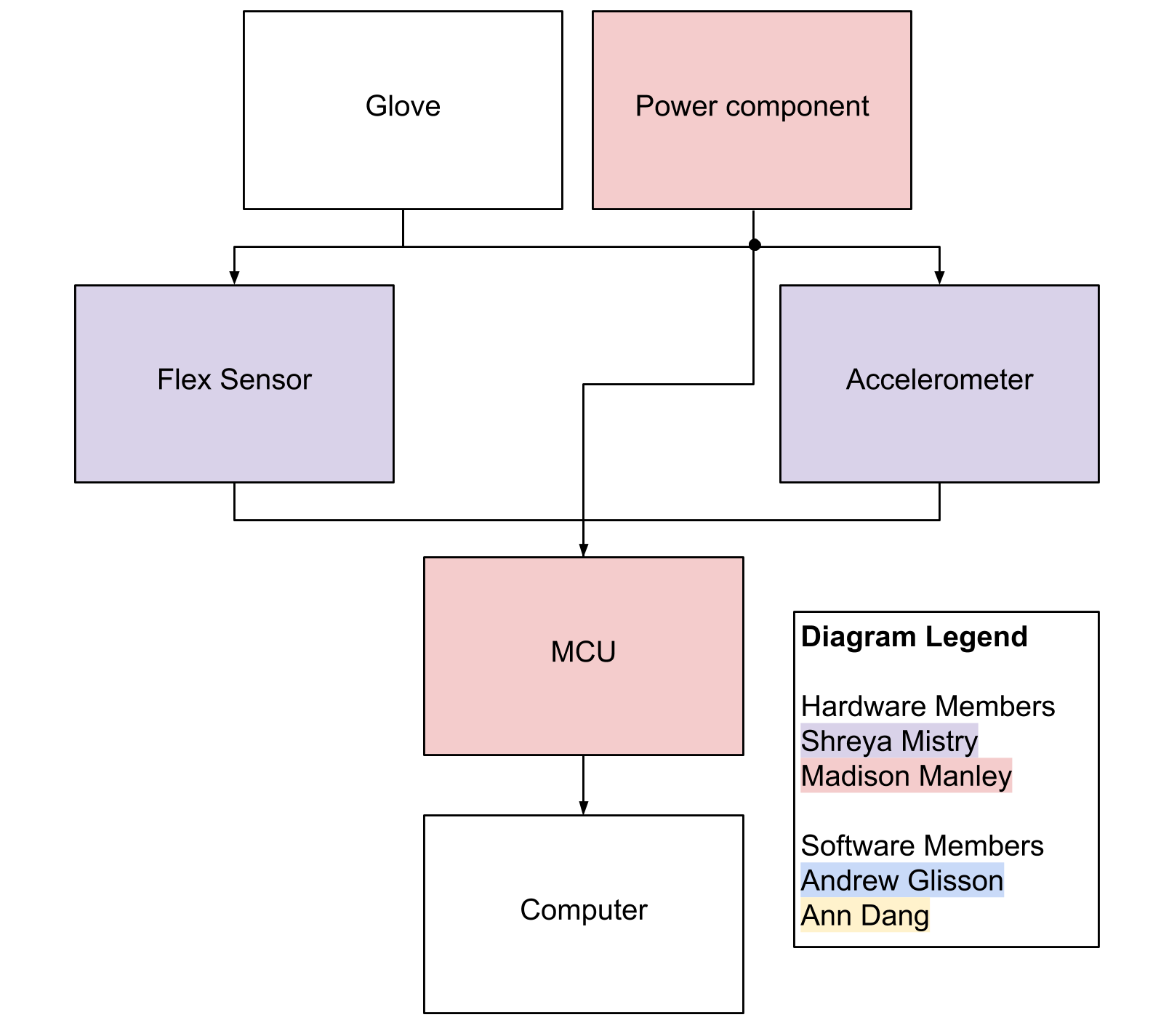


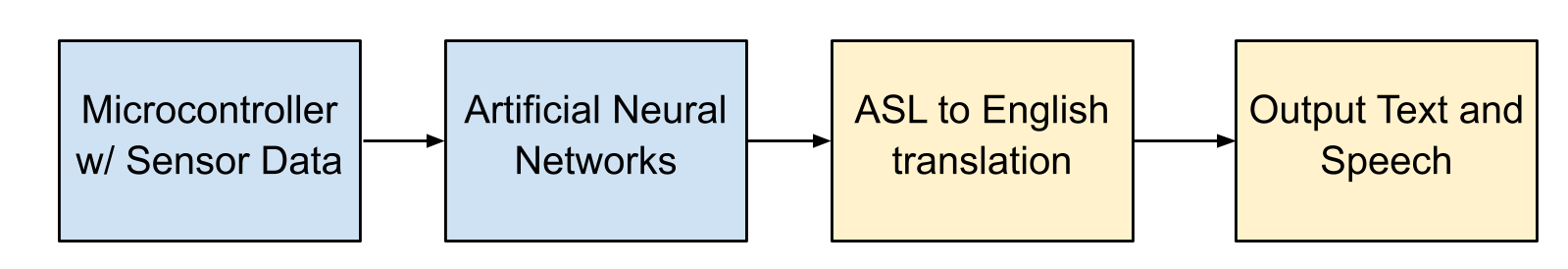
**Figure 3.3-1: American Sign Language Alphabet and Numbers. Reprinted with permission from National Institute on Deafness and Other Communication Disorders (NIDCD).**

## 

## 3.4 BLOCK DIAGRAM

Shown below in Figure 3.2.1-1 and Figure 3.2.1-2 are the block diagrams for the hardware and software components, respectively. The sensors and accelerometer are connected to the MCU which connects to the computer using bluetooth.

  
**Figure 3.2.1-1: Hardware Block Diagram**



**Figure 3.2.1-2: Software Block Diagram**

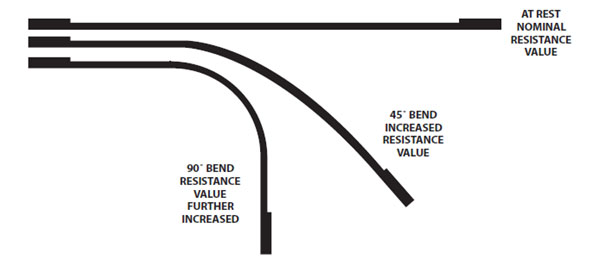
## 3.3 HARDWARE

### 3.3.1 FLEX SENSOR

In this section, different types of flex sensors are discussed and compared to determine which flex sensor was chosen in the execution of the Babel glove. The first section will provide a brief introduction on flex sensors and important considerations for the flex sensors pertaining to the Babel glove specifically will be discussed. Finally, different flex sensors will be reviewed to determine if they meet the requirements of the project.

#### 3.3.1.1 INTRODUCTION

The flex sensor will be the primary element to collect data from the glove. A flex sensor is a device that relates how much the element bends to a change in resistance. One flex sensor will be placed on each finger component of the glove and as the user bends their fingers, the flex sensor will read the change in resistance and send this data to the microcontroller. The greater the bend, the higher the resistance value. The different combinations in the change of resistance will be used to identify each letter from the alphabet the user is signing. Flex sensors can be either unidirectional or bidirectional. A unidirectional flex sensor will change resistance as it is bent in one direction, while a bidirectional flex sensor will measure the change in resistance at both directions. For this design, a unidirectional flex sensor will be used since it is not necessary to measure the finger bending in both directions. Figure 3.3.1.1-1 below shows the different bend resistance in a flex sensor.



**Figure 3.3.1.1-1: The Bend Resistance in a Flex Sensor**

There are three main types of flex sensors. The first is an optical flex sensor. This sensor consists of a flexible tubing with a reflective interior wall. At one end of the tube, a light source is placed and at the other end, a photosensitive detector is placed. This is used to detect the different combinations of light as the flexible tube is bent. [1] Some properties of a optical flex sensor can be seen below:

* Most common
* High repeatability
* Bend in any direction
* Usually unipolar
* Expensive

The second type of flex sensor is the conductive ink based flex sensor. A strip of resistive ink is placed on a flexible plastic substrate. At rest, the resistance is measured as an intrinsic resistance. When it bends, the resistive materials inside are stretched apart, causing fewer adjacent resistive particles to come into contact with one another and this increases the resistance. The resistance changes as there is deflection and this creates a flexible potentiometer. The range of resistance is typically between 10 kΩ and 50 kΩ. Most of these sensors are unidirectional, however, placing two back-to-back will create a bidirectional sensor. Some properties of a conductive ink based flex sensor are seen below:

* Noise neglectable
* Resistance is a function of curvature
* High temperature tolerance
* Low cost
* customizable [7]

The third type of flex sensor is a conductive flex sensor. It can be fabric, thread, or polymer based. It is made by placing a layer of resistive material in between two pieces of conductive material. As pressure is applied to it, the conductive material gets closer and the resistance decreases. It is very similar to a force sensor. Some properties of a conductive flex sensor are seen below:

* Pressure-sensing
* Slow response
* Poor accuracy and repeatability
* High temperature tolerance
* Customizable
* Cheap

#### 3.3.1.2 CONSIDERATIONS

On the market today, there are many different types of flex sensors available. For this project, there are several different specifications that should be met. First, the flat resistance should be 25 kΩ. This means that when the sensor is flat, the resistance should measure to be about 25 kΩ. The resistance tolerance should be plus or minus thirty percent. The bend resistance range should be between 45 kΩ and 100 kΩ. The power rating should be 1 watt peak with a .5 watt continuous rating. There are several different flex sensors that meet these specifications.

#### 3.3.1.3 BI-DIRECTIONAL FLEXIBLE BEND SENSORS

This flex sensor is from Images Scientific Instruments. It is di-directional, which means that it can bend in two directions instead of one and the resistance will gradually decrease as the sensor is bent in either direction. This sensor is 4.5 inches long, with a width of .375 inches and a thickness of .038 inches. It is available in three different resistor ranges. The first range is the low range with a nominal resistance between 1 kΩ and 20 kΩ. The medium resistance range is between 20 kΩ and 50 kΩ. The high resistance range has a nominal resistance between 50 kΩ and 200 kΩ. It has a flat resistance of 10 kΩ. This sensor is also pressure sensitive, so it can be used as a force or pressure sensor as well. This could prevent the need to purchase a force sensor, which will be discussed later on in the paper. Some applications of this sensor include being used in VR gloves.

Some advantages of this sensor is that it can also work as a force sensor. Although Images Scientific Instruments is not as well known as other vendors, it has been around for over thirty years and this company sells many other products as well. Although there are many advantages to selecting this product, it will not be chosen for the design. This is a bidirectional sensor and for this project, only a unidirectional sensor is needed.

#### 3.3.1.4 SPECTRA SYMBOL 4.5” FLEX SENSOR

This flex sensor is a simple flex sensor that is 4.5 inches long. As the sensor is bent, the resistance increases. The technology is by Spectra Symbol. This is a unidirectional flex sensor that meets three of the four specifications. It has a flat resistance of 10 kΩ, a resistance tolerance of plus to minus thirty percent, and a power rating of .5 watts continuous and 1 watt peak. The only specification that does not match is the ben resistance range. The bend resistance range for this flex sensor is 60 kΩ to 110 kΩ, which is higher then what the specifications for this project is. This sensor can be connected to an analog input to the microcontroller or it can be connected to a digital input if a 0.1 uF capacitor is used.

Some advantages of using this sensor is that it is available from many different vendors and reasonably priced. These sensors are commonly used and come from a well known supplier. This is the product that has been picked to be used in the design of the BabelGlove.

#### 3.3.1.5 ONE-DIRECTIONAL FLEXIBLE BEND SENSORS

This is a one directional flex sensor by Images Scientific Instruments. It has two different styles. Style A has a nominal resistance of approximately 10 kΩ. As the sensor is bent in one direction, the resistance increases. The range of resistance for Style A is between 10 kΩ and 40 kΩ. Style B has a nominal resistance that is infinitely high. As the flex sensor is bent, the resistance decreases. The range of resistance of Style B can go below 10 kΩ depending on the degree that the sensor is bent.

Some advantages are that it has two different options for flex sensors. Its applications also include being used in a VR glove. This is another sensor by Images Scientific Instruments. However, this flex sensor will not be used in the design.

#### 3.3.1.6 LITTLEBITS BEND SENSOR

This sensor is by the littleBits brand. It sends a signal when the strip is bent. This sensor has been used in many projects. It also can come in a kit with other components. The products by the littleBits brand are designed to be used in independent student projects.

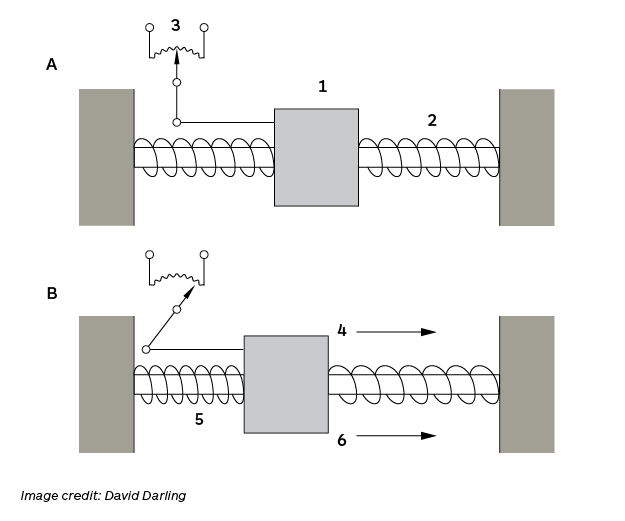
Some advantages to this sensor is that it has been used in many different projects reliably. It is compatible with other littleBits products. This sensor will not be used in the design due to its simplicity.

### 3.3.2 ACCELEROMETER AND GYROSCOPE

In this section, different types of accelerometers are discussed and compared to determine which accelerometer was chosen in the execution of the Babel glove. Gyroscopes are also discussed in terms of its functions and how it will be used alongside the accelerometer. The first section will provide a brief introduction on accelerometers and important considerations for the accelerometer pertaining to the Babel glove specifically will be discussed. Then, gyroscopes will be introduced. Finally, different accelerometer and gyroscope combinations will be reviewed to determine if they meet the requirements of the project.

#### 3.3.2.1 INTRODUCTION

The accelerometer is used to help improve the motions that are being sensed using acceleration. It will be able to detect other motion, such as moving the hand at different angles, by measuring the change in acceleration. Accelerometers are usually very sensitive and will be able to pick up small changes in acceleration. There are two types of forces it can measure. The first is static forces, which are constant forces. Dynamic forces are caused by movement. There are three main types of accelerometers.

****

**Figure 3.3.2.1-1: Accelerometer Function**

The first type is the capacitive MEMS accelerometer. This type of accelerometer has become increasingly popular in everyday life. It can be seen in various different technology applications, such as in iPhones and Wiis. MEMS accelerometers have a low cost and small size. This type of accelerometer is best used to measure low-frequency vibration, motion, and steady-state acceleration. However, they have a bad signal to noise ratio, a small bandwidth, and are restricted to smaller acceleration levels.

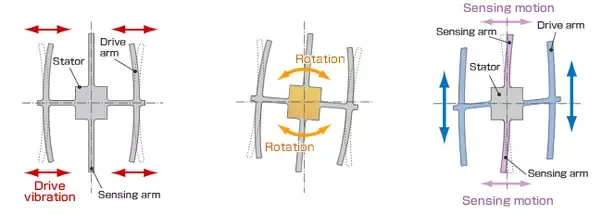
The second type of accelerometer is the piezoresistive accelerometer. Piezoresistive accelerometers are good for measuring high frequency shock events due to having a big bandwidth. Piezoresistive accelerometers measure to zero hertz. This is useful to accurately calculate velocity and displacement. Piezoresistive accelerometers are best used for impulse and impact measurements due to their high frequency range and amplitude. However, piezoresistive accelerometers have a low sensitivity, so it is difficult to measure vibrations accurately. It is also sensitive to temperature. Piezoresistive accelerometers are more expensive than MEMS accelerometers.

The last type of accelerometer is piezoelectric accelerometers. These are used most often in vibration measurement. They have a wide frequency response, good sensitivity, and they are easy to incorporate into a design. They are also available in a variety of sensitivities, weights, sizes, and shapes. Piezoelectric accelerometers have low noise levels. One disadvantage is the fact that they are AC coupled, so it is difficult to use when trying to get data for velocity and displacement. Below is a table that helps characterize when best to use each type of accelerometer.

|  |  |  |  |
| --- | --- | --- | --- |
| Application | Capacitive MEMS | Piezoresistive | Piezoelectric |
| Static Acceleration | ✓ | ✓ |  |
| G-Force | ✓ | ✓ |  |
| Seismic |  |  | ✓ |
| Low Frequency Vibration | ✓ | ✓ | ✓ |
| General Vibration | ✓ |  | ✓ |
| High Frequency Vibration |  |  | ✓ |
| General Shock | ✓ | ✓ | ✓ |
| High Impact Shock |  | ✓ | ✓ |
| Extreme Shock |  | ✓ | ✓ |

**Figure 3.3.2.1-2: Comparison of Different Types of Accelerometers [5]**

A gyroscope is a device that will measure or maintain rotational motion, known as angular velocity. A triple axis gyroscope can measure rotation on the x, y, and z axis. These are small, inexpensive devices that, when combined with the accelerometer, will help improve the measurements. The data collected from these two sensors will be more accurate and help provide more information to distinguish each letter or phrase. Some specifications to consider about gyroscopes are range and sensitivity. As the sensitivity of a gyroscope increases, the range decreases. All of the values for the specifications can be seen on the data sheet for each gyroscope. [9]

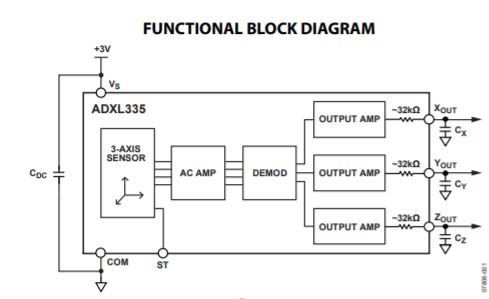


**Figure 3.3.2.1-3: Gyroscope Functions**

#### 3.3.2.2 SPARKFUN TRIPLE AXIS ACCELEROMETER BREAKOUT - ADXL335

This is a three axis accelerometer device. It is the newest model in this line of analog sensors. It has a power consumption of 302 uA. The sensor has a full sensing range of plus and minus 3g. There is no on-board regulation. The power provided should be between 1.8 and 3.6 V DC. This is due to it not any integrated voltage regulation. The board comes fully assembled and tested with external parts installed. This is the 0.1 uF capacitor. The bandwidth of each axis is set to 50 Hz. The size of the board is 0.7” by 0.7”.

There are many advantages to this breakout board. It has an extremely low noise and power consumption. It also comes fully assembled with the external components installed. Also, these types of accelerometers have been used in many projects over the years and are very reliable. There are many different accelerometers with different specifications from this supplier. All the appropriate documentation is included for both the board as a whole and the individual accelerometer component.

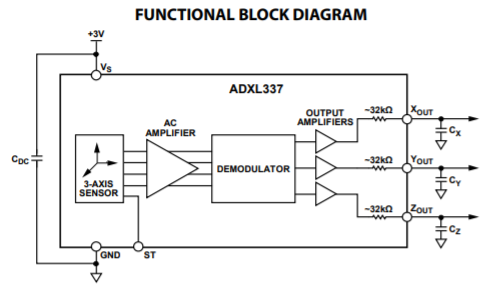


**Figure 3.3.2.2-1: Functional Block Diagram of ADXL335**

#### 3.3.2.3 SPARKFUN TRIPLE AXIS ACCELEROMETER BREAKOUT - ADXL337

The ADXL337 is a 3 axis accelerometer with a signal conditioned analog voltage output that measures acceleration. This is a three axis accelerometer device. It is the newest model in this line of analog sensors. It has a power consumption of 302 uA. The sensor has a full sensing range of plus and minus 3g. There is no on-board regulation. The power provided should be between 1.8 and 3.6 V DC. This is due to it not any integrated voltage regulation. The board comes fully assembled and tested with external parts installed. This is the 0.1 uF capacitor. The bandwidth of each axis is set to 50 Hz. The size of the board is 0.7” by 0.7”.

There are many advantages to this breakout board. It has an extremely low noise and power consumption. It also comes fully assembled with the external components installed. Also, these types of accelerometers have been used in many projects over the years and are very reliable. There are many different accelerometers with different specifications from this supplier. All the appropriate documentation is included for both the board as a whole and the individual accelerometer component.



**Figure 3.3.2.3-1: Functional Block Diagram of ADXL337**

#### 3.3.2.4 SPARKFUN 6 DEGREES OF FREEDOM BREAKOUT - LSM6DS3

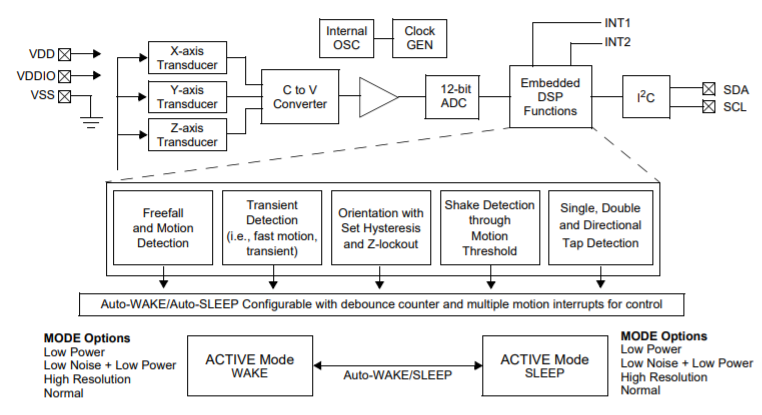
The LSM6DS3 is an accelerometer and a gyroscope. It has a 8 kb FIFO buffer and embedded processing interrupt functions. This can be used to detect shocks, tilt, motion, taps, count steps, and read temperature. It is capable of reading accelerometer data up to 6.7 kS/s and gyroscope data up to 1.7 kS/s for more accurate movement sensing. It can host other sensors and drive interrupt pins due to the built in FIFO. One side of the board features power and I2C functionality while the other side of the board has pins that control SPI functionality and interrupt outputs. If a voltage greater than 3.6 V is applied, it can permanently damage the IC.

There are several advantages for using this product. It contains both the accelerometer and the gyroscope. It is low cost and small. It has many capabilities. It is very flexible and can be configured for many different applications.

#### 3.3.2.5 SPARKFUN TRIPLE AXIS ACCELEROMETER BREAKOUT - MMA8452Q

This is a three axis MEMS accelerometer device. It has 12 bits of resolution. It has embedded functions with flexible user programmable options. It is configurable to two interrupt pins. The embedded interrupt functions allow for it to save power, relieving the host processor from continuously polling data. It has a user selectable full scales of plus and minus two, four, and eight. It has high pass filtered data as well as non filtered data available in real time. The device can be configured to generate inertial wake up interrupt signals from any combination of the configurable embedded functions. This allows the breakout board to monitor events and remain in a low power mode when it is inactive. The board breaks out the ground, power, I2C, and two external interrupt pins. It is also available with headers.

It has a 1.95 V to 3.6 V supply voltage requirement and an interface voltage of 1.6 V to 3.6 V. The output data rates from 1.56 to 800 HZ. There is a 12 bit and an 8 bit digital output choice. The I2C digital output interface operated to 2.25 MHz with a 4.7 kΩ pullip. There are two programmable interrupt pins for six interrupt sources. There are three embedded channels for motion detection and an orientation detection with set hysteresis. The orientation is portrait and landscape. There is a high pass filter that delivers data in real time. The current consumption is between 6 μA and 165 μA. There are many advantages to this breakout board. It is a smart, low-power board. It is user friendly. It also comes from a reliable source which has many other products.



**Figure 3.3.2.5-1: Block Diagram of MMA8452Q**

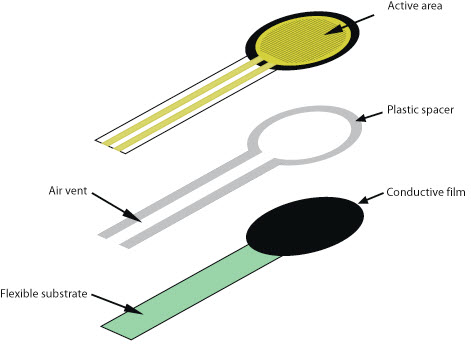
### 3.3.3 FORCE SENSOR

Since there are some letters and phrases in the ASL alphabet where the fingers are in contact with each other, a force or pressure sensor will be needed. This will also help when some signs are not distinguishable by data from the flex sensor and accelerometer. In this section, different types of force sensors are discussed and compared to determine which force sensor was chosen in the execution of the Babel glove. The first section will provide a brief introduction on force sensors and important considerations for the force sensors pertaining to the Babel glove specifically will be discussed. Finally, different forces will be reviewed to determine if they meet the requirements of the project.

#### 3.3.3.1 INTRODUCTION

A force sensor or force sensing resistor is a conductive polymer that will decrease in resistance as force is applied to the surface. [11] The resistance measured will vary depending on the amount of pressure applied to the sensor. The more pressure that is applied, the lower the resistance value measured. The output is produced as an electrical signal. Unlike a flex sensor, a force sensor does not measure the resistance as the sensor bends. It will only measure the resistance when pressure is applied to the active area. If there is no force being applied to the active area, the resistance will be large. The relationship between force and conductance is linear.

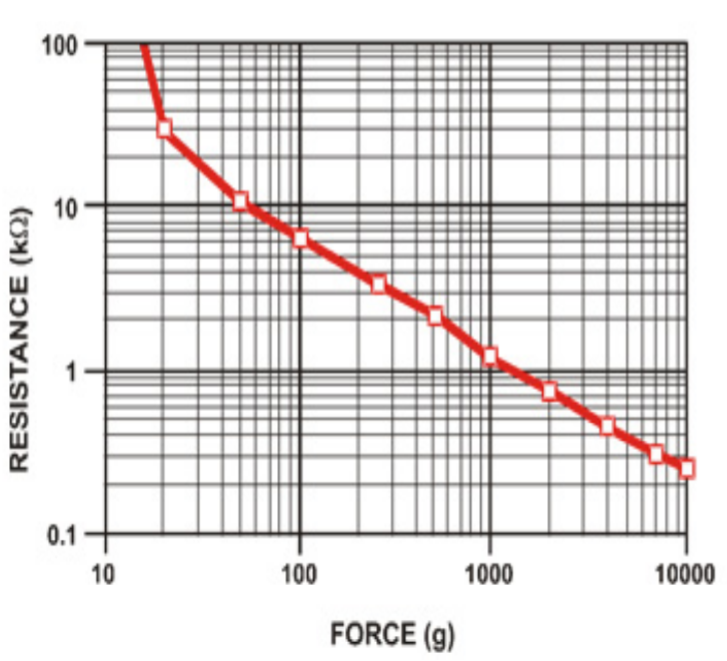
Force sensors are low cost and easy to use, but not very accurate. The response from a force sensor comes in a range. Also, if pressure is applied to a force sensor for an extended amount of time, it will be damaged. There are several benefits for using this type of sensor. For instance, it can be used with any type of board and can be customized for different specifications. By adding force sensors to our design, there will be less error when it comes to determining the letter of phrase that is being signed.



**Figure 3.3.3.1-1: Force Sensor**

#### 3.3.3.2 FORCE SENSITIVE RESISTOR - SMALL

This is a small force sensor from Interlink Electronics that has a 0.16 inch diameter. The overall width of the sensor is 0.28 inches. It will vary resistance based on the amount of pressure applied to the sensing area. The more force that is applied, the lower the resistance. Then no pressure is being applied, the resistance will be larger than 1 MΩ. When the maximum amount of pressure is applied, the resistance will be 2.5 kΩ. It is breadboard friendly with two pins extending from the bottom of the sensor with a 0.1 inch pitch. It has an actuation force as low as 2 grams.  
  
Some advantages are that it is simple to set up and good for pressure sensing. However, these types of sensors are not very accurate.



**Figure 3.3.3.2-1: Typical Force Curve of FSR 400 Series Round Force Sensing Resistor**

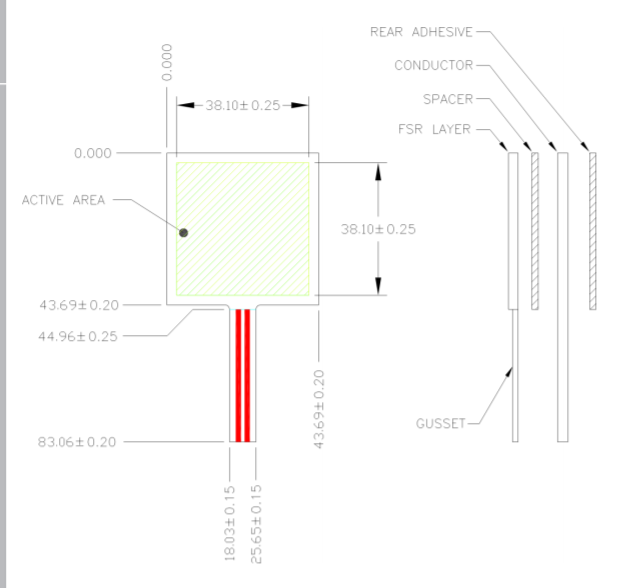
#### 3.3.3.3 FORCE SENSITIVE RESISTOR 0.5”

This is a force sensor that has a 0.5 inch diameter sensing area. It will vary resistance based on the amount of pressure applied to the sensing area. The more force that is applied, the lower the resistance. Then no pressure is being applied, the resistance will be larger than 1 MΩ. This sensor can sense force being applied in the range of 100g to 10kg. This sensor is breadboard friendly with two pins extending from the bottom of the sensor with a 0.1 inch pitch. There is a peel-and-stick rubber backing on the sensor to mount it. This sensor should not be soldered directly to the exposed silver traces. The solder joint will not hold and the substrates can easily melt and distort during soldering. It is recommended to use male or female clincher connectors instead.

Some advantages are that it is simple to set up and good for pressure sensing. However, these types of sensors are not very accurate.

#### 3.3.3.4 FORCE SENSITIVE RESISTOR - SQUARE

This is a square force sensor that has a 1.75 by 1.75 inch sensing area. It will vary resistance based on the amount of pressure applied to the sensing area. The more force that is applied, the lower the resistance. Then no pressure is being applied, the resistance will be larger than 1 MΩ. This sensor can sense force being applied between 100g to 10kg. It is breadboard friendly with two pins extending from the bottom of the sensor with a 0.1 inch pitch. There is also a peel-and-stick rubber back to mount the sensor.   
  
Some advantages are that it is simple to set up and good for pressure sensing. However, these types of sensors are not very accurate.



**Figure 3.3.3.4-1: Schematic Design of Square Force Sensor Resistor**

#### 3.3.3.5 PHIDGETS TOUCH SENSOR

This is a force sensor from Phidgets. This a capacitive touch sensor and can detect touch through plastic, glass, or paper. The recommended material thickness is 0.5 inches. The sensor can work in close proximity to other sensors. It can sense objects with a distance of up to 0.5 inches from the board in all directions without direct contact. On the bottom side of the sensor is a small exposed metallic pad. This can be used to make a soldered connection to increase the size and dimensions of the sensing area. This has a sensing chip that recalibrates 45 seconds after contact.

It has a current consumption of 500 uA maximum. The output impedance is 1 kΩ. The supply voltage has a range from 1.8 V DC minimum to 5.5 V DC maximum. It can be read by any Phidget with an analog input or VINT Hub port. Some advantages are the ability to increase the sensing area and that it can be read through other materials, such as the lining of a glove. It may not be as compatible with other products besides Phidgets.

### 3.3.4 MICROCONTROLLER UNIT (MCU)

In this section, various microcontrollers are presented and compared to determine which MCU was ultimately chosen in the execution of the Babel glove. The first section will provide a brief introduction on microcontrollers, then important considerations for the MCU pertaining to the Babel glove specifically will be discussed. Finally, different microcontrollers will be reviewed to determine if they meet the requirements of the project.

#### 3.3.4.1 INTRODUCTION

A microcontroller is essentially a computer on an integrated circuit (IC) chip. They usually consist of one or more central processing units (CPUs), program memory that can be in the form of random access memory (RAM) or read only memory (ROM), and peripherals, which are hardware modules such as input/output (I/O) pins, timing, and data converters. Microcontrollers are low-power devices that are usually small and low cost. They are dedicated to run one specific task and are used in embedded applications such as implantable medical devices and remote controls.

Because of their low cost, high level of integration, and composition of various components, the cost and size of the whole system is reduced. They are flexible in the sense that the user can reprogram its function and are easy to use. Many MCUs use assembly or high-level languages; for example, the MSP430 uses the C programming language. However, it is worth mentioning that even though there are many great advantages to using microcontrollers, there are also many limitations to be aware of. Microcontrollers are limited by the number of executions they can perform simultaneously, the speed at which they can process data, as well as their memory size.

#### 3.3.4.2 CONSIDERATIONS

When choosing an appropriate microcontroller for the Babel Glove, there are many basic considerations that should be accounted for. Because this project is using a mix of flex sensors, accelerometers, and pressure sensors, the MCU needs to at least have six analog input pins that each have an analog to digital converter (ADC). The power consumption of the processor should also be considered as it is desirable for the glove to employ only one battery to power the whole system. It would then be desirable to choose a unit that doesn’t consume a large amount of power.

Furthermore, because the Babel glove is processing data from flex sensors, accelerometers, gyroscopes, and pressure sensors and using a machine learning algorithm to turn the user’s gesture to text and speech, the data bus size and RAM of the unit need to be large enough to meet these needs. One consideration that is often overlooked is time. This project is to be completed in two semesters; therefore, it would be more convenient to consider units that come with their own programming environments and pre-written serial communication functions such as UART or I2C. Finally, cost will remain as the biggest consideration. With a tight budget, it is highly desirable to choose a microcontroller that meets the project’s basic needs while also being affordable.

In the next few sections, the advantages and disadvantages of Texas Instruments’ MSP430, Atmel’s ATMega32u4 and ATSAMD21G units will be presented. At the end, their specifications will be compared.

#### 3.3.4.3 MSP430

The MSP430 was created by Texas Instruments and is a 16-bit low power microcontroller. It can be programmed using Code Composer Studio, an environment by Texas Instruments. It has a low supply-voltage range from 1.8V to 3.6V and has ultra-low power consumption. The unit has a 16-Bit RISC Architecture and has five low-power modes, allowing it to have extended battery life in portable systems. The CPU includes 16 registers and a constant generator that help maximize code efficiency and reduce instruction execution time. The registers include a program counter, stack pointer, status register, constant generator, and general-purpose registers. There are six operating modes that can be configured by software, including one active mode and five low-power modes. Peripherals are connected to the CPU through data, address, and control buses and include modules such as ADCs, comparators, UART, and I2C. The MSP430 also utilizes a real time clock and three other basic clock module configurations.

One advantage that should be mentioned is that all the group members working on this project have experience in using the MSP430 with Code Composer Studio. Each member has taken the class “Embedded Systems” at the University of Central Florida and has completed the laboratory component that included experiments using this unit. Using this microcontroller can help save a considerable amount of time since all members are familiar with the environment, Code Composer Studio, compared to learning a whole new environment from scratch.

#### 3.3.4.4 ATMEGA32U4

The ATMEGA32 was created by Atmel and is a high performance, low power AVR 8-bit microcontroller that uses RISC Architecture and has 2.5KB of SRAM and 32 KB of flash memory. It has an operating voltage of 5V and recommended input voltage of 7-12V and operates at the speed of 16MHz. There are 40 pins total, 2 of which are used for power and two for oscillator, one for reset, and three for providing power and reference voltage to its ADC. The other 32 pins include 20 digital I/O channels and 12 analog I/O pins. There are multiple communication options such as two wire interface, USART, and serial peripheral interface. Other peripheral modules that are included are an analog comparator, external interrupts, and USB communication. To help reduce power consumption, there are multiple sleep modes that can shut down unused modules.

Although the ATMEGA32UA is an 8-bit microcontroller and not as powerful compared to others, such as the MSP430, it is very beginner friendly and easy-to-understand. It is a very popular microcontroller that is used in Arduino development boards and has a large community of users. This helps make the learning curve of this microcontroller low.

#### 3.3.4.5 ATSAMD21G

The ATSAND21G is another microcontroller developed by Atmel. It is a 32-bit low power microcontroller that has 4KB of SRAM and 32 KB of flash memory. The device operates at a maximum frequency of 48MHz and includes flexible and intelligent peripherals. It includes features such as an in-system programmable flash, twelve-channel direct memory access (DMA) controller, 24 digital I/O channels,14 analog I/O channels, and 16-bit Timer/Counters (TC) that can be configured to perform accurate program execution timing. There are multiple Serial Communication Modules (SERCOM) that can be configured to act as an UART, SPI, and I2C. Other peripheral modules include a 12-bit ADC with programmable gain and oversampling up to 16-bit resolution, as well as analog comparators.

The unit has an accurate and low-power external and internal oscillator that can be configured as the source for the system clock or run at different frequencies. Doing so will help reduce power consumption while also maintaining a high CPU frequency. There are also two sleep modes that can be selected in software, known as idle and standby. In the idle mode, all functions are running except for the CPU. In standby mode, all the clocks and functions are not running except for those selected. Finally, there is a SleepWalking feature that wakes up the CPU when needed.

#### 3.3.4.6 MSP430 VS ATMEGA32U4 VS ATSAMD21G

In this section, the three microcontrollers are compared with each other. The advantages and disadvantages are summarized below:

* The MSP430 only has 512B of RAM compared to the ATMEGA32UA which has 2.5KB of RAM and ATSAMD21G that has 4KB of RAM. As a machine learning algorithm is being used, a high amount of memory may be necessary to store information.
* The MSP430’s data bus is 16-bits while the ATMEGA32UA is only 8-bits. The ATSAMD21G’s data bus is 32-bits. Essentially, the wider the data bus, the more amount of data can be processed by the clock. Because many sensors are being used, it is important to have a wide enough data bus that can process all the incoming information.
* The MSP430 has a much lower power consumption compared to the ATMEGA32UA and ATSAMD21G.
* The MSP430 only has 8 analog I/O Channels while the ATMEGA32UA has 12 and the ATSAMD21G has 14. This can play a significant role depending on the total number of sensors used. If there are not enough channels, then ADC modules will have to be added outside of the microcontroller, thus increasing the size of the system.
* As stated earlier, the group is already familiar with Code Composer Studio, which is used by the MSP430. This can save significant time in terms of programming the microcontroller instead of having to the ATMEGA or ATSAMD21 environment from scratch.

|  |  |  |  |
| --- | --- | --- | --- |
| **SPECIFICATION** | **MSP430** | **ATMEGA32UA** | **ATSAMD21G** |
| Storage | 16 KB | 32 KB | 32 KB |
| RAM | 512 B | 2.5 KB | 4 KB |
| Data Bus | 16 bits | 8 bits | 32 bit |
| Speed | 16 MHz | 16 MHz | 48 MHz |
| Digital I/O Channels | 8 | 20 | 24 |
| Analog I/O Channels | 8 | 12 | 14 |
| Cost | $1.28 (TI.com) | $3.98 (Digikey) | $3.06 (Digikey) |

**Figure 3.3.4.6-1: Comparison of MSP430 vs. ATMEGA32UA vs. ATSAMD21G**

### 3.3.5 WIRELESS COMMUNICATION

This section introduces three different types of wireless communication systems that are being considered: Near Field Communication, ZigBee, Wi-Fi, and Bluetooth. The Babel Glove will be converting the hand gestures of the user into text, which will be shown on an external display. As the group is aiming for the glove to be as comfortable as possible for the user, the group wants to avoid having to use long cables and wires that are not only uncomfortable for the wearer, but also downright unaesthetic. Therefore, a wireless communication system is needed to help make the glove lightweight and portable.

#### 3.3.5.1 NEAR FIELD COMMUNICATION

Near Field Communication (NFC) is a method of wireless data transfer that lets devices communicate with each other using radio frequency (RF) field technology. It is quick and easy and lets two devices transfer data between the two by being within about 4 centimeters. This makes NFC especially popular for contactless payment systems such as Apple Pay and Google Wallet, which are also known as tap-and-go services. Because this technology requires close contact, the receiving device will read the data instantly when sent and helps reduce human error, as this action must be performed intentionally.

NFC is based on the relationship known as inductive coupling. This is when electrons flow through a conductor, thus creating a magnetic field. When this magnetic field changes, it can cause electrons to flow through a conductor. Radio frequency identification (RFID) tags is an application of this relationship and NFC is an application of RFID. There are two types of NFC technology: active NFC and passive NFC. An active NFC device uses power to generate a magnetic field which allows it to both read and generate information while a passive NFC device does not have its own power supply and therefore, can only read information.

As NFC is a standard, it follows specifications such as having a transmission frequency of 13.56 MHz. It can send data at three different speeds: 106, 2012, or 424 KB per second, making it unsuited for large amounts of data that need to be sent at a time. There are also three modes of operation for an NFC device. One mode is read/write mode which just lets a device read an NFC tag (e.g. a barcode). Another mode is peer-to-peer which allows two devices to communicate to exchange information. Finally, there is a card emulation mode, which as the name suggests, the device imitates a smart card (e.g. tapping your card to enter a public transportation system).

This technology is very popular in other countries, most notably Japan, but it has not become widely adopted in the United States mainly due to its potential security risk. With NFC being used to emulate a smart card, it is vital to prevent others from getting the user’s financial information. Therefore, encryption is used to help keep information safe from outsiders.

In summary, NFC should not be used as the wireless communication system for the Babel Glove because of the lack of security and short operating distance. As the user will most likely be much further than 4 cm from the computer, it would be impractical to use NFC.

#### 3.3.5.2 ZIGBEE

Another, more niche, wireless communication option is ZigBee. Zigbee communication is a low powered radio frequency build for control and sensor networks on the IEEE 802.15.4 standard for wireless personal area networks (WPANs). Our group is considering using this technology as the developers claim it increases flexibility for users and developers and delivers secure efficient local connections for low power, low data rate, and close proximity wireless ad hoc network.

ZigBee was designed for Internet of Things devices and connecting smart homes and similar setups to a secure control system. There are a lot of security protocols implemented in the standards of ZigBee, it transfers data over long distances by passing data through their mesh network of devices to reach farther away ones. ZigBee uses this mesh network to optimize its low data rate application, long battery life and secure network. The protocols are very secure compared to bluetooth and wifi, they use secure 128 bit symmetric encryption keys, along with the transport of cryptographic keys, cyfering frame and controlling devices. Implementation of the system would work very well with our system and add an extra layer of control and security compared to Bluetooth and regular WiFi.

On the other side ZigBee comes with added IoT features and sophisticated control systems for multiple devices, these protocols would not benefit us and would add more complexity to our system. Our goal is to keep it simple and easy to use for our users, we do not want to make BabelGlove any more complex than it already is.

#### 3.3.5.3 WI-FI

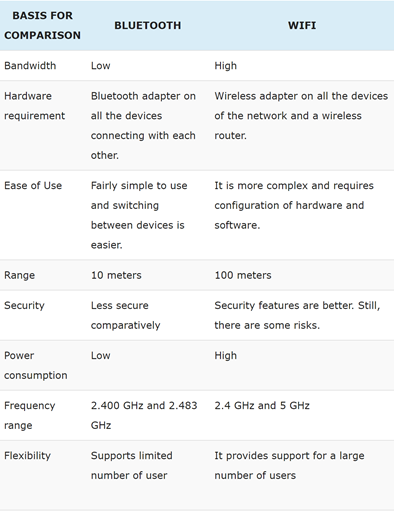
Wi-Fi uses radio waves to send and receive information usually over distances ranging from meters to miles. It operates at frequencies between 2.4 GHz – 5 GHz and has high data rates but consumes a large amount of power. There are two operation models: infrastructure mode and ad-hoc mode. In infrastructure mode, there is at least one access point (AP) in the wireless network that is usually linked to the wired network infrastructure and multiple wireless end stations. In ad-hoc mode, wireless stations interact direction with each other with no access point or wired network connection.

Wi-Fi is also one of the more secure wireless communication technologies and has two types of authentication: open system authentication and shared key authentication. Open system authentication (OSA) is when a computer requests to gain access to a wireless network that uses Wired Equivalent Privacy (WEP) protocol. Shared Key authentication provides even more security than OSA that uses a shared secret key to gain access to the network.

When using wireless connectivity, we use frequency bands. These bands have different uses, there is on one band that fits all of our wireless connectivity applications. As our group researches which wireless connection to utilize in our system, we need to look more into frequency bands, which is how any wireless connection is made such as wifi or bluetooth. The radio spectrum is the radio frequency (RF) portion of the electromagnetic spectrum. In the United States the radio spectrum is regulated by government agencies, the Federal Communications Commission (FCC) and the National Telecommunications and Information Administration (NTIA).

Frequency bands are distributed and allocated by these agencies for private and public use, more specifically for terrestrial or space radiocommunication services. As of now the frequency bands between 9kHz and 275 GHz have been allocated and in use, the FCC maintains a Table of Frequency Allocations, which is a compilation of those frequencies and their specific allocation. There are many applications for the range of frequency bands, including Industrial, scientific and medical uses. The industrial, scientific and medical (ISM) applications utilize radio and microwave frequencies that are clustered around 2.4GHz, These devices range from MRI machines, and radio telescopes to microwave ovens and wireless routers, which is where our system falls under. The frequency band 2.4GHz is a popular band because there is no need for licensing devices to use it, it is unlicensed by the FCC. Therefore, it is the perfect frequency band for development. Also another reason why the frequency band 2.4GHz is so popular for development is that the cost of setup is cheap compared to other radio frequencies.

Although Wi-Fi is a viable option to use for the Babel glove, there are also many limitations. At a higher frequency, more data is sent at faster speeds at the cost of higher power consumption. The system would also require more complex hardware and software design compared to other methods, such as Bluetooth. Together, this would make the design bulkier and more cumbersome. The table below further summarizes why Bluetooth was ultimately chosen over Wi-Fi.

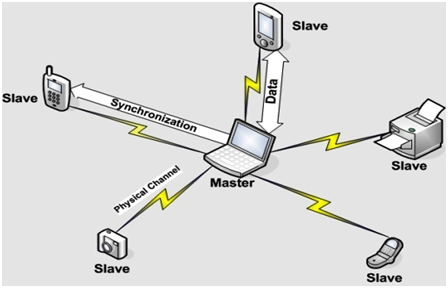
****

**Figure 3.3.5.3-1: Table comparing Bluetooth vs. WiFi**

#### 3.3.5.4 BLUETOOTH

Bluetooth uses a similar radio-wave technology as Wi-Fi but consumes virtually no power and is for shorter distances. It sends and receives radio waves at 2.45 GHz and usually has 79 available channels. Devices can automatically detect and connect to each other and use a technique known as spread-spectrum frequency hopping to randomly pick a channel if they want to talk to one another. Pairs of devices also shift the frequency they’re using, up to thousands of times per second, to help minimize the risks of interference from other electronics and improve security.

Bluetooth uses a scattered ad-hoc topology. It uses a cell called a Piconet that has four states: Master, Slave, Stand by, and Parked/hold. The master is one device that is the overall controller of the network while the slaves will obey the master’s instructions. Bluetooth is generally less secure compared to Wi-Fi, but communications are usually encrypted and there are many levels of security such as device-level security and service-level security.

****

**Figure 3.3.5.4-1: Illustration of Bluetooth topology**

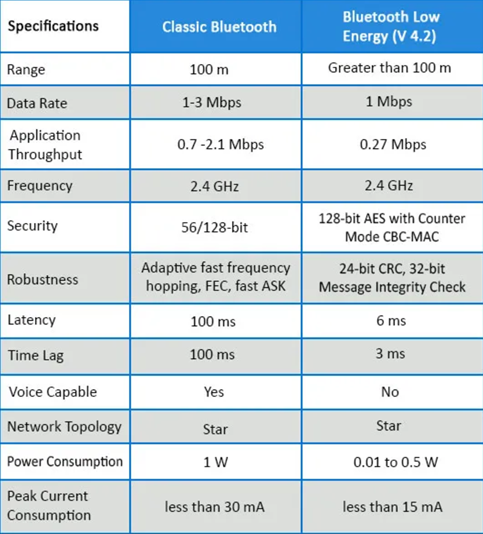
With this, the group has decided that Bluetooth technology should be used for the Babel glove. The user of the glove is expected to always be within a few meters of the computer; therefore, it is unnecessary to use Wi-Fi which is designed for greater distances. Although Wi-Fi is also more powerful than Bluetooth and can send a larger amount of data, it also consumes a larger amount of power. Wi-Fi will also cause the hardware design of the system to be a bit more complicated which can cost a large amount of time. These trade offs make Bluetooth the more desirable communication system to be used as it provides a great middle ground between Wi-Fi and NFC technology.

As there are multiple types of Bluetooth, the group has narrowed down their options to Classic Bluetooth and Bluetooth Low Energy (BLE). Bluetooth and BLE essentially operate in a similar way, except for one key difference. BLE has a significantly lower power consumption compared to Classic Bluetooth. BLE remains in a sleep mode until a connection is initiated and is typically used for applications that do not need a large amount of data.

Devices such as wireless keyboards and mice and other Internet of Things devices. An important factor in designing and building our glove system is the power factor. Implementing Bluetooth LE will greatly lower the power cost versus other wireless communication options, while obtaining that important secure connection between the glove and the user’s computer. Every design in the Bluetooth LE standard is to achieve the lowest power consumption, the lowest economic cost, low power, low bandwidth, and to keep it as simple as possible. To achieve the lowest power consumption, the device is kept in sleep mode when its not in use and the peak power consumption is less than 15mA with the average power consumption around 1µA. If we implement our system a Bluetooth Low Energy module, not only will it save us presious power consumption, but it will last longer than other implementations.

Bluetooth Low Energy has lots of robust features, with the low power consumption and cost, it also has a strong connection range. The connection setup and data transfer rate is very fast for BLE, it can support the connection as quickly as 3ms. This is a great speed for our system, as we need a quick burst of data transfer between the glove and the computer. A module can have a range of over 100 meters, which allows the user greater flexibility and freedom to use the glove. So the user does not have to be right next to the computer when signing, the user can place the computer in front of the person they are trying to talk to, so other person can see the translation happen in real time. Data packet security is not a necessity in our system, but BLE comes with fully integrated AES-128 encryption using CCM, which provides strong encryption and authentication to our data being sent from the glove to the computer. It is not a requirement that the data transfer be that secure, as we think it would be highly unlikely for a third party to interrupt or view the transfers, but with BLE we will have that security.

The group aims for the Babel glove to be lightweight and portable. Though Classic Bluetooth is more powerful than Bluetooth Low Energy, the group has decided that the high-power consumption could potentially cause issues. Therefore, the group has decided to use a BLE as the wireless communication technology. The table below further explains the differences between Classic Bluetooth and BLE technology and why BLE was chosen over Classic Bluetooth.

****

**Figure 3.3.5.4-2: Table comparing Classic Bluetooth and Bluetooth Low Energy**

### 3.3.6 POWER SUPPLY

The Babel glove will need a power supply to run the microcontroller, thus a power supply needs to be designed. Although the MCU can be powered via a USB connection, an external power supply is desired in the form of batteries to make the glove design portable. Since an external power supply is being used, a large enough voltage should be supplied to make sure that the microcontroller will regulate at the correct voltage.

#### 3.3.6.1 BATTERIES

In this section, different kinds of rechargeable batteries will be introduced. Rechargeable batteries are preferred over non-rechargeable batteries because they are safer, more reliable, and create less environmental waste. Batteries play a significant role in the project, so it is important to carefully review the different types of batteries and how they are charged to make the best final decision.

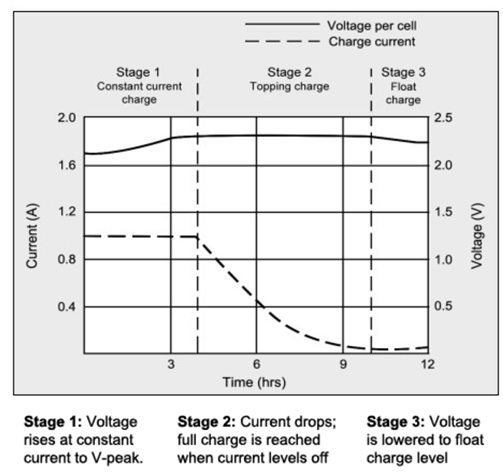
##### 3.3.6.1.1 LEAD ACID

Lead acid was invented in 1859 and was the first rechargeable battery used commercially. It is still widely used today because of its dependability and low cost-per-watt. The battery is made from a lead alloy and other metals (most commonly, antimony, calcium, tin, etc.) are added to improve electrical properties. Compared to other types of batteries, such as nickel and lithium, lead acid is heavier and less durable when deep cycled. A lead acid battery typically provides up to 200 to 300 discharge/charge cycles, of course depending on the depth of discharge. This is due to the corrosion on the positive electrode and depletion of the active material.

In the mid-1970s, sealed lead acid batteries were invented. The design of this battery resembles nickel and lithium-based systems and lets the battery operate in any orientation without leakage. They are able to create water to prevent drying out during cycling. Some of the most common sealed lead acid batteries are valve-regulated lead acid (VRLA) and absorbent glass mat (AGM). AGM batteries have a faster charging and instant load currents compared to conventional lead acid systems. It is typically used as a mid-range battery and unsuited for large systems compared to VLRA.

Although lead acid batteries are easy to charge, it is important to choose the correct voltage limits. Too low a voltage will shelter the battery but will cause sulfation buildup on the negative plate, causing poor performance. Too high a voltage will improve performance but will corrode the positive plate. These batteries are also slow chargers, taking up to 14-16 hours to be fully charged, but they have a moderate life span, losing approximately 40% of their stored charge in one year.

Using a constant current constant voltage (CCCV) charge method, lead acid batteries are charged in three stages: constant-current charge, topping charge, and float charge. In the constant-current charge stage, the battery is charged up to 70% in 5-8 hours. In the topping charge, the battery is charged the remaining 30% in about 7-10 hours. The stage is important and helps the battery to continuously accept a full charge. Finally, the float charge helps maintain the full charge of the battery. The figure below, reprinted by Battery University, helps further illustrate this process.



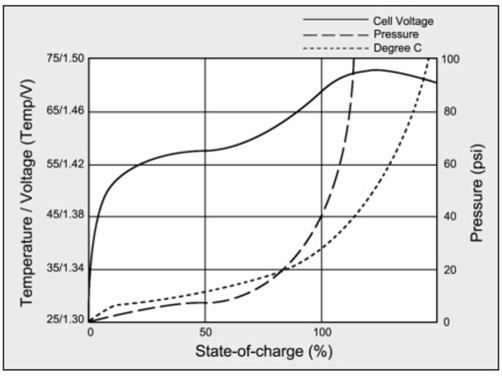
**Figure 3.3.6.1.1-1: Stages of charge of Lead Acid batteries**

In summary, lead acid batteries are inexpensive and have a low cost per watt-hour. They have the lowest self-discharge compared to other rechargeable batteries and are capable of high discharge currents. In contrast, they take up to 14-16 hours to fully charge and have a limited cycle life. Even more, they are not very environmentally friendly.

##### 3.3.6.1.2 NICKEL CADMIUM

Nickel Cadmium (NiCd) batteries were invented in 1899 and were mainly used for portable devices until nickel metal hydride (NiMH) batteries took over as they helped solve the toxicity problem of NiCd batteries. These batteries have a high cycle count if maintained properly and can charge ultra-fast with little stress. They have a good load performance and long shelf life and are economically reasonable. In contrast, NiCd batteries have a low specific energy compared with other systems and the metal, cadmium, is toxic and cannot be disposed of in landfills. They also have memory effect which causes a loss of capacity; therefore, they need full discharges.

To charge NiCd batteries, a constant current is used, and the voltage can rise freely. A slow charge for 16-24 hours is recommended for new batteries. This means that all cells in a battery pack are brought to an equal charge level. The battery cells will reach optimal performance after several charge/discharge cycles, where peak capacity will occur between 100-300 cycles. These batteries should include a safety vent that will release excess pressure if it is incorrectly discharged. One method to check if the battery is fully charged is by temperature. Low-cost chargers use temperature sensing to end the fast charge, where a temperature of 50°C is used as a cut-off. More advanced chargers use delta temperature over delta time, or dT/dt, to take advantage of the rapid temperature increase towards the end of charge to terminate the charging process. Other chargers use a defined voltage signature, meaning the charger will look for a voltage drop that occurs when the battery is fully charged. This is known as the negative delta V (NDV) method. During the first 70% of charge, the efficiency of NiCd systems are close to 100% as the battery absorbs almost all the energy. As the charge percentage increases, the charge efficiency drops and gases build up, pressure rises, and temperature increases. This can be seen in the figure below, reprinted from Battery University.

****

**Figure 3.3.6.1.2-1: Charge characteristics of NiCd batteries**

In conclusion, the group has decided not to use Nickel cadmium batteries in this project as the disadvantages far outweigh the advantages for the project.

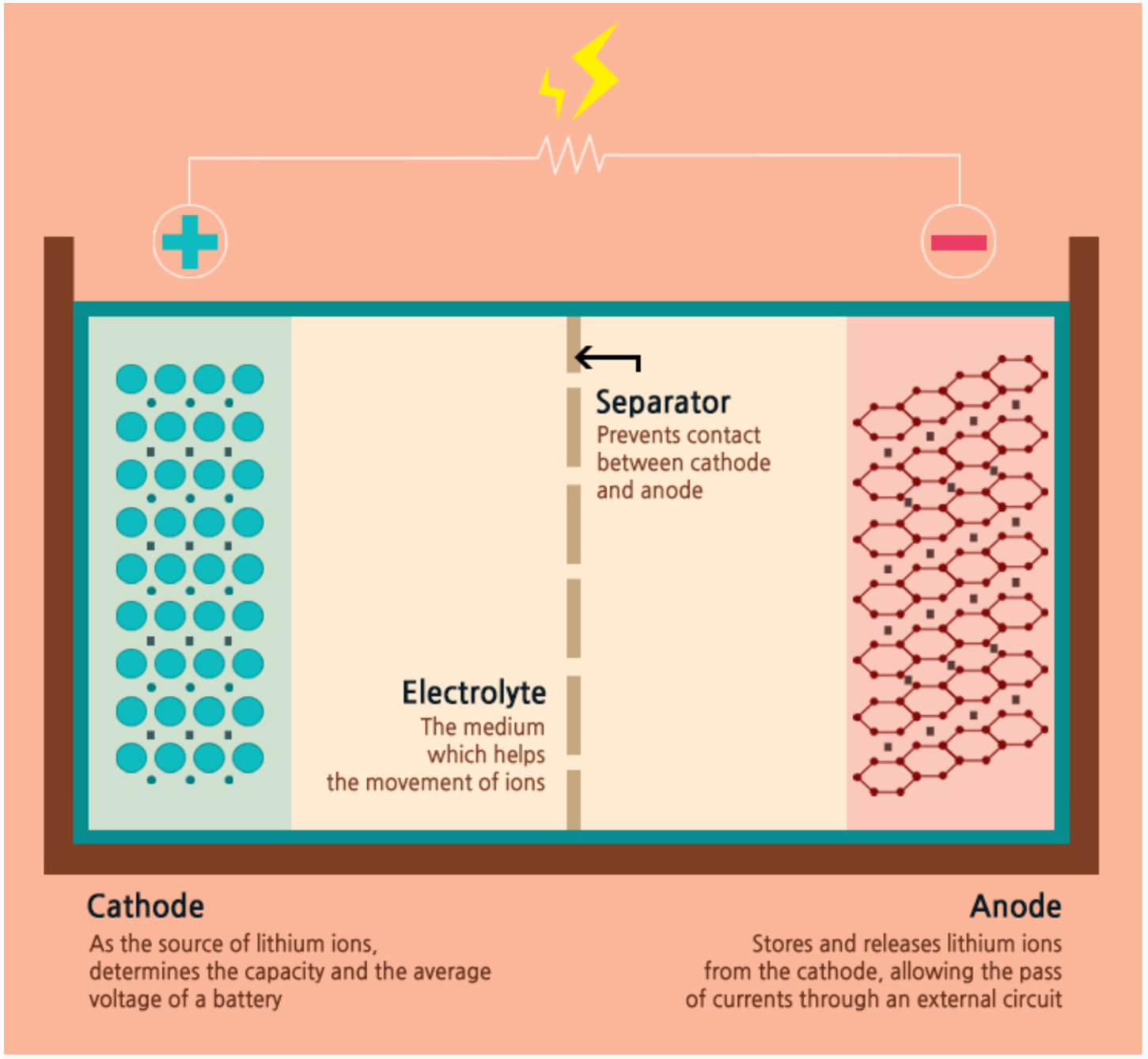
##### 3.3.6.1.3 NICKEL METAL HYDRIDE

Nickel-metal-hydride (NiMH) provides about 40% higher specific energy than NiCd batteries. It is a rechargeable battery that is readily used by consumers. Many AA and AAA are NiMH based and are both durable and low-cost. They can be rejuvenated and less prone to memory compared to NiCd and are more environmentally friendly as they contain only mild toxins. However, these batteries have a 20% self-discharge in the first 24 hours after charge and 10% per month after. Even when not in use, they get “flat” and sometimes will need to be recharged before use.

Charging NiMH batteries is similar to NiCd systems except NiMH is a bit more complex. NiHM chargers include Negative Delta V (NDV) to respond to a voltage drop of 5mV per cell or less. These chargers also include temperature threshold and time-out timers to detect when a full charge is reached. More advanced chargers will rest a few minutes to allow the battery to cool down when a certain voltage threshold is met. The current will continue to reduce as the charge continues until the battery is fully charged.

##### 3.3.6.1.4 LITHIUM ION

Lithium ion batteries are made up of 4 important components: cathode (positive electrode), anode (negative electrode), electrolyte, and separator. This can be seen in the figure below.

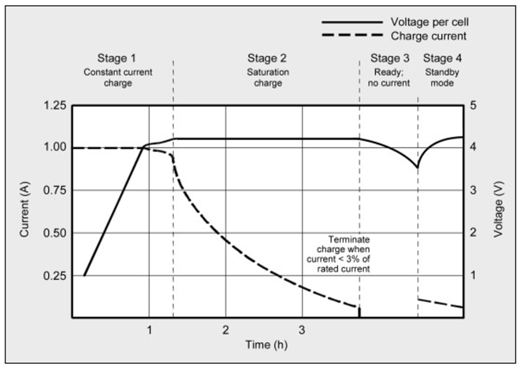
****

**Figure 3.3.6.1.4-1: Illustration of lithium ion battery components**

These batteries are low-maintenance and have no memory and do not need a full discharge to remain in good shape. They have a high specific energy and high load capabilities with Power cells. They also have a long cycle and shelf-life and low self-discharge. In contrast, these batteries require a protection circuit to prevent thermal runaway if stressed and they degrade at high temperatures and high voltage.

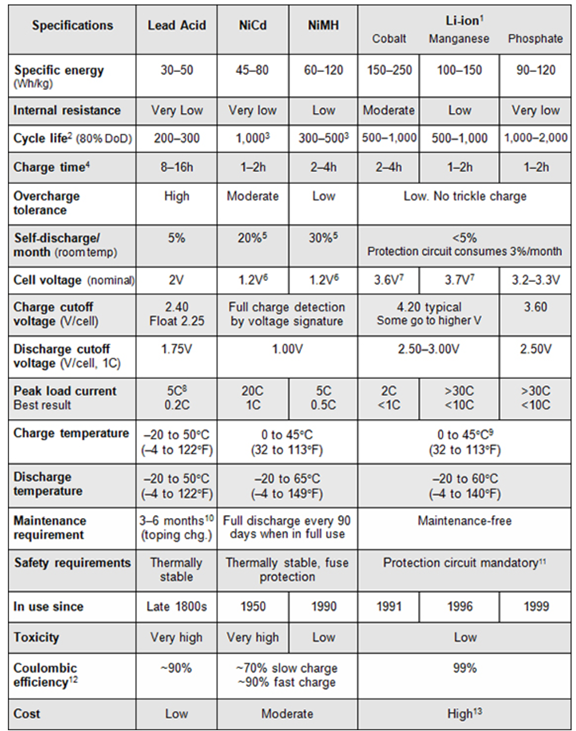
Lithium ion batteries are charged using a voltage-limiting device that is like lead acid systems. They use a higher voltage per cell, tighter voltage tolerances, and are strict on the voltage cut-off because Li-ion cannot be overcharged. Using traditional cathode materials such as cobalt or nickel, Li-ion charges about 4.20V/cell with a +/-50mV/cell. Increasing the voltage will increase capacity but this will stress the battery, therefore, protection circuits are built into the charger so it will not exceed the set voltage. The figure below shows the voltage vs. current as a lithium ion passes through 4 stages. The battery is fully charged when the current decreases to between 3%-5% of the Ah rating.

Li-ion takes about 2-3 hours to be fully charged and have a 99% charge efficiency. Using a higher current will make State 1 shorter, but the saturation change in Stage 22 will take longer. These batteries also do not need to be fully charged as the high voltage can stress the battery, therefore, it is sometimes better to choose a lower voltage threshold to prolong the battery life at the cost of a longer charge time.



**Figure 3.3.6.1.4-2: Stages of charge of Lithium ion batteries**

In conclusion, a lithium ion based system is desired for the Babel glove because they are low maintenance and have a fairly quick charge. The table below, reprinted from Battery University summarizes the four different kinds of batteries researched.

****

**Figure 3.3.6.1.4-3: Table comparing different rechargeable batteries**

#### 3.3.6.2 VOLTAGE REGULATORS

In this section, three types of voltage regulators will be discussed to determine which regulator should be used for the Babel glove. Voltage regulators are designed to maintain a constant voltage level even if the input voltage or load current change. They are used in many electronic devices to maintain the DC voltage used by the processor and also help minimize power consumption. In this project, a voltage regulator will be used in the power supply to make sure all components are receiving the correct voltage to function.

When selecting a voltage regulator, it is important to consider at least five factors. First, the input voltage and output voltage of the Babel glove should be known to choose what specific output voltage the regulator chip should operate at. Secondly, the dropout voltage should also be considered. Dropout provides a buffer between the output and input voltages, so if the Babel glove has a small difference between the input and output voltage, a low-dropout voltage regulator (LDO) or even ultra LDO may be considered. Thirdly, voltage regulators can produce output noise, so the sensitivity of the system needs to be considered.

The response time of the voltage regulator should also be considered. It is important for the Babel glove to respond to the user’s gestures in a reasonable time frame. Finally, the power draw of the voltage regulator should be considered. If the current becomes too high, the heat generated can become too high. In conclusion, these five factors should be considered when choosing the voltage regulator to be used for the Babel glove.

##### 3.3.6.2.1 SERIES VOLTAGE REGULATORS

The series voltage regulator is also known as the series pass regulator. It is one of the most commonly used regulators for a linear regulated power supply. These regulators provide a high level of performance when there needs to be low noise, ripple, and transients in output. It is made up of a variable element that is in series with the load. When the resistance of the series element is changed, the voltage of the element can be varied so the regulated output across the load remains the same. The main advantage of using a series voltage regulator over other types is that the amount of current drawn is effectively the same that is used by the load; therefore, it is more efficient. Although, it should be noted that these regulators are much less efficient compared to switch mode power supply but have the advantage of being much simpler and the output does not have switching spikes.

##### 3.3.6.2.2 SHUNT VOLTAGE REGULATORS

As the name suggests, this regulator works by the regulating element shunting the current to ground. The shunt voltage regulator maintains a constant voltage and uses the excess current to maintain the voltage across the load. The load is in series with a resistor and the voltage source and shunt regulator is in parallel with the load. The shunt regulator will draw negligible current at maximum load current and full current at minimum load current. This can make shunt voltage regulators inefficient compared to other regulators.

##### 3.3.6.2.3 SWITCHING VOLTAGE REGULATORS

The switching voltage regulator works by moving bits of energy from the input voltage source to the output. This is done by using an electrical switch and a controller which can regulate the rate at which energy is transferred. Essentially, the switching element transforms the input power into a pulse voltage that is smoothed using elements such as capacitors and inductors. Once the output voltage reaches the desired value, the switch element is turned off so input power is reduced. Switching regulators can have up to 85% efficient, making them one of the more energy efficiency regulators.

### 3.3.7 FPGA

Field Programmable Gate Arrays (FPGAs) are semiconductor devices with programmable components. For this project, the most interesting programmable component is the logic gates. The Babel glove needs to implement a ‘standby’ mode for the glove. This mode is for when the user may be wearing the glove but does not want to actively use it. Therefore, the glove should not be sending any information to the computer in this mode. This can be implemented by using hardware or software. If a hardware implementation is chosen, then FPGAs should be used.

To program FPGAs, a developing environment needs to be used. There are many developing environments, but one of the most popular programs and one that the group has familiarity with, is Vivado. Vivado is a product of Xilinx and lets the user program FPGAs that use logic gates by physically connecting all components together to build the circuit or use the hardware description language (HDL), Verilog. Verilog is the preferred method as circuits can become large and complicated, making the process tedious and time consuming. A testbench can also be created to simulate the design before downloading to an FPGA development board.

To implement this design into the Babel glove, an FPGA chip will need to be added into the printed circuit board design. An FPGA development board will not be needed since it includes unnecessary features such as LED lights and switches. It should also be mentioned that one member is very familiar with using Verilog in Vivado because of the class EEL 4783: HDL in Digital System Design. Therefore, time isn’t a great constraint to be worried about if using a hardware implementation. If a software implementation is used, then the ‘standby’ mode can be added to the microcontroller program.

In conclusion, the group is planning to use a software solution for the ‘standby’ mode since adding an FPGA chip will cost extra money and time, when it can simply be implemented into the microcontroller.

### 3.3.8 PCB

A printed circuit board (PCB) will be designed to include the microcontroller components and power supply components. This will have to be worn by the user; therefore, it needs to be of reasonable size. More specifically, it needs to fit on the glove while not being too bulky and allowing flexibility for the user to make hand gestures. The PCB will need to be carefully placed on the glove so the user won’t accidently damage it.

There are many manufacturers around the world for printed circuit boards. For this project, since time and cost are the main constraints, it is recommended to complete the PCB design as fast as possible to have ample time for shipping and testing. The PCB also needs to be at a reasonable price, therefore there needs to be a balance between time and cost. It is usually cheaper to ship PCBs from oversea manufacturers, but this is at the cost of time. Therefore, the PCB will need to be designed earlier, compared to ordering from a domestic manufacturer which will save time at a much higher economic cost. The worst-case scenario should also be considered, which is that a PCB is ordered but it does not work, so another PCB will have to be designed and ordered. In this section, different PCB manufacturers will be compared to determine which manufacturer the group should use.

#### 3.3.8.1 EXPRESS PCB

ExpressPCB has free softwares to design schematics and circuit board layouts. ExpressPCB Classic software is for 2-4 layer boards and silkscreen top layer. ExpressPCB Plus is for 2-6 layer boards and silkscreen on 2 sides (top & bottom). Using this free software is helpful in that an external circuit board design software will not have to be used and transferred. Unfortunately, the average price for a 2-layer PCB board that is no larger than 12in x 14in is $160. This price includes shipping.

#### 3.3.8.2 BAY AREA CIRCUITS

Bay Area Circuits is a PCB manufacturing company located in California. Their location in Silicon Valley is fully functional, which is important to note during this time. They offer a two-layer special where you can get 3 printed circuit boards, up to 50 square inches per board, for $30 each, for a total of $90. It is then shipped within 5 business days.

#### 3.3.8.3 PCB ZONE

PCB Zone is a manufacturing company located in New Zealand. Due to the COVID19 virus, at this time they are only accepted orders related to government, education, research, and COVID19 purposes. For a two-layer PCB board, the cost is $145.67 which includes 3 day shipping. Although the turnover rate is quick, the price is quite high.

**3.3.8.4 COMPARING PCB MANUFACTURING**

Keeping in mind the current circumstances of COVID19 and how it is affecting the prices and shipping time of PCBs, the group has decided to use Bay Area Circuits due to its cheap price and location. As the PCB is planned to be around 3in x 4in, the group will be able to take advantage of the two-layer special Bay Area Circuits offers.

|  |  |  |
| --- | --- | --- |
| **Manufacturer** | **Price** | **Location** |
| Express PCB | $160 | USA |
| Bay Area Circuits | $90 | USA |
| PCB Zone | $145.67 | New Zealand |

### 3.3.9 SERIAL COMMUNICATION

#### 3.3.9.1 ANALOG TO DIGITAL CONVERTERS

The data being sent from all the sensors used on the Babel glove will be analog, but the MCU is only able to process digital data. The analog signal will have to be converted to a digital signal, which is usually done with an Analog to Digital Converter (ADC). Therefore, it is important to be aware of how many sensors are exactly being used and how many analog-to-digital pins the selected microcontroller has. The Bluetooth module will also use pins, so an ADC may be needed for the Babel glove.

There are many types of ADCs as they come in different speeds, use different interfaces, and have different degrees of accuracy. One of the most common types is flash ADC. It is the fastest type and uses comparators, one per step voltage, and resistors. The comparator outputs are connected to logic that determines the voltage output based on the high and low value of the comparators. Although flash ADCs are fast, they can become quite large because of the number of comparators and draw large currents. If an ADC is needed for the glove, the group will most likely use this type.

### 3.3.10 GLOVE

#### 3.3.10.1 INTRODUCTION

The glove is one of the most important parts of this project. It is what will hold all the different electronics and sensors and make this a usable device. There are many different considerations to think about when choosing a glove.

#### 3.3.10.2 CONSIDERATIONS

The glove will need to be made from a comfortable and flexible material. It will also need to be effective against moisture. Leather is a natural material that is durable and can withstand moisture quite well. There are many different types of leather and there is also synthetic leather. However, leather gloves can be quite thick. PVC is a plastic material that has good durability and can be protective against moisture. Nitrile is a synthetic rubber that is very durable. However, small holes form in the coating of the glove during production and this can allow water to pass through. Latex is another common material to make gloves that is durable and has a high elasticity. However, latex can cause some people to have allergic reactions. Cotton is another material that is comfortable and durable but does absorb moisture. Another consideration when looking at the type of material is the option to have two layers. With this option, the electronic and hardware elements can be positioned in between the two layers. This could lead to more accurate sensing and data collection. It will also look better as the electronics will not be exposed. The material will also need to be able to withstand heat. As the electronics run, heat will be produced and can affect the material.

For the size of the glove, it will fit the average hand that is approximately 7.3 inches. This should be big enough to hold all the sensors and the electronics with the exception of the PCB. The glove should also be lightweight. With all the electronics and hardware attached to it, it should not weigh more than three pounds. Also, the glove needs to have five full individual finger slots that are not attached together. It is important that part of the finger slots should not be cut off as this will affect the sensor placement and that none of the fingers should be attached to one another as this would restrict movement. The back of the glove should be empty with no strap of velcro to secure the glove. This will impact the sensor placement. These are all considerations to look at while choosing the appropriate glove for the purpose of the project.

#### 3.3.10.3 NIKE ALL WEATHER

#### 

The Nike All Weather gloves are golf gloves that are made from nylon, polyurethane, spandex, and silicone. They are water-resistant, breathable, and stretchy. This will make the glove comfortable to wear. There are silicone moldings over the knuckles to allow for more movement. This pair of gloves is the most expensive out of all the options.

#### 3.3.10.4 NIKE HYPERWARM ACADEMY

The Nike HyperWarm Academy gloves are soccer gloves designed to help the hands stay warm in cooler weather. Although this may be good in some situations, this might cause trouble for this project. As the electronics produce heat, this could get trapped in the glove and cause it to be hot and uncomfortable to wear for the user. The silicone zones on the fingers and palm help with grip which is not a specification that is needed. The flexible cuff helps to secure the fit of the glove. It is made from polyester, nylon, rubber, and spandex. Overall, this pair of gloves ranks in the middle in terms of price.

#### 3.3.10.5 UA RADAR SOFTBALL GLOVES

These gloves are women’s softball gloves by Under Armour. It is sold in pairs and made from leather with synthetic overlays. It uses Under Armour’s HeatGear fabric with 4-way stretch construction. This allows for better movement in every direction. The perforated synthetic overlays add support and durability. The perforations also increase the breathability of the glove. There is an elastic wrist cuff that adds wrist support and helps to secure the glove in place. This glove is the second most expensive option. This option could also be described as the most visually pleasing out of all the options since it is not black and contains some color.

#### 3.3.10.6 ULINE ULTRA-LITE POLYURETHANE COATED GLOVES

These gloves are by Uline. They have a sheer liner that can fit small parts in between. It is made with material that allows it to stay cool. These are large gloves with a wrist dimension of four inches. The cuff contains natural rubber latex. This product comes in a carton of twelve pairs of gloves.

#### 3.3.10.7 TACTILE RUNNING GLOVES

These gloves are from Kalenji sold by Decathlon. These are the cheapest pair of gloves out of all the options. It is made from polyester cationic and spandex. These gloves have touchscreen compatible tips which can allow the user to use a screen on a touchscreen device while not having to worry about removing the glove.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Summary of Features | | | | | |
| Model | Nike All Weather | Nike HyperWarm Academy | Under Armour Radar Softball Gloves | Uline Ultra-Lite Polyurethane Coated Gloves | Kalenji  Tactile Running Gloves |
| Price | $28 | $24.97 | $24.99 | $26 for 12 pairs | $7.99 |
| Material | Nylon, polyurethane, spandex, silicone | Polyester, nylon, rubber, spandex | Leather, synthetic overlays, HeatGear fabric | Polyurethane, rubber latex | polyester cationic, spandex |
| Features | Water-resistant, silicone molding | Flexible cuffs | Support, durability, breathability | Liner, cooling material, | lightweight |

**Figure 3.3.10.7 - 1: Summary of Features**

### 3.3.11 BLUETOOTH LOW ENERGY MODULE

In this section, the pros and cons of different types of Bluetooth modules will be compared to determine which module should be used in the project. The Bluetooth module will collect data from the sensors and send it to the computer using radio frequency.

As there are two modes a Bluetooth module can utilize, single mode or dual mode, the group has decided the module needs to use single mode. Single mode only uses the Bluetooth low energy module while dual mode will use both Bluetooth Classic and Bluetooth low energy. To keep the costs and power consumption of the design low, single mode will be used.

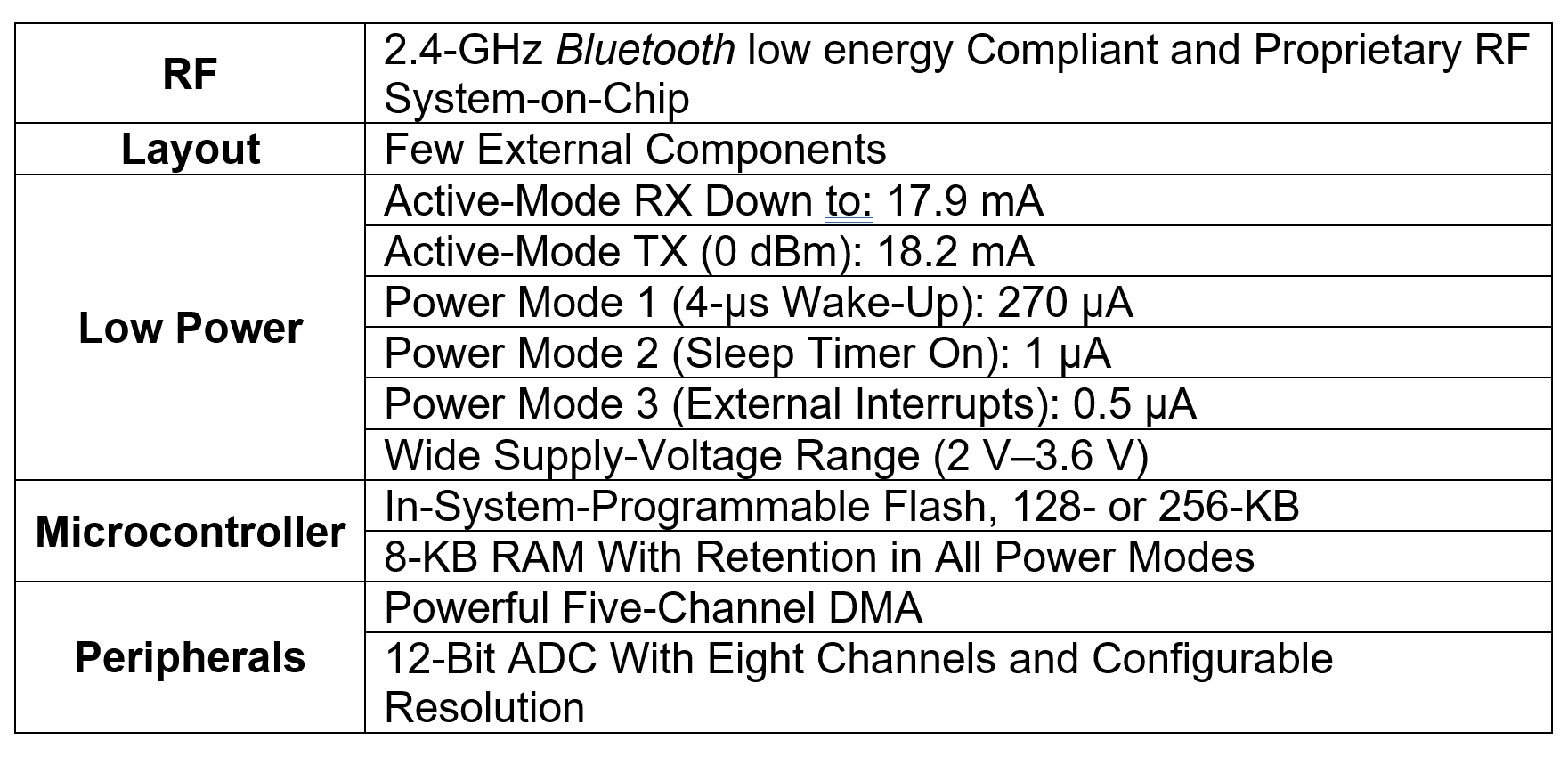
Another specification that should be considered is the package options of the Bluetooth modules. Bluetooth modules come in either Quad Flat No Leads (QFN) or Wafer Level Chip Scale (WLCSP or CSP). QFN packaging is larger and therefore, has a higher cost compared to CSP. It will also require more than a two-layer circuit board making the cost of the PCB much higher. In conclusion, the group has decided that CSP packaging is more desirable for the project.

#### 3.3.11.1 TEXAS INSTRUMENTS CC2541

The CC2541 chip made by Texas Instruments lets developers use the main advantages of BLE. It is used in many applications such as light control and wireless sensor networks. It is a power-optimized system-on-chip (SoC) solution for BLE and 2.4 GHz applications. It’s suitable for systems that require ultralow power consumption, which is specified by various operating modes. There are also short transition times between these operating modes to help the system consume even less power.

The chip has in-system-programmable flash of 128KB or 256KB. It has 8KB of RAM with retention in all power modes. There are also few external components and includes two active modes and three power modes. It also has a wide supply-voltage range of 2V-3.6V. The CC2541 also includes many peripherals. There is one 16-bit and two 8-bit general-purpose timers and a 32KHz sleep time with capture. It includes a 12-bit ADC with eight channels and configurable resolution. There are two USARTs with support for many serial protocols and 23 I/O pins. Finally, it supports I2C interface, which supports slave and master operation.

Below is a table of features of the CC2541 chip.



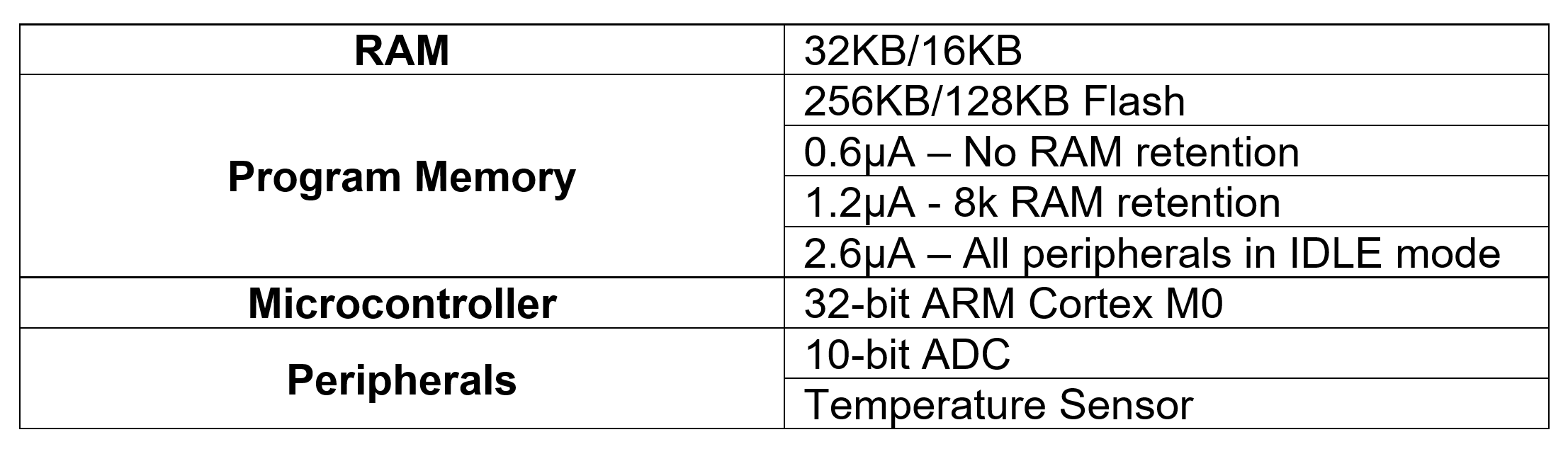
**Figure 3.3.11.1 -1: Table of CC2541 Chip’s features**

#### 3.3.11.2 NORDIC SEMICONDUCTOR nRF51822

The nRF8001 chip made by Nordic Semiconductor is a multiprotocol BLE and 2.4GHz proprietary system-on-chip. It is used in many applications such as wearables, CE remote controls, sports and fitness sensors, healthcare and lifestyle sensors, and BLE applications. The nRF51822 has a separation between application code development and embedded protocol stacks. The BLE stack is a pre-compiled binary that is available from Nordic Semiconductor and the application code is compiled stand-alone.

The nRF51822 uses a 32-bit ARM Cortex M0 MCU and has 128KB flash and 16KB RAM or 256KB flash and 32KB RAM. There are 31 GPIO pins and a full set of digital interfaces that include SPI and UART. The chip includes many peripherals such as a 10-bit ADC, temperature sensor, and RTC. It has a low power 16MHz crystal and RC oscillators and ultra-low power 32KHz crystal and RC oscillators. There is a wide supply voltage range of 1.8V-3.6V and individual power management for all peripherals.

Below is a table of features of the nRF51822 chip.



**Figure 3.3.11.2 -1: Table of nRF51822 Chip’s Features**

#### 3.3.11.3 MICROCHIP RN4020

The RN4020 chip made by Microchip uses a Bluetooth version 4.1 module and on-board BLE 4.1 stack. It is used in many applications such as health/medical devices, sports activity and fitness, wearable smart devices and accessories, and Internet of Things Sensor tags. It is environmentally friendly and features remote commands over-the-air. One of the most important features is how compact the device is: 11.5 x 19.5 x 2.5mm.

The chip has a single operating voltage from 1.8V to 3.6V (3.3V typical) and temperature range of -30C to 85C. It has low-power consumption and simple UART interface. It includes an integrated crystal, I2C interface, and multiple IOs for control and status, GPIOs, and an ADC. It also has three Pulse Width Modulation (PWM) outputs.

The figure below summarizes key features of the RN4020 chip.

## 

**Figure 3.3.11.3 - 1: RN4020 Block Diagram**

Reprinted from the Microchip RN4020 datasheet.

## 3.4 SOFTWARE

### 3.4.1 GESTURE RECOGNITION

There are a couple of different algorithms that can be used to implement the type of gesture recognition we need for our project. In previous projects, we’ve seen algorithms that used conditional statements (“hard coding”) and algorithms that used machine learning to interpret gestures. When researching the algorithms that used conditional statements, we found that it would be quick and easy to implement, but hard to scale if we wanted to translate more gestures. On the other hand, using a machine learning algorithm wouldn’t be as straight-forward but it would be easier to add more gestures once we had the training in place. For purposes of scalability, we chose to use machine learning for the gesture translation. In this section we explore the various options in choosing a programming language for the machine learning algorithm. Choosing which kind of machine learning algorithm to utilize in our gesture recognition software.

### 3.4.2 PROGRAMMING LANGUAGES

### 

Picking the best programming languages for our system is a tough choice, there are many options. On the hardware side, the microcontroller we end up choosing will have its default language to program it with. On the computer side, there are many choices to program the user interface and the artificial intelligence that will be determining the ASL letters and phrases from the sensors on the glove. In the data science world, the top programming languages for machine learning are Python, R, C/C++ and Java. The most decisive factor when choosing a programming language is the application area. Where we are running the code for this project is on a computer, so we have flexibility for the language environment.

Another important factor we are considering is ease of use. When we finish the software and the prototype glove, we want the user to be able to easily use the product. The Java programming language requires a virtual machine (Java Virtual Machine) to run its programs rather than creating an executable file. It compiles into an intermediate language called Java bytecode and runs it on its portable machine. While Java is portable and can be run on multiple machines, it adds an extra level of complexity to the set up process of the system. The user might not be able to download and install the JVM if they are using a shared computer, such as in a library. Therefore, Java is not the programming language we will choose.

In our development system, timing is an important factor. It is important for the software to keep up with the user while the user is signing. Millisecond timing is not that important in our software application, but the timing should be reasonably fast, for user accommodation and software use. Specifically the software should be behind two or three words at most. In this case C/C++ is not necessary, as that language focuses on optimization and speed. Considering these cases, Python is the best language for us to develop with, as it is easy to program and efficient in machine learning development.

#### 3.4.2.1 PROGRAMMING ENVIRONMENTS

As we research which programming language to use and implement in our system, it is important to find the best integrated development environment (IDE) to actually build and test the code in. There are a lot of different IDEs to consider, some of the factors we are considering are ease of use, powerful and flexible and debugging features. Looking at some of the popular IDEs for python, most of them are cross-platform, multi-language and multi-functional editors. The coding environments that our team is considering are:

* IDLE, the Python developer’s IDE
* Visual Studio Code, Microsoft’s versatile IDE
* Eric, flexible IDE written in Python, for Python
* PyDev, large plugin in for the Eclipse IDE

IDLE is python’s default IDE, it is very minimalist and designed so it does not ‘get in your way.’ It is very simplistic, which makes it flexible, and being made from python, by the developers of python, it most likely works flawlessly with python. It hosts some impressive features that any developer would want:

* Has an integrated Python shell windows
* A native python debugger
* Smart text editor

It also comes with a powerful debugger, with simplistic and easy to use settings. IDLE does not offer a lot of additional features though, it lacks project management facilities, which is important for our development. While IDLE would be easy to start developing, once our project grows in size with a lot of files, it will be harder and harder to keep track of it using this IDE.

Visual Studio Code (Code) is an amazing IDE that supports multiple languages and supports plugins where you can add even more features to help support development. It has many features that has only increased its reputation as the most versatile IDE in the industry:

* Built in Git commands
* Endless community extensions
* IntelliSense, a state of the art smart text editor

It is a very powerful and flexible software, but also comes with a minimalist mode, where the developer can hide Code’s user interface resulting in a less noisy development space. It also has a strong debugger, with a built-in terminal so you could just use Code for development, without needing extra softwares and programs. Visual Studio Code is a strong choice for building the software side of our system, as it is very customizable, where you can make it very simplistic or very complex.

Eric is a powerful python IDE that supports a few other languages, while the design is not the most clean, it makes up for its powerful features. Eric supports a hex editor, SQL browser and more for ‘serious coding.’ It has excellent support for real-time collaboration on the code, some multiple developers can be editing and texting the code at once, like sharing a document google docs. Eric is a naturally powerful editor and a real tool house for python specific functions.

The last IDE our group is considering is PyDev, it is a popular open source editor that runs in Eclipse, another popular IDE that has a wide range of support features. Some of those strong features are listed:

* Free and Open Sourced
* A strong debugger
* Works with popular IDE, Eclipse
* Djando integration

PyDev gives lots of coding support to the developer, auto-completion, coding templates, and a great testing area for your code. With all of these features, there is potential for the application to get bloated and hard to use. PyDev is another strong contender for a strong python IDE.

#### 3.4.2.2 VERSION CONTROL SYSTEMS

In our plans to develop and build our software for the system we will need a repository to store and share our files. There are numerous options for this application. To narrow the selection down, we know we want something online so we as a group can easily access it, and the more popular it is the more support it will have.

It is important to our group, as developers, to control the different and changing version of our code, so during the development of our system we will be using Git. Git is a ‘free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.’ Git makes it easy to control the versions of your development, create and manage staging areas and manage multiple workflows. All of the online repository options integrate Git into their systems. Also support in our IDEs is a must, direct access to upload the code and see the changes and download new code. A few who meet these requirements are GitHub, BitBucket, and GitLab.

GitHub, owned by Microsoft, is definitely the most well known of this list, it is very popular among developers and is supported by most IDEs, or there are plugins for it. This online version control system supports many features that has lead to its popularity:

* Synchronization to the cloud
* Smart automated security software
* Integrated desktop file system
* Free and open sources, used by millions of developers

GitHub makes uploading your repository very easy and seamless, with native Git integration. It is the go to Git online repository for developers and programmers worldwide. GitLab is another strong contender, it is fully open source and you have more control of your repository and host instances than GitHub or BitBucket. It is well respected in the developer community for its strive to being open source and dedication to give control to the individual developers.

Our last consideration is BitBucket, it shares most of the features of Github and GitLab with a few newer features, like native support for Mercurial repositories, but we will not be using those. It was a startup recently to compete against GitHub, then got bought by a larger company, as it is in use by many programmers, it is not as seamless and useful as its competitors. GitHub looks to be the best choice for our plans, and as a group we are mostly familiar with Github, which will help with ease of use.

### 3.4.3 MACHINE LEARNING

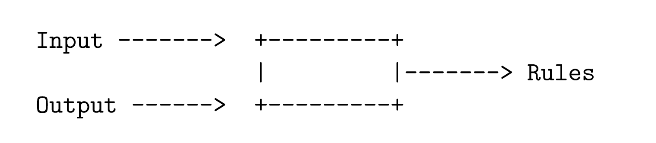
#### 3.4.3.1 ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) is a branch of computer science that attempts to simulate intelligent human behavior normally not found in a computer system. AI is the overarching term for our other, more specialized terms. Programming AI typically puts the input data and the rules that go along with the input data, into the algorithm and then the algorithm computes the output. The goal of many AI developers is to create an expert system that exhibits intelligent behavior and learning. The ultimate philosophical goal is to make computers think more like humans. To simulate the many neurons in our brain all connected and communicating with each other. The science of AI is based on lots of other branches of science, such as sociology, neuron science, and biology. A popular application of AI is neural networks, a system of layers that learn given an input, it was designed based on the complex structures of our brain. Supposedly simulating the neurons in our heads talking to each other.

AI has lots of applications in the real world. To image and speech recognitions, gaming and in business. We are starting to see large corporations invest in artificial intelligence to help find patterns and models in your vast data that they collect from their consumers. With this application they can create consumer behavior forecasts and predict who is going to buy what when. In our application of AI we will be focusing on machine learning, which is a branch of artificial intelligence.

#### 3.4.3.2 MACHINE LEARNING

Developers at Google say machine learning is “the process of training a piece of software, called a model, to make useful predictions using a dataset.” The model can then make predictions about unseen data given by a user or new data. Machine learning is dynamic and can learn on its own without human intervention. That makes the program strong and less reliant on user intervention. Machine learning is defined as the field of study that gives computers the ability to learn without being explicitly programmed. In machine learning the programmer provides input data and output data that relates and then the program outputs the rules or how the data sets relate. Figure 3.4.3-1 below relates the concept of machine learning to an simplistic image.



**Figure 3.4.3-1: The Basic Function of Supervised Learning**

Machine learning has two paradigms, supervised and unsupervised learning, but in practical settings machine learning is more of a spectrum, with supervised and unsupervised on either side. Supervised learning is where the model is provided with labeled training data. Unsupervised learning is where you provide the model with the training data but no labels to match. This means the algorithm has to figure out the pattern of the input data on its own. In machine learning, a label is what we seek to predict, such as the type of animal or price of stock, or in our case the letter or phrase in ASL. A feature is an input variable that is used to predict the label. A model is the relationship between the feature and the labels of the data, when training a model, you are creating or learning the model, or learning the relationship between the features and the labels.

Supervised learning is most beneficial to our system because it finds patterns between the data and labels that can be expressed mathematically and then output to the correct ASL letters and phrases. On the supervised learning side of the machine learning spectrum, we can improve the accuracy of the program and give it the data sets needed to create a relevant model that accurately predicts future unseen data. We will need to create these data sets for the program to learn and develop the models. In supervised learning there are two model algorithms, classification and regression.

Classification is identifying new data and statistically categories it into a set of sub-populations on the basis of a training set from supervised learning. The algorithms associated with classification are known as linear classifiers. They assign a score to each possible category on the basis of the input as vectors with weights for each component from the designated input. The category with the highest score is the most likely category associated with the input. We are considering a number of classification algorithms to classify the user’s hand motions, some are Neural networks, Navid Bayes, and Fisher’s Linear discriminant.

Another category under supervised machine learning, is regression. A regression model predicts continuous values, such as what is the probability that a user will click on an ad or what is the value of a house in california. Regression is most used for finding the relationship between variables and forecasting. A simple two feature linear regression algorithm is shown below:

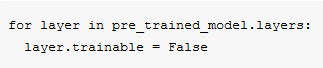
Where Y is the dependent variable that we are trying to predict based on the two independent variables, , and . is the bias or the intercept, which is always constant, and is the weight of , the same as the slope in the traditional equation of the line. Under supervised training, the algorithm trains the equations to find the best bias value and the best weight value for the feature variable. Then when presented with new data, newly trained algorithms can infer or predict the output, , for the new independent features, , and . With this equation you can create a more sophisticated model that can rely on multiple features each with a seperate related weight.

#### 3.4.3.3 FINE-TUNING

Training a model from scratch for a real world application offers real world limitations. There is an endless supply of network parameters the network needs to get its optimal set of parameters, so there is an endless supply of data needed. Which requires huge computing power, the network will train over multiple iterations with all the data. As our team has researched different machine learning applications to best prepare ourselves for our machine learning application we learned about fine-tuning. Our team is limited by our budget, so we cannot afford to buy a large GPU computing rig to train our models for our glove. Using a pre trained model might help us tremendously with time and accuracy. We are considering it as an option for training our model for our glove system.

A pre-trained model is a model that already has this optimized parameters from its training data in its application. Usually the initial layers learn very general features and the higher levels have patterns more specific to its training data. So when using a pretrained model, you can tweak it or fine-tune it to your application. Fine-tuning a Pre-trained model eliminates the two limitations of training your own model.

When Fine-Tuning a Pre-Trained Model, there are a few necessary steps before you can use the model to train your own data. You must freeze the layers that you want to keep. This code snippet below shows an example on how you might implement freezing your layers.



**Figure 3.4.3.3 -1 Python code snippet for Fine-Tuning**

Then once you have the layers you want frozen, you can create your new model. Adding whatever need in your machine learning application, optimizer, classifiers, etc. Then you can train the model using your training and testing data. This fine tuning greatly increases the time and decreasing required computing performance than having to train a model from scratch.

### 3.4.4 USER INTERFACE

Once the gestures get recognized and translated, we will be using a graphical user interface to provide feedback to the user. We will have to consider whether we will be using a desktop or a mobile platform to put our application on. In the following sections we will discuss the possibilities of the different platforms we can use for our application.

The main purpose of the application will be to take the translation and display the translated gesture in the interface. Since our project is oriented more towards people who are learning ASL, our project will have two modes. In the first mode, the user will display whatever letters or phrases the user signed with the glove and in the second mode, the user interface will prompt the user to sign a letter or phrase and the user will perform a gesture matching the prompt.

As one of the main visual components that users interact with to get feedback from the system, the functionality as well as the attractiveness of the application that can be made on the platform should be considered when choosing an environment to develop our application in.

It is important that we examine each option for the different mobile and desktop platforms to develop our user interface for since each of the different options may change either our timeline or budget.

#### 3.4.4.1 MOBILE PLATFORMS

There are a couple of different mobile platforms out there, but the main ones that are the most accessible and reach the widest audience are Android and iOS. Some of the pros to using a mobile application is that it is portable and accessible to most people. Some of the cons for using a mobile application is that we would have to develop two separate apps for the two different platforms, or figure out a way to port an application on one platform to another without breaking any of the features. Another con is not having every group member able to test the interface. Because our application will be used solely for the purpose of being a user interface, most phones will have the hardware necessary to run our application.

##### 3.4.4.1.1 ANDROID

One of the main mobile platforms for app development is Android, which is owned by Google. In 2019, the market share for Android was estimated to be around 70%. In the United States, the market share for Android is a little lower at around 60%. Android runs on phones, tablets, gaming systems, and more. There are a lot of free open-source materials that can help developers create an application for Android. Android Studio is the official IDE for Android and contains everything that is needed to build an Android app. It provides tools for developing, debugging, and conveniently packages Android apps. One of the tools that is very useful is that you can use an Android phone you already have on hand to test your application, or you can use an emulator which creates a virtual device to mimic certain Android devices. Having this feature makes it easier to design and develop the app for multiple Android devices with different screen sizes, such as phones and larger tablets. Because Android is so popular, there is a lot of information and documentation online that will help developers navigate through issues easily. It is easy and free to publish apps on the Google Play Store.

One con to developing an Android application is that none of our group members have an Android device. While we would be able to use the emulator in Android Studio to build and test the app, it would be hard to test the whole system altogether without a physical device. While we would be able to acquire an Android device easily for testing purposes, this would drive up our budget a considerable amount, which we may need to consider.

##### 3.4.4.1.2 iOS

Another main mobile platform for app development is iOS, which is owned by Apple. In 2019, the market share for iOS was estimated to be around 29.5%. In the United States, the market share for iOS is a little higher at around 39.7%. iOS runs on phones and tablets. Similar to Android, Xcode is iOS’s official IDE that allows apps to be developed on the iOS platform as well as for Mac OS. Similar to Android Studio, Xcode provides iOS developers all the tools for developing and debugging your iOS applications. It also provides a simulator so you are able to see how your application would look and run on different Apple devices. There are several resources online that can be used to help developers debug their apps as well as the iOS Developer Library which has a lot of helpful resources for developers.

Unlike the Android mobile platform, publishing apps in the App Store is not free. To publish your iOS application, you must meet Apple’s App Store Guidelines, and become a member of Apple’s Developer Program. Registration of the Developer’s Program is not free and can cost $99 per year, at the cheapest plan. These costs must be considered when choosing a platform for our application for the user interface.

#### 3.4.4.2 DESKTOP APPLICATION

Creating a desktop application could also be a great choice for creating our application for our user interface. One of the main pros to creating a desktop application instead of using one of the previously discussed mobile platforms for apps is that it will be cheaper. Since members in our group do not own Android devices, and we are not already members of Apple’s Developer Program which costs $99 to join, it is more cost-effective and accessible to develop a desktop application for the user interface.

A couple of options we have to create an application that runs on a Windows Desktop, is a Java Swing application. Swing is a GUI widget toolkit for Java which has the ability to provide a sophisticated set of GUI components to create Java-based front-end applications. One of the pros to using these tools to create an application that holds the user interface is that there is a lot of documentation online and these tools are familiar and have been used in some of our group member’s previous classes.

Another option we have to create an application that runs on a Windows Desktop is creating a Universal Windows Platform app. The Universal Windows Platform is a common app platform on every device that runs Windows 10. This is pretty significant since Windows is the dominant desktop OS, with the largest market share of Windows 10 being around 50% in the United States. Developing an application on the Universal Windows Platform would require the app to be developed in either C++, C#, Microsoft Visual Basic, or JavaScript. A pro to developing a Universal Windows Platform app versus a Java Swing application is that the app can be published to the Windows Store for free and can be downloaded from any device that runs Windows 10, including tablets. Having the option to be able to have our app on either a full-fledged computer or a tablet makes it so that we don’t have to make the decision to go mobile for portability. A con to creating a Universal Windows Platform app is that none of our group members have had the opportunity to develop an app using this platform. Additional time would need to be allocated to learn how to develop apps on the platform.

# 4. CONSTRAINTS

## 4.1 DESIGN CONSTRAINTS

Throughout designing and researching our project, we have come across numerous design constraints in every aspect of our system. These concentrations cause limitations in the design of the glove and will need to be taken into consideration when creating the design. This section describes in detail the various design constraints our glove has to operate around. We strive to find the most realistic constraints so our system will relate and reflect the outside environment in which it will operate in. These realistic factors include: Ethical, Social, Manufacturability, Sustainability, Economic, Environmental, and Health.

### 4.1.1 ETHICAL AND SOCIAL

A big limiting factor is language, specifically American Sign Language. The glove is limited to only the ASL alphabet and simple phrases that can be completed with one hand. The system is unable to translate more complicated and complex words and grammar of ASL. ASL also relies heavily on facial expressions that our system is unable to capture, thus greatly limiting what can be translated. To that end, we hope that our glove is accurate enough so the user can learn the basics of ASL and then through practice and use, the inept ASL user can graduate from our glove to signing with other ASL users. We must be wary of the deaf community as well. ASL is their language and we must respect and learn from them.

There are ethical constraints that our group must abide by, as well as any respectable engineering in the industry. As we research and develop our system, every group member must be honest and original in their ideas and designs. We are aware of the previous projects that have built an ASL translator using a glove and microcontroller, but we strive to keep to an original design as we built our BabelGlove. Also in our research we strive to always cite all the authors, articles, paper, etc that we reference in our paper. We will keep all referenced writing in Appendix B. An important ethical constraint that was given to us by our professors, is to receive permission for all images and figures we use in our paper. We must reach out to the authors and artists and ask for explicit permission and add proof of said permission at the end of the document in Appendix A.

### 4.1.2 MANUFACTURABILITY AND SUSTAINABILITY

While designing the glove, the team factored in the important constraint of keeping the glove to be lightweight while holding all the sensors. The position of the sensors and hardware needs to be placed so that it is comfortable for the user as well as function properly. How the glove looks is also a factor that contributes to the comfort of the user and manufacturability. All of the hardware and electronics will be exposed on the top of the glove. Ideally, this would be hidden. There are several different ways that this could be achieved. Fabric could be sown over the sensors and other electronics to hide them. Another option is to use two gloves and have the hardware and other electrical components placed in between the two gloves. This would be difficult to do without possibly damaging a part. Also, if there is an issue with a part after the glove is all put together, they will not be as easily accessible to fix the problem.

Another concentrate is material used to make up the hardware. The hardware is made of sensitive material that can be easily damaged. One thing that can damage the hardware is an electromagnetic pulse. Also, weather and the handling of the equipment is also a factor that contributes to the sustainability of the hardware. For this project, the hardware will be placed on the top of the glove in an exposed location. It is important that the glove be handled with caution and care as not to damage the equipment and prevent the glove from functioning properly. One option is to create a cover for the equipment. However, this could impact the movement of the glove.

The accuracy of the system depends on how accurate it can interpret the user’s movement. Because each person has a different hand size, the same gesture will be slightly different for each person. Therefore, the hardware and software need to work together to make the correct translations. The machine learning algorithm will need to work with the system too for different hand sizes and to react to the different hand gestures a person with a larger hand size might make. This can be done with more training data for the supervised machine learning algorithm. We will use subjects with different hand sizes to get more accurate training data, so the trained model knows how to react to the small differences in the user’s hand gestures with different hand sizes.

Another manufacturability constraint is the battery. The battery chosen is a lithium battery. It is important that the right battery is selected for the glove and that it is positioned correctly. The battery needs to be able to last a long amount of time to be practical to use. Also, if the battery is not rechargeable, the added cost of having to replace the battery occurs. If the battery is rechargeable, the PCB will need to be designed with a rechargeable port. Another option is to have a removable battery with a charger outside the glove. This will cause restraints in the design of the glove due to the space and also temperature limitations.

In order to make our system sustainable, its parts must be built from common parts that can be found in store or on popular electronic websites. We want our system to be easily manufacturable, not just for us, but for people who maybe will replicate our system and maybe try to improve on it and add more features. Our goal is to design the system so it does not require that we build our own components. For the sake of time, cost and sanity, our team chose to keep our design simple and build with components purchases from third-party vendors. During our prototype phase we will wire and piece together the optimized components.

### 4.1.3 ECONOMICS AND TIME

The group’s choice to self-fund this project will greatly limit the cost of the system. In choosing our parts, we must be careful in making sure they work together and give the correct output. Limiting the cost will also limit the performance and accuracy of the system. The parts that are the least expensive tend to be the ones that are less accurate and less performing. If a part doesn't work or is wrongly purchased, the group will have to spend more money to fix this issue. Also, when designing the PCB, it is important to try and design it as accurately as possible in the first try so as not to spend more money changing the design and getting it manufactured possibly multiple times. However, our group is confident in our ability to choose the correct parts that will lead to a working system. We have analysed other similar projects and their parts list and how they choose their specific parts. Through our detailed research, we are determined to find the best working parts for the least amount of money. Our goal is to build our system using less money than the previous project group used. On the opposite side of the spectrum, if we choose to reach out to a company and have them sponsor our project, we possibly would have more access to money and afford the more expensive parts. If that was the case, we could increase the performance and accuracy of the system, with the increased costs. This can include buying more sensitive sensors and more powerful hardware.

Our time schedule for the project is limiting. The project begins in our Senior Design 1 course (EEL 4914) and ends with our Senior Design 2 course (EEL 4915). We started Senior Design 1 in January of 2020 and we will finish Senior Design 2 at the end of July 2020. That gives us around seven months for building this project (refer to the administrative content section for our milestones in more detail). Our glove system must be designed, developed, and built during this seven month period. Due to this limited time, we will have to sacrifice certain features that may not work or features that can’t be implemented within the time frame. While it is a suitable amount of time, we must plan ahead to ensure that our goals are met. Our timeline in the Administrative Content section shows a detailed plan and explicit tasks for each group member to complete their part of the system in a timely fashion. A constraint that we did not expect when planning our timelines, is a national quarantine due to the high contagious COVID-19 virus. The specific factors on the impact of the virus are discussed in the next section.

### 4.1.4 ENVIRONMENTAL AND HEALTH

A constraint that surprised the entire group in the rising pandemic of the COVID-19 virus, also popularly known as the coronavirus. Our team is working in America and like the rest of the country, we were blindsided by the impact this virus has caused to society, the economy, and to our coursework. Our university has closed its campus, which hinders our team’s ability to meet and talk in person. We are forced to hold our meetings via video chats. The lack of meeting in person hinders our creativity and being able to talk about our ideas for the glove system we ease. Also, with the campus being closed, it hinders our ability to conduct tests. For the preliminary testing of the products, equipment such as multimeters are needed. This is not something that is available at home to test the products with there. This will impact the time to build the project and get the testing and improvements done. The virus has also impacted the economy and in turn the electronics industry, making it harder or impossible to order the parts we need. Factories and distribution centers are shutting down due to an outbreak of the virus. Our team will have to wait longer to receive our parts, which will limit our building and testing time period.

Safety is a very important constraint that every engineer must follow and design for. A device that poses a risk and danger to the user and others around them is a failure in our eyes. In designing our system, we will choose the parts with the best safety ratings. The power supply system will not pose any risk to the user and our team while building the glove system. Given that our system is a glove, we must be wary about the heat dissipation the power system will produce. Too hot and the glove will be uncomfortable or cause pain when in use for an extended period of time. This is a key constraint. As we design the PCB and choose the battery system, we must strive for the system to stay at a cool temperature so it does not cause alarm to the user. The steps to achieve this task are making sure the wires in the system are connected properly and separately, the PCB has a strong heat dissipation design, and choosing the correct batteries.

With the many hardware components on the glove, it is important to not have any exposed wires that could come into contact with the skin and produce an electric shock to the wearer. The glove needs to be big enough to cover exposed skin to also prevent this from happening. Enclosures can be created to cover the hardware and make sure nothing comes into contact with the hand and cause safety risks for the user.

## 4.2 STANDARDS

### 4.2.1 BLUETOOTH

Bluetooth was originally standardized as IEEE 802.15.1. A task group was formed to establish a communication standard for low-rate information exchange between compatible devices over short distances using ISM radio frequency. But IEEE 802.1 no longer maintains the standards. Instead, the Bluetooth Special Interest Group (SIG) oversees Bluetooth standards and the licensing of Bluetooth technologies. It would be helpful to understand and heed the recommendations of Bluetooth technology for this project, as Bluetooth plays a large role.

### 4.2.2 LITHIUM-ION BATTERIES

As a polymer li-ion battery will be used to power the Babel glove, it is important to read and understand the safety standards of Lithium-ion batteries. The group should especially use these standards to follow proper management and maintenance of the polymer li-ion battery used. The following standards are for small lithium batteries.

* UN/DOT 38.3 5th Edition, Amendment 1 – Recommendations on the Transport of Dangerous Goods
* IEC 62133-2:2017 – Safety requirements for portable sealed secondary lithium cells, and for batteries made from them, for use in portable applications – Part 2: Lithium systems
* UL 2054 2nd Edition – Household and Commercial Batteries

### 4.2.3 SOFTWARE STANDARDS

On the machine learning software side of our system, there is not an official set of standards for machine learning and artificial intelligence systems. It is still a new and developing technology. As there are no official standards in the machine learning industry, there are community standards that programmers and developers have taken upon themselves to solidify and spread to the community. These standards or guidelines will help our team develop our machine learning algorithm and guide it to produce the best results.

There are multiple community machine learning software out there that support the ‘community standards.’ Our team will be following these frameworks as we develop and build out machine learning model for the BabelGlove. Such software includes Keras, TensorFlow, and Pytorch. These frameworks will help our team build and develop a strong and more reliable machine learning model, that can only help our users when they are translating their American Sign Language on our BabelGlove. The development of these frameworks is more recent as the popularity of deep learning and machine learning has risen. There are multiple different machine learning frameworks to choose from, which gives our team a hard time to decide. With our research and the advice of professors and fellow student, we have decided to work with a combination of Keras and TensorFlow. Our team believe that with this combination we can speed up our training time, create more accurate results, and be able to build a better reliable model.

# 5. DESIGN

## 5.1 HARDWARE DESIGN

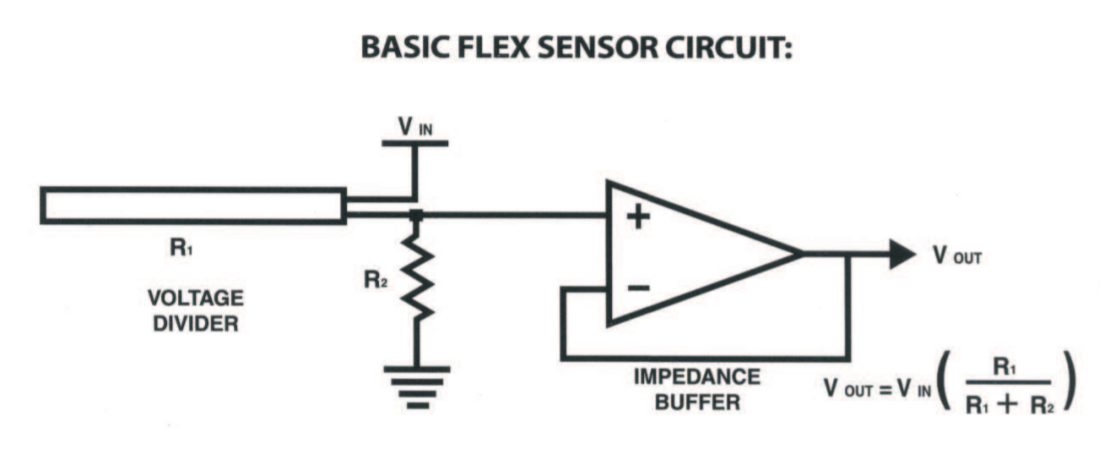
### 5.1.1 FLEX SENSORS

#### 

#### 5.1.1.1 SELECTION AND INTEGRATION

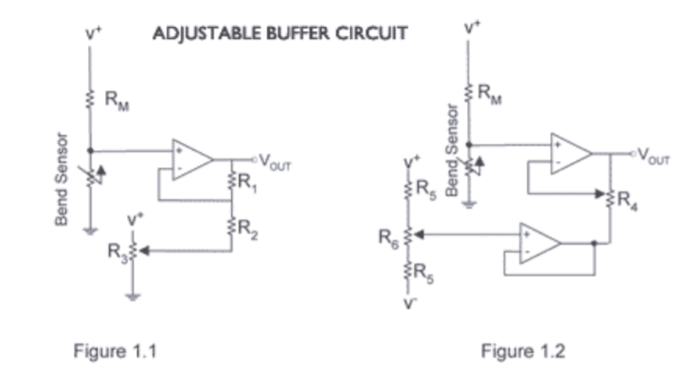
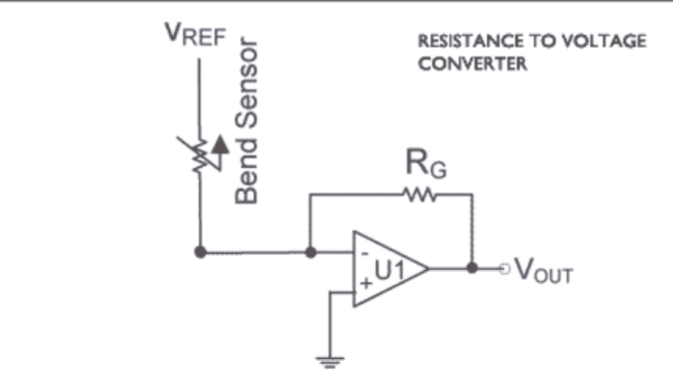
For this project, the team decided to use the 4.5 inch flex sensor by Spectra Symbol. This flex sensor was chosen due to its reputation as a reliable part and also that it matched the specifications needed for this project. These sensors have been used in a variety of real life applications, such as the Nintendo Power Glove. This sensor will be long enough to the fingers on the glove and has a good range to detect the motions as the fingers move in various ways. If needed, modifications can be made to the length of the flex sensor to better fit each finger placement. Five separate flex sensors will be placed on each finger of the glove. An adhesive will be used to secure the sensors in place on the glove. The flex sensors will be the main source of data collection. As the sensors are bent, the resistance will change and an output voltage will be measured. The different voltages will be measured and recorded. This will be used to help differentiate between the different letters and phases.

There are a few different options to integrate the sensor into the circuit. One option is to use a voltage divider with an impedance buffer. The impedance buffer is used to reduce error due to source impedance of the flex sensor as voltage divider. This is done with the low bias current of the op amp. The impedance buffer is a single sided op amp. The suggested op amps to use are the LM358 or LM324.



**Figure 5.1.1.1 - 1: Basic Flex Sensor Circuit**

Another option is to use an adjustable buffer. This uses a potentiometer added to the circuit. It will be used to adjust the sensitivity range. The last option is a resistance to voltage converter. The sensor is the input of the resistance to voltage converter. It uses a dual sided supply op amp. A negative reference voltage will give a positive output. This circuit should be used when a low degree of bending is required.

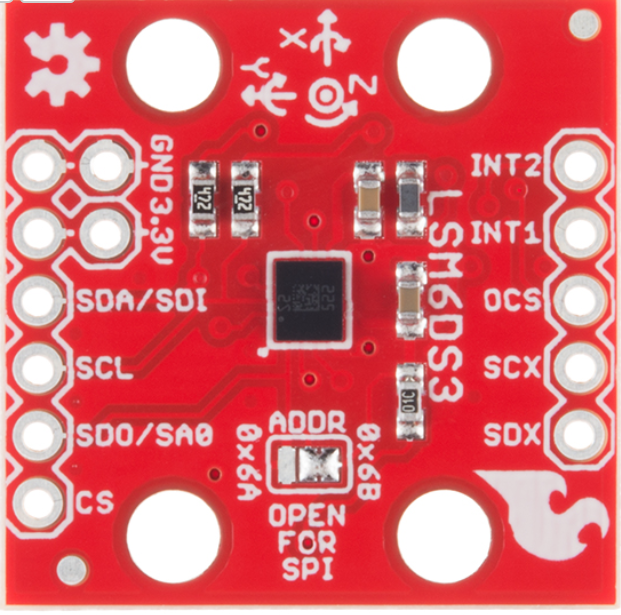
 

**Figure 5.1.1.1 - 2: Adjustable Buffer Circuit and The Bend Sensor**

### 5.1.2 ACCELEROMETER AND GYROSCOPE

#### 5.1.2.1 SELECTION AND INTEGRATION

For this project, the team decided to use SparkFun 6 Degrees of Freedom Breakout - LSM6DS3. This is an accelerometer and a gyroscope sensor. By having both of these elements, the data collected from them will be more accurate. Each sensor has three axis of movement and when combined, this becomes six axis. This combination will allow data to be recorded with six degrees of freedom. It can detect shocks, tilt, motion, taps, and other things. It is small, low cost and has a low power consumption. Both the gyroscope and accelerometer are built into the board. By having the two elements in one part, it is easier to use. It is easier to program and easier to integrate into the circuit. It is designed to be easily configured for many applications. The LSM6DS3 can read accelerometer data up to 6.7 kS/s and gyroscope data up to 1.7 kS/s. This allows for accurate movement sensing. This product has the ability to buffer up to 8 kB of data between reads and can also host other sensors.



**Figure 5.1.2.1 -1: LSM6DS3**

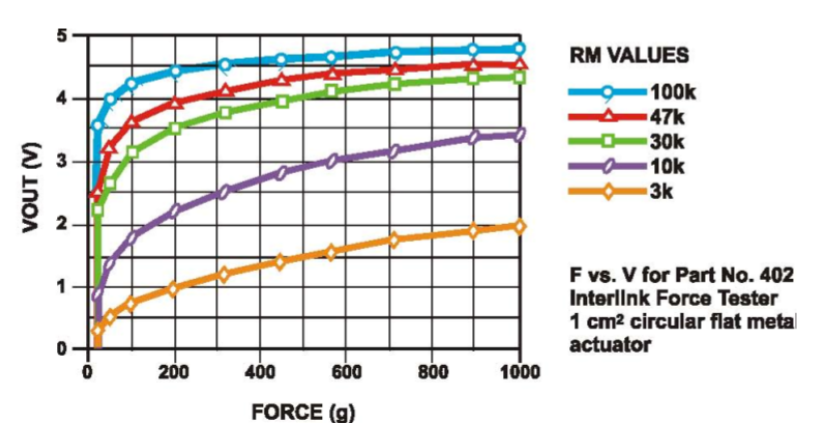
This board is designed to be easily applicable to many applications and should not be difficult to integrate into the project. The easiest way to connect the LSM6DS3 to an arduino compatible board is to use the I2C interface. The wires can be soldered directly to it and plugged into the PCB. The location of this sensor will be on the back of the hand towards the center of the glove.

### 5.1.3 FORCE SENSOR

#### 5.1.3.1 SELECTION AND INTEGRATION

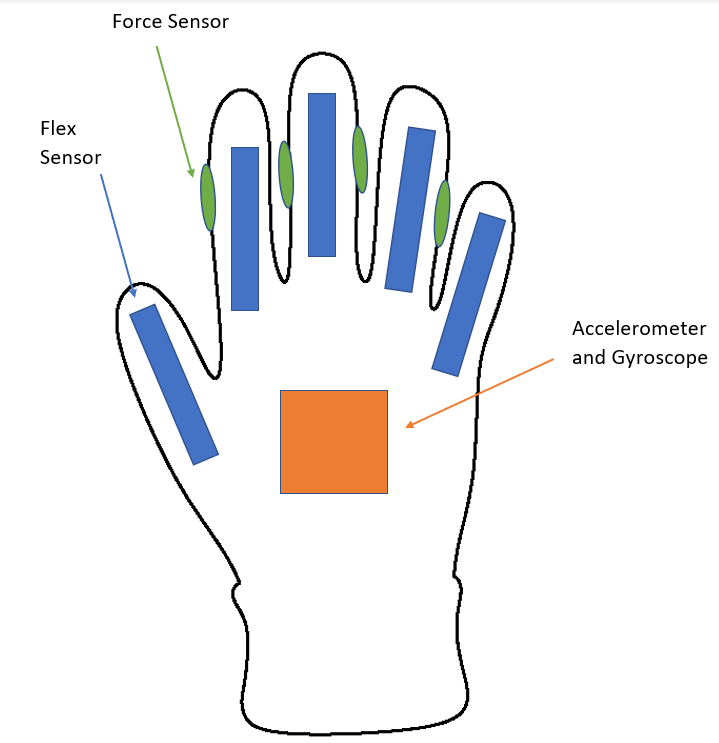
For this project, the team decided to use the force sensitive resistor (FSR) from Sparkfun. The force sensors will be used to determine if there is any contact between the fingers. This is important to help determine what letter or phrase is being signed. These sensors will be placed between the fingers on the glove. The sensor that we choose has a .5 inch diameter sensing region. The resistance will vary depending on how much pressure is being applied to the sensing area. These sensors are simple to set up.

There are several different ways to integrate this into the circuit. The simplest way is to do a force-to-voltage conversion with a voltage divider. The FSR is tied to a measuring resistor and the output voltage is measured. As the force increases, so does the output voltage. The measuring resistor will be chosen to maximize the force sensitivity range and limit the current. This circuit could be followed by an op-amp depending on the impedance requirements.



**Figure 5.1.3.1 - 1: Visual Representation of the Force Sensitive Resistor**

Below is a diagram of the placement of all the sensors on the glove.



**Figure 5.1.3.1 - 2: Custom Diagram of our Glove with Force Sensors, Flex Sensors and The Accelerometer and Gyroscope**

### 5.1.5 PCB

#### 5.1.5.1 DESIGN

The PCB design for the Babel glove will integrate the major components of the glove. This includes the MCU, Bluetooth module, power supply, charging circuit, and the sensors used by the glove. The group has decided to use a two-layer board that is about 3in x 4in due to its basic complexity and low cost compared to using a four-layer board. Through Hole Mounting (THM) will be used for the majority of components as the group members don’t have much experience with soldering in general and through hold components are easier to replace, making testing a prototype easier compared to surface mounted components.

Eagle CAD software will also be used to design the schematic for the printed circuit board. It will also be used to design the layout of the board and generate the Gerber files. It is a popular program for PCB design and allows the user to add additional libraries. The group members also have basic experience in using Eagle CAD through EEL 3926L Junior Design course at the University of Central Florida.

To design the PCB board, additional libraries for the specific parts being used will need to be downloaded into the Eagle CAD software. <include parts>. The autowire feature will be used to make the layout design of the PCB board and the remaining connections will be routed manually using vias from the bottom and top layers.

### 5.1.6 MICROCONTROLLER UNIT

#### 5.1.6.1 SELECTION AND INTEGRATION

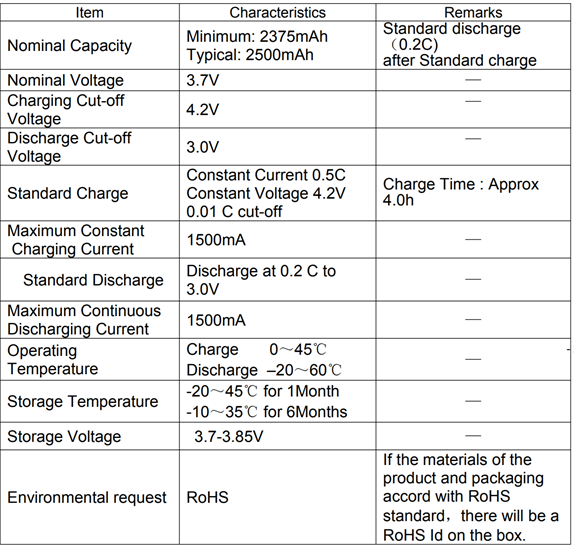
In the end, the group decided to go with the ATMEGA32UA as the Babel glove’s MCU. This is mostly because the ATMEGA32UA is already configured with a USB bootloader compared to the ATSAMD21G which does not. Already having a USB bootloader makes the programming of the MCU a bit easier and helps save time. The MCU will easily be designed on the PCB.

### 5.1.7 POWER SOURCE

#### 5.1.7.1 BATTERY

The group has decided to use a lithium ion polymer battery for the Babel glove because they are light, thin, and powerful. More specifically, the group has decided to purchase a 3.7V lithium ion polymer battery – 2500 mAh from adafruit.com. The battery comes pre-attached with a 2-pin JST-PH connector and includes the necessary protection circuitry. This keeps the battery voltage from overcharging or over-use and will cut-out when the battery is dead at around 2.8V. A Li-ion/Li-Poly constant-voltage/constant-current charger is recommended to charge the battery at a rate of 1200 mA or less.

The battery model is LIPO785060 weights 43g and is 47mm x 61mm x 6.7mm in size. Below is a table of the battery specifications reprinted from the LIPO785060 datasheet.



**Figure 5.1.7.1 - 1: LIPO785060 Battery specifications**

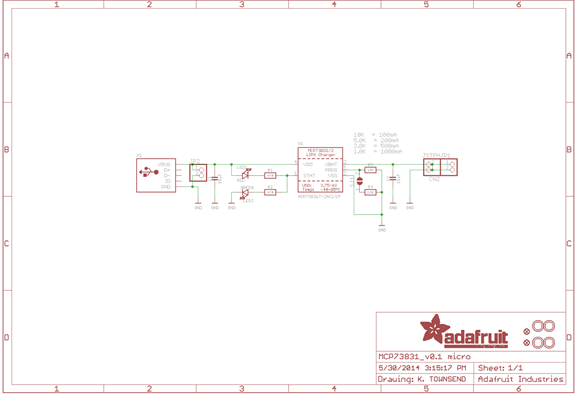
Reprinted from LIPO785060 datasheet.

#### 5.1.7.2 CHARGER

The group has decided to use the Adafruit Micro-Lipo Charger for LiPo/LiIon Battery w/ MicroUSB Jack. This charger is small and easy to use by simply plugging the MicroUSB cable into a USB port and 3.7V lithium ion rechargeable battery into the JST plug on the other end. There are two LEDs to let the user know when the battery is fully charged (green) and when the battery is charging (red).

Charging is done in three stages: preconditioning charge, constant-current fast charge, and constant-voltage trickle charge. The charge current is 100mA by default, which was concerning at first, but this value can be easily changed to 500mA by soldering closed the jumper on the front. Thus, the 3.7V 2500 mAh battery from Adafruit would take 4.5 hours to fully charge.

Below is the schematic of the Adafruit Micro-Lipo charger reprinted from the datasheet.

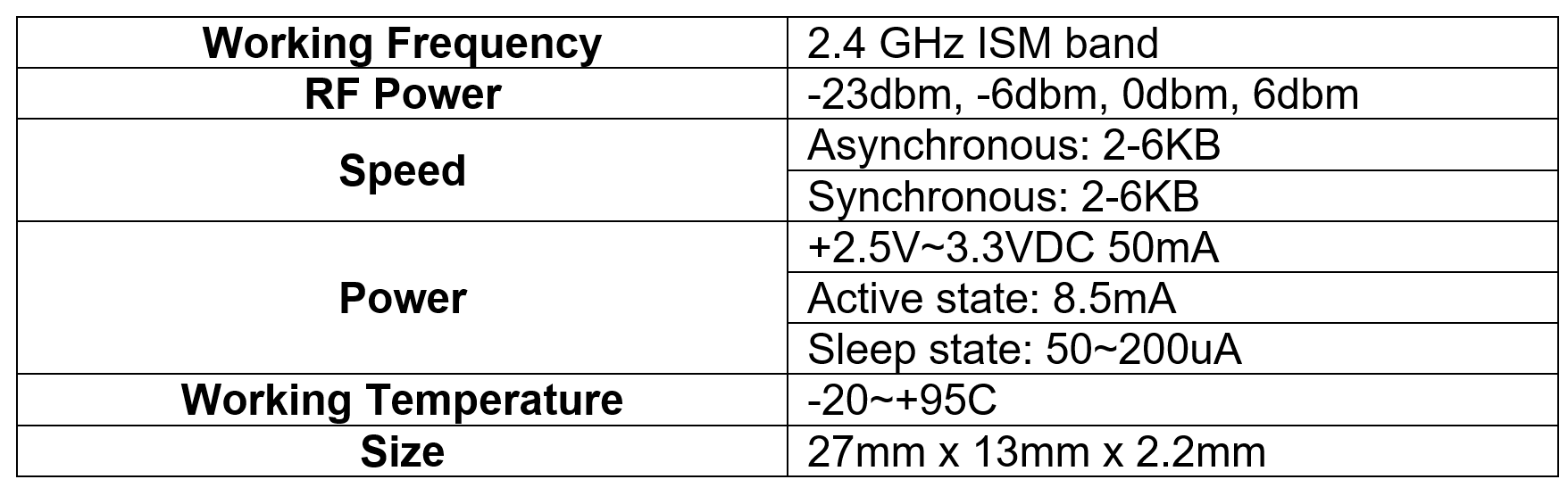


**Figure 5.1.7.2 - 1: Schematic of Micro-Lipo charger**

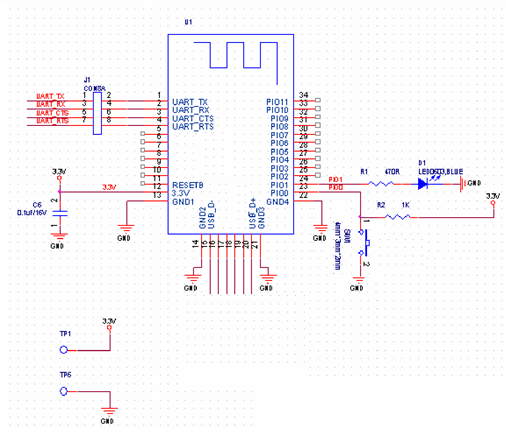
Reprinted from datasheet.

### 5.1.8 BLUETOOTH MODULE

The group has decided to use the HM-10 4.0 BLE module from DSD Tech. The HM-10 uses the CC2541 chip by Texas Instruments. Newer versions of HM were considered, but they use newer versions of Bluetooth such as 4.2 and 5.0, which not all computers support. The HM-8 BLE module was also considered but the only reason to use the HM-8 would be if the group is developing an app on iOS. Below is a table that includes properties of the HM-10.



**Figure 5.1.8 -1: Table of characteristics of HM-10**



**Figure 5.1.8 -2: Schematic of HM-10 BLE module**

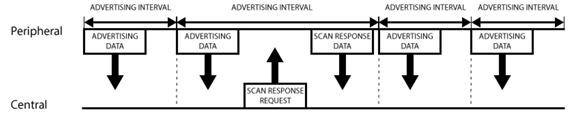
Reprinted from HM-10 datasheet

# 

#### 5.1.8.1 BLUETOOTH GENERIC ACCESS PROFILE

Generic Access Profile, or GAP, controls the connections and advertising in Bluetooth. It determines if two devices can interact with each other and makes the device visible. It defines various roles for devices, but the most important ones are: Central and Peripheral. Central devices are usually more powerful devices, such as a phone or computer while peripheral devices are small, and resource limited. These devices have to connect to a more powerful central device and include things such as a heart rate monitor.

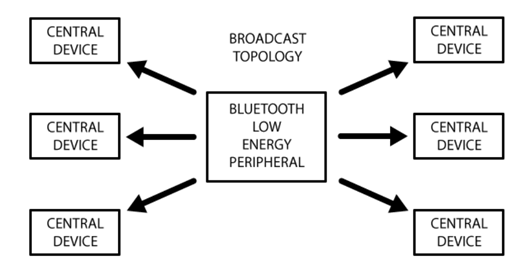
GAP can send advertising in two ways: Advertising Data payload and Scan Response payload. Both the payloads are identical and contain up to 31 bytes of data. What makes them different is that advertising data payload is mandatory since it will be constantly transmitted out so central devices know it exists in range. Scan response payload is something central devices can request and lets device designers have more information in the advertising payload. The way this process works is by a peripheral setting a specific advertising interval to retransmit its main advertising packet. If a listening device is interested, it can request the scan response payload and the peripheral can respond.



**Figure 5.1.8.1 -1: Advertising Process**

Reprinted with permission from Adafruit

There are other situations where only a peripheral is wanted to send data to multiple devices at once. This is done using the advertising packet. By having a small amount of custom data in the 31-byte advertising/scan payloads, a BLE peripheral can be used to send data one-way to multiple devices. This is also known as broadcasting in BLE.



**Figure 5.1.8.1 -2: Advertising Process**

Reprinted with permission from Adafruit

#### 

#### 5.1.8.2 BLUETOOTH GENERIC ATTRIBUTE PROFILE

Generic Attribute Profile, or GATT, is when two BLE devices transfer data back and forth using Services and Characteristics. It uses a generic data protocol known as Attribute Protocol (ATT), which stores Services, Characteristics, and other related data in a lookup table that uses 16-bit IDs for each entry. Once a connection has been established between two devices, GATT starts to become used. It should also be noted that the connections are exclusive, meaning a BLE peripheral can only be connected to one central device at a time.

The peripheral is known as the GATT Server and holds the ATT lookup table and service and characteristics definitions while the GATT Client sends the request to the server. The client for example can be a phone or tablet. The master device, or GATT client starts the transactions and receives responses from the slave device or GATT Server. To establish a connection, the peripheral suggests a ‘Connection Interval’ to the central device which will try to reconnect to see if any new data is being sent.

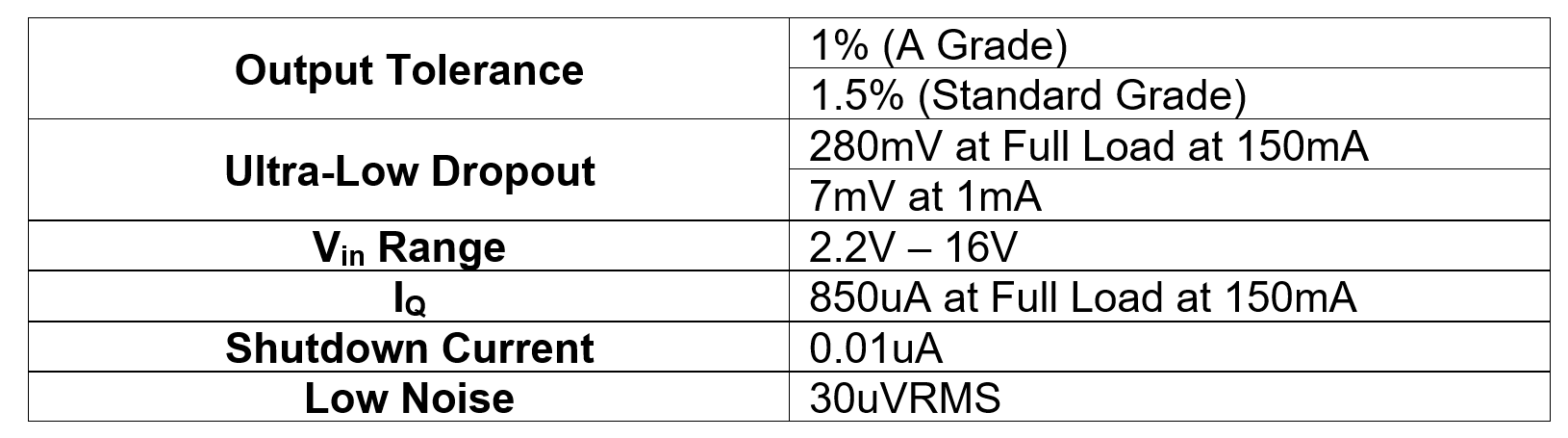
In BLE, GATT transactions are based on nested objects. Profiles are a pre-defined collection of Services that have been compiled by the Bluetooth SIG or peripheral designers. Services break up data into logic entities and contain data called characteristics. Services can have multiple characteristics and each service has a unique ID. Characteristics are the lowest level and is a single data point. They also have a predefined ID and are also used to send data back to the BLE peripheral.

### 5.1.9 VOLTAGE REGULATOR

#### 5.1.9.1 SELECTION

The group has decided to use the LP2985 regulators made by Texas Instruments. These low-noise, low-dropout regulators with shutdown have two grades. A grade has an output tolerance of 1% while standard grade has an output tolerance of 1.5%. They both have an ultra-low dropout of 280mV at full load of 150mA. They are affordable and have great performance in portable applications, making them a great fit for the Babel glove.

The LP2985 regulators are available in voltages ranging from 1.8V – 10V. They have a wide Vin range of 2.2V min to 16V max and a low IQ of 850uA at full load at 150mA. The shutdown current is typically 0.01uA and they have a low noise of 30uVRMS with 10 nF Bypass Capacitor. Overcurrent and overtemperature protection are also included. Below is a table of features of LP2985 regulators.



**Figure 5.1.9.1 - 1: Table of features of LP2985 Regulators**



**Figure 5.1.9.1 -2: Functional Block Diagram**

Reprinted from LP2985 Datasheet.

### 

### 5.1.10 GLOVE

#### 5.1.10.1 SELECTION

The glove selected for this project by the team is the UA Radar Softball Glove. The motive behind this decision was the material the glove is made out of and the ease of movement it provides. Since it is made with leather, there is no concern about possible allergies due to the material. Also, this material is able to support all the electronics and hardware that will be attached to the glove. With its HeatGear fabric, it has a 4-way stretch that improves movement in every direction. The budget was also a big consideration. With the perforations in the material, it will provide comfort and will prevent the glove from getting too hot due to the heat released by the electronics and hardware.

#### 5.1.10.2 INTEGRATION

The glove will be the main component that the hardware and other electronics besides the PCB will be attached to. Many of the sensors will be attached with adhesive material. Other attachment methods will be used if needed.



**Figure 5.1.10.2: The Glove**

## 5.2 BILL OF MATERIALS

Below is a list of materials that were used to create the Babelglove. Some items, such as resistors, wiring, and other consumable items, have been omitted from the list. As seen throughout the paper, there were a lot of considerations to take into account and research done to find the best products that meet the specifications of the project. One of the factors that went into picking the materials is cost. This project is fully funded by the members of the group. This is the cost to build one glove.

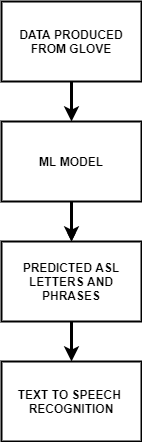
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Vendor** | **Part Number** | **Quantity** | **Cost** |
| Flex Sensor | SparkFun |  | 5 | $15.95 |
| Accelerometer and Gyro | SparkFun |  | 1 | $10.95 |
| Force Sensor | SparkFun |  | 5 | $6.95 |
| MCU - ATMEGA32U4 | Digikey |  | 1 | $3.98 |
| HM-10 Bluetooth 4.0 BLE Module | Amazon |  | 1 | $9.99 |
| UA Radar Softball Gloves | Under Armour | Style #1299550 | 1 | $24.99 |
| Adafruit 328 Battery, Lithium Ion Polymer, 3.7V, 2500mAh | Amazon |  | 1 | $20.59 |
| Adafruit Micro-Lipo Charger for LiPo/LiIon Battery w/ MicroUSB Jack | Amazon |  | 1 | $19.39 |
| LP2985 | TI |  | 5 | $0.55 |
| PCB | Bay Area Circuits |  | 3 | $90 |

**Figure 5.2: Bill of Materials**

## 

## 5.3 SOFTWARE DESIGN

This section dives into our decisions for our software designs. We discuss the software prototype development, programming language used, the environments we have chosen, the machine learning algorithm, and the graphical user interface based on the research done in the research section.



**Figure 5.3.1 - 1 Overview of the Software Prototype**

### 5.3.1 SOFTWARE PROTOTYPE

It is important that we create a software prototype in our system’s early developmental phases. This is so we can create the hardware and test alongside the software and optimize the software to the limitations of the hardware. The software prototype development will start alongside the hardware development at the end of our spring semester and into the start of the summer semester. We are striving for a fully developed prototype by the start of our summer semester. On the software side of the prototype it must include the machine learning algorithm, as it is a very intense part of the software with lots of complex mathematical functions. We also have a strong accuracy in our system choosing the translation of the ASL letters and phrases, so the more time our team can have to improve on the machine learning part, the better. The figure below shows the overall view that we have decided for the software prototype:

### 5.3.2 PROGRAMMING LANGUAGE

The main programming language chosen is Python. Python is our team’s programming language of choice because of its ease of use, wide range of application and support library for machine learning problems. We will use Python for the development of the protype, as our goal for the prototype development is to build a working system as guilty as we can, so we have time and resources to find the major faults or add other features to improve upon our original design. Programming our software in Python will help our team achieve that goal. Our software developers on our team are very familiar with Python and can build and develop a working prototype in a timely manner.

The language used on our board uses Arduino. Arduino is a set of C/C++ functions that you can call in the code. The Arduino board is a AVR development board, stands for Alf and Vegard’s RISC processor, they are a family of microcontrollers that use flash memory for program storage. This is a strong benefit for our system as it means our software embedded into the board will executive fast and more efficiently than other boards. The board that we have chosen can be programmed in ARV C or C++, but for developing our prototype we have decided to stick to the Arduino code, as it is designed to be quickly implemented. Our team is confident in using the Arduino programming language to programming our microcontroller to collect the glove sensor data and transmit it to the computer.

### 5.3.3 CODING ENVIRONMENT

In our research section of this paper, our team discusses the best and most useful Integrated Development Environment (IDEs) for developing our system’s software. Some of our favorites were Visual Studio, PyDev and IDLE. Our team’s choice is to develop in the Visual Studio Code environment. Our team feels confident in developing and programming in Visual studio’s custom coding environment. Visual Studio Code offers many features that we considers in our decision:

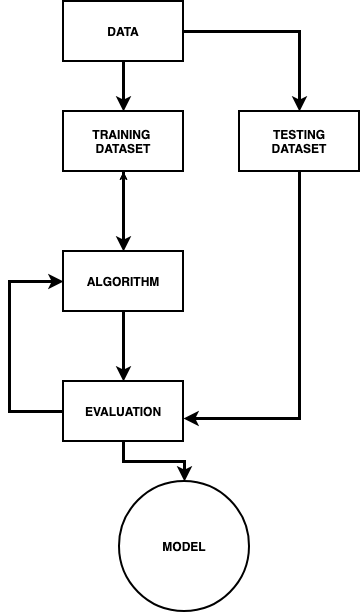
* Strong Source Control interface with Git
* Interactive Debugging solutions
* Vast and useful libraries, especially for python
* Open Source, honest and secure

It supports many community made plugins that can help enhance the development of our Python based software. The software engineers on our team have used Visual Studio in the past and the python based plugins. They are also excited to work with the machine learning plugins that the community has made for a fast and more effective implementation of machine learning algorithms. They also favor the minimalist aesthetic that Visual Studio features, helping our programmers focus on their important programs.

The coding environment for our ATMEGA32UA microcontroller will be in its native Arduino IDE. The Arduino IDE streamlines the task of writing the code and uploading it to our board. It will allow the developer of the board to write and execute their code on the board in real-time. Arduino IDE gives lots of on board functions that help with ease of use when coding for the MCU. Its goal is to help the developer focus on the code and develop the core function of the MCU. The IDE is also open sourced which we love as a team. Open source software is usually better and more honest than other software, also it is free to use, which ties in with our cost constraint. The Arduino IDE comes with extensive libraries and examples of code that will help fast track our prototype development, but we have to be careful with these examples, as they will not be optimized for the purpose of our system. We will focus on optimization and fine tuning our code during the final development of the system. The Arduino IDE is different from other IDEs, as it's made for its own programming language, the programs written with the IDE are called sketches. The sketches are what the IDE imports into the microcontroller to program it. The editor has many features for manipulating the user’s program and gives useful feedback while saving and exporting. Our team is excited to use this intuitive IDE to program the advanced arduino MCU for our system.

### 5.3.4 MACHINE LEARNING

This section goes into the detailed analysis of the design of the chosen machine learning algorithm used in our glove system. Our team has decided to design a machine learning algorithm to determine the ASL hand signs that our users will be making with the glove. We choose machine learning as opposed to hard coding the hand signs because machine learning offers greater scalability. Our team could write a program for each letter of the alphabet in ASL and export to our MCU, but we want the system to be more. With machine learning we can show the computer new hand signs and it will know how to recognize them for future users, this leads to less work for our team having to hard code every new hand sign. The problem with machine learning is designing it properly, that requires an algorithm that can learn and classify the different signs and accurate dataset for training.



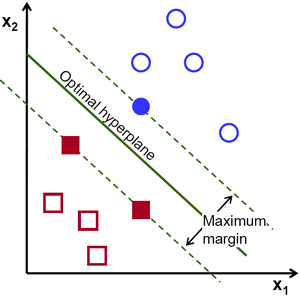
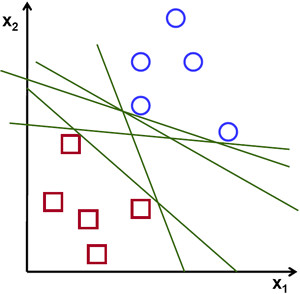
**Figure 5.3.4 - 1 State Machine for Supervised Machine Learning**

In order to train the machine learning algorithm, our team has to create a dataset for the hand signs we want the computer to recognize. With the glove designed properly, we can make the hand signs and get the sensor data. Once we have a lot of similar differentiation of the same hand sign, our team will label the data as the appropriate hand sign and then train it in the supervised machine learning system. We will do this for every hand sign that we want the computer to recognize. It is important to get an accurate data set, but we do not want them too accurate. We can not simply make the ASL hand sign once and then fill the dataset with a copy of that instance data. Our team must be careful and create multiple variations of the same hand sign, which means completing the hand sign multiple times to receive those small variations in data for the algorithm. The machine learning algorithm will see those variations and find the range of data the hand sign is to be in. We must do this because our users will not have the exact same finger and hand placement each time. There will be strong variations, which the computer will struggle with, unless we help it with these imperfect datasets. Another tactic we can use to vary the datasets to help train the machine learning model is to use different hand sizes when collecting the data. Same as the finger and hand placement with each user, the user’s hand will be different each time. If we use different hand sizes in the training dataset we can train the algorithm to respond better when presented with a user with a vastly different hand size than the ones trained in our dataset. Our dataset will be split 80-20, 80% dedicated to training and 20% dedicated to testing.Figure 5.3.4 -1 shows the flow of the machine learning system. From splitting the original data set and training the model based on the training set and evaluating the algorithm’s with the testing set.

There is a wide variety of different machine learning algorithms. In our team’s research we have narrowed it down to the category of classification. We are looking for a strong multi-class classifier for an algorithm as we will have our model designate different ASL hand signs or classes in our system. We will look at different classification algorithms used by the data science community that would benefit our system and our machine learning problem. The most possibing classification applications that our team can implement and work with are Support Vector Machines and Neural Network. There is more research about the two implementations and their pros and cons below.

#### 5.3.4.1 SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is used in the data science community as a regression and classification algorithm. In our team’s case we will be using its classification features. The simplest version of SVM's main goal is to find the widest gap between the different categories. SVM excels at high dimensional spaces, or datasets with a large number of attributes, which will benefit our problem. Our system takes five flex sensors and four force sensors, the SVM can process those attributes and find a hyperplane in the N-dimensional space (N being the number of features) that distinctly classifies the data points. Figure 5.3.3.1-1 below shows the possible hyperplanes between two distinct classes.



**Figure 5.3.4.1-1: The Possible Hyperplanes in SVM**

The hyperplanes created by SVM are decision boundaries, the separating line of data points that classifies those data points into a category, in our class a hand sign or another. The algorithm uses support vectors to influence the position and orientation of the separating line or the hyperplane. Support vectors are the data points that are closest to the hyperplane, they are the points that build the SVM and determine its goal to maximize the margins of the classifier. The best case scenario with using SVM in our team’s case is the machine learning model will learn to classify and separate the ASL letters and phrases from each other so it can properly identify what the user is signing. The farther away the classes are from each other the easy and more flexibility the user will have at signing. There will be a larger margin of error for the user in his or her signing.

Using the SVM algorithm, our team will also be implementing the loss function known as hinge loss. Hinge loss helps maximize the margin between the data points and the hyperplane, which is our goal in our application. An equation for hinge loss can be seen below:

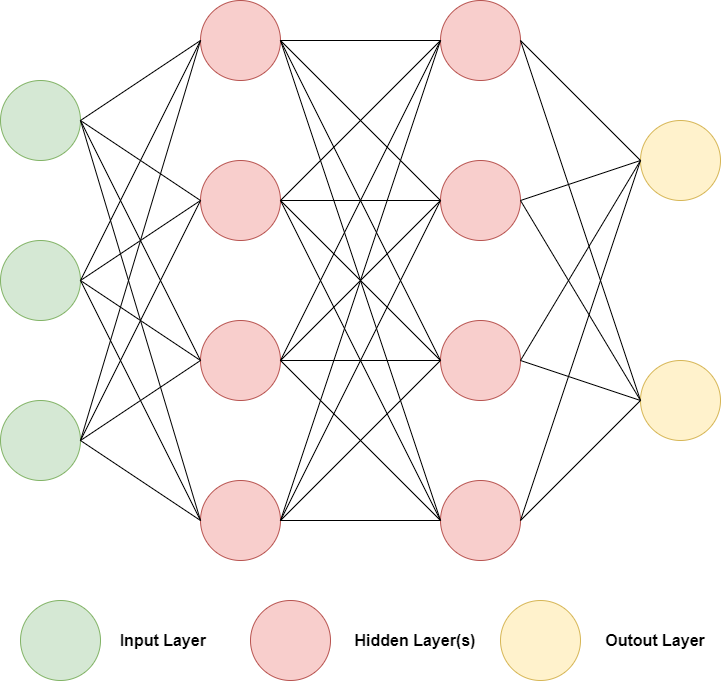
The hinge loss function is a convex function which will make working with convex optimizers a bit easier in the machine learning domain. The function works well at penalizing points that are misclassified, those points would be placed far away from their other points in that class. The loss calculated when misclassified increases exponentially which results in the SVM algorithm focusing on fixing the misclassification. The loss function allows for some margin of error in the SVM algorithm. As shown in Figure 5.3.4.1 - 1, the SVM algorithm creates a margin of error between the classes. When a point goes over that margin of error the hinge loss function attempts to bring it back and fix the mistake.

#### 5.3.4.2 NEURAL NETWORKS

The other option our team has researched about is the method of Neural Networks. An Artificial Neural Network is a layer of networks that can learn to transform its input raw data into more abstract and composite representations. This implementation of artificial intelligence falls under the Deep Learning category. Deep learning essentially is about using multiple layers to extract high level features from the input data, it was designed based on the complex structures of our brain. Supposedly simulating the neurons in our heads talking to each other. Neural Networks have become a lot more popular in recent years as we have had more data to process and with that increase in data we have had access to a vast support of computing power to process all the data. There are lots of tools out there that developers can pick up and start programming deep learning models. TensorFlow and Keras are the most popular for building large neural networks.

The architect of a neural network is complex. There are hidden layers filled with neurons that are all connected with each other. The last hidden layer output the values from the values of the connected neurons. Figure 5.3.4.2 - 1 shows the complex interconnection of a hypothetical two layer neural network. The flow of a neural network doesn't just go from the input data, through the hidden layers and then out at the end. In order for the neural network to learn and adjust itself it has the ability to go back through and adjust the weights and biases of the neurons in the hidden layers. This is called Backpropagation, as opposed to forward propagation when the network goes through all the layers to output its predictions. An artificial neural network is defined by these parameters:

* Interconnection, the connection of each of the neuron in their hidden layers
* Forward Propagation, the network moving through the layers and predicting its weights and bias
* Backpropagation, the network learning from its mistakes and adjusting its weights and biases.



**Figure 5.3.4.2 - 1: Architecture of a Neural Network**

Neural Networks are very versatile. Developers and engineers can use them with different machine learning methods and some other artificial intelligence methods. For our implementation of neural networks, our team would use the supervised learning approach, which is good for pattern recognition. This would suit our need for sequential data or gesture recognition.

### 5.3.5 GRAPHICAL USER INTERFACE

In Section 3.4.4, we researched and discussed the pros and cons of creating an app on a mobile platform versus a desktop platform. In this section, we will be discussing our chosen platform for application development.

The platform we will be developing our application for the user interface on will be a desktop platform due to the costs associated with the mobile platforms listed below:

* For the Android platform, we would need to obtain an Android device
* For the iOS platform, we would need to join the Apple Developer program which has a yearly membership fee of $99

We decided that the costs associated with developing an application for a mobile platform weren’t worth it for what we needed our application to do. We would be able to design an application with the same features at a lower cost on a desktop platform. Between creating a Java Swing application and a Universal Windows Platform application, we decided the app would be more accessible on the Universal Windows Platform since it can be downloaded on any device that runs Windows 10.

An important part of any product or service in the commercial market is that it has to be user-friendly. An easy way to make sure that our product is marketable and user-friendly is to have a good layout and design of the application to go along with the glove. The purpose of the interface will be to provide the user visual feedback based on the gesture that was made with the glove. By using the Universal Windows Platform, we can also use Microsoft’s Fluent Design System which gives developers a set of tools to make attractive applications.

To make our project more user-friendly, we will opt for a more minimal and simple user interface so that the user can focus on the function of the product, rather than trying to figure out how to navigate the interface. The main design considerations when it comes to the user interface is the application layout, functionality, and appearance. We believe that the tools associated with the Universal Windows Platform will be very beneficial in creating an easy-to-use application for our user interface.

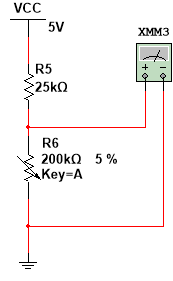
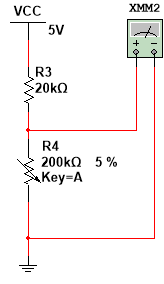
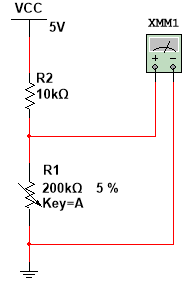
## 5.4 HAND GESTURE RECOGNITION

### 5.4.1 INTERPRETING DATA FROM FLEX SENSOR

The first step in allowing the glove to recognize hand gestures is to interpret the data from the flex sensors that are going to the microcontroller. The data taken from the flex sensors will be varying resistor values depending on how much the fingers bend for each finger when signing a letter or phrase. For the microcontroller to differentiate between the resistor values for each letter and phrase, there cannot be other data interfering with it. This includes making sure the current is constant. The microcontroller will perform some calculations to determine the resistance across the flex sensor. A voltage divider circuit will be used to help measure the change in resistance of the sensor. The sensor will be placed in series with a resistor of a fixed value and a constant voltage will be applied across both resistors. As the sensor is bent, the resistance value across the sensor will change and the ratio of the voltage across the sensor will change. This voltage value will be the output data that gets sent to the microcontroller.

For the voltage divider circuit, there are several things to consider. The value for the fixed resistor will need to be calculated. The flex sensors have a flat resistance range of 10 kΩ and a bend resistance range of 60kΩ to 110kΩ. The voltage level that is hooked up to the two resistors needs to be optimized and the optimal voltage level that the microcontroller can handle also needs to be considered when choosing the value for the fixed resistor. The voltage source for the voltage divider will be from the power source.

Simulations were run to determine the ideal value for the fixed resistor. A voltage divider circuit was simulated using three different values for the fixed resistor. For the flex sensor, a variable resistor of 200kΩ was used. The three different values used for the fixed resistor were 10kΩ, 20kΩ, and 25kΩ. A voltage of 5 V was used. First, the value of the variable resistor was set to 5%, which will give a value of 10 kΩ resistance. This is the value of the flat resistance. Then, to get a value of 60 kΩ, the variable resistor was changed to 30%. This is the lower limit of the bend resistance range. Also, the variable resistance was changed to 55%, which would give the value of a 110 kΩ resistor. This is the upper limit of the bend resistance range. The values for the output voltage are seen in the table below.



**Figure 5.4.1 - 1: Simulated Resistance Values**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Flex Sensor/Variable Resistor Values** | | |
|  |  | **10 kΩ** | **60 kΩ** | **110 kΩ** |
| **Fixed Resistor Values** | **10 kΩ** | 2.5 V | 4.286 V | 4.583 V |
| **20 kΩ** | 1.667 V | 3.75 V | 4.231 V |
| **25 kΩ** | 1.429 V | 3.529 V | 4.074 V |

**Figure 5.4.1 - 2: Table of the range of resistance values**

For all of the values of the fixed resistor, there is a good range of values for the output voltage. These values are all within the normal range of voltages for the average microcontroller. There is also a big enough difference between these values to tell what position the sensor is in. The resistor values may change as the group conducts testing and builds the prototype, however, these seem to be the best values for the resistors at the moment. The microcontroller will receive these voltage values from the flex sensor and from there, these values will be used to help differentiate between the different letters in the alphabet and different phrases.

Finding the ideal voltage levels for the voltage divider is very important. If the wrong values are used, there is a risk of damaging the microcontroller or the data not being sent to the microcontroller. The maximum output voltage will occur when the flex sensor is fully straight. The flex sensor will have a resistance of 10 kΩ. The output voltage needs to be low enough not to damage the microcontroller or be out of the range of voltage that the microcontroller can handle. When the fingers are fully bent, the resistor value of the flex sensor will be at the maximum of 110 kΩ and the voltage output will be the minimum. This voltage value needs to be high enough to still be in the range of voltages of the microcontroller but low enough to have a big range of values to accurately be able to distinguish the data.

The values found in the simulation may have to be adjusted as testing is done using the actual sensors, resistors, and voltage source. Based on the simulation using a voltage source of 5 volts, the ideal fixed resistor is around 20 kΩ. This will create a good range for the output voltage that will be going into the microcontroller. When the sensor is fully straight, the output value will be about 1.67 volts and when the sensor is fully bent, the output voltage will be about 4.23 volts. There is a big difference between these values, which will make it easy to identify the change. This is also a good range for the microcontroller.

The output voltage value will go into one of the pins of the microcontroller. An analog to digital converter will be needed since the microcontroller is a digital processor and the voltage from the sensors is an analog signal. The microcontroller can only interpret digital data so an analog to digital converter is needed. The analog to digital converter can either be included on the microcontroller or it will need to be an external on the PCB.

### 5.4.3 CALIBRATION

Calibration of the glove is a critical step in making sure the glove will be able to differentiate between the different letters and phrases in ASL. There are several factors to consider. One is that the size and shape of everyones hand is different. Another is that each person does not gesture in the same exact way. Everyone is different, whether it is in the hand motion or how much one person bends their fingers in comparison with another person. To help solve the issues that the differences in motion might create, the glove will need to be calibrated before it can be used. This will help the glove become familiar with the motion and gestures that the person who is using it will make and help improve the accuracy of the glove.

There are several different pieces of data that will need to be collected from the user to calibrate the glove. The first is the range of motion of the person who will be wearing the glove. To get this data, the person will have to wear the glove and power it on. Then, the software will initiate the function for calibration. One part of the calibration is having the user bend all their fingers so that the values for the maximum and minimum amount of bend for each finger can be recorded. This is important because each user might bend their hand by a different amount. If this step were not done, even if two people were making the same gesture, it would not be recognized as being the same due to the different values recorded when bending the fingers. By having the user bend each of their fingers in the calibration stage, the system will make a note of the values and will adjust the data accordingly for each letter and gesture. The system will get a reference for the maximum and minimum amount of bend and make decisions about the hand gestures according to this information.

Since not everyone keeps their fingers straight when they are not used for gesturing, the minimum amount of bending is very important. By calibrating the minimum amount of bending of a person's fingers, this will replace the value for the fully straight hand position. This will allow the person to be able to fully relax their hand without having to worry about keeping their fingers fully straight when they are not gesturing. Once all this data is collected, the program will be able to use this to help differentiate between when a person is at rest and not gesturing to when a person is bending their fingers and gesturing. Then, using the incoming data when a person is gesturing, mapping will be used to distinguish between each gesture. Depending on the minimum and maximum values of the resistance of the flex sensor for a person, there will be a range of output voltages produced. These values will be used to tell the letters and phrases apart.

The calibration process starts when the microcontroller is first started up. If the glove needs to be calibrated again, the data will need to be reset by resetting the microcontroller. This can be done by pushing the reset button on the PCB or restarting the microcontroller. This will allow the microcontroller to restart again and allow the calibration process to reinitiate.

### 5.4.4 DIFFERENTIATION BETWEEN SIMILAR HAND GESTURES

### 

One of the challenges that will occur is differentiating between hand gestures that are similar and uncommon. Some hand gestures require movement that will not be captured by the flex sensors or will be too similar to distinguish just based on data taken from the flex sensors.To prevent these issues, the glove will be equipped with other sensors to collect more data.

One other sensor included on the glove is the accelerometer and gyroscope.The accelerometer will measure the hand’s position with respect to the x, y, and z axis of the Earth. With this data, the glove will be able to sense if a hand is tilting in a certain direction. Some letters require hand motion such as this. For example, the letter J requires the letter to be drawn in the air with the pinky. This motion will be detected by the accelerometer and the data will be sent to the microcontroller.

Since the accelerometer can only measure the changes when the fingers bend, the accelerometer will be important when gestures that also require the hand to change orientation occur. Experimentation will have to be done with the accelerometer to see what data can be gathered from it and how it can be used to help differentiate between letters and phrases. The data gathered from this along with the data gathered from the flex sensors will be used to tell what letter or phrase is being used.

Another challenge is that some letters are very similar. For example, the letters K, U, R, and V are very similar in that they both use the middle and index finger in an unbent position. By just using the data from the flex sensors, there would be nothing to distinguish between these letters. The glove needs more sensors to distinguish between these letters. Looking at the letters R and U, these are almost identical. The distinguishing factor is that the index finger overlaps the middle finger for the letter R. When looking at these two letters, it is clear to see that the amount of bend in the fingers is almost the same and it would be hard to distinguish the letters just based on data from the flex sensors. One way to solve this problem is to use pressure sensors. This would be able to tell if the fingers were touching. With the pressure sensors, the fingers would not have to be in full contact with each other. The data gathered from the pressure sensor along with the other data from the flex sensors will be able to help distinguish the letters.

# 6. CONSTRUCTION

## 6.1 TESTING

When creating a test plan for the hardware and software components of our glove, it is important that we develop ways for us to test the components independently so that we don’t waste time waiting for one component to be ready. It is especially important due to recent events relating to the Covid-19 pandemic that we are prepared to test the components separately since our group may not be able to meet face-to-face until later in the summer semester.

One consideration for testing is where it will take place. Since most of the hardware will be exposed on the top of the glove, it will be important to conduct testing in a location where weather conditions will not affect the functionality of the glove. It is also important to test somewhere where the appropriate equipment is available and any changes can be made quickly and efficiently if needed.

## 6.1.1 HARDWARE TESTING

Most of the hardware can be tested independently of the other components of this project. The hardware components will be used together to provide data to the microcontroller to distinguish between the letters. All of the hardware sensors will be connected to the outer part of the glove. The hardware will be tested at least once before all the components are put together for the glove. The hardware components include the flex sensors, the accelerometer and gyroscope, and the pressure sensors.

### 6.1.1.1 FLEX SENSORS

To test the flex sensors, a voltage divider circuit will be assembled as previously mentioned. Power will be supplied to the circuit and the voltage across the flex sensor will be measured. This will be done by connecting a multimeter in parallel to the flex sensor. As the flex sensor is bent, the voltage should be decreasing. The change in voltage is what will be collected and recorded by the MCU.

Once the prototype is created, the MCU will be able to collect the data collected from the sensors and interpret it. The different voltage outputs from the voltage divider circuit will be linked to a certain letter. The varying voltages will be due to the different degree of bending from the flex sensors. As the fingers are bent more, the flex sensors will bend more and the voltage reading will lower. Using the program, the first round of testing will be using the flex sensors to see if the letters can be recognized just by bending the fingers. The motion for each finger will be repeated several times until the correct letter is registered and displayed. If there are still inconsistencies in the data, the flex sensors can be adjusted. This will be repeated until the correct letter is displayed.

### 6.1.1.2 ACCELEROMETER AND GYROSCOPE

### 

Since there are two different sensors attached to one board, more than one test will need to be done to make sure both components are functioning properly. First, the accelerometer will be tested to ensure that it is able to identify the orientation and tilt of the hand. This will be important to help distinguish the different letters and phrases since many individual letters and phrases use these hand movements. One letter that can be used to test the accelerometer is p. To create the letter p, the hand is in a downward position. If the accelerometer is connected correctly and the program is running correctly, then the letter p should be registered when the hand is in a downward position. If the hand is not in the downward position, it should not read the letter p even if the fingers are in the right position.

The gyroscope will be tested in a similar way to the accelerometer. A different letter will be used to test the movement. The letter j requires the hand to move in a swinging motion. If the glove recognizes this movement then that means that it is functioning properly. One difficulty of this is that the letter i is very similar to the letter j. The team will need to make sure that the glove can recognize the motion and distinguish between these two letters. Several repetitions will need to be done to make sure that it reads properly. This will also help increase the accuracy of the readings in outputting the right letter.

### 6.1.1.3 FORCE SENSOR

The force sensor can be tested in a very similar way to the flex sensor. Using the voltage divider circuit, the voltage will be measured across the force sensor. Instead of bending the sensor as is done with the flex sensor, pressure will be applied to the sensing area of the force sensor. It might be difficult to apply a consistently increasing force but as long as it is increasing and the measurements are correct, then the sensor is working correctly. As the force applied increases, the voltage should decrease.

With the prototype, the main purpose of the force sensors is to collect extra data to help distinguish the different letters. For example, the letters k, u, and v are very similar to one another. The main difference between the k and t is that the thumb rests in between the middle and index fingers for the k and is not for the v. The difference between the u and the v is that for the u, the fingers are touching each other while for the v, the fingers are spread apart. The force sensor will be able to help differentiate between these letters in this scenario. Many repetitions will need to be done to ensure that the glove can tell these letters apart.

## 6.1.2 SOFTWARE TESTING

Similar to the hardware testing, the software can be tested independently of the other components of the project, with some added limitations. Because we are using the signals from the sensors to train the glove, we will need to have some hardware portions of the glove completed before we can begin testing the machine learning portion of the software to gather data. An issue that we expect to run into is having a low accuracy when translating gestures. Since some of the letters in the ASL alphabet are very similar to each other (see Figure 3.3-1: American Sign Language Alphabet and Numbers), we expect that the software will have some initial difficulty in telling apart the gestures. Due to the nature of our software and our utilization of Machine Learning, the more testing that we do will increase the accuracy of the gesture recognition.

### 6.1.2.1 MACHINE LEARNING

Testing the machine learning algorithm comprises a test harness. The goal of the test harness is to quickly and consistently test algorithms against a fair representation of the problem being solved. Results of the testing harness will give a fair assumption to how learnable our problem, translating ASL to english text and speech, is. Our team’s software developers will be testing a variety of machine learning algorithms to find the best one for our problem. There is also the question of performance testing for our machine learning problem. We can optimize our program by implementing rigorous performance testing. In our team’s case, we are using a classification algorithm, there are many standard performance measures designed by the machine learning community, such as classification accuracy for the model choosing the correct classification.

Given that our team will test and optimize our machine learning algorithm to the best of our ability, we will never get a completely accurate model. That is the nature of machine learning, it is not an exact science. A factor that can increase the model’s accuracy is, testing with new data, rather than the original training data. It is also important to split our data into a training set and into a test set. The Training set should be most of the data, around 80% is a good amount, and we input it into our algorithm for the neural network to actually train the model. The test set should be a smaller set usually around 20%, we use this set to test the algorithm and then use those results to update the parameters of the algorithm as needed. This important check and balances more efficiency trains the model to find the hidden patterns in the data. The bottom line is, as a team we have to be humble and accept the fact that our model is not going to be perfect, we will have to settle for an accuracy lower than 100%.

In developing our training data and training a machine learning model we must be wary of all the different variables to keep track of that can skew our team’s results. An important factor is generalization. Generalization is the model's ability to give reasonable results to new unseen data. The goal of our machine learning models is to generalize well, that means it has to have a balance between overfitting and underfitting the training data. If a model overfits the particulars of our training data it will lead to misplacing the prediction of new data. This can be caused by overcomplicating the model, as it has a low loss on the training data but a high loss on the test data. In machine learning we strive for a low loss on the test data because it is the unseen data with labels that test the model on how well it can process new unseen data in the environment. Underfitting is the opposite of overfitting, when the machine learning model stops the learning too early and doesn't learn enough about the patterns in the training data. This can lead to the similar problems in overfitting.

During our team’s testing of machine learning, we might find that the results and accuracy are not enough and there is nothing we could do to increase the accuracy of the planned system. In that case our team will try to change the machine learning system that we planned to do. There are lots of different ways to get better results, our team might find a better and more efficient way to complete the system. Another classifier algorithm or a different loss function. The data that we receive from the glove might be a bit different than what we planned for, so we reserve the right to change and adjust our machine learning system.

### 6.1.2.3 USER INTERFACE

The user interface that we will be using to display the output to the user should be able to be tested independently of other components of the glove. The layout of the desktop application will be able to be tested with placeholder information/data so that it will not depend on the machine learning library to be created first before testing the interface. Multiple tests will have to be done when the machine learning portion is complete to make sure that the translation of the gestures is being output properly on the desktop application.

## 6.1.3 PASS/FAIL CRITERIA

To make sure that the entire system is functioning properly, our overall test plan for the system will ensure that our glove’s accuracy meets a certain threshold. For our project a pass would be defined as the system being able to correctly interpret the gesture and translate the gesture into text, and a fail would be defined as the system incorrectly interpreting the gesture.

Our goal for this project is to have an overall 85% accuracy rate which means that when a user uses our glove, 85% of the gestures that they create with the glove should be given a pass, which means that the gesture was interpreted correctly.

We will be keeping track of each of our testing sessions and make note of which gestures are consistently not being interpreted correctly. From these tests we will be able to go back and make adjustments to the system to reduce the fail rate. By the end of the testing, we should be getting around an 85% accuracy rate.

## 6.2 FACILITIES

To complete this project, the group plans on using the Idea Lab in the UCF Engineering II building. This room has glass walls that can be written on with whiteboard markers and has fun chairs and great lighting. A lot of the members have used this lab for their own projects from other classes and club projects. It’s located right across the Texas Instruments Innovation Lab making it easy to quickly brainstorm ideas and prototypes then going immediately to implement them.

The Senior Design Lab located in Engineering I will be extensively used to help research, develop, and test the final prototype. The lab is open 24/7 for senior design students and includes workspaces that are equipped with a computer that has useful software such as Eagle Desktop, and has a function generator, oscilloscope, multimeter, and breadboard with resistors and capacitors readily available. This lab will be extensively used to troubleshoot equipment and help individually test parts.

Finally, the group plans on using the Texas Instruments Innovation Lab. This lab gives students access to 3D printers, laser cutters, and soldering irons. Other miscellaneous tools are also provided to help students build prototypes for their projects. Perhaps the resource that will be utilized the most in the lab is the soldering iron. This will be needed to put together the electrical components on the PCB.

It should also be noted that all these resources and labs are located at UCF, which is currently closed due to the Corona Virus outbreak. Therefore, the group is heavily relying on the reopening of the UCF campus during the summer to use the equipment as the members do not have access to their own equipment.

## 6.3 SUPPLIERS

For this project, the group plans on buying parts from various vendors. For the sensors, the group is planning to buy from SparkFun Electronics, which is located in Niwot, Colorado. This company is quite popular, especially for beginners and hobbyists. They sell microcontroller development kits and boards and all products are open-source.

For the glove, the group plans to buy a glove made by Under Armour. This glove was picked due to meeting many of the requirements wanted for the glove. Also, Under Armour is a reputable company that is known to make good products that last long.

Amazon is planned to be utilized as a third party website to order the Bluetooth module. Amazon usually offers free shipping on most products and is fast and reliable. Due to Covid-19, their shipping has been significantly delayed.

The group is also planning to buy from Adafruit Industries. This company includes tutorials on how to use their products and is very popular and accessible for beginners. They are open source and located in New York City. Unfortunately, they are only accepting essential queries relating to Covid-19, therefore the team may have to order from other companies if many businesses are still shut down.

Bay Area Circuits was selected for the PCB assembly. They are located in California and ship within 3 business days. Although they can be expensive, the group plans on taking advantage of a special, which allows the buyer to get 3 PCBs for $90 as long as the board is less than 50 in2. As the dimensions of the PCB is planned to be no larger than 3in x 4in, the group believes this is the best route. It should also be noted that their Silicon Valley location is unaffected by Covid-19, therefore they are still taking orders.

# 

# 7. ADMINISTRATIVE CONTENT

## 7.1 PROJECT MILESTONES

Listed below in Tables 7.1-1 through 7.1-3 are the proposed milestones which show a tentative schedule for the project during the Spring 2020 semester. Each task is assigned a start and end date, as well as who is responsible for each task. Table 7.1-3 shows a similar table for the Summer 2020 semester. Due to Covid-19 it is possible that our timeline may shift and to account for that, we will be ordering parts earlier than expected. Throughout the rest of the Spring semester following Spring Break, we will be conducting our meetings online through Google Hangouts.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start** | **End** | **Who** |
| **Research** | | | |
| Group Formed | 1/6/2020 | 1/10/2020 | All of group |
| Divide and Conquer | 1/20/2020 | 1/31/2020 | All |
| Finalize Divide and Conquer | 1/31/2020 | 2/14/2020 | All |
| Microcontroller | 1/31/2020 | 3/6/2020 | Madison |
| Power System | 1/31/2020 | 3/6/2020 | Madison |
| User Interface | 1/31/2020 | 3/6/2020 | Ann |
| Sensors | 1/31/2020 | 3/6/2020 | Shreya |
| AI training | 1/31/2020 | 3/6/2020 | Andrew |
| Contact with ASL club | 2/14/2020 | TBD | Madison |
| 60 page SD1 report | 3/9/2020 | 3/20/2020 | All |
| Final SD1 report | 4/6/2020 | 4/21/2020 | All |

**Table 7.1-1 - Spring 2020 Project Milestones: Research**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start** | **End** | **Who** |
| **Design** | | | |
| Microcontroller | 3/6/2020 | 4/10/2020 | Madison |
| Power System | 3/6/2020 | 4/10/2020 | Madison |
| User Interface | 3/6/2020 | 4/10/2020 | Ann |
| Sensors | 3/6/2020 | 4/10/2020 | Shreya |
| AI training | 3/6/2020 | 4/10/2020 | Andrew |
| 100 page SD1 report | 3/23/2020 | 4/3/2020 | All |
| 120 page SD1 report | 4/3/2020 | 4/21/2020 | All |
| Order Parts | 4/21/2020 | 4/30/2020 | All |

**Table 7.1-2 - Spring 2020 Project Milestones: Design**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start** | **End** | **Who** |
| **Implementation** | | | |
| PCB/MCU | 5/18/2020 | 6/19/2020 | Madison |
| Power System | 5/18/2020 | 6/19/2020 | Madison |
| User Interface | 5/18/2020 | 6/19/2020 | Ann |
| Sensors | 5/18/2020 | 6/19/2020 | Shreya |
| AI Training | 5/18/2020 | 6/19/2020 | Andrew |
| Test working prototype | 6/22/2020 | TBD | All |
| Final SD Report | 6/22/2020 | TDB | All |

**Table 7.1-3 - Summer 2020 Project Milestones: Implementation**

## 7.2 BUDGET

The proposed budget shown below in Table 7.2-1 outlines the costs associated with the project. Additional costs for extra or replacement parts are also considered. The prices listed below are estimates based on research done about the products and by looking at the cost of similar projects. As of right now, there is no sponsor and the cost of the project will be split between the group members. The costs can increase or decrease due to a number of factors such as complications or improvements. Given that all of the previous projects that involved a glove that translates hand gestures to text have been over $550, our aim is to get our project under the $450.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Vendor** | **Part Number** | **Quantity** | **Cost** |
| Flex Sensor | SparkFun |  | 5 | $15.95 |
| Accelerometer and Gyro | SparkFun |  | 1 | $10.95 |
| Force Sensor | SparkFun |  | 5 | $6.95 |
| MCU - ATMEGA32U4 | Digikey |  | 1 | $3.98 |
| HM-10 Bluetooth 4.0 BLE Module | Amazon |  | 1 | $9.99 |
| UA Radar Softball Gloves | Under Armour | Style #1299550 | 1 | $24.99 |
| Adafruit 328 Battery, Lithium Ion Polymer, 3.7V, 2500mAh | Amazon |  | 1 | $20.59 |
| Adafruit Micro-Lipo Charger for LiPo/LiIon Battery w/ MicroUSB Jack | Amazon |  | 1 | $19.39 |
| LP2985 | TI |  | 5 | $0.55 |
| PCB | Bay Area Circuits |  | 3 | $90 |

**Table 7.2-1 - Budget**

## 7.3 DIVISION OF LABOR

The BabelGlove team comprises two computer engineering and two electrical engineering students from the University of Central Florida. Each member has designated roles and responsibilities throughout the entire process of development and prototyping. In general, Ann Dang is responsible for the software side of the project, creating the graphical user interface. Andrew Glisson is also a part of the software side, with more development on the machine learning algorithms. Madison Manley is in charge of the microcontroller and the power systems, making sure the glove gets the correct amount of power and sending the data from the glove to the computer. Lastly, Shreya Mistry is responsible for the multiple sensors on the glove that gather the data for the translation. Each member is expected to contribute that same amount of work in their respective roles, to create a solid balanced group dynamic.

## 7.4 PROFILES

**Ann Dang:**

Ann Dang is a first generation undergraduate Computer Engineering student at the University of Central Florida. In the summer of 2014 she participated in the High School Research & Engineering Apprenticeship Program at UCF’s Center for Research in Computer Vision learning about different projects and papers that applied computer vision. Along with her STEM activities, she has also held leadership positions in campus organizations such as Managing Editor and Editor-in-Chief in Sparks Magazine, which helps with her editing and documentation skills. Ann is a current Systems Engineering UCF/Lockheed Martin College Work Experience Program student supporting the Digital Sensor Systems team in modeling and simulation. Ann wanted to get more experience with designing user interfaces and applications, so Ann is in charge of creating the user interface of the system.

**Andrew Glisson:**

Andrew Glisson is an undergraduate student at the University of Central Florida, majoring in Computer Engineering. He has been taking classes at the university for four years under the Electrical and Computer Engineering Department. Andrew’s focus on the project is building the machine learning algorithm used to decode what the user’s hand gestures translate to in english. He has taken circuit analysis and software engineering classes, but feels most comfortable with the software side of this project. Andrew has taken multiple Artificial Intelligence and machine learning classes, so he would use that knowledge and experience to build a large practical machine learning algorithm.

**Madison Manley:**

Madison Manley is an Electrical Engineering student at the University of Central Florida. Since her freshman year, she has been involved in undergraduate research at UCF which involves fabricating and testing nanoscale devices using 2D materials for neuromorphic computing applications. She has also had the opportunity to do research at other universities such as Columbia University and UC Berkeley ranging from device physics to hardware design. Madison was interested in learning more about hardware, so she was in charge of designing the MCU and power supply.

**Shreya Mistry:**

Shreya Mistry is an Electrical Engineering student at the University of Central Florida. She has been taking classes at the university for four years under the Electrical and Computer Engineering Department. She is a member of various engineering clubs on campus such as SWE, IEEE, AIAA, and ASME. Shreya was in charge of researching and selecting what sensors to use for this project that would best fit the requirements and specifications.

# APPENDIX A: COPYRIGHT PERMISSION

## Permission to use figures from National Institute on Deafness and Other Communication Disorders (NIDCD)

Re: Use of Image in Student Work

NIDCD Info <NIDCDInfo@nidcd.nih.gov>

Thu 4/2/2020 12:43 PM

To:

* Ann Dang <annchristy@Knights.ucf.edu>

Dear Ms. Dang,

Thank you for contacting the National Institute on Deafness and Other Communication Disorders (NIDCD) Information Clearinghouse with your question about using an image on our website. The NIDCD supports and conducts research and research training on the normal and disordered processes of hearing, balance, taste, smell, voice, speech, and language, and provides health information, based upon scientific discovery, to the public.

The NIDCD website is a primary source of online information about human communication disorders. Most of the information on this site is in the public domain. Unless otherwise stated, documents and files on the NIH web servers can be freely downloaded and reproduced. The NIDCD asks only that no changes be made to the content of the materials. Most of the documents on this server are sponsored by the NIH; however, you may encounter documents that were co-sponsored by private companies or other organizations. Accordingly, other parties may retain all rights to publish or reproduce these documents or to allow others to do so. Some documents available from this server may be protected under U.S. and foreign copyright laws. Permission to reproduce may be required.

We would appreciate acknowledgement of the information that came from the NIDCD and a notation about the website address.

We hope this information is helpful.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[NIDCD Information Clearinghouse](https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.nidcd.nih.gov%2F&data=02%7C01%7Cannchristy%40knights.ucf.edu%7Ccaf6333878fb45b64c0208d7d724f2af%7C5b16e18278b3412c919668342689eeb7%7C0%7C0%7C637214425997399202&sdata=sw%2BPOQUurj3diY57xrinUcZvm6DMNHh3jJlD6hSLsDM%3D&reserved=0)

1 Communication Avenue

Bethesda, MD 20892-3456

Toll-free Voice: (800) 241-1044

Toll-free TTY: (800) 241-1055

Email: nidcdinfo@nidcd.nih.gov

*The NIDCD is one of the National Institutes of Health (NIH). The NIH—the nation’s medical research agency—includes 27 Institutes and Centers and is a component of the U.S. Department of Health and Human Services. It is the primary federal agency for conducting and supporting basic, clinical, and translational medical research, and it investigates the causes, treatments, and cures for both common and rare diseases. For more information about the NIH and its programs, visit* [*www.nih.gov*](https://nam02.safelinks.protection.outlook.com/?url=http%3A%2F%2Fwww.nih.gov%2F&data=02%7C01%7Cannchristy%40knights.ucf.edu%7Ccaf6333878fb45b64c0208d7d724f2af%7C5b16e18278b3412c919668342689eeb7%7C0%7C0%7C637214425997399202&sdata=8HTAA3CW%2FoYQfccJ2FpCKJS%2Bv%2F7%2FhroQF8LVDLdiNeQ%3D&reserved=0)*.*

**From:** Ann Dang <annchristy@Knights.ucf.edu>

**Sent:** Tuesday, March 31, 2020 9:20 PM

**To:** NIDCD Info <NIDCDInfo@nidcd.nih.gov>

**Subject:** Use of Image in Student Work

Hello,

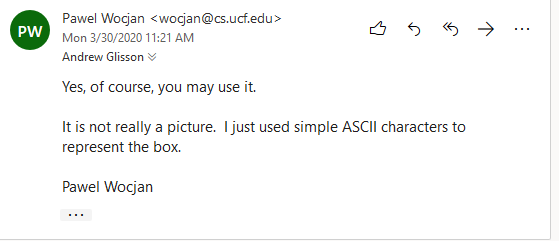
My name is Ann and I’m an undergraduate computer engineering student at the University of Central Florida. I am currently enrolled in Senior Design I and am creating a project that translates ASL into text (for language-learning purposes) and was wondering if I would be able to use one of your images that have the ASL alphabet chart in our design document. Accreditation would be attributed properly and your permission would be greatly appreciated. Please let me know if we can use your image in our research documentation.

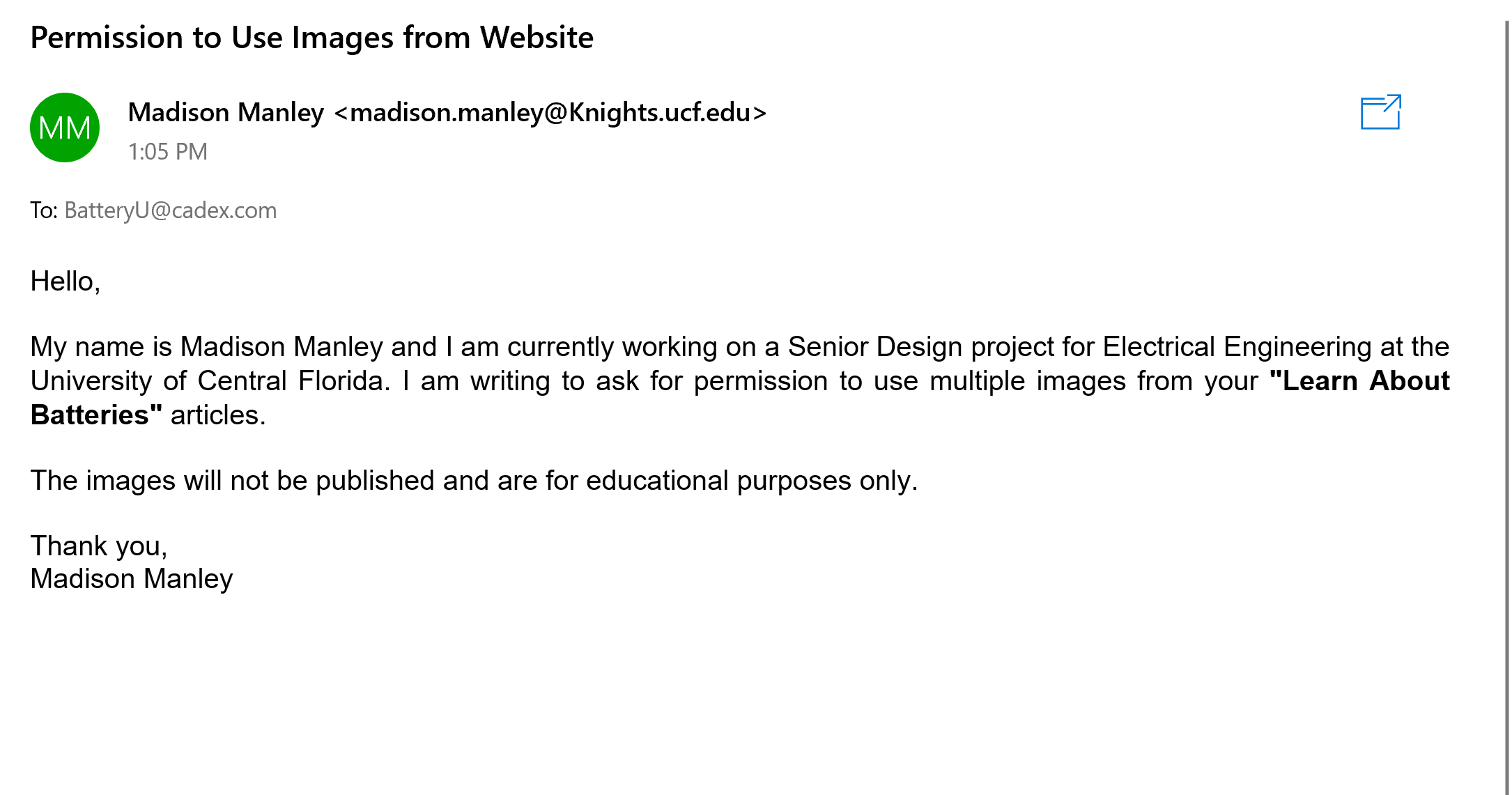
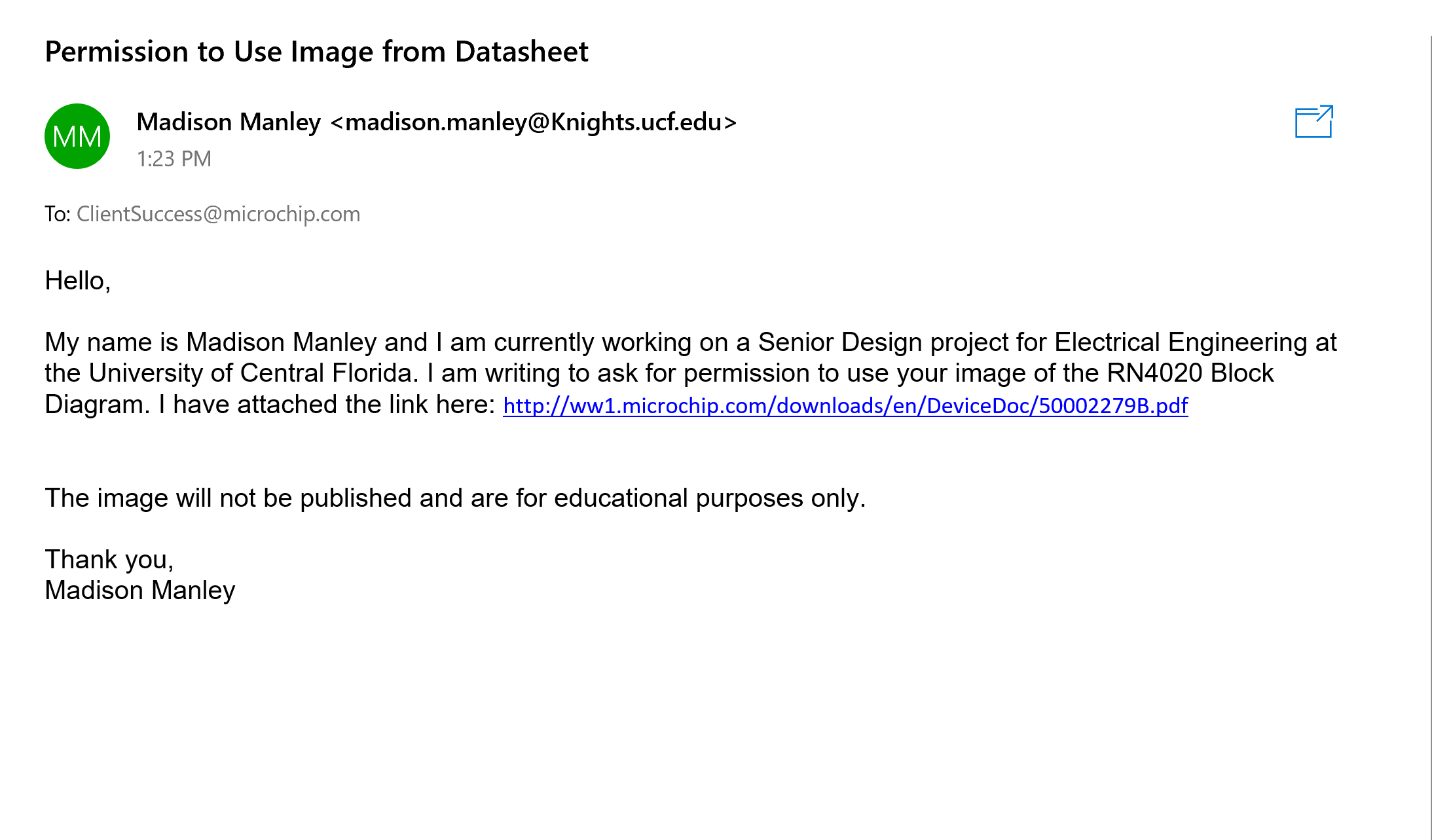
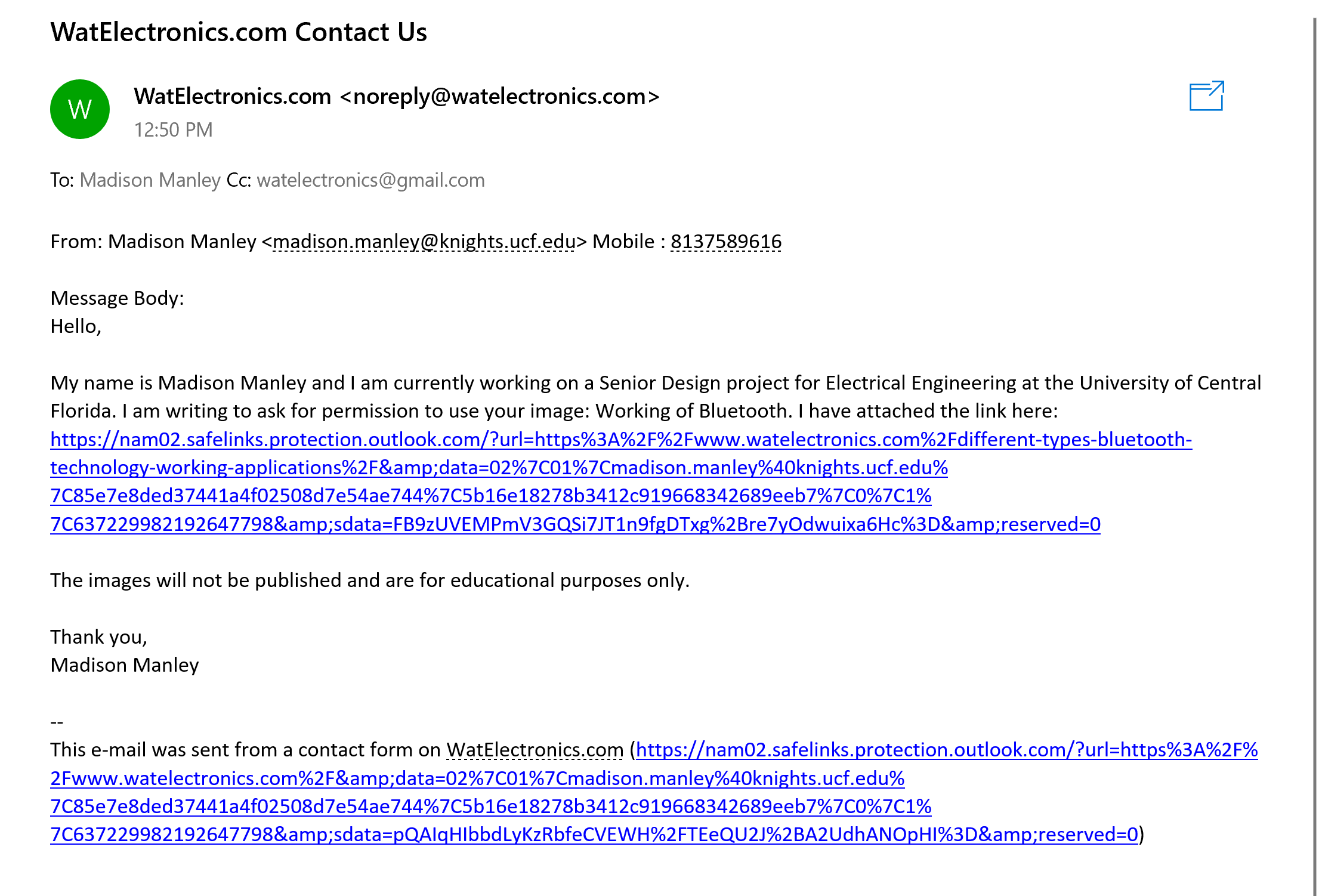
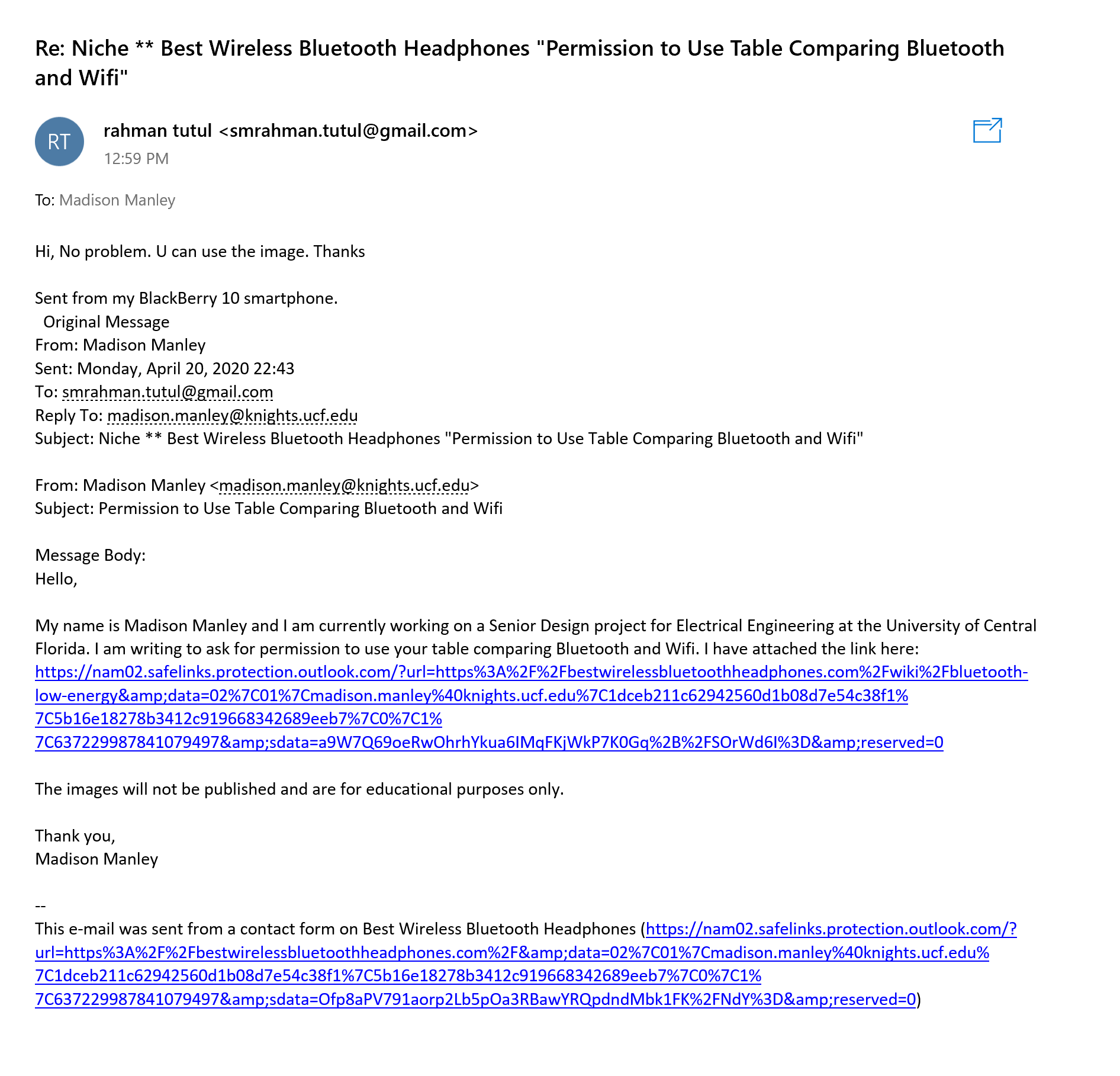
I’ve attached a link to the image.

[https://www.nidcd.nih.gov/health/american-sign-language-fingerspelling-alphabets-image](https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.nidcd.nih.gov%2Fhealth%2Famerican-sign-language-fingerspelling-alphabets-image&data=02%7C01%7Cannchristy%40knights.ucf.edu%7Ccaf6333878fb45b64c0208d7d724f2af%7C5b16e18278b3412c919668342689eeb7%7C0%7C0%7C637214425997409197&sdata=I2COVqxop1FTaX6YT1nL076Rc%2B4QWpO8dboNorUCh6M%3D&reserved=0)

Thank you so much,

Ann Dang





To: Techdifferences.com

Name: Madison Manley

Subject: Permission to Image from Website

Message:

Hello,

My name is Madison Manley and I am currently working on a Senior Design project for Electrical Engineering at the University of Central Florida. I am writing to ask for permission to use your table comparing Bluetooth and Wifi. I have attached the link here: <https://techdifferences.com/difference-between-bluetooth-and-wifi.html#:~:text=The%20main%20difference%20between%20Bluetooth,provides%20high%2Dspeed%20internet%20access.&text=Bluetooth%20is%20used%20when%20speed,bandwidth%20is%20allocated%20to%20it>.

The images will not be published and are for educational purposes only.

Thank you,

Madison Manley

To: Wirelessbluetooth Headphones

Name: Madison Manley

Subject: Permission to Use Table Comparing Bluetooth and Bluetooth Low Energy

Message:

Hello,

My name is Madison Manley and I am currently working on a Senior Design project for Electrical Engineering at the University of Central Florida. I am writing to ask for permission to use your table comparing Bluetooth and Bluetooth Low Energy. I have attached the link here:<https://bestwirelessbluetoothheadphones.com/wiki/bluetooth-low-energy>

The images will not be published and are for educational purposes only.

Thank you,

Madison Manley

To: Adafruit

Name: Madison Manley

Subject: Permission to Use Images from Website

Message:

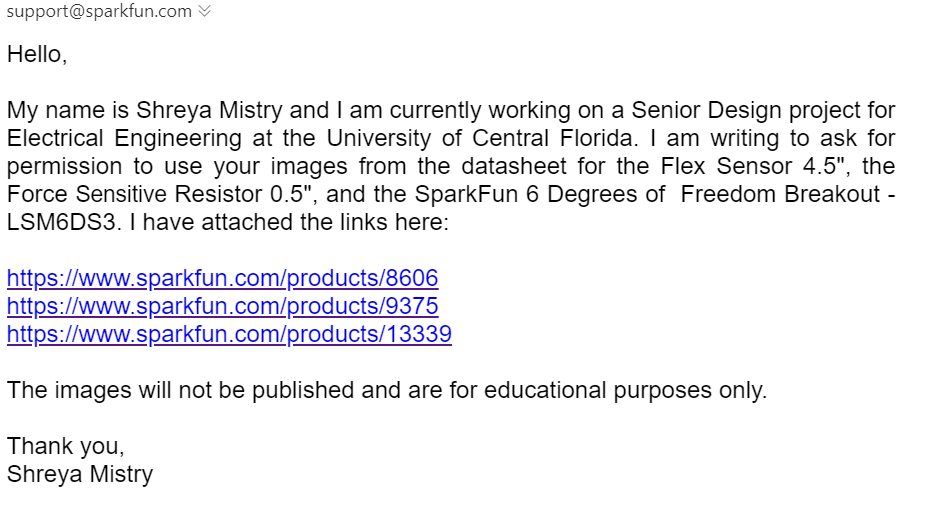
Hello,

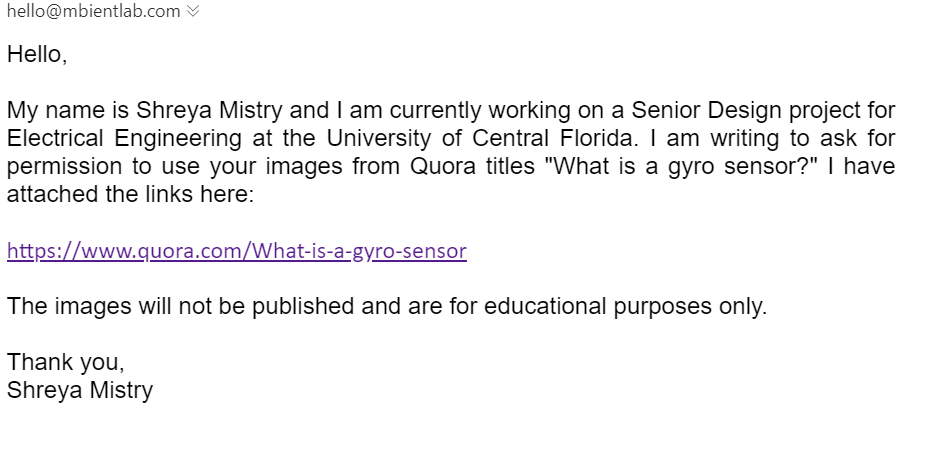
My name is Madison Manley and I am currently working on a Senior Design project for Electrical Engineering at the University of Central Florida. I am writing to ask for permission to use your images in your GAP | Introduction to Bluetooth article. I have attached the link here: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap

The images will not be published and are for educational purposes only.

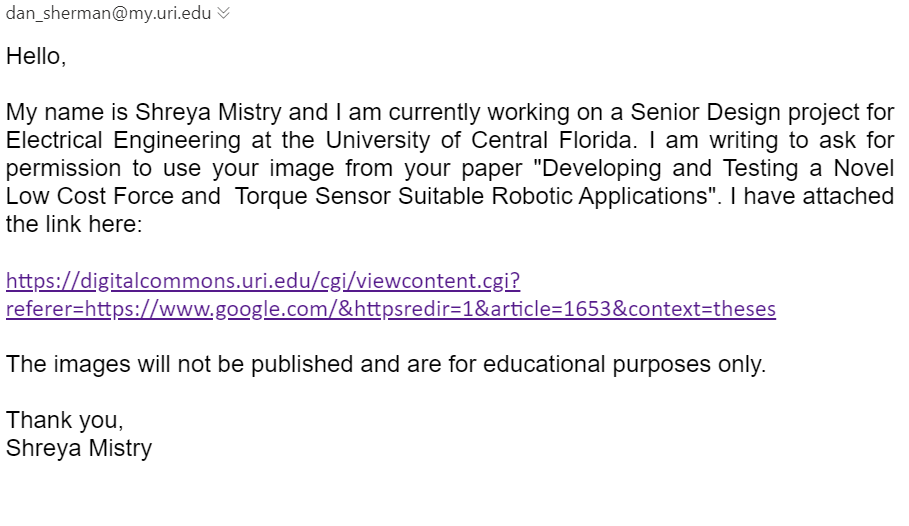
Thank you,

Madison Manley









# APPENDIX B: REFERENCES

**Sensors**

<https://www.researchgate.net/publication/318850816_A_Review_on_Applications_of_Flex_Sensors>

<https://www.semanticscholar.org/paper/Wireless-data-glove-system-developed-for-HMI-Saggio-Bocchetti/b0e4394dcca9c0053f309ffeca6741f44427479a>

<https://www.researchgate.net/publication/281066609_Smart_Glove_With_Gesture_Recognition_Ability_For_The_Hearing_And_Speech_Impaired>

<https://www.researchgate.net/publication/322911030_Wearable_technologies_for_hand_joints_monitoring_for_rehabilitation_A_survey#pf9>

<https://blog.endaq.com/accelerometer-selection>

<https://sensorwiki.org/sensors/flexion>

<https://insights.globalspec.com/images/assets/263/1263/Accelerometers-02-fullsize.jpg> (accelerometer picture)

<https://learn.sparkfun.com/tutorials/gyroscope/all>

<https://www.quora.com/What-is-a-gyro-sensor> (gyro picture)

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8469081&tag=1>

<https://www.elprocus.com/force-sensing-resistor-technology/> (force sensor picture)

<https://home.roboticlab.eu/en/examples/sensor/force>

<https://www.phidgets.com/?&prodid=102>

**Microcontroller**

<https://electronics.howstuffworks.com/microcontroller1.htm>

**Batteries**

<https://batteryuniversity.com/learn/article/lead_based_batteries>

<https://batteryuniversity.com/learn/article/lithium_based_batteries>

<https://batteryuniversity.com/learn/article/nickel_based_batteries>

<https://batteryuniversity.com/learn/article/charging_the_lead_acid_battery>

<https://batteryuniversity.com/learn/article/charging_nickel_based_batteries>

<https://batteryuniversity.com/learn/article/charging_nickel_metal_hydride>

<https://batteryuniversity.com/learn/article/charging_lithium_ion_batteries>

**Near Field Communication**

<https://www.techradar.com/news/what-is-nfc>

<http://nearfieldcommunication.org/>

**ZigBee**

<https://www.digi.com/resources/standards-and-technologies/zigbee-wireless-mesh-networking>

**WiFi**

<https://computer.howstuffworks.com/wireless-network.htm>

<https://www.fcc.gov/engineering-technology/policy-and-rules-division/general/radio-spectrum-allocation>

<https://www.edn.com/the-basics-of-bluetooth-low-energy-ble/>

**Bluetooth**

<https://www.explainthatstuff.com/howbluetoothworks.html>

<https://bestwirelessbluetoothheadphones.com/wiki/bluetooth-low-energy>

<https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap>

<https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>

**Voltage Regulators**

<https://www.baldengineer.com/switching-voltage-regulator-tutorial-addohms.html>

<https://www.electronics-notes.com/articles/analogue_circuits/power-supply-electronics/linear-psu-shunt-regulator-circuit.php>

<https://www.electronics-notes.com/articles/analogue_circuits/power-supply-electronics/linear-psu-series-regulator-circuit.php>

**FPGA**

<https://www.researchgate.net/post/What_are_the_advantages_and_disadvantages_of_FPGAs_compared_to_micro-controllers>

<https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>

**ADCs**

[https://www.eetimes.com/analog-to-digital-converters/#](https://www.eetimes.com/analog-to-digital-converters/)

**Power Source**

<https://www.newbiehack.com/MicrocontrollersAlternativePowerSources.aspx>

**Machine Learning**

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

<https://www.edureka.co/blog/classification-algorithms/>

<https://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/>

<https://github.com/schneider128k/machine_learning_course/blob/master/slides/2_b_slides.pdf>

<https://developers.google.com/machine-learning/crash-course/ml-intro>

<https://en.wikipedia.org/wiki/Arthur_Samuel>

**Programing Languages**

<https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7>

<https://data-flair.training/blogs/advantages-and-disadvantages-of-python/>

<https://www.techradar.com/best/best-ide-for-python>

[https://git-scm.com](https://git-scm.com/)

<https://opensource.com/article/18/8/github-alternatives>

# APPENDIX C: DATASHEETS

<http://ww1.microchip.com/downloads/en/DeviceDoc/50002279B.pdf>