A.S.L.A.

American Sign Language Assistant

Group E - Senior Design Spring/Fall 2016

Isabel Atanacio	Electrical Engineering
Colin Fox	Electrical Engineering
Charlie Munoz	Electrical Engineering
Daniel Sarmiento	Computer Engineering

1 Executive Summary

The objective of this project is to design a Glove that converts American Sign Language to computer generated speech to help those who are hearing-impaired communicate in everyday life. Connecting via Bluetooth to a computer system to output the spoken speech. By using a computer, it would need a microcontroller to communicate with it. The computer will have its own software to process the data received from the microcontroller and to output the computer generated speech. The glove will be able to recognize every letter in the alphabet and basic full word signs. In order to make it more personal for the user, the design will allow the ability to add new words or phrases to the already existing library.

The battery life on the gloves also need to last for at least one day of normal use so to not inconvenience the user with multiple charges daily. Ideally, the glove would also need some battery indicator to show the user how much time is left until it needs to be put to charge. The gloves have to be something comfortable to wear when going out in everyday life.

There will be a flex sensor for each finger that will be used to detect flexing and bending in one direction. By using accelerometers to measure acceleration, one can find out the angle the device is being tilted. Another sensor being used will be a gyroscope sensor that will help determine orientation when detecting sign language. Both gloves need to communicate to a central microcontroller to record to send to the computer. The program will process the data and map it to known letters and phrases used commonly in American Sign Language.

This project has the intention to make the lives of those not so fortunate to be able to live a regular life. This project is design to make the communication with others easy and simple. The reason why it is said to be simple is because it can either be displayed on a computer screen or it can be programed to translate the hand gestures into speech. This feature would immediately facilitate a variety of activities that a disabled person cannot usually do with the simple addition of a pair of gloves. These gloves will be designed to be portable, flexible and very practical. This brings somewhat of a challenge because our design must be compact enough to be able to be work as just another accessory of our body.

In terms of our compact design, the glove must certain requirements and specifications. This means that all of the multiple parts of our design must be able to fit in the small section of a human wrist. In other words, the microprocessor, IMU, flex sensors, pressure sensors or contact sensors, fuel gage, voltage regulators and Bluetooth module will all be able to be integrated in a single sized glove. These limitations require of a very optimal PCB design that can allocate space for all of these parts. Many of which are large enough to deal with. For example, the Bluetooth module, or even the regular lithium battery rechargeable port.

This project involves multiple parts, most of these parts will have to be integrated into one. To facilitate this process a running type of glove is the best choice for this project. This is because it is made up of a flexible material with breathable features to it. Also, the additions of the sensors will be easy since this glove will be made up of a couple layers. When adding these sensors, the possibility of adding extra material to cover these is also a good choice. To do this we will place the sensors, then a piece of cloth on top of it will be sewed on each finger to allow the glove to look very uniform and not with a whole lot of wiring and sensors sticking out. Also, there might be another possibility of adding a wrist piece which will contain all of the important devices like the microprocessor on the PCB along with all the other components like the fuel gage, rechargeable battery and the Bluetooth module. This would make it more practical and interchangeable.

Although this is not the first time a project of this nature has been implemented, this project offers different features than the previous projects done at this school. Previous projects that have been done at the University of Central Florida consisted of implementing only the alphabet in American Sign Language. The implementation of such project only needed the creation of one glove to implement the American Sign Language alphabet. This in other words makes the whole project a lot simpler since the database will be smaller due to the small library of letters. Also, it means less sensors and analog signals to work with.

Other similar previous projects include an American Sign Language translator glove that used an Android application to display these letters. To this project they only needed a small database which allowed them to program everything to their memory flash on their PCB. For our project we would not be able to do this since our project requires a very large data base for the implementation of words. Therefore, we will use a computer to compute and save all the gestures into an the already existing library.

The new feature our group is bringing to this project which makes it stand out from other projects done at the school, is the ability to implement two gloves to be able to translate American Sign Language sentences instead of just words. The feature of allowing personal gestures to the open library is another improvement to previous.

The biggest improvement of all is the introduction of text to speech. This allows the generated gestures to be translated to speech. This new feature is the one that separates our project from other. Our project has different levels of development and this was just a brief description of all of them. This document will explain in detail every part and their involvement with this project.

2 Project Description

2.1 Motivation

The American Sign Language Assistant glove was an idea that originated based on the group background. Group E decided to tackle this project since it was an interesting idea for each of the group members. This project is favorable enough due to the group build. It involves a good amount of electrical work with a lot of hardware implementation as well as an extensive amount of software work. One of the motivations to this project was to do something that would truly benefit society and help those with disabilities, while at same time giving the group a sense of fulfillment for helping people in need. Also, the group wanted to do something enjoyable to work with rather than to do something that would not interest the group. Some of the reason this project was selected was in order for the group members to acquire some knowledge on microcontrollers, sensors, and wireless communications as well as the software implementation of these. Although, our group had the chance to go with a funded project, it was for the best interest to go with a self-funded project and later try to get funding from interested companies. As mentioned before part of the motivation was to help those in need by translating sign language to text and even to speech to facilitate even further. Since at UCF past projects have developed similar projects, one incentive is to make it cost effective and to add some improvements like maybe the implementation of two gloves to fully translate complete sentences rather than just letters. Other additions might include the development of a phone application to carry out the sign language translation. As engineers this group motivation is to apply previous knowledge and develop new engineering skills building this project.

2.2 Team Members

Team E is made up of one Computer engineer and three Electrical engineers. One of the Electrical Engineers, will be in charge of the selecting and researching the right microprocessor that could accommodate for the different analog signals and translating them to digital signals. The second Electrical engineer will be in charge of the selecting the different sensors and power needed for the translator glove. These include selecting the right sensors that would track all hand gestures, also picking the most suitable power source for the project. The third and last Electrical engineer is in charge of the wireless communication for the glove with the user interface. This means choosing the best way to communicating to the computer without using wires which allows the possibility to use Wi-Fi or Bluetooth. The Computer Engineer is in charge of programing the microcontroller and setting up the interactive data base for the different hand gestures.

Group E will work simultaneously together in order to achieve the best results. Communication between all members is key especially during the first stage of selecting the parts. This is key because the type of sensors selected will dictate what microprocessor will be used. In regards to the communication type, it all depends on the type and amount of data needs to be transmitted, and as well as the distance needed.

Isabel Atanacio	Electrical Engineering	iatanacio0406@knights.ucf.edu
Colin Fox	Electrical Engineering	foxcolin@knights.ucf.edu
Charlie Munoz	Electrical Engineering	cm65@knights.ucf.edu
Daniel Sarmiento	Computer Engineering	dsarmiento@knights.ucf.edu

2.3 Objectives

The project intention is to develop one glove at first and then possibly a second glove to fully translate hand gestures from the American Sign Language to text and speech. First, this translation would be directed to a computer and later it can be implemented by a phone application. At the beginning the glove will be connected to the computer by a wired connection, but the objective is to have it wirelessly communicate to these devices. Also, since there is a possibility to have the two gloves, then these two must communicate with the computer and later with the cell phone application. One important goal of this project is to use low power consumption. Regarding the wireless communication, a Bluetooth module that uses low power is one important objective. This is because our group needs a device that does not consume a lot of power as a result of the restraints and limitations of having a lightweight and flexible glove with minimum batteries.

Part of the objective is to implement the four basic American Sign Language parameters. Although, there are five parameters one of them will not be taken into account since it is the use of facial expressions. These four parameters include hand shape, orientation, location and movement. All which need to be taken into account when storing each hand gesture in the database. Since the goal is to track these four parameters, to optimally do this our group decided to use two different types of sensors; flex sensors and pressure sensors. As part of tracking these four parameters using a gyro/accelerometer is also a goal of this project.

The American Sign Language glove requires of a very complete implementation of the whole alphabet in sign language. This major aspect in our design that requires a lot of attention from each group member. As part of this task, our group has the objective to learn and masters American Sign Language considering that this project involves knowing all of these hand gestures to be added to a database using a glove. With respect to storing these gestures, it is definitely important to for the glove to translate these hand gestures properly. One goal that's depends on reaching previous goals is the implementation of two gloves for sign language since to get the full experience two hands are needed to carry out sentences. Consequently, this goal gives the full translation of sentences not just letters like it would be using one glove. Although, this is part of the project, the idea depends on reaching all of the milestones leaving room for further development of the project.

Other goals and objectives included the ability for our project to store other hand gestures not part of the American Sign Language. For example, a new gesture that is not part of this library but that can be added to our database. Therefore, the ability for our database to have room for new additions to our library.

2.4 Requirements and Specifications

The required specifications for this project can be divided to the function that it applies to. In this project our group has branched them into two categories: A general objectives section, and a functional section. These two categories will be explained accordingly as follows.

General Requirements	
1.	The glove must be flexible enough to allow for easy hand gestures.
2.	The glove must weigh 2 lbs or less.
3.	Each finger of the glove must have one flex sensor.
4.	A gyro/accelerometer with the glove, a second one if the second glove

is implemented
5. A low power microcontroller unit must be used.
6. The microcontroller must contain a low power Bluetooth module.
7. The glove must have a pressure sensor in each fingertip.
8. The glove must contain a flex sensor across the palm of the hand.
9. The PCB must be small enough to allow for easy movements of the hand.
10. The microcontroller must communicate via Bluetooth to user interface.
11. The battery must last at least one day.

Table 2.4 – 1: General Requirements

These are the most general requirements for our design for the American Sign Language translator glove. In order to acquire a good translation, these rules and specifications must work together as specified above. With a flexible glove, it makes the whole hand gesture movements easier to perform with no constraints. The reason why the glove must be lightweight about two pounds or less is because the goal of this design is for practical use. A really heavy glove will limit hand gestures and will be uncomfortable for the user and this will have no future practical use. Also, to capture each movement accurately, each finger must be tracked individually by using a flex sensor per finger. The Gyro/Accelerometer will accurately track the positioning and movement of the hand. Since a low power microcontroller will be used, it is specified to use also a low power Bluetooth module. The reason why a flex sensor will be placed across the palm of the hand is because some hand gestures require the bending of the palm. Also, this extra sensor allows to differentiate between closely related hand gestures. The requirement to have a small printed circuit board (PCB) is to give room for easy bending of the hand that will not put the PCB at risk. A large PCB will restrict the hand movements.

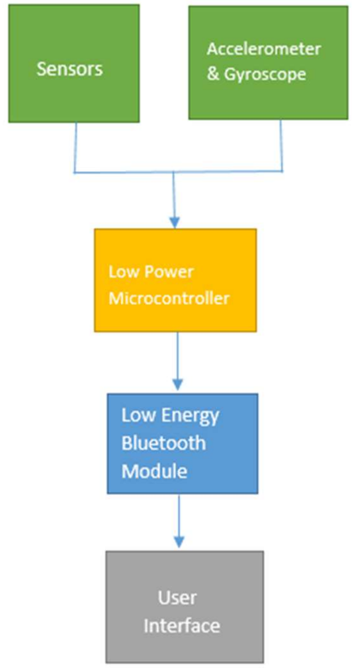


Figure 2.4 – 1: Architecture of Sign Language Glove

Functional Requirements and Specifications
1. Device must recognize positioning and movement of the hand.
2. Device must recognize all letters in the alphabet (26 letters).
3. The system recognize full sentences if second glove is implemented.
4. The system should store each separate hand gesture in database for each letter.
5. The system should allow to add new words/phrases if necessary to existing library.
6. The Bluetooth module must link the microcontroller and the computer.
7. The computer will do the data processing and text generation at first.
8. A limit for each gesture to be displayed on user interface.
9. The generation from speech to text.

Table 2.4 – 2: Functional Hand Gesture Requirements and Specifications

The above specifications dictate the how far our system can go in terms of hand gesture recognition. For a correct detection of gestures, the system must first find out the positioning of the glove. Part of these specifications include the recognition of at least the 26 letters in the alphabet by our system. If possible then our group will add the requirement to also detect full sentences if the second glove is implemented. In which case, the database library should be capable of storing new hand gestures to the already existing library. In the first case of just implementing just the alphabet letters, the database should be able to create the library by storing each hand gesture. After this database library is done, the system should be able to detect each hand gesture already stored in the library when implementing the sign language alphabet. This recognition also has certain specifications. Since for real life practical use the detection has to be carried out in a timely manner. In other words, the translation must be fast so there will be a time limit for each detection. This assures that the system is suitable for real time sign language translation. Lastly, a method of communication between the glove and the user interface must be specified. In this case, the communication will first be done manually for trial and error purposes and later a Bluetooth communication will be set. This communication via Bluetooth has to ensure the low power use to maintain the general specification of at least one day battery life.

Figure 2.4 – 2: Functional System Path

2.5 Design Constraints

Designing a project that could make possible translating American Sign Language to text and then to speech using a computer involves many technical and behavioral factors. These factors also come along with many limitations that cause constrains for our design and that should be definitely taken into consideration. These multiple factor constraints include economic, environmental, social, political, health, safety, manufacturability, sustainability, and even health constrains.

2.5.1 Social and Political

The social and political constraints in our projects are not many but should definitely be defined. Some social concerns when starting the project where the amount of languages that we could implement. This means that only one language and this is the American Sign Language will be able to implement due to time constraints.

Some political issues involve the careful determination of the project name in terms of if it is an invention or if it already has been implemented by someone else. For our case, this project is school based and it has already been developed by other students even in other universities. So we will not be considering it an invention

2.5.2 Economic

Some of the constraints of our project include economic factors. The reason for this limitation in that our group has no funding. This factor limits since the only way to fund our project is for each group member to contribute and invest some money. Instead of changing the PCB design multiple times, this would limit the amount of changes allowed to make since it would cost more money. Also, our project was initially designed to be cost effective to prove that it could be implemented and even be manufactured at a low cost. Consequently, this requires of finalizing the selection of the best suitable parts for our project. This means selecting the best microcontroller for the functions needed when translating different analog signals. When it comes down to selecting sensors, choosing the right amount of flex sensors and pressure sensors and staying under budget.

Also, even though it is not a huge constraint in the design of our project, the Bluetooth module required also has some constraints. This is because our project requires a light and flexible design for the glove or gloves if the two are implemented. This would limit the size of the Bluetooth module we decide to select. It must be small enough that it can fit on top of the wrist of both of the gloves. This is somewhat of a challenge since there are lots of Bluetooth modules that are used for different applications. Although, for some devices price is a big constraint, for the Bluetooth module in this case it will not be one since these modules are usually reasonably cheap.

Another challenges when selecting the parts such as the microcontroller processor needed for such complex project. One of the limitations is the size of memory of a microcontroller. This would in turn cause a delay due to the lack of memory causing a wait time until data is ready for use. Also, part of the limitations of a microcontroller is that it cannot connect directly to sources of high power. When programming this device, a good strong programming foundation is required since to get the microcontroller to perform such complicated task like our project, it definitely needs extensive programing. So a huge constraint could arise when choosing the final programming language since it involves having the skills to program such a high level microcontroller. We must take into account that a microcontroller can be set up using a variety of languages which also brings another limitation which is merging two different languages which becomes difficult at times. This brings up one of the biggest issue of a microcontroller which is the amount of instructions and executions that it can perform. This problem along with the speed limitation should be taken into great consideration when selecting the right microcontroller.

2.5.3 Sustainability

When it comes down to describing our system sustainable capabilities, we have to address many factors which include the hardware side of it. A more hardware driven constraint for a microcontroller is sensitive material that it is made up of. A usual microcontroller can be made up of metal oxide which can be damaged by a series of things. One of these being a static charge or even an electromagnetic pulse EMP could cause damage to this chip. Although there exist a solution for the last mentioned risk of an EMP, this is to protect the device from electromagnetic radiation. In the most extreme of the cases placing the device in a metal container such as a metal garbage can could definitely protect against such a destructible threat. This method is called a Faraday cage if the lid is closed, which is based on protecting whatever is inside of the can against electromagnetic radiation. What this method does is that it protects from electric fields but not just magnetic fields. So in the most extreme case this would be a great solution for these constraints.

Another part of the sustainable driven constraints is the fact that our whole project must be carefully made to last a definite amount o. This means that even though it is only a senior design project, it must be designed in such a way that it will last more than the presentation amount of time. This is because in a realistic model, you do not built a device to last a few hours. Instead, a project of this magnitude is designed to last an indefinite amount of except for the battery.

Also, the whole project including the software must be designed to have durability in order for future additions to the library. What this really means is that it has to be sustainable enough to allow future hand gesture additions to the library database. In addition, the hardware connections of the PCB including the voltage regulator, fuel gage are to be design that they are not to get disconnected at any time. Constant voltage and current when the device is on is also required to avoid any kind of future malfunctions.

When it comes to hardware implementation, there are a lot of drawbacks of implementing with sensors. One of them being that most of these sensor signals will be analog. Therefore, the challenge lies in translating these all of these signals to digital in order to be processed and compared to other data. One of the challenges is that for this specific project there will be a lot of signals that will have to be translated to digital which involves having great knowledge of sensors and analog to digital background as mentioned before the challenge and limitations are definitely seen on this part of the project since it is the most critical part. Translating these signals perfectly is a must in order for recognition of our hand gestures. Therefore, selecting the best microprocessor is an important decision that must be made carefully. All of these factors put together contribute to the reliability of this device to always work under different conditions.

2.5.4 Health and Safety

Part of the design constraints is the temperatures of the devices connected to the glove or gloves of our project design. For example, the Bluetooth module must be temperature compatible with our glove. This means that the Bluetooth module cannot exceed temperatures that would put in jeopardy the fact that it has to be practical enough to be work by a human hand.

Another part of the design constraints concerning the health and safety issues. This simply means that our glove design must be free of stripped wires that could be made in contact with the skin and therefore causing electrical shock to the person. This limits the PCB design to be small and compact enough to cover any loose wires. This also means choosing a perfect glove for the project that can hide and cushion any materials that could cause harm to the skin. To ensure that everything is covered, we must create enclosures within the glove that would make sure nothing is making contact with the hand or that nothing is showing for appearance purposes.

2.5.5 Manufacturability

Also, the design of our project has to be practical enough to allow for the implementation of a lithium battery. This means that a major constraint is the positioning and as well as the proper selection of the right battery for our design. This battery has to last a decent amount of time for us to give a lengthy presentation of the multiple sentences and words that our glove can do and implement. And in the practical use of our project, the battery of each glove must last at least one day for it to be a good enough product to be manufactured. Even if it does not last up to one day it should at least be able to last 12 hours. This also brings up the other design constraint of a rechargeable port implementation. What this means in terms of design constraint is that the PCB design must be flexible enough in terms of room for another addition of a new device which will be the rechargeable port. The other option is to have a removable battery that can be charged separately by taking it of the glove when it is dead or drained and then plugging it into an outside charger and then proceeding to plugging it back into the glove part where it belongs. This basically means that our glove design might be reduced and extra space will be available to other devices implementation. Therefore, the battery presents somewhat of a constraint to our design in terms of space and temperature limitations. This being one major constraint in the future design of our circuit printed board. The decisions made from this point on will be very crucial in what constraints our design will present. Every device must be carefully selected, this includes an optional Bluetooth module along with the right type of sensors like flex sensor accelerometer and gyroscope or an IM to simply replace these two. This project offers multiple limitations and constraints that must be monitored.

2.2.6 Ethical

Among all of the above explained types of constraints, there is one more important constraint which is the ethics of this engineering project. These constraints include the moral principles. These principles are based on not copying previous group designs, as well as not using their senior design documents to directly mirror what they did. Some other ethical constraints include the proper use of the American Sign Language and implementing the right hand gestures that anyone using this language can understand. Even if this means extra work because some gestures might be more difficult than others.

3 Project Management

3.1 Milestones

Our team is taking Senior Design 1 during the spring 2016 semester and Senior Design 2 during the fall 2016 semester. Which makes our milestone timeline from March 2016 to December 2016, giving us ten months. Three of which (May - July) will be in between semesters and some of us will be out of town or doing internships. During this time production of our product will come close or come to a complete halt. The next two tables are subject to change with issues or complications that may arise during the documentation writing and product development process and is set as a guideline we should follow.

Date	Task
3/3/16 - 4/27/16	Research American Sign Language
3/3/16 - 4/27/16	Research sensors
3/3/16 - 4/27/16	Research microprocessors/microcontrollers
3/3/16 - 4/27/16	Research wireless communication
3/3/16 - 4/27/16	Research software APIs
3/3/16 - 4/27/16	Research power systems
3/31/16	**Draft of our documentation is due**
4/11/16 - 4/15/16	Part list for one glove prototype without wireless
4/14/16 - 4/18/16	Part list for two glove prototype without wireless
4/18/16 - 4/22/16	Develop a one glove schematic without wireless
4/21/16 - 4/25/16	Develop a two glove schematic without wireless
4/27/16 - 5/15/16	Develop the software prototype for alphabet
4/27/16 - 5/6/16	Build the one glove prototype without wireless
4/28/16	**Final documentation is due**

Table 3.1 – 1: Milestone chart for the spring 2016 semester.

Date	Task
8/15/16 - 9/16/16	Develop/debug the software prototype with common words and phrases
8/22/16 - 9/2/16	Build the two glove prototype without wireless
9/19/16 - 10/16/16	Develop/debug the software prototype with library extension
10/3/16 - 10/10/16	Integrate wireless into the two glove prototype
10/10/16 - 10/16/16	Final design and wireless testing
10/14/16 - 10/16/16	Finalize PCB designs
10/17/16 - 11/4/16	Order PCBs
10/31/16 - 11/28/16	Final product and software testing/debugging
12/1/16	**Senior Design Showcase**

Table 3.2 – 2: Milestone chart for the fall 2016 semester.

3.2 Group Management

Group management is very important towards the success of the project. Some activities include:

- Using GitHub to host all code for version control and synchronized file sharing of code between the team members[1]
- Communication via a group chat hosted by Facebook
- Hold bi-weekly meetings (twice a week) for progress reports, and general updates
- Use a Google Drive shared folder to synchronize all project design files, besides code, and documentation
- Each team member is responsible for his or her subsection and integration between subsections are the responsibilities of those people in charge of the subsection. Subsections include:

[1] See appendix A – 1

- Sensors - Isabel Atanacio
- Microcontrollers/Microprocessors - Colin Fox
- Wireless Communication - Charlie Munoz
- Software - Daniel Sarmiento
- Power system - Isabel Atanacio
- Schematics/Printed Circuit Board - Colin Fox

3.3 Facilities and Equipment

Facilities and Equipment available to us include the following

The Texas Instruments (TI) Innovation Lab, located at University of Central Florida, Engineering 2 Room 112. Equipment there includes:

- Metcal MFR-1100 Soldering Iron
- Aoyue Int 968 Repairing system
- Microscope for surface mount soldering
- Keithley 2000 Multimeter
- Tektronix AFG 3022C Dual Channel Arbitrary/Function Generator, 25 MHz
- Keithley 2231A-30-3 Triple Channel DC Power Supply
- Tektronix MDO3034 Mixed Domain Oscilloscope, 350 MHz, 4 Channel
- Universal Laser Systems ILS 12.75 Platform Laser Cutter
- Dimension sst 1200es 3D Printer
- Dell Optiplex 960 Computer

The lab also has multiple TI processors and components available for student use and rental.

The UCF ECE Senior Design Lab, located at University of Central Florida, Engineering 1 Room 426. Equipment there includes:

- Tektronix MSO 4034B Digital Mixed Signal Oscilloscope, 350 MHz, 4 Channel
- Tektronix AFG 3022 Dual Channel Arbitrary/Function Generator, 25 MHz
- Tektronix DMM 4050 6 ½ Digit Precision Multimeter
- Agilent E3630A Triple Output DC Power Supply
- Dell Optiplex 960 Computer

3.4 Costs and Budget

All current costs and budget are assumed maximum estimates, and can increase or decrease depending on risks taken and complications that may arise during the development of our project.

Item	Amount	Quantity	Total
Flex Sensor	$13	5/hand	$130
Pressure Sensor	$7	5/hand	$70
Accelerometer/Gyro	$20	2/hand	$80
MCU	$20	1/hand	$40
Bluetooth	$20	1/hand	$40
Power System	$40	1/hand	$80
Gloves	$20	1 pair	$20
PCB	$15	1/hand	$30
Shipping	$50		$50
Misc.	$60		$60
Total			$600

Table 3.4 – 1: Estimated costs

4 Research

4.1 Sign Language

Sign language uses manual communication to transmit ideas and thoughts instead of spoken language. Mostly used by people hard of hearing, or complete hearing loss. In 2013 approximately four million people are deaf in the United States, this accounts for 2.1% of the population.[2] There are over 137 sign languages accounted for since 2013, we are going to focus on American Sign Language.[3] This

[2] See Appendix A – 2

[3] See Appendix A – 3

encompasses the standard English alphabet along with multiple words and phrases used by English speakers.

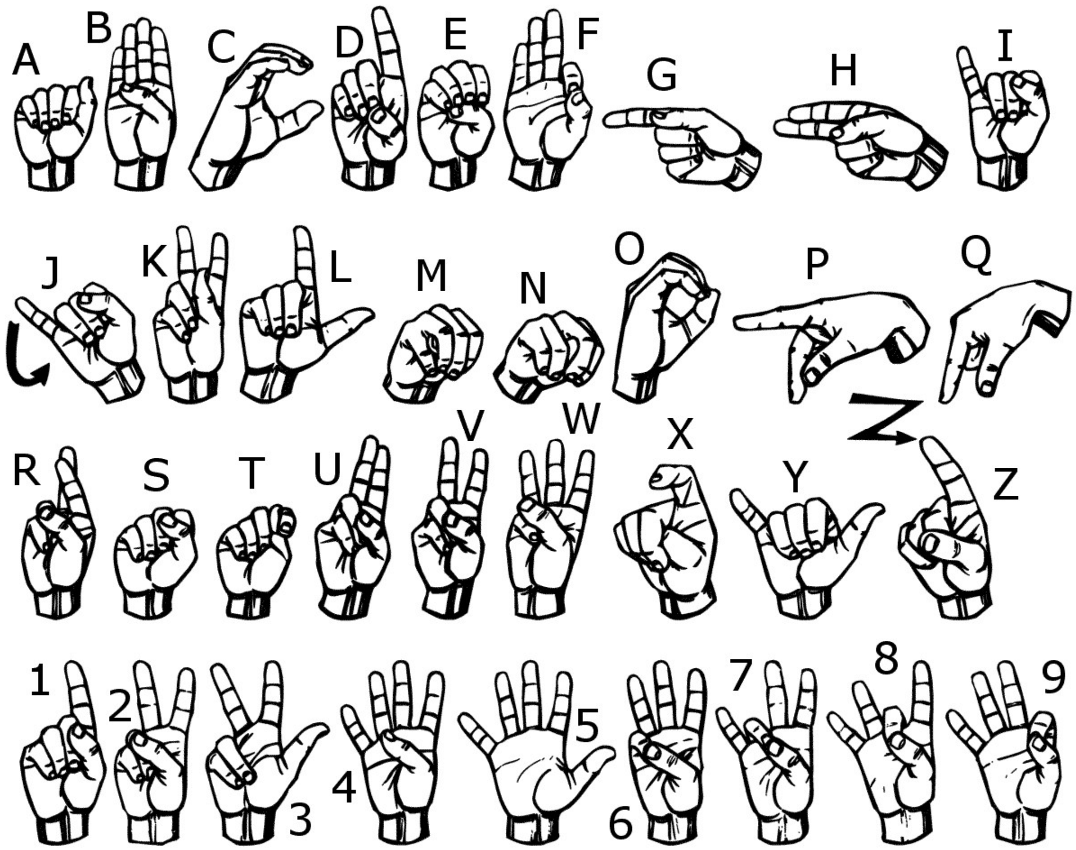


Figure 4.1 - 1: Fingerspelling of the alphabet and numbers[4]

4.2 Sensors

In order for the sign language glove to read all different sign language gestures possible, we need to implement many different types of sensors to our design. We will consider a different variety of sensors that could benefit the design we have come up with. For example:

- Flex Sensors
- Force Sensors
- zContact Sensors
- Accelerometer
- Gyroscope

[4] See Appendix A – 4

- Battery Voltage Sensor

4.2.1 Flex Sensors

There will be multiple flex sensors in each hand that will be used to detect flexing and bending in one direction. These sensors were ideal because they can be placed on a stationary or on a flexible surface. As the flex sensor is being flexed, the resistance across it increases as the angle decreases. Bending outside the usual range can be done but precaution should be taken to make sure the sensor will not be damaged. Since bending the sensor more than 90 degrees can permanently damage it. Flex sensors can be unidirectional or bidirectional. The bidirectional flex sensor can measure deflection in two opposing sides, but having it bend two ways is not really needed for a finger. Flex Sensors usually come in 3 types:

- Conductive Ink
- Fiber Optic
- Conductive Fabric/Polymer

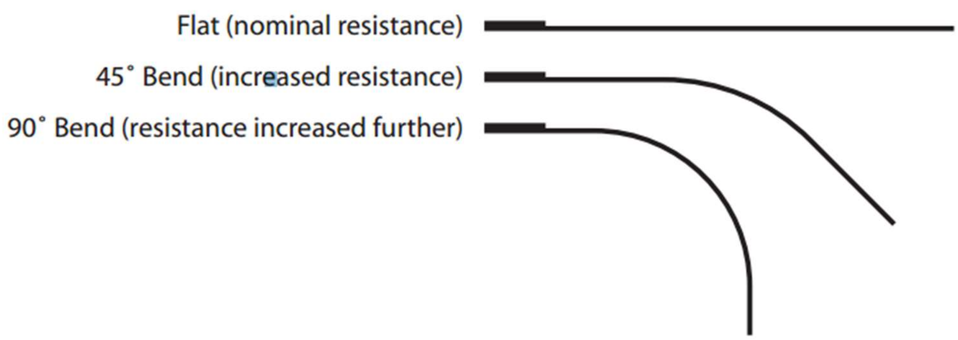


Figure 4.2.1-1: Resistance/Bend for Flex Sensor[5]

4.2.1.1 Conductive Ink

The conductive ink based flex sensor is made by layering a strip of resistive ink on a flexible plastic surface. It can have a nominal resistance between 10k-50k ohms and can increase by a factor of 10 at maximum deflection. Most ink based sensors can measure only in one direction, so resistance doesn't change if bent in the opposite direction. By placing two ink based sensors back to back, it will allow bidirectional detection. The longer the sensor is, the resistance at maximum deflection also increases. Instead of measuring angle at a point, the ink based flex sensor measures the radius of curvature. This sensor can be found at a relatively low cost.[6]

[5] See Appendix A – 5
[6] See Appendix A – 6

4.2.1.2 Fiber Optic

Another common type is a fiber-optic flex sensor that has an abraded section and also is photosensitive. It uses light that is emitted into the fiber at one end and sensed at the other end. While the sensor is being bent, it results in a loss of light. This sensor has the ability to be bent in any direction and can also be expensive.

4.2.1.3 Conductive Fabric/Polymer

The third type of sensor is the conductive fabric/polymer-based flex sensor. The sensor is consisted of 2 conductive materials and in the middle is a resistive material. The flex sensor stretches as it is being bent, causing the two conductive materials to be pushed closer together. When pressure is placed on the material, the resistance of the sensor decreases. This type of flex sensor is also considered a pressure sensor, the only difference is that this sensor also measures deflection. One con of using the fabric/polymer-based flex sensor is that it lacks accuracy and it has a slow response rate. This sensor can be found at a relatively cheap price[7].

4.2.2 Force Sensors

At times, some fingers will come in contact with each other when making sign language gestures. By definition, a force sensor is a transducer that takes in a mechanical input such as force and converts it to an output as an electrical signal. The resistance in the force sensor will vary according to how much pressure is being applied. The stronger the force that is being applied to the sensor, the lower the resistance becomes. This type of sensor is flexible but it does not measure the resistance when it is being flexed, unlike a flex sensor. The resistance will only change when pressure is being applied to the circular region in the top of the sensor. When no force is being applied to the sensor, the resistance will be relatively large. This can confirmed by using a multimeter to measure the differences in resistance when pressure is being applied. The relationship is linear in terms of conductance versus force.

One thing to consider when using this sensor is that it is not very accurate when trying to measure different pressures. It is great when simply measuring if pressure is being applied but not how much pressure is being applied, so it isn't good to use this sensor as pressure scale. At the bottom of the sensor are two pins that are used to connect to any type of board. A benefit of using this sensor is that it can be customizable to meet any specific needs, like the shape of the sensor. The force sensor can also be used a switch by allowing the applied force to act as a trigger for the switch. The characteristics of force sensors allow it to be used in applications

[7] See Appendix A – 7

such as grip for humans/robots and equipment for medical devices. By adding force sensors to our design, it will help create less error when trying to read/detect specific sign language gestures.[8]

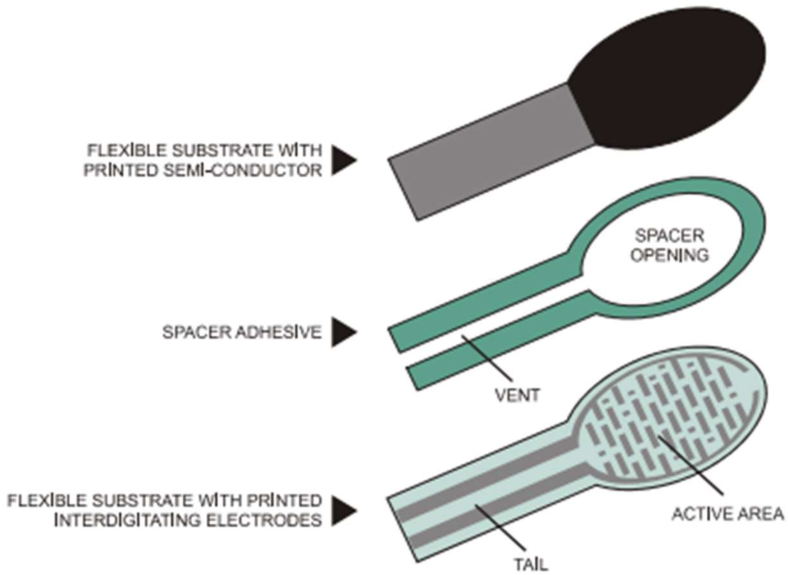


Figure 4.2.2-1: Pressure Sensor Diagram[9]

4.2.3 Accelerometer/Gyroscope

An accelerometer is capable of measuring acceleration forces, which will help detect hand movement. This device is very sensitive, which allows it to even measure the smallest changes in acceleration. These forces can be either static or dynamic. Static forces that are continuous, like gravity, or dynamic forces that can detect movement. By measuring the gravitational pull, the accelerometer can determine the tilt with respect to the Earth. The accelerometer has the capability to detect if an object is moving upwards/downwards, horizontally/vertically and if it is falling over.[10]

This small device can function in a couple of different ways, two of the more common ways are the piezoelectric effect and by the capacitance sensor. In the piezoelectric effect, acceleration forces can cause tiny crystals, embedded in the device, to become stressed. When under stress, the crystals give off a voltage which the accelerometer can then use to determine the orientation and velocity of the device. The capacitance method can detect change in resistance. Change in

[8] See Appendix A – 8
[9] See Appendix A – 9
[10] See Appendix A – 10

capacitance occurs when acceleration forces cause microscopic structures to move. After this, the accelerometer uses the capacitance to determine voltage. One thing to consider when selecting an accelerometer is the option of choosing between analog or digital outputs. Accelerometers that are analog have a continuous output voltage that is proportional to the acceleration force. When the accelerometer is digital, it uses a square wave input so when voltage is at its max, it is proportional to the acceleration. An accelerometer alone is not enough in order to best measure hand/finger orientation.[11]

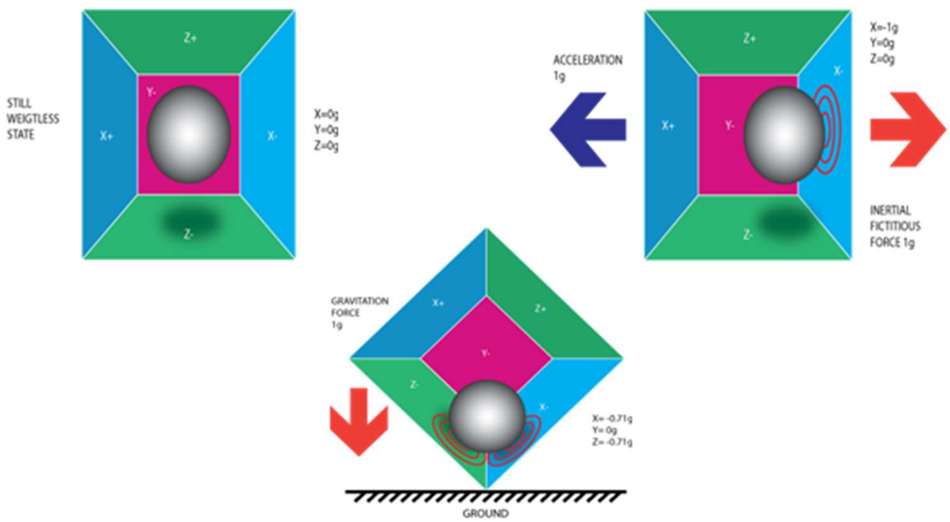


Figure 4.2.3-1: Accelerometer Function[12]

Unlike the accelerometer, the gyroscope measures angular velocity around a particular axis. This device can be used for measuring changes in orientation and to sense rotational motion. This is possible by using Earth's gravity to measure orientation. It becomes harder analyze the reading when the device is moving. When in free fall, the acceleration should read zero. A gyroscope measures rotation in all 3 axes: X, Y and Z. Gyroscopes are tri-axial and in order to get the best orientation results, an initial orientation must be known. The gyroscope has a rotor, a rotating disk that is attached onto a larger static wheel. When the axis is turned, the rotor becomes still, which can tell us the gravitational pull and also tell us which direction is down. The gyroscope can have drift errors. A gyroscope alone is not

[11] See Appendix A – 11
[12] See Appendix A – 12

enough in order to best measure hand/finger orientation. When combined with an accelerometer, they can provide more precise information for the glove to use.[13]

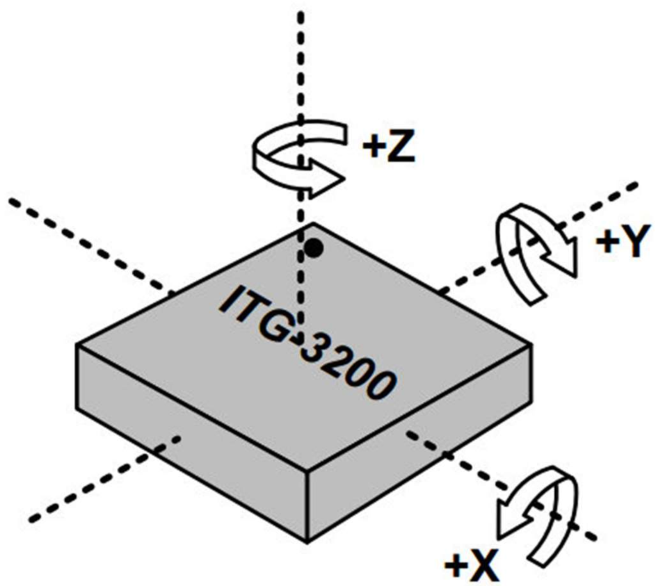


Figure 4.2.3-2: Gyroscope Function[14]

4.2.4 Battery Voltage Sensor

The ASLA will be running wirelessly to ensure easy and fluid usability for the user. Because of this feature, the glove will have to run on an external battery power source. To avoid the battery power of the glove from unexpectedly giving out during the middle of use, a battery voltage sensor should be added and connected to the glove to consistently measure the battery levels and warn users when the levels get too low.

4.3 Microcontroller/Microprocessor

The control unit for this project is very important because it will have to receive all the inputs from the sensors of the system and transmit them wirelessly to a computer where the language processing takes place. This section will discuss all the possibilities of microcontrollers and processors that would be suitable for this project including data-path size, clock speed, power consumption, usability, size, and cost. The final product should be made for a real world application so it should

[13] See Appendix A – 13
[14] See Appendix A – 14

be portable and run off battery power, so the focus of the controller research will be on low power microcontrollers that have relatively high processing power at the same time. Microprocessors will require other components including memory hardware to be bought separately so this should be another aspect attention should be paid when deciding on what is appropriate for the project.

One aspect that must be considered when choosing a microcontroller is the packaging. Some of the packing specifications of the microcontrollers are limited causing issues when it comes to the integration of the microcontroller onto the printed circuit board (PCB). This is a very important factor in determining the appropriate microcontroller for it will dictate whether the microcontroller can be placed into the system during assembly.

Texas Instruments (TI) also provides tools for hardware development as well. They give access to development kits like LaunchPads and BoosterPacks at low costs. The LaunchPads incorporate certain microcontrollers onto boards where there are peripherals like LEDs and buttons already integrated, as well as a USB interface for programming purposes. These LaunchPads give you the ability to prototype devices before you have to lay out a PCB, but this should not be confused with the microcontroller itself. The LaunchPad can also have BoosterPacks added to them for more features/modules adding to the functionality of the board, but not all of these peripherals are necessarily on the microcontroller themselves.

Arduino is a software based company that designs and manufactures hardware, software, and microcontroller boards that are open-sourced and meant for ease of development for projects and devices. Arduino, like TI, allows for expansion boards to be plugged onto an existing board, further expanding the capabilities of a system for testing and building purposes. These shields are easy to mount and cheap to buy. There are even shields with wireless capabilities; using the Xbee shield allows multiple boards to communicate with each other wirelessly. But again like the LaunchPads and BoosterPacks TI makes, these shields are really only meant for testing and quick build purposes.

4.3.1 Arm Processors

ARM is one of the industry’s most advanced microprocessor developers and suppliers. They provide the microprocessor cores that are integrated into many different electronic devices with over 50 billion processors sold. They range from very powerful processors with 64-bit architectures used in the latest smartphones, to power efficient 32-bit architectures used in microcontrollers and safety equipment.

One of the concerns with using just a processor is that it has no real memory unit or any of the other peripherals that a microcontroller may have. The processor may be integrated into a system with external memory and other peripherals but this requires researching many more components, and a more time consuming design as well as a possible increase in the overall size of a board that would be used to implement the overall system.

4.3.1.1 Cortex-A

The ARM Cortex-A is for the highest end performance used in applications requiring more advanced operating systems and complex computational tasks. Cortex-A has two architectures, the ARMv7-A architecture which runs all their 32-bit processors, and the ARMv8-A architecture which runs all their 64-bit processors. The ARMv8-A has a specialized state that allows it to run certain ARM 32-bit applications making the 64-bit processors backwards compatible. The processors come in options from single-core to quad-core, and each core may range from 8-stage pipelining up to 15+ stage pipelining. These processors have a lot of computing power, but they do not have the best power efficiency out of all the ARM processors. The other cores may be more suitable to look at for a tradeoff between high processing power and power efficiency.[15]

4.3.1.2 Cortex-R

The ARM Cortex-R is based off a 32-bit architecture, ARMv7-R, and is used high performance embedded applications. These processors are meant for delivering highly reliable, real-time responses in deterministic processes. They're meant for long run times with little rest states for applications such as wireless baseband, and automotive safety where real-time response is essential. Most of the processors in the Cortex-R series can be configured for Asymmetric Multi-Processing, while Cortex-R7/R8 may also be configured for Symmetric Multi-Processing drastically increasing the performance. This core architecture is closer to one suitable for our project, but may still a little too robust for our needs.[16]

4.3.1.3 Cortex-M

ARM Cortex-M was designed specifically for the use in ultra-low power, scalable embedded applications. While still supplying a powerful 32-bit process, it has reduced overall gate count and smaller memory size than the Cortex-R process, reducing the overall cost and energy used. These processors are also designed to have sleep modes with fast wake up times, enabling longer battery life. It is also

[15] See Appendix A – 15
[16] See Appendix A – 16

designed to be easy to debug and code instructions for. Out of the three cortices this seems to be the best fit for our project. To reiterate though, buying a microprocessor alone means the other peripherals of the system, like memory, must be bought separately and integrated into the system. Some microcontrollers use the Cortex-M microprocessors so one of those may be a better option.[17]

4.3.2 TI Controllers

TI is one of the leading companies when it comes to high performance, low power microcontrollers. The MSP430 family is a set of mixed-signal microcontrollers that have low power consumption, and low cost. One concern about the MSP430 family is that even though it meets the majority of the specifications, it only comes in 16-bit packages and with the number of sensors necessary for the project a 32-bit microcontrollers may be needed. The new MSP432 may be a better option because it has 32-bit packages while still offering low power consumption yet greater processing power.

TI breaks down their low power MSP430 microcontrollers into sub groups based on certain specifications. One of the peripherals that TI uses to differentiate one group from another is with the device type script FR, which stands for FRAM. This is used to because it differentiates a new set of microcontrollers that uses FRAM memory units which combines the best attributes of both SRAM and Flash. It is non-volatile, high speed, has very low power consumption, reliable and stable, as well as having high durability for number of writes. It is also a unified memory element with the ability to update security keys near instantaneously to encrypt and keep the data safe. One of the main aspects mentioned in the technical specifications about microcontrollers with FRAM is that they are normally meant for applications that have a lot of down time (stay in low-power mode often), where depending on the amount of speaking an individual does in a given day may not be the best option either.

The G2x/i2x device types of the MSP430 represent TI's high performance, low cost line. They generally have lower pin counts but still incorporate Flash-based memory, using ultra-low power consumption. These device types also integrate internal pull-up/pull-down resistors and a VLO for low power oscillations.

The last device type of the MSP430 family is the F group. They use the same ultra-low power designs as the previous devices, yet allow for further integration of peripherals for even greater performance. These devices also tend to have slightly

[17] See Appendix A – 17

higher processing speeds than the G2x/i2x devices and larger general purpose I/O pin count.

The MSP432 is a newer technology that incorporates the ultra-low power technology as well, but was designed for higher processing speeds. Since there are only two models of the MSP432 there is limited peripherals and pin options for these devices, as well as less open source work that has been done with them.

Apart from the device type, the MSP MCUs come in all different package options. The most common package types used for integration by hand onto circuit boards would be dual in-line package (DIP), surface mount device (SMD) for smaller mounting, and ball grid array (BGA) for even smaller but more permanent mounting. Of the more common package types for PCB mounting, DIP is the easiest and can generally be found for MSP devices.

4.3.2.1 16-bit

The MSP430 is a family of low powered microcontrollers which has some of the most diverse and specified control units in the world. They have one of the largest selections of peripherals to buy integrated into the microcontroller that sometimes it is easy to get carried away looking at all of them and thinking you will need more than you really do. Their intuitive selection tool helps you separate the microcontrollers based on the peripherals and control units specifications you want and need.

	MSP430 Family				
	F2xx	G2xx	x4xx	x5xx	x6xx
Power Specs					
RAM Retention	0.1 µA	0.1 µA	0.1 µA	0.1 µA	0.1 µA
Real-Time Clock Mode	0.7 µA	0.7 µA	0.7 µA	2.5 µA	2.5 µA
Standby Mode (VLO)	0.3 µA	0.4 µA	N/A	N/A	N/A
Per MIPS Active	220 µA	220 µA	200 µA	165 µA	165 µA
Wake-up Speed	<1 µs	<1 µs	<6 µs	<5 µs	<5 µs
Device Parameters					
Flash Options	1-120 KB	0.5 - 56 KB	4-120 KB	Up to 512 KB	Up to 512 KB
RAM Options	128 B - 8 KB	128 B - 4 KB	256 B - 8 KB	Up to 66 KB	Up to 66 KB
GPIO Options	10, 11, 16, 24, 32, and 48 pins	10, 16, 24, 32 pins	14, 32, 48, 56, 68, 72, 80 pins	29, 31, 47, 48, 63, 67, 74, 87 pins	74 pins
ADC Options	Slope, 10 & 12-bit SAR, 16 & 24-bit Sigma Delta	Slope, 10-bit SAR	Slope, 10 & 12-bit SAR, 16-bit Sigma Delta	10 & 12-bit SAR	12-bit SAR, 16 & 24-bit Sigma Delta
Other Important Peripherals					
USCI Module	x	x	x	x	x
USART Module			x	x	x
16x16 Multiplier			x		
32x32 Multiplier			x	x	x
Temperature Sensor	x	x	x	x	x
Comparator A	x	x	x		
Comparator B				x	x

Table 4.3.2 – 1: MSP430 Family General Comparisons[18] [19]

The MSP430F5x/6x series are the comparable 16-bit microcontrollers to the MSP432 32-bit controller. The two are very similar in all of their device specifications with the exception of a few of their integrated peripherals. The max clock speed for both devices is 25 MHz, but can be lowered for better power consumption. The MSP430 also has the ability to operate in six different power modes further lowering power consumption. They have up to 512 KB Flash memory on chip as well up to 66 KB RAM, and can operate at low voltage levels from 1.8-3.6 V. Their power consumption is relative to the processor speed with a relationship of 195 µA / MHz while active. One of the features that sets apart these two MSP430s from the others is the USB 2.0 peripheral. The MSP4306x series even has an LCD peripheral integrated on chip.[20]

Some peripherals that may be important for some of the applications for the project include multipliers, comparators, as well as analog-to-digital converters (ADC). The multipliers may be used if it is necessary to filter a signal once it has already been converted to a digital signal. It may be necessary to use a comparator to compare to input signals with each other, and could help differentiate between finger and hand positions for forming words. Since there are going to be multiple analog inputs coming into the microcontroller from the sensors, it would be necessary to convert them to digital signals for the microcontroller to process and transmit to the computer for analysis.

4.3.2.2 32-bit

The MSP432 is the new 32-bit version of the MSP430 and provides high-performance at 48 MHz while still maintaining the low power consumption like the previous versions. With the maintained low power levels, long battery life was maintained. With the increase in clock speed, floating-point performance was increased as well helping execute advanced algorithms. It utilizes ARMs Cortex-M4F discussed early. The increase in both the processor speed and performance allows the MSP432 to carry out real-time processing which is good for analysis of data from multiple sensors. It also allows for better analog integration than the other microcontrollers again improving the performance of sensor analysis for motion and change. The MSP432 also has a large number of general purpose I/O (GPIO) pins.[21]

[18] See Appendix A – 18
[19] See Appendix A – 19
[20] See Appendix A – 20
[21] See Appendix A – 21

4.3.3 Atmel Microcontrollers

When it comes to competing with TI in microcontroller sales and distribution, no other company than Atmel comes to mind. They lead the industry in their advanced design and manufacturing processes of microcontroller systems. Like the TI controllers Atmel controllers integrate peripherals for components such as nonvolatile memory, mixed-signal, advanced logic, and radiofrequency (RF).

Apart from the Atmel microcontrollers that integrate some powerful ARM processors, they were really the first microcontroller company to integrate on-chip flash memory through a modified Harvard architecture. The original AVR microcontroller, named after the creator's, was originally an 8-bit single-chip microcontroller. There have since been modifications to add new peripherals and make the systems more power efficient, as well as the creation of 16-bit and 32-bit AVR versions for even more power. Due to this highly efficient architecture, Atmel leads in low powered, high speed system integration while providing security and support.

4.3.3.1 8/16-bit

The more common ATmega Atmel family offers the largest selection of devices for choosing the right one with the specific peripherals needed. On this 8-bit microcontroller you have the ability to choose based on memory size, pin count, power consumption, and other peripherals. The ATmega family also has a number of analog capabilities including an advanced programmable analog amplifier, an analog comparator, and ADC, and a DAC. With the higher pin count models and integrated analog components, the ATmega may be an option.

The next of the Atmel AVR family is the ATtiny, and like the ATmega it only comes in 8-bit packages. Since this device was designed with the specific purpose of having the smallest package size, and lowest power consumption it has some trade-offs that make this microcontroller an option that will not be suitable. Its pin count will be too low and it won't be able to incorporate all the necessary peripherals.[22] [23]

Thirdly the ATxmega may be the most suitable, for it still incorporates as small of a packaging size as the ATmega but offers a 16-bit version. It also has models that incorporate larger memory options than the ATmega, and a more powerful 12-bit ADC with gain stage built in with 4MSPS. Has better real-time performance with direct memory access and complete libraries of device drivers. Within the ATxmega

[22] See Appendix A – 22
[23] See Appendix A – 23

device family there are five families. The A series is their standard ATxmega, where the B series incorporates an LCD peripheral, the C series using less power with USB, the D series uses less power without USB, and the E series is even lower in power consumption with faster wake up times and the smallest packaging.[24]

4.3.3.2 32-bit

Like some of the other Atmel MCU families, the 32-bit AVR UC3 microcontroller family has many different series. As a whole the UC3 family is the most efficient 32-bit MCU to date. It combines the Atmel AVR processor architecture, to maintain the power efficiency, and large selection of peripherals for large data throughput and device connectivity. The 32-bit comes with higher GPIO pin counts than the other Atmel MCUs as well as larger flash memory options and device connectivity.[25]

4.4 Wireless Communication

There are several ways of establishing connection between two devices. One way is to hardwire it and make the connections with conventional wire from point to point. The second and newer way is to make the two devices communicate wirelessly. This means establishing a connection using radio waves. These radio waves can be sent at different frequencies. Below will be shown a detailed explanation of the different radio frequencies allowed and as well as the different methods that use these radio frequencies.

4.4.1 Wi-Fi

Wi-Fi is usually known as trademark phrase which simply means IEEE 802.11x. This wireless technology is based on the use of radio waves to provide internet connectivity on a network, the purpose of this new technology is to introduce high speed internet connection. But the main definition of wireless was introduced by the 'Wi-Fi Alliance'[26] who registered Wi-Fi as trademark name to define 'wireless local area network' or WLAN. The way it works is by establishing a connection between a sender and a receiver. Since no physical connection is needed, Wi-Fi uses radio frequencies; these RF currents are given to the antenna which creates a magnetic field that propagates through space. To do this the use of a wireless adaptor is needed to create a hotspot. A hotspot is basically where this Wi-Fi connection is accessible based on the available networks. These networks can

[24] See Appendix A – 24
[25] See Appendix A – 25
[26] See appendix A – 26

either be public or private depending on how they were set up. The next part of the connection requires an access point for these hotspots to connect. The access point job is to broadcast the signal to the devices able to detect it only by using a wireless adaptor. Also, these Wi-Fi networks work in two ways, by sending and receiving signals. Therefore, after a signal is received by the wireless adaptor, the same signal is also sent to a router using an antenna to be decoded and then sent to the internet through a wired connection known as Ethernet. The same process happens when receiving internet date, it first goes to the router, and it is coded and sent to the wireless adaptor. The radio frequency signals that Wi-Fi works on are 2.4GHz and 5GHz depending on the use that is needed. It is important to mention that Wi-Fi is very popular due to the fact that is compatible with most of the new devices such as computers, cell phones, printers, game consoles, etc. in general, Wi-Fi can be implemented using any of the 802.11 specification standards which vary on throughput and speed. These use of these standards depends on what the user wants of their Wi-Fi connection. These standards will be explained below based on their advantages and disadvantages.

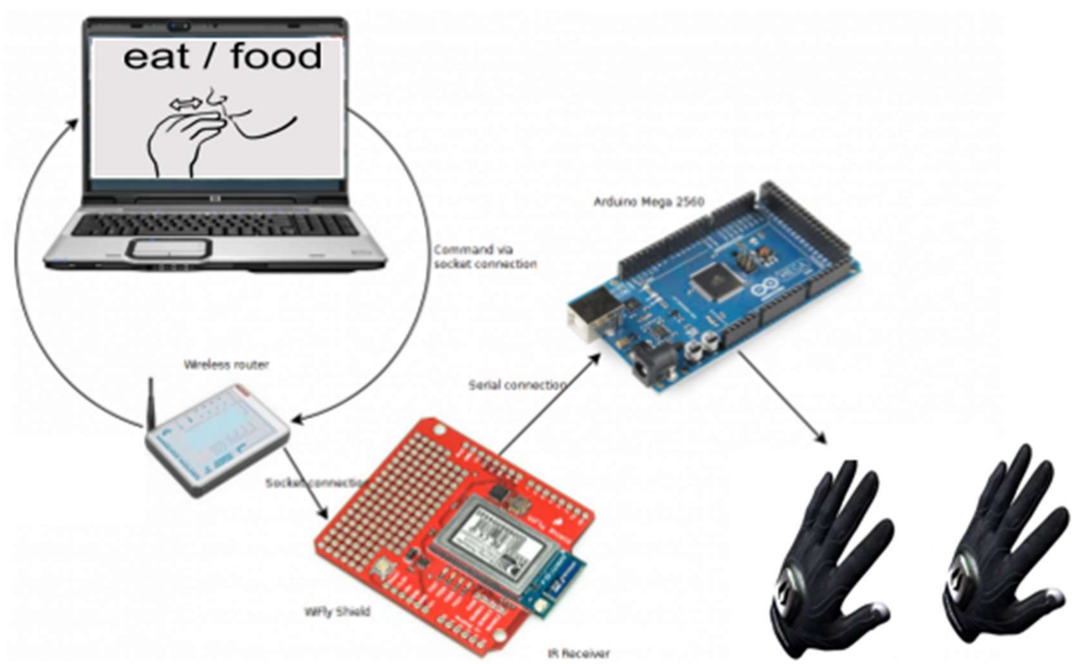


Figure 4.4.1 – 1: Wi-Fi Wireless Communication Diagram

4.4.2 2.4 GHz vs. 5GHz

Wireless connectivity implies the use of frequency bands, these bands have a specific use because one band does not fit all. Frequency bands are regulated by agencies like the Federal Communications Commission (FCC) in the United States. In Europe, it is called the Conference of Postal and Telecommunications Administrations (CEPT). These agencies regulate the use of frequency bands by allocating them for certain uses. The usable bands in the spectrum are called licensed which basically means that the user needs to buy the license from local regulator in order to operate a rio transmitter in these frequencies. A good example is cellular communications companies who buy these licenses in order to operate. When buying these licenses, government uses what is called government auctions to sell these spectrum bands in a fairly manner for commercial use. It is important to mention the other use of frequency bands which is for Industrial, Scientific and Medical uses (ISM). These bands are allocated the International Union's Radio communication sector (ITU-R). Some of these bands include 433 MHz, 868 MHz, 915 MHz and 2.4 GHz which are basically used for wireless communications such as remote controls, and Wi-Fi. Although, these spectrum bands allocations vary from country to country.

Worldwide, the 2.4 GHz band became popular due to its facility of used in all regions as unlicensed. Therefore, due to its universality the 2.4 GHz band makes easier the development and use of products much easier with other nations. Another reason why 2.4 GHz is so popular is because the cost of setup is cheap when compared to other spectral bands. This makes the 2.4 GHz band become a crowded place as more wireless devices are added to our everyday lives. In some reports, it has been said that this band is so crowded and noisy that it results in some dropped connections. As mentioned before, the 2.4 GHz band has become a very busy place where the most common wireless devices operate. These devices include cordless phones, baby monitors, garage remotes and many others. It is important to mention that even appliances such as microwaves as they get older begin to emit some radiation outside the unit which cause noise in the 2.4 GHz band that causes a lot of interference. Of course, these spurious emissions are not supposed to happen since they are also harmful for human health. Much of this interference causes a huge negative impact on access points and routers. Consequently, this heavy traffic causes a significant decrease in speed and throughput of a wireless network. Also, as technology increase so does the increase of smartphones which can now access the Wi-Fi 2.4 GHz networks for internet access. This also creates a lot more traffic on this band. Although, an advantage of using the 2.4 GHz band is the range that it can achieve. Since lower frequency radio waves can propagate further, this creates a huge advantage in closed places where there are a lot of walls and solid materials. But the rule is that lower frequency radio bands have less bandwidth for channels so this creates a challenge depending on what is trying to be implemented. For applications that do not use a lot of bandwidth like internet browsing and emails, it is suitable to use the 2.4 GHz band. As mentioned before, the 2.4 GHz band supports the standard specifications of the wireless 802.11b, 802.11g, and 802.11n. The design of this band is three non-overlapping channels which does not give a lot of room to work with depending on its usage. As a whole this band is a great option depending on what use it is giving since it offers a good range and easy accessibility due to its unlicensed use, but this band also comes with a lot of interference because of its popularity. So when using the 2.4 GHz band make sure to pick the right channel in order to obtain maximum throughput. For example, depending on the location it might be good to use 2.4 GHz when living at an apartment complex since these have many wireless routers. In which case it will be better to go with the 5 GHz band.

The 5 GHz band offers different features due to its high frequency. This high frequency simply means a shorter range of transmission. So solid objects like walls, plaster, glass, and steel will decrease the transmission range significantly. This band also offers a great advantage which is minimal noise interference because of the less usage of this band. This aspect generates higher data transfer rate or speed. Also, since this spectral radio band is not as crowded, it provides fewer disconnects which allows for a better experience. Unlike the 2.4 GHz band, common wireless devices and Bluetooth will not cause any interference with this band. This decrease in noise is because sources of noise like microwaves do not operate in such high frequencies. In addition, the 5 GHz band allows for a faster throughput or transmission rate, uses less power consumption which generates better network execution. One disadvantage of using this spectral band is that not all devices have the option to operate on the 5 GHz band. Another disadvantage of using 5 GHz is that when two signals are transmitted through similar antennas while using the same power, the signal using higher frequencies will cover less distance that if the same two signals were using the radio spectral band of 2.4 GHz. Consequently, nearby devices will not cause interference, in other words, in a neighborhood this will be an advantage because neighbors wireless devices will not cause any interference since the high frequencies will not affect the others. Factors such as location affect the use of 5 GHz as mentioned before. When living in a secluded area with neighbors at least one fourth of a mile away from each other such as a country home, then it might be better to use the 2.4 GHz band since there are not many devices that could interfere with it. In suburban areas with small open spaces it is a better option to use the less populated 5 GHz radio spectral band. In general, the 5 GHz band offers greater speeds and throughput but it limits the transmission range significantly. Although this is a big advantage over the populated 2.4 GHz band, it turns out that it is proven to be very costly to implement this radio technology. Many reports say that some find no difference between the two radio spectral bands, but this becomes a significant factor when transmitting large pieces of data like when streaming videos where a fast throughout is needed.

Wi-Fi Standard	Frequency	Wireless Speed (Max)	Wireless Distance (Max)
802.11a (1999)	5 GHz	54 Mbps	390 ft
802.11b (1999)	2.4 GHz	11 Mbps	460 ft
802.11g (2003)	2.4 GHz	54 Mbps	460 ft
802.11n (2009)	2.4/5 GHz	300 Mbps - 900 Mbps (Combined)	820 ft (2.4 GHz) / 460 ft (5 GHz)
802.11ac (Draft - 2012)	5 GHz	433 Mbps - 1,733 Mbps	Up to 820 ft (Amplified)

Figure 4.4.2 – 1: Wireless Specifications and their Frequencies [27]

4.4.3 Bluetooth

This is a standard of technology which allows the exchange of data over a short distance. Bluetooth uses Ultra High Frequency radio waves which means that it has short wavelengths which explains why Bluetooth only works on short distances. This technology only works in the ISM band described in the last section. Specifically, Bluetooth works on the radio spectral bands between 2.4 GHz to 2.485 GHz. It was invented and introduced by Ericsson as an alternative wireless communication technology. One of the reasons it stands out is because it allows for easy connection of devices without synchronization problems. Bluetooth has network of devices known as 'piconet'. This network can range from two devices up to eight devices connected simultaneously. Each of these devices have a role, at least one of them acting as the master controlling all other devices, and the rest are called slaves which are driven by the master.

4.4.3.1 Bluetooth BR/EDR

Bluetooth versions of specifications support what is called downward compatibility. This means that new specification versions support older versions. As part of the Bluetooth specifications, one of the most common used is known as Bluetooth

[27] See appendix A – 27

Basic Rate or Enhanced Data Rate. This specification is also known as the Bluetooth Core Version 2.1. The main aspect of this specification is that it allows devices to select an option of 'add Bluetooth device' making this device visible to connect to others at any time. Also, this feature complements previous Bluetooth versions like v1.2 in terms that it improves peak data rates to up to 2 Mbps and 3 Mbps which is a huge increase from the basic data rate of 1 Mbps. The way this works is by using phase shift keying (PSK) implementation which in turn improves the efficiency of the data rate speed up to three times the original one. It is important to mention that all the different modulation types like PSK and QPSK use the same 1 Mbps symbol for basic rate BR and for Enhanced Data Rate EDR. With the only difference being that Enhanced Data Rate utilizes more spectral bandwidth. This is because EDR modulation uses root-raised cosine filters instead of Gaussian filters used by the basic rate format which are narrower, therefore; using less bandwidth. The federal communications commission FCC has allowed the Enhanced Data Rate EDR to work on the 2.4 GHz band by allowing the -20 dB spectral requirement from 1 MHz to 1.5 MHz.[28]

Since Bluetooth allows the connections of personal devices with each other. The EDR allowed the introduction of faster and better performances. This was generated due to the consumer need to connect with multiple devices. At first Bluetooth was designed to serve as personal devices such as portable headsets, handheld devices. This in turn served as great way to replace wired connections giving more mobility to the consumer. These Bluetooth devices can be synchronized to work in a Bluetooth personal area network PAN. Since the consumer needs added more devices requiring more power and faster speed rates. These new applications also added the need for more and longer battery life. The main problem of Bluetooth was that users need to run more than one application at the same time and connected to the same PAN or Personal Area Network. Other examples of the needs for the betterment of the Bluetooth technology are applications that increase the bandwidth usage such as printers, cell phones, headsets, wireless keyboards and mice. In our project the devices would simply be each glove, the computer and even a phone device if the android application is created. With the use of EDR these devices could work properly with all of the required needs to perform such as giving them enough bandwidth room and better data rates for overall performance. The reason EDR is needed is because in scenarios like our project where multiple devices need to be connected wirelessly and work at high data rates with low power consumption. It is important to mention the effects of the Enhanced Data Rates on power consumption. Since there's a lot more transfer of data or in other words higher data transfer, the

[28] See appendix A – 28

wireless devices do not need to be on for a long time. This allows for less power usage and decreases power consumption between the multiple devices in a piconet and personal area network or PAN. Therefore, battery life is increased, in our project this is an extremely useful technology development because for practical use of our glove we need to have long battery life and also make sure that the battery is small enough to fit in our glove design without altering our glove specifications. A good example of a possible scenario that could be design by our group in terms of wireless communication between all the required devices will be shown in the diagram below. This diagram connects all devices in one PAN with simultaneous communication.

Figure 4.4.3.1 – 1: Possible Wireless scenario using Bluetooth EDR

This figure depicts a perfect scenario between possible devices that can be used in Bluetooth piconet for our project. All of these devices would consume little power or basically will be off when they are not in used. Although they are all synchronously connected at the same time, all of them will not be used at the same exact time. For example, when we are only using the just the translation and processing of signals to the desktop, just the gloves and the desktop will be in use. This means that the other devices in the Bluetooth piconet like the headset, the cell phone device, the keyboard and mouse will not be in use. In other words, these devices will not be consuming any power and will basically be off until further notice.

Another possible scenario of the scenario from the above diagram could be when using the sign language translation from text to speech. This scenario would require the use of the two gloves each with a separate Bluetooth module, headset, desktop computer, and even the mouse and keyboard depending on what the user wants to do with the translation.

The last and possible scenario of our project would include developing an Android application that would allow the translation to be directed to the cell phone device that which of course has Bluetooth EDR capability. The possibilities of using Bluetooth EDR are really endless and is a growing technology nowadays with in home devices and even car personal area networks PAN.

4.4.3.2 Bluetooth Smart

This new Bluetooth technology is also known as Bluetooth Low Energy LE, or just BLE. The first intention of this new technology was for the health related field and for even industries, and even for security and applications, designed by the Bluetooth Special Interest Group SIG. The main reason it stands out from other old standards like the Classic Bluetooth technology is that it provides cheaper implementation for the public and also that it uses less power consumption and it still keeps a similar range compared to the classic Bluetooth.

The origin of this new technology come from Nokia in 2006 which introduced this Bluetooth technology as Wibree. Nokia developed this technology for low power devices to be able to use Bluetooth for communications. Although, when it first came out it was thought of a new technology as a competitor of the normal Bluetooth, later the technology was merged and added to the Bluetooth specifications. To this date Android, IOS, Windows phone, OS X, Windows 8, and Linux all support Bluetooth smart and it is predicted by SIG that by the year 2018 almost more than 90% of all Bluetooth capable devices will be able to support Bluetooth smart.[29]

Bluetooth Smart or BLE is also known as the Version 4.0+ in the Bluetooth specification. This new specification is highly impressive because of the new feature that allows less power consumption, this being the main reason for the update to the Bluetooth specifications. Another very good improvement is that it allows for more frequent transfer of data. The frequency that it can send data now would be up to five times per second and in the normal case of just one time per hour. This causes the battery life to increase at least five or ten times more than the original battery life of the desired device. It has been said that devices using this kind of

---

[29] See appendix A – 29

technology like keyboards and mice that can last up to five years before the battery has to be replaced since the power consumption is so efficient.

This technology has been incorporated into everyday things that are used sporadically. These include devices that measure your heart rate and steps like a smartwatch, these devices communicate with your cell phone device through the Bluetooth smart technology. Other devices include the new and popular Fitbit which is a small device that looks like a small wristband which measures steps and calories everyday while synced to the cell phone device. Basically many of our everyday lives already use wireless communication that controls appliances like tablets, doors, and even smart TVs.

An excellent feature of Bluetooth smart is that it is very versatile. This is because developers have allowed already Bluetooth devices to be able to accept new Bluetooth technologies. This features is based on the fact that Bluetooth Low Energy BLE or Bluetooth smart is developer friendly. This just means that the development architecture is very flexible which makes it cost effective to develop. Limitations are really only set by humans because the possibilities to implement this technology are really endless. The last feature it provides is the access for the tightest security that has a 128-bit AES data encryption.

	Classic Bluetooth technology	Bluetooth low energy technology
Data payload throughput (net)	2 Mbps	~100 kbps
Robustness	Strong	Strong
Range	Up to 1000m	Up to 250m
Local system density	Strong	Strong
Large scale network	Weak	Good
Low latency	Strong	Strong
Connection set-up speed	Weak	Strong
Power consumption	Good	Very strong
Cost	Good	Strong

Table 4.4.3.2 – 1: Bluetooth Smart vs Classic Bluetooth[30]

[30] See appendix A – 30

4.5 Software

4.5.1 Programming Languages

There are multiple programming languages available for use. Some are better than others for certain applications. Our application, while millisecond timing is not necessary, a long delay is g languages, each with their own benefit and drawback. The main languages that we are considering are C, C#, C++, Python, and Java. While millisecond timing is not crucial for our software application, it should have no more than a reasonable delay from input to output. Reasonable being two word behind or less.

The software is split into two parts, software on the microcontroller and software on the computer. For the main computer program Python will be used for prototyping stages, while C++ will be used for the fully developed end product. If because of time the C++ final product cannot be developed, the most stable version of the Python prototype will be used. Most of the microcontrollers under review are coded in either C++, C, or Assembly.

Two languages, C# and Java, inherently have a downside due to the fact that they need to run on virtual machine in order to run the program. Rather than creating an executable what the two languages do is create an intermediate language, called bytecode, so that the virtual machine can run the program. While this is good for the portability of the program, there are no guarantees that the user is going to be able to install the virtual machine on the computer they need to use at the time of use, such like a school computer in a lecture room. This adds another layer of complexity to the product that the end user does not have to deal with if dealt with from the development phase. Not only this but Java has been notorious for multiple security flaws and we do not wish to force the user to install potential hazardous software onto their computers.

4.5.2 Software Description

The gloves must retrieve sensor data at a reasonable rate to ensure accuracy. Timing is critical for the microcontroller on the gloves. The computer software is going to be able to acquire data from the gloves, and map it to known letters or phrases in the database. Using machine learning to learn new phrases and words and decision making when the input is not ideal. The user interface must be simple and easy to navigate/use for the user.

4.5.2.1 Data Samples

The rate of sampling data points is important, too fast and we are just wasting execution time on gathering data that is not necessary. Too slow and the desired output from the software will be incorrect at worst, slow at best. An Institute of Electrical and Electronics Engineering published paper in Biomedical and Health engineering recommended a sample acquisition rate of 400-500 Hz for electromyography sensor sample rates for able-bodied and amputee subjects.

> "Our results showed that decreasing the sampling rate from 1 kHz to 500 Hz only caused 0.8% reduction of the average classification accuracy over five able-bodied subjects and 2.2% decrease over two transradial amputees. When using a 400 Hz sampling rate, the average classification accuracy decreased 1.3% and 2.8% in able-bodied subjects and amputees, respectively."[31]

Using this as a guideline for our design in having able-bodied subjects performing. We will try to achieve a rate of 400-500 Hz, as testing of the prototype commences we will adjust the sample rate as needed, possibly because the limiting sensor data acquisition rate. Listing the set of data:

- Accelerometer X, Y, and Z axis data
- Gyroscope X, Y, and Z rotational data
- Flex sensor resistance data
- Contact sensor resistance data

The data points will be read over a time frame and mapped to a dictionary of known letters or words with machine learning. This will generate the text needed to drive the text-to-speech API.

4.5.2.2 Machine Learning

Machine learning is the "study of pattern recognition and computational learning theory in artificial intelligence."[32] This gives the program the ability to make decisions and be able to learn without need of the programmer. Machine learning is a branch of artificial intelligence, and the theory behind it is computational statistics. There are many different machine learning algorithms in development, all with different focuses, ways of training the software, and how the data is handled. The software learns new words and phrases from new data that the user

---

[31] See Appendix A – 31

[32] See Appendix A – 32

will input via the library expansion part of the software, this is called supervised learning.

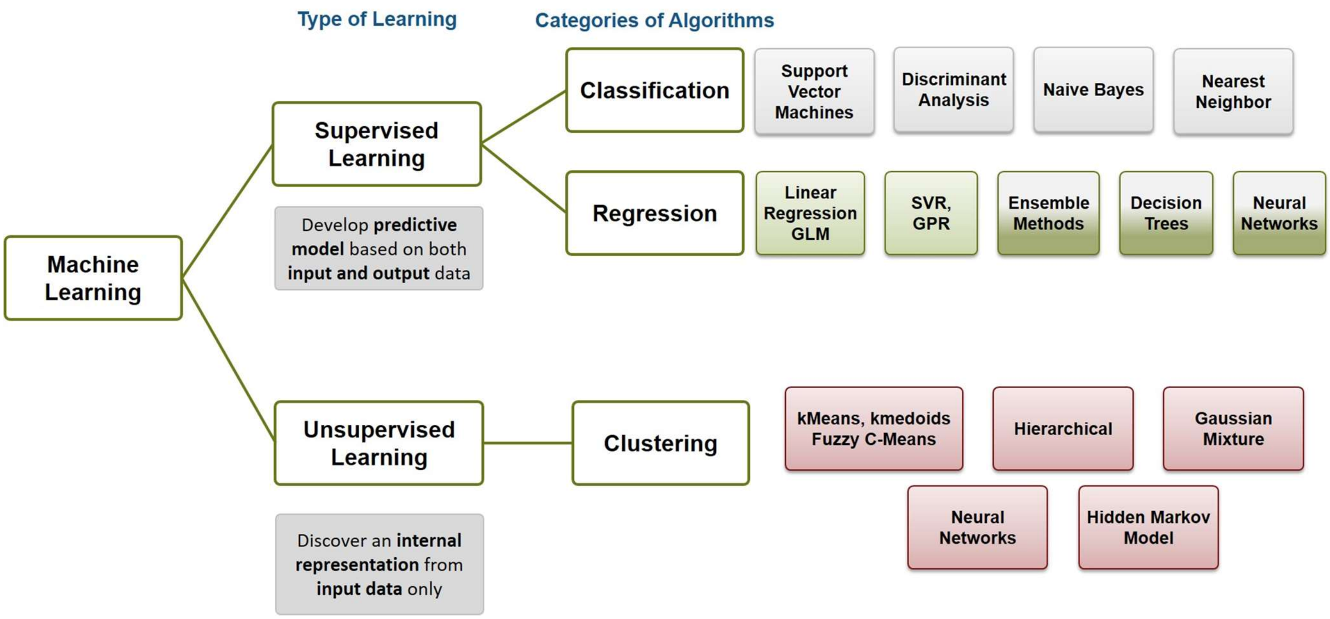


Figure 4.5.2.2 – 1: Machine learning algorithms classifications[33]

Supervised learning is needed and more beneficial to our project because we can improve accuracy and we have to show the program the data sets needed for the correct output to be displayed by the software. Within supervised learning there are two categories of algorithms, Classification and Regression.

Classification is “the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations whose category membership is known.”[34] The observation being what the software captures in terms of data from the sensors and category being what letter/word/phrase it belongs to. A large number of algorithms in the category of classification can be considered linear classifiers. A linear classifier is a linear function that assigns a score to each possible category compared to the input as a vector with weights for each individual component that contributes to the input. The category most likely associated with the input has the highest score. This kind of algorithm has the general form of:

$$score \quad _i, k) = \beta_k \cdot X_i$$

[33] See Appendix A – 33

[34] See Appendix A – 34

Where $X_i$is the input vector, k is the category we are checking against, β_k is the vector of weights for the category we are checking against, and $score(X_i, k)$ is the score value associated with the input when compared to the category k. Here is a list of well known classification algorithms we are considering:

- Support vector machines
- Nearest neighbor
- Naive Bayes
- Neural network (depending on implementation)

Regression is another category under supervised machine learning. Regression is a statistical process for estimating the relationships among variables. Outputs from this method are continuous. The focus is this is to find the relationship between a dependent value and one or more independent variables. But due to the nature of regression there may be some relationships between a dependent and independent variable that may not actually be there. Regression is mostly used for prediction and forecasting. The general regression model is in the form:

$$Y \approx f(X, \beta)$$

Y is the dependent variable that we are trying to relate, X are the independent variables we are checking, and β are unknown parameters, which may be either a scalar or vector. The function f is based on the knowledge on the relationship between Y and X that does not require actual data. If relationships are observed between Y and X the number of them is N, when N is less than the length of β(let's call it k) the regression model cannot be performed, as there is not enough data to find β. If N is equal to k and the function f is linear, the equation can be solved for exact values rather than approximate. If f is nonlinear then there might be no solution, or multiple solutions. The most common case is when N is greater than k, there is enough information to estimate β that fits the data in some way. These are some algorithms using a regression model:

- Decision tree
- Neural networks (depending on implementation)

4.5.3 Text-To-Speech

Speech synthesis, or more commonly known as text-to-speech system. Synthesized speech is generated by combining bits of pre-recorded speech based on the word being generated.

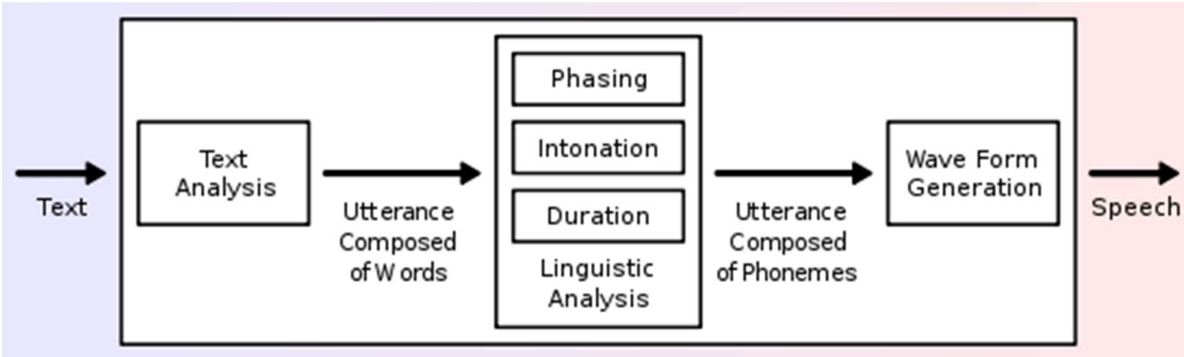


Figure 4.5.4 - 1: Overview of a text-to-speech system[35]

4.6 Power System

Power systems by definition is a network of systems supply power to the entire project design. We will discuss in this section the multiple ways to power the sign language glove, for example:

- Batteries
- Chargers
- Voltage

4.6.1 Batteries

This project requires the use of batteries in order to power the system. In this section, we will be discussing the most common types of batteries that can power a microcontroller that will be used in our project. We decided which battery was best to use considering all the different pros and cons of each type of battery. We will be discussing these types of batteries:

- Lithium Ion
- Lithium Polymer
- Nickel Metal Hydride
- Nickel Cadmium

Since this is a portable device, we need a battery that is powerful enough to power the glove without any complications. It must also be lightweight and also relatively thin is ideal when trying to make the glove easy for the handler to use.

[35] See Appendix A – 36

4.6.1.1 Li-ion - Lithium Ion

Lithium Ion batteries can store a lot of energy because they are made up of carbon and highly reactive lithium. This type of battery is one of the most known when it comes to portable electronics and it is one of the fastest growing systems. Lithium Ion is the best when looking for something that is lightweight, has a high energy density and a low self-discharge. This type of battery can be found in a thin size, which is ideal when trying to make the glove as compact anf lightweight. An advantage to using a Lithium Ion battery is that it has the ability to recharge at a relatively high rate. A lithium Ion battery has a fairly good cycle life and can have a max of 1000 cycles. It is low maintenance in that it does not need periodic discharge and no memory. One downside to a Lithium Ion battery is that it is fragile and needs a protection circuit in order to keep it safe. The protection circuit limits voltage and current. Aging is also another problem that can be considered in that after 2 or 3 years, the Lithium Ion battery performance will decrease with time, especially when under hot temperatures. A use for Lithium Ion batteries includes laptop and cellular phone batteries.

4.6.1.2 Li-Po - Lithium Ion Polymer

One benefit of this type of rechargeable battery is that it has the ability to be custom made. Manufacturers don't have to follow a certain standard design and are allowed freedom in choosing the shape, size and form needed for any type of design or project. The design can even be as thin as a credit card. A Lithium Polymer battery can have a max of 500 cycles. The Lithium Polymer battery is similar to the Lithium Ion battery but there are significant differences that differentiate them. This type of battery is much thinner than the Lithium Ion battery but it also has a higher cost. It also has a lower energy density than a Lithium Ion battery. One downside to the Lithium Polymer battery is that it is low in conductivity and can't conduct enough current needed for a modern communication device. The conductivity can increase but only by making the battery heat up to about 140°F and that is not beneficial when working with portable devices. Most lithium polymer batteries produced are hybrids, meaning gelled electrolyte are added to the battery system to make it more conductive. In reality, by adding the gelled electrolyte, the correct name for the battery becomes lithium ion polymer.

4.6.1.3 NiMH - Nickel Metal Hydride

This type of rechargeable battery uses hydrogen to store energy. One benefit of using a Nickel Metal Hydride battery it uses environmentally friendly metals so it is not as hazardous as other types of batteries. A NiMH battery can have up to 40% of a higher energy density than Nickel Cadmium. Even though it beats NiCd in energy density, NiMH is not as durable as NiCd and it has a much lower cycle life compared to Lithium Ion and NiCd batteries. One negative aspect of using NiMH is that it has a high self-discharge rate. This type of rechargeable battery costs about 20% more than a NiCd battery. It also requires the battery to be maintained regularly by fully discharging to prevent formation of any crystalline. NiMH has become more popular to use than NiCd when it comes to wireless communication and in the mobile industry. Li-ion and NiMH can actually hold about the same amount of charge.

4.6.1.4 NiCd - Nickel Cadmium

Nickel Cadmium batteries have a fast charge and discharge rate and also function well at low temperatures. One great thing to note is that NiCd has a low internal resistance, about less than half of NiMH. This type of rechargeable battery can be found at a low cost when implementing it on a lower power design. A weakness can be found when looking at how susceptible NiCd is to memory effect. Memory effect is when the battery remembers how much was discharged and limits the next charge accordingly. NiCd batteries can be damaged if they are overcharged. Its cell nominal voltage is relatively low at 1.2V compared to alkaline cells at 1.5V. The metals used to make a NiCd battery are toxic and even some countries are limiting/banning the commercial use of it because it isn't environmentally friendly. The max discharge rate for an AA size NiCd battery is about 1.8A and it increases as the size of the battery also increases. The terminal voltage only changes a bit when it is being discharged.

	NiCd	NiMH	Lead Acid	Li-ion	Li-ion polymer
Gravimetric Energy Density (Wh/kg)	45-80	60-120	30-50	110-160	100-130
Internal Resistance (includes peripheral circuits) in mW	100 to 200[1] 6V pack	200 to 300[1] 6V pack	<100[1] 12V pack	150 to 250[1] 7.2V pack	200 to 300[1] 7.2V pack
Cycle Life (to 80% of initial capacity)	1500[2]	300 to 500[2,3]	200 to 300[2]	500 to 1000[3]	300 to 500
Fast Charge Time	1h typical	2-4h	8-16h	2-4h	2-4h
Overcharge Tolerance	moderate	low	high	very low	low
Self-discharge / Month (room temperature)	20%[4]	30%[4]	5%	10%[5]	~10%[5]
Cell Voltage (nominal)	1.25V[6]	1.25V[6]	2V	3.6V	3.6V
Load Current - peak - best result	20C 1C	5C 0.5C or lower	5C[7] 0.2C	>2C 1C or lower	>2C 1C or lower
Operating Temperature (discharge only)	-40 to 60°C	-20 to 60°C	-20 to 60°C	-20 to 60°C	0 to 60°C
Maintenance Requirement	30 to 60 days	60 to 90 days	3 to 6 months[9]	not req.	not req.
Typical Battery Cost (US$, reference only)	$50 (7.2V)	$60 (7.2V)	$25 (6V)	$100 (7.2V)	$100 (7.2V)
Cost per Cycle (US$)[11]	$0.04	$0.12	$0.10	$0.14	$0.29
Commercial use since	1950	1990	1970	1991	1999

Table 10.2.1: Rechargeable Batteries Specifications[36]

4.6.2 Chargers

Since our project is portable, we need to have a charger that will charge the batteries for our device. We will discuss the different chargers that are compatible with the batteries listed above. There are two different ways to charge the batteries for our system. We could either have a removable able system for the batteries. This would mean the battery can be removed and charged separately through a charge dock. There is also another method to charge the batteries that is just as

[36] See Appendix A – 37

useful if not better than the previously mentioned. It is to have a USB charging mechanism integrated into the design. This eliminates the need to constantly remove the battery from the glove, which could cause wear and tear.

4.6.2.1 Li-ion Charger

Since the base for this charger is Li-ion, both the Li-ion and Li-Po batteries are compatible with this charger. This type of battery charger works by using a constant voltage charging method. From the figure below, we can see the charge state for a Li-ion charger. At first, the system charges at a constant current until the voltage per cell reaches 4.2V. After the battery reaches 4.2V, it will hold that voltage and charge at a constant voltage instead. The charger lowers the current necessary to keep the charge at a constant voltage until current reaches about 10% of the initial charge rate. One of the only ways the battery charger can overcharge is if the actual system or charger malfunctions, which can accidently apply a higher voltage to the system causing a shorter life span. This is why setting a max voltage is used to prevent from overcharging. It is best to stop charging the batteries once they reach full charge to prevent any problems.

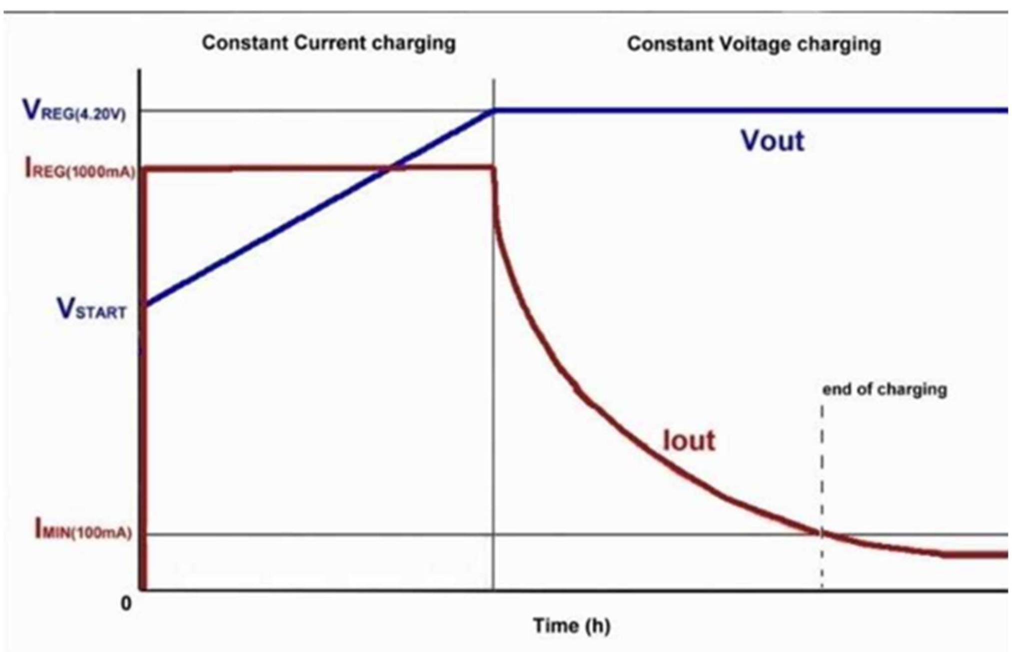


Figure 4.6.2.1-1: Charge Stage for Lithium Ion[37]

[37] See Appendix A – 38

4.6.2.2 NiMH/NiCd Charger

Since NiMH and NiCd batteries share similar characteristics; they can be charged using the same type of charger. Charging these types of batteries is much more difficult than charging Li-ion batteries. The most cost efficient way of charging these batteries is by using the overnight charging method, which chargers at a rate of C/10. Cells now a days have oxygen recycling catalyst that prevent from overcharge damage to the batteries. If the rate is higher than C/10, it won't be able to help prevent from overcharging. The minimum voltage necessary to fully charge is dependent on the temperature. For example, at 20 degrees Celsius, you need about 1.4V/cell in order to be at full charge. A charge rate that is less than C/10 will extend charging time and it might not charge at all if the charge rate is really low.

Another method used for charging NiMH and NiCd batteries is the rapid charge/trickle charging method. From the figure below, it can be seen that the voltage/temperature rapidly increase until reaching full charge. At the end of the charge, the charger measures dT/dt to calculate the temperature rise that can be used to terminate the charge. After reaching a designated charge rate, the fast charge is stopped and is changed to the trickle charging method. This new method is used to avoid overcharge and to prolong battery life. It keeps the charge rate high enough to fully charge but it keeps it low enough not to overcharge the batteries. The trickle charge rate is usually about 10% of the maximum battery capacity. One reason people use this method is to replace the charge that naturally discharges from the batteries. It is useful in keeping batteries always charged at any time when needed, which is not damaging if used correctly.

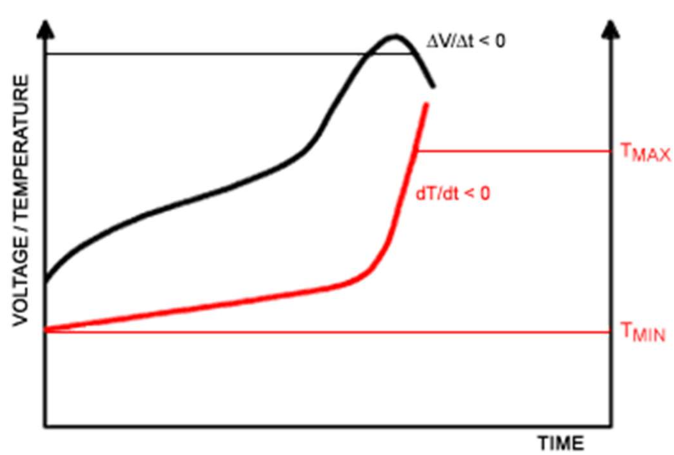


Figure 4.6.3 – 1: Charge Stage for NiMH/NiCd[38]

[38] See Appendix A – 39

4.6.3 Voltage Regulator

Any electronic device that needs to maintain a voltage within specified limits will need a voltage regulator. A voltage regulator is used to maintain a constant output voltage for the system even if changes are made to the input or impedance. Limiting the possible voltage variations help protect the system from any possible damage. We will discuss and decide on which type of voltage regulator will be implemented in our design. The two types of voltage regulators are Linear and Switching.

4.6.3.1 Linear Voltage Regulator

A linear voltage regulator acts like a variable resistor and changes the voltage divider accordingly so that the output voltage is maintained constant. It works by using a voltage controlled current source to apply a constant DC voltage. The difference in voltage between the input and output will be dissipated out of the system as heat. This difference is called dropout voltage, the voltage difference between the input and output. The cheaper types of voltage regulators have a bigger dropout voltage and the more expensive ones have a relatively small dropout voltage. The dropout voltage is important because it manages what input voltage will be necessary to include. For example, if 5v is required for the output voltage and you have a 2v voltage regulator, then you have to input a voltage of at least 7v. So the smaller the dropout voltage, the less energy is wasted and the more efficient it is. Another thing to consider is the amount of time required to "fix" the output voltage after changes are made. This is called transient response, it measures how fast it takes for the regulator to regulate the steady-state conditions after changes to the load are made.

One benefit of using a linear voltage regulator instead of a switching voltage regulator is that it produces a lot less noise for the output. This type of voltage regulator is the best when a fast response is needed between the input and output. They tend to be cheaper and occupy a small amount of space in a PCB compared to a switching voltage regulator. Since a battery will be used to power our design, a low dropout regulator (LDO) is best to use because it will utilize the input voltage fully.

4.6.3.2 Switching Voltage Regulator

A switching voltage works by taking small portions of energy from the input source and transferring it to the output. This is done by using an electrical switch to manage the rate the energy is being transferred from the input to the output. The energy lost during this process is relatively small to even consider. When switched on, the regulator is completely functioning. When switched off, it dissipates relatively no power. This on/off mechanism is the reason this type of regulator has a high efficiency. One thing that sets this apart from a linear voltage regulator is that it has the ability to amplify the input voltage, resulting in a higher output voltage.

The four most used topologies for switching voltage regulators are buck, boost, buck-boost and flyback. A buck regulator is used to convert a DC voltage input to an even lower DC voltage for the output with the same polarity. This is accomplished by using a transistor as a switch that will connect/disconnect the input voltage from the inductor. It is beneficial to use this topology when max load current is relatively low. The next topology is the boost converter, it produces an output voltage, of the same polarity, that is higher than the input. The schematic for the boost voltage converter can be seen below. When the switch is turned on, the voltage goes across the inductor making the current running through the inductor increase. When the switch is off, the diode will become forward bias causing the capacitor to charge up the voltage, resulting in a higher output voltage. The next topology is the buck-boost regulator, it receives a DC voltage and produces an output voltage of opposite polarity. It has the ability to use the buck and boost topologies in that it can produce an output voltage that is either lower or higher than the input. When the switch is turned off, the current lost in the inductor causes the voltage at the diode to become negative. The last topology is the flyback regulator, it has the ability to create multiple output voltages that can have different polarities. By using a low dropout voltage regulator, it regulates the output voltage well enough to have only a small amount of efficiency loss.

4.6.3.3 Comparing Voltage Regulators

Compared to the linear voltage regulator, the switching voltage regulator can have efficiencies in the 90% range if used properly. Since the efficiency is not dependent on the input voltage, it can power from very high voltage sources. This type of regulator is ideal when efficiency is the priority. One drawback is the noise output is much stronger in a switching voltage regulator compared to a linear voltage regulator. The voltage regulator we will use depends a lot on the type of battery we will use to power the system. It has the ability to step-up, step-down and invert the voltage if needed. We would need to use the step-up method if the input voltage is lower than needed. One drawback from using the step-up method is the loss of current. For the most part, this might be a possible choice for our design.

5 Related Standards

In order for the developing world where technology is growing and advancing so fast it is pertinent to have sets of standards for developers to follow such that new technologies are able to be adopted by the current platforms and accepted across a wide variety of groups. There a large organizations of professionals that come together and agree on setting standards for developers to follow such that issues with compatibility and other legal issues do not arise.

5.1 Microcontroller/Microprocessor

The major standards that are relevant to understand when talking about the microcontroller are the different types of communication interfaces. Understanding the way these standard interfaces function is pertinent to the programming the microcontroller properly and making sure that these interfaces are running properly and that the coming in is accurate. A couple of the key factors that differentiate these standards apart from each other is the number of wires which are required for communication, and how the source of the clock that runs the data transfer speed is chosen.

5.1.1 I^2C

The I^2C interface is a protocol that is meant to give the control of multiple "slave" electronic circuits to one or more "master" circuits. That means that there is one or a few master device clock signals that are used to control many peripheral devices; the technology really has the potential to control a nearly unlimited number slave devices in theory, but in the realization of control over an infinite number of devices would require infinite memory sizes and an infinite range of resistances. The technology is also a short range communication protocol, intended for the use within a single device to connect a core to some peripherals. In reality the I^2C protocol can support up to 1008 slave devices using only two wires. Also when using more than one master, only one is available to communicate with the slave devices at a time, and there is no master-to-master communication.[39]

[39] See Appendix A - 41

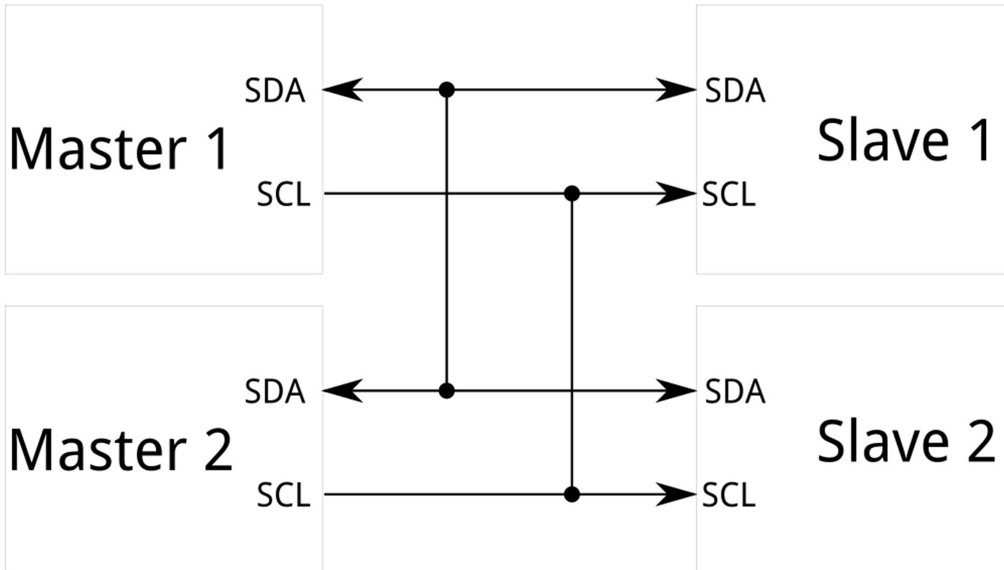


Figure 5.1.1 – 1: Basic Multi-Master Multi-Slave Communication Network[40]

I^2C has the ability to communicate either a rate of 100 kHz, 400 kHz, 1 MHz, 3.4 MHz, and 5 MHz, but these speeds are all dependent on the specific device and not all speeds are really attainable. The I^2C bus consists of an SCL and SDA line that contain the clock signal and the data signal; the clock signal is always generated by the master device that is controlling the bus at the time, and although the slave device is not controlling the clock directly it can "stretch" the clock time by forcing the clock low for a necessary delay. Pull-up resistors are a necessary component to use when connecting a device to the I^2C interface, this is because the I^2C device requires a high signal and if there is no device asserting a low, the resistor will restore the high.[41]

[40] See Appendix A – 42
[41] See Appendix A – 43

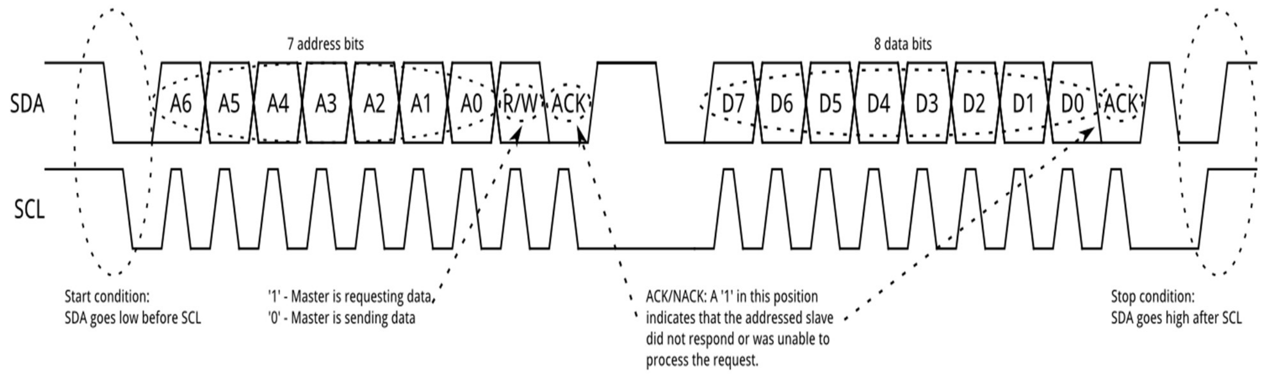


Figure 5.1.1 – 2: I^2C Communication Bit Representation[42]

When messages are sent along the SDA line they are broken up into two segments: the address frame which dictates the address of the slave device the master is communicating with, this is followed by the data frames which can be 8-bit messages of data. The data follows bidirectional communication, so the data can be being transmitted from the slave to the master or vice versa. When communication begins between the devices, the address frame really consists of nine bits of data, the first seven bits are for the address of the slave device, the eighth bit is for determination of a read or write mode of operation, and the ninth bit is the NACK/ACK bit used by the master device to see whether or not the slave device is receiving the data or not. Following the address frame are the data frames, which have no definite duration and is determined during operation by the slave or master device depending on which is communicating data. Once the data frames have stopped being generated a stop condition is reached and the once the condition is reached the data stops.[43]

5.1.2 Universal Synchronous and Asynchronous Receiver and Transceiver (USART)

USART is a block of hardware that allows for serial communication to happen both synchronously and asynchronously. The interface is meant to be used between parallel and serial interfaces that would not normally be able to communicate directly. They utilize a bus interface and a control wire on one side, and only two serial wires at the other.[44]

---

[42] See Appendix A – 44
[43] See Appendix A – 45
[44] See Appendix A – 46

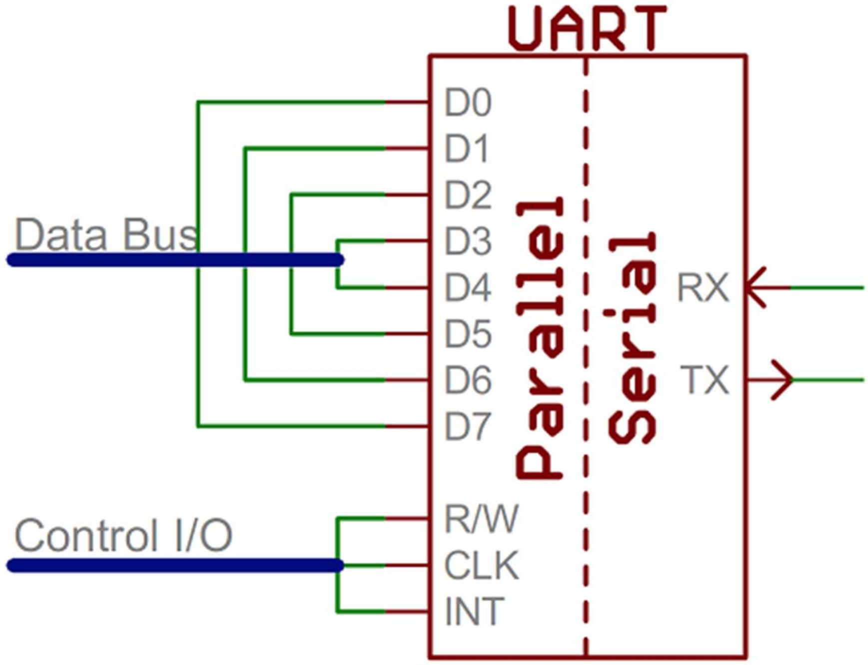


Figure 5.1.2 - 1: Simplified Model of a UART System[45]

The RX line represents the receiver line of the UART system, while the TX represents the transceiver. This generates bidirectional communication like the I^2C interface but the hardware implementation of this system is much more difficult to design and implement, as well as more costly.

5.1.3 Serial Peripheral Interface (SPI)

SPI is a standard that uses synchronous serial communication, this means that the timing of the clocks on both the transmitting and receiving end are controlled and matched such that there is no corruption of data. This also means that the data is transferred at the same rate as the clock and that the only memory elements needed for the receiving end is a shift register, which is both cost and space efficient. The interface works by selecting the clock of either the master or the slave device, and with the selection of a read or write modes which can actually happen simultaneously using SPI. This simultaneous bidirectional communication is referred to as full duplex. SPI usually requires four wires, one for the clock, two for the bidirectional communication, and one for the slave select.[46]

---

[45] See Appendix A – 47

[46] See Appendix A – 48

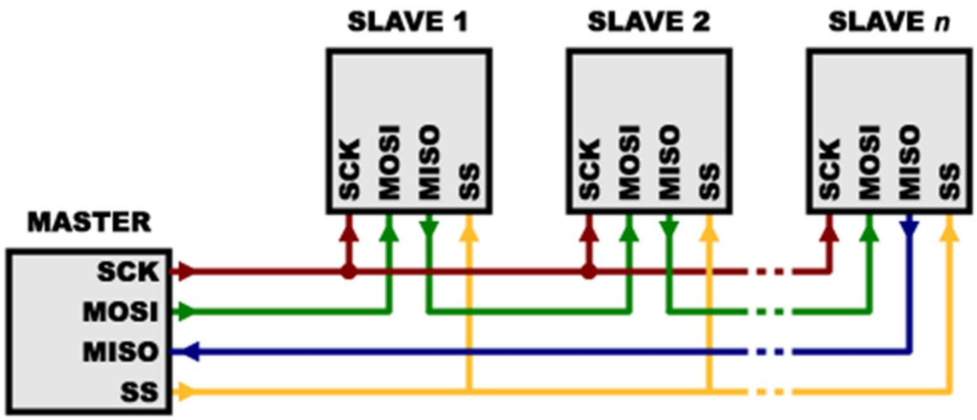


Figure 5.1.3 – 1: Model of a SPI Interface System[47]

5.2 Wireless Communication

Wireless communications have come a long way. Along with new technology so did new standards have been introduced. IEEE along with other sources have decided to come together and organized each newer standard into a new type and named standard. Most of the needed standards to be researched in this project have been part of the IEEE 802.11 standard for wireless communication. Many of these standards had only a few changes but over the years they were classified into separate ones. Below is an introduction and description of the most important ones that have some type of relation to this project.

5.2.1 Wireless a/b/g/n/ac

Wireless specification 802.11a was part of a series established by IEEE (Institute of electrical and Electronics Engineers) standard technologies on 1999. Although, the original 802.11 already existed at the time, it was very slow and did not achieve popularity. Therefore, 802.11a was incompatible with the original since they both have different radio technologies. This specification allows the transmission and reception of data at a speed up to 54 Mbps. Since error correction must be made the actual throughput is about 22 Mbps. Also, 802.11a operates in the 5.15 GHz 5.35 GHz radio spectrum band giving it a better performance advantage over other specifications since the 5 GHz band is relatively unused. Even though this is great advantage, this high carrier frequency brings a huge disadvantage which is the limited range of this band because it only covers one-fourth of the area compared to specifications 802.11b and 802.11g. Theoretically, 802.11a signals tend to be absorbed by walls and solid objects such as brick walls or construction objects making it a disadvantage due to the lack of penetration compared to specification

[47] See Appendix A – 49

802.11b. Another advantage of the 802.11a specification is that these signals are less prone to interference since at a local scenario they have fewer signals to interfere with which creates better throughput. While this specification seems to be very attractive, the cost of achieving such throughput comes at a very high cost due to the expensive hardware needed. Ultimately, specification 802.11a was phased down in order for newer standards to take over.

Specification 802.11b also part of the IEEE standard technologies was also part of one of the first wireless home networks product. This specification is also incompatible with 802.11a due to the use of different radio technologies. Operating at the unregulated 2.4 GHz radio spectrum band, it encounters a lot of interference because of the high use of these frequencies by other wireless home products that use same frequencies such as cell phones, microwave ovens, garage doors openers, Bluetooth, security radios, baby monitors, and others. A good advantage of the frequency used is the range that it can achieve of about 100 feet. A disadvantage is that it can only support a speed up to 11 Mbps which makes it significantly slower compared to newer technologies. But compared to Ethernet (10 Mbps), it is slightly faster; although, it is not faster than the new technology of Ethernet called Fast Ethernet. This standard proved to be the least expensive wireless LAN. Even though specification 802.11b was widely used, it then became outdated due to the new standards 802.11g and 802.11n.

The next specification is 802.11g which emerged in 2002 and 2003 as a part of the WLAN products supporting a new standard. This standard combined both specifications 802.11a and 802.11b. It uses the same transmission speed rate of 802.11a with 54 Mbps which after error correction becomes 22Mbps throughput. It uses the same radio spectrum band of 2.4 GHz just like specification 802.11b in order to obtain a greater range. It is also backwards compatible with 802.11b which means that the 80.11g network adapters and 802.11b access points will work with each other and the other way around. This feature reduces the throughput by ~21% when compared to 802.11a due to legacy issues. In other words, when a 802.11g network allows activity of 802.11b, this will reduce the data rate significantly. Just like 802.11b this specifications also encounters the interference disadvantage due to other similar home devices operating in the same radio spectrum band like wireless keyboards. In general, 802.11g allows good data rates and the signal range is not easily blocked, but it also proves to be costly and regular home appliances tend to cause interference.

One of the newer specification standard 802.11n was ratified and published in October 2009 by industry standard groups.[48] It is also known as 'Wireless N' and was intended to improve initial 802.11 standards. The way of improving these standards is by introducing Multiple-input-Multiple-output antennas known as MIMO technology. This technology uses multiple wireless antennas instead of just a single one. Therefore; this specification uses both radio spectrum bands seen before, the 2.4 GHz and the 5 GHz. Although, support of the lesser used 5 GHz radio spectrum bands is optional. The specification 802.11n operates at speed data rate from 54 Mbps to up to 700 Mbps. before it was even ratified, industry enterprises were already moving towards this new technology on the Wi-Fi Alliance. Also, it has a good range due to its high signal intensity. As a new technology, it is backwards compatible with 802.11b and 802.11g. A new feature of this specification is that it offers 64-QAM modulation which means Quadrature Amplitude Modulation also known as Amplitude shift keying (ASK). This modulation can be analog or digital and it yields up to 64 different combinations. The result of using this modulation its transmission rate is six times the signal rate. In addition, the 802.11n introduces a technology called beamforming but was not standardized so it made it an issue. Incase beamforming modulated the amplitude and phase in order to leave a small non-interfered space with beam. This method can be implemented by routers and mobile devices. Overall, some of the pros of using this technology is the high maximum speed rates and the maximum signal range that it can achieve and also the impedance to other signal interference from outside sources. Some cons include the cost of it since it has not been fully finalized, and the due to MIMO it creates some signal interference with neighboring networks such as 802.11b and 802.11g.[49]

One of the newest Wi-Fi technologies, the 802.11ac specification was introduced in 2013 and it improves on what the standard 802.11n offered. Since 802.11n could only support 4x4 MIMO spatial streams, this new technology offers eight spatial streams and a wider channel of 80 MHz or 160 MHz when they are combined. In other words, this standard offers 8x160 MHz of spectral bandwidth to deal with compared to the 802.11n standard which only offered 4x40 MHz of bandwidth. This is a big difference when transmitting data across airwaves. Another big improvement of the 802.11ac specification is that it uses 256-QAM modulation instead of the 802.11n 64-QAM modulation. The advantage of this modulation is that it puts 256 signals over the same frequency by using Amplitude shift keying which molds each signal into a different phase. Therefore, this amplifies the spectral signal band capabilities by quadrupling it compared to the modulation

48 See appendix A – 50
49 See appendix A – 51

used by the standard 802.11n. The improvement in the specification efficiency is not that important in the 5 GHz band since the channels are rather wide, about 20 MHz per channel, but it is important in other frequencies like the 2.4 GHz or cellular bands where channels are only 5MHz making spectral efficiency extremely important. This standard also introduces the feature of beamforming which means that it specifically directs a signal to a specific device. In return, this allows for a better throughput and power consumption. Beamforming uses a smart antenna that can either do two things; it can physically move to track the device or it can modulate amplitude and phase of the signals making them interfere with each other so that they only leave a small space without interference with beam. In perspective the specification 802.11ac on the 5GHz band using beamforming is just as fast as the 802.11n without beamforming. This is expected since the 5GHz band has a smaller range that the widely used 2.4 GHz band. This is part of the tradeoff needed since the 2.4GHz band cannot support the high speeds used by this standard. The speed of the 802.11ac can be defined in two ways; theoretical and practical. Theoretically, the max speed is about 7 Gbps which is a transfer rate of 900 bits per second. Practically, after channel connection issues the speed comes down to 1.7 Gbps to 2.5 Gbps. In order to achieve some of these impressive speeds, the top routers in the market must be used like D-Link AC3200 Ultra Wi-Fi Router, the Linksys Smart Wi-Fi Router AC 1900, and the Trendnet AC1750 Dual-Band Wireless Router. These routers help achieve some high speeds but they still won't replace those Gigabit Ethernet network speeds. The future of the specification 802.11ac is that it will get faster since there is still a lot of room to achieve this theoretically maximum speed. Also, there is no doubt developers will come out with new link speed of about 2Gbps or more. In using these speeds, developers will have to come out with a new way to implement four or more 802.11ac streams. Although, it is already possible that communications companies such as Qualcomm, Intel, and Broadcom are already working on developing ways to implement four and up to eight 802.11ac streams.

Below is a possible representation of a Wi-Fi network that could be resemble in our project in order to process the whole translation to the computer or user. As depicted below it would require multiple devices to take about the wireless communication. This communication would also be using the specifications discussed above such as 802,11b/g. The reason we will probably will not go in this route is because it would require a router, a Wi-Fi shield and then connect it to the Arduino board. Also, this set up would require a lot more coding and protocol setup that is somewhat complicated. Although, nowadays some of these Arduino boards come with Wi-Fi shields already installed, these shields communicate with the board through UART serial connection. In general, it would not be cost effective to use this route but it is important to have some knowledge of it in case we do decide to adopt this wireless communication. Also, having two gloves will make Wi-Fi implementation of them difficult.

5.3 Software

Here are the following standards for software.

5.3.1 ISO/IEC 9899:2011

ISO/IEC 9899:2011[50], or also known as C11, specifies the form and establishes the interpretation of programs written in the C programming language. It specifies:

- The representation of C programs
- The syntax and constraints of the C language
- The semantic rules for interpreting C programs
- The representation of input data to be processed by C programs
- The representation of output data produced by C programs
- The restrictions and limits imposed by a conforming implementation of C

5.3.2 XML

XML or Extensible Markup Language is a markup language that has a format that is both human-readable and machine-readable. It is defined by W3C XML 1.0 specification.[51]

5.3.3 CSV

CSV or comma separated value is a standard in data storage. It separates each value by a comma and a new set of data is on a new line. Its easily manipulated in spreadsheet programs like Google Sheets or Microsoft Excel. It is defined by the RFC 4180 standard.

5.4 Power System

Here are the related standards for power systems.

5.4.1 IEC 62133

In every electronic device, there are always standards that must be met. Lithium batteries have certain standards that are given on an international level. This

[50] See Appendix A – 52
[51] See Appendix A – 53

standard is used to prevent against possible safety issues for rechargeable batteries. In May 2012, the committee that sets standards decided that all Li-ion batteries must be tested/certified to this standard. An exception to that rule includes batteries that are in IT equipment and audio/video. Battery cells must pass this standard in order for the battery to become CB certified. This standard is also important when exporting these batteries. The first version of this standard was created in 2002. The second version of this standard because of these additional benefits:

- Clarification on the different types of sample pretreatment.
- More help in designing Li-ion batteries.
- More requirements on transportation safety.
- Internal short circuit test.
- Clarification on the tests for lithium and nickel batteries.

In this old standard, the tests were outdated because battery technology was growing rapidly and required more advanced tests. By having updated standards, it promotes better requirements that the manufacturers can use to improve production and widen battery application so it can be used in different types of fields.[52]

5.5 Printed Circuit Board

Here are the following standard for printed circuit boards.

5.3.1 Gerber Format

The Gerber format was developed by Gerber Systems Corp, on August 27, 1980. This is an open ASCII vector format for 2D images. It is the standard for printed circuit board images. The current format is named Extended Gerber or RS-274X.[53]

6 Sensors

These sensors will provide enough data for the MCU to read and differentiate. They all provide different information that is useful for our project. From Acceleration forces to orientation in 3 axes and also detecting which finger is moving or if it is still. All this data is crucial in detecting specific signs. Since most of the sensors use

[52] See Appendix A – 54
[53] See Appendix A – 55

analog pins to connect to the microcontroller, we have to make sure there is enough space allocated to them.

6.1 4.5'' & 3'' Flex Sensors

The flex sensor manufactured by Spectra Symbol is one of the best flex sensors available on the market. These specific sensors have been known to be used in robotics, bio-metrics, gaming, and other applications. Spectra Symbol allows buyers to specify design modifications to the flex sensors to fit the buyer's specific needs, such as length, resistance modification and substrate layer modification (for functionality in different temperatures). The material used to produce these flex sensors is a bendable plastic base in conjunction with conductive ink. This allows for functionality on both straight and flexible surfaces, as well as in more extreme temperatures (-35°C to 80°C). The sensor can also be purchased at a specific lengths ranging from 1 inch to 5 inches, allowing flexibility in terms of sensor and glove size. Since not all finger lengths are the same. We will be using three 4.5''[54] and two 3.0'' flex sensors for one hand.

Flex Sensor	Flat Resistance (Ω)	Max Resistance (Ω)
4.5''	10k	20k
3'	25k	125k

Table 6.1 – 1: Resistance Difference

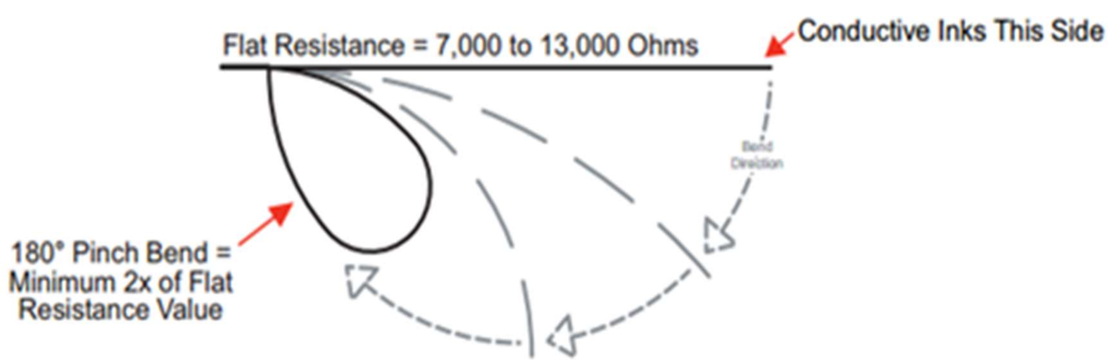


Figure 6.1 – 1: Describing a 4.5'' Flex Sensor[55]

Each sensor has a life cycle of over 1 million. Based on the design schematics of the sensor, there are a few opportunities to adjust the sensor. The buffer can be

[54] See Appendix A – 56
[55] See Appendix A – 57

adjusted by adding a potentiometer to the system, which adjusts the sensitivity range. A variable deflection switch can also be added, which uses the flex sensor as a switch and bypasses having to use a microcontroller for the switch. Finally, a resistance to voltage converter is the last option to add to the system. This converter uses a dual-sided supply op-amp to measure readings when bending at low degrees.

6.2 Contact Sensor

In sign Language, often times, there are signs that are extremely similar. In order to differentiate them, we will need to use a contact sensor. This sensor is used to detect if contact has been made. It usually doesn't require power. It is able to handle more current and are typically easier to diagnose. Contact sensors are similar to force sensors. One of the biggest differences is that contact sensors will not use analog pins to connect to the MCU. Instead, the contact sensors will connect to the digital pins. One reason for this is because we have run out of analog pins to use. We ran out of analog pins to use because of the flex sensors, IMU and battery voltage sensors take up all the analog pins.

Instead of purchasing a contact sensor, we will be making the sensor. The sensor that we will build will consist of copper tape and wires that will be used to create a binary input that will be sent to the MCU. The MCU will then know if contact has been made or if no contact was made. The contact acts very much like a switch.

6.3 Inertial Measurement Unit (IMU)

In order to have the most precise reading, both an accelerometer and gyroscope will be needed to measure the most accurate results. Using both sensors together helps compensate for each other's weakness being either noise or drift. By combining a 3-axis accelerometer and a 3-axis gyroscope, it then becomes capable of processing complex 6-axis algorithms that will give the most accurate result. The IMU has already both accelerometers and gyroscopes in it already, making the process of using both sensors much easier.

One of the more popular and cheaper options of IMU's on the market is the LSM6DS3, manufactured by SparkFun. This IMU has an accelerometer and gyroscope sensor directly built into the board for convenience of data recording. By having both of these components built into one board, the programmability becomes easier and the configuration becomes more flexible to accompany more applications. The IMU has 6 degrees of freedom, allowing it to detect 6 axis in the X, Y, and Z directions as well as the positioning in these directions. This includes detection of tilt, motion, shocks, taps, and counting steps (and has embedded temperature sensor, although irrelevant in this case). Having the ability of collecting and reading accelerometer data at 6700 samples per second (1700 for gyroscope) the LSM6DS3 is a very capable IMU that can keep up with fast and large amounts of data acquisition. The board itself has two sides of pins to utilize, with one side offering power and I^2C functionality and the other side offering SPI functionality and interrupt outputs. The maximum voltage that can be applied to the LSM6DS3 board is 3.3 volts. Applying any more than 3.6 volts can damage the sensor. The board consumes 0.9mA in its standard mode and 1.25 mA at high-efficiency mode. Typical values for analog supply voltage ranges from 1.71 volts to 3.6 volts.[56]

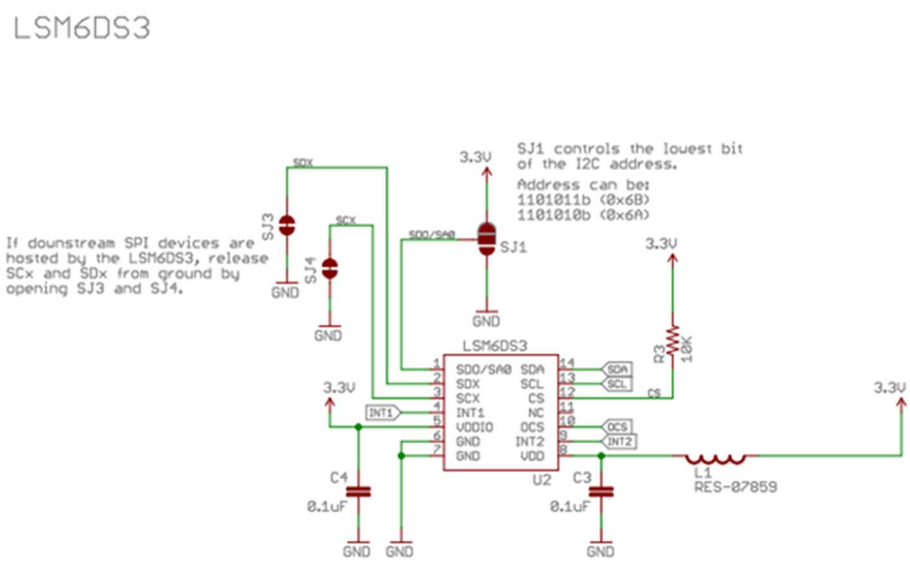


Figure 6.3 – 1: IMU Schematic[57]

Another viable option for an IMU is an upgrade of the LSM6DS3 model, called the 9 Degrees of Freedom Razor, or 9DOF, also manufactured by SparkFun. This model includes an additional magnetometer along with the accelerometer and gyroscope. The magnetometer acts as a virtual compass to further gauge the orientation of your recorded data. With this addition, it is possible to record data on a full 3D scale. The 9DOF model also has an upgraded 9 degrees of freedom from the previous LMS6DS3 model, which only featured 6. This allows it to detect 6 axis in the X, Y, and Z directions and the positioning in these directions, as well as the additional tilt that may accompany any movements in these directions.

[56] See Appendix A – 59

[57] See Appendix A – 60

Although the 9DOF encompasses more detailed measuring than the LSM6DS3, it is the lesser choice out of the two options. Sign language, for the most part, is a 2D hand movement gesture. This renders the magnetometer obsolete, as 3D functionality is not required. Also, the separate flex sensors on the system should pick up any movement that might have been recorded in conjunction with the magnetometer. In addition to its impracticality, it is also almost 4 times more expensive than the LSM6DS3 model.[58]

6.4 Battery Voltage

Currently, the LiPo Fuel Gauge, by SparkFun, is the top choice on the market. Its simplistic design and affordable pricing makes it the most appealing choice for users. The model consists of a very small printed circuit board around 2.5 cm in length and 1cm in width that is connected in circuit with the battery. It works by using an algorithm to detect the state of charge the lithium polymer battery is in and measures the battery voltage. It requires an I^2C interface and is conveniently equipped with an alert pin that triggers when battery levels go below a certain percentage. In our case, the gauge is accurate to within ±12.5 mV for 5 volt batteries. Once connected, the sensor will be able to relay the current battery levels of the ASLA glove onto the computer display.[59]

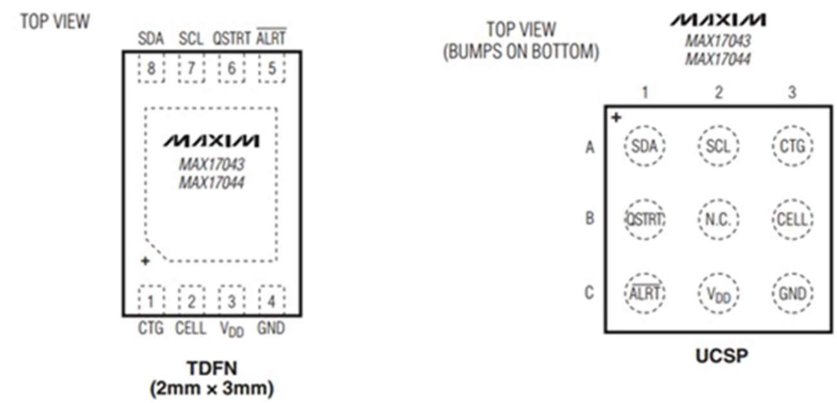


Figure 6.4 – 1: Battery Voltage Sensor Pins[60]

7 Microcontrollers/Microprocessors

After compiling some of the research from all the sections we decided to look at the Atmel low power microcontroller units over the Texas Instruments MSP low

58 See Appendix A – 61
59 See Appendix A – 62
60 See Appendix A – 63

power microcontroller units because it was found that the Atmel chips, in general, used less power and had more online community support. The final two microcontrollers that we are taking a look at will be the ATmega328P and the ATmega2560.

Key Parameters	**ATmega328P**	**Atmega2560**
Flash	32 kB	256 kB
CPU	8-bit AVR	8-bit AVR
Max Frequency	20 MHz	16 MHz
Max I/O Pins	23	86
ADC Channels	8	15
ADC Resolution	10 bit	10 bit
ADC Speed	15 ksps	15 ksps
Analog Comparators	1	1
I/O Supply Class	1.8 to 5.5	1.8 to 5.5
PWM Channels	6	15
SPI	2	5
TWI (I^2C)	1	1
UART	1	4

Figure 7-1: Comparison Table of Key MCU Parameters[61] [62]

As shown in figure 7-1, both 8-bit AVR Atmel microcontroller units that both have 10-bit resolution analog-to-digital converters. For the project we have analog sensors to determine the position and orientation of the fingers and hand, this requires the microcontroller to have the ability to convert the analog inputs into a digital value such that the data acquired can be transmitted wirelessly and analyzed

[61] See Appendix A – 64
[62] See Appendix A – 65

on the computer. So the number of analog input pins and a digital output pin are also necessary to look at when deciding. Both of the microcontrollers have an input/output pin supply class of 1.8 to 5.5 V, which is sufficient for the analog input signals. The ATmega328P only has six analog input pins with a six channel ADC, while the ATmega2560 has sixteen analog input pins with a sixteen channel ADC; here using the ATmega2560 would guarantee that there are enough input pins and a large enough ADC, but the ATmega328P should have enough input pins with the current design expectations. The ATmega2560 has fifty-four digital input-output pins, fifteen of which can be used as pulse width modulation outputs. The ATmega328P fourteen digital input-output pins, 6 of which support pulse width modulation output. Either one of these controllers has the capability of working for the project, so the last two things to consider are the cost and ease of use.[63] [64]

Each of these two microcontrollers has an Arduino development board that may be used for prototyping purposes, both with an operating voltage of five volts. The Arduino UNO, which houses the ATmega328P, is one of the Arduino boards that has been around for a while and is very well known for being used in personal projects, so there is quite a vast amount of open source libraries and community support. The Arduino 2560, which utilizes the ATmega2560 is a little larger so it can be used to do more at once, but has a little less open source material and community support. The price of the Arduino Mega 2560 Rev3 is $45.95 which is a little expensive for a prototype board, while the Arduino UNO Rev3 is only $24.95 which is better.[65]

Since one of our group members already has an Arduino UNO the real net cost since the start of the project would remain zero if the ATmega328P were selected, so we have decided to use it. The ATmega328P does fulfill the necessary functions needed from a microcontroller unit for the project, while maintaining low cost and low power consumption. Apart from the benefit of saving on the expense of the control unit, there will be an ease of the programmability of the device since the device is already known; the coding and debugging of the microcontroller will be easier than choosing the other ATmega controller.

---

[63] See Appendix A – 66

[64] See Appendix A – 67

[65] See Appendix A – 68

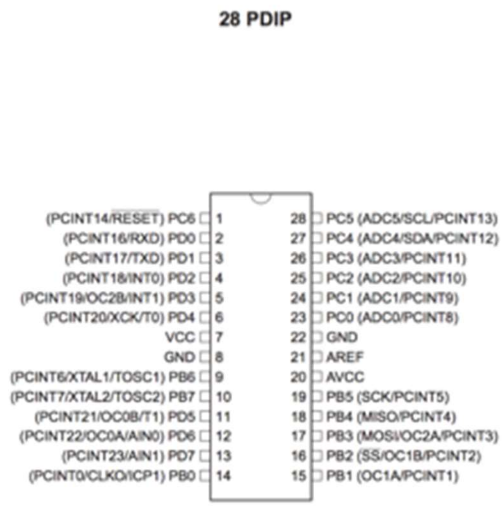


Figure 7-2: Pinout ATmega328P Configuration[66]

Figure 7-2 shows the pin layout of the ATmega328P and the configuration that would be present in both the Arduino board and the microcontroller if purchased separately. The Arduino is going to have its own board layout separate from that which is going to be designed as part of the printed circuit board portion of the project. So when prototyping it important to look at the board layout generated by Arduino and later match the connections made to that of the printed circuit board developed later on. This will ensure that if the device was working properly during the prototyping stage then if the printed circuit board is developed properly it too should work.

7.1 Communication Interfaces

Apart from only taking in the analog inputs through the analog specific input pins for conversion, the ATmega328P also has two serial peripheral interface ports, one two-wire serial interface port, and one Universal Synchronous and Asynchronous serial Receiver and Transmitter port. The datasheet also provides detailed descriptions and shells of code to help initialize the use of the interfaces.

7.1.1 Serial Peripheral Interface

The SPI allows for high-speed synchronous data transfer using a full-duplex three-wire connection. This means that the communication of the SPI is going to happen between the master device in the microcontroller and the slave device(s) in a bi-directional and simultaneous manner. The SPI also has a very fast communication

[66] See Appendix A – 69

speed when it is compared with other types of bus configured communication protocols. Another advantage of using SPI over other integrated communication interfaces is that they usually use less components and dissipate less power. So if applicable to the components selected, SPI should an interface to note for use.[67]

7.1.2 Two-Wire Serial Interface

The TWI is meant for flexible two-way communication. It has a 7-bit address space and can transfer data at speeds up to 400 kHz. The TWI is compatible with I^2C protocol and has circuitry design to suppress/reject noise spikes on the bus lines. The system is designed in such a way that it should be able to support up to 128 separate devices. There are a couple components already selected that will utilize the ATmega328P TWI.[68]

7.1.3 Universal Synchronous and Asynchronous serial Receiver and Transmitter

The USART is meant for flexible serial communication with the controller that can be run in a synchronous or asynchronous manner. There are four modes on the ATmega328P in which the clock may operate: Normal asynchronous, Double Speed asynchronous, Master synchronous, and Slave synchronous. The USART asynchronous modes are meant to transfer data between parallel and serial forms; since transmitting data in serial form only requires one wire, whereas parallel requires multiple, the serial transmission is usually smaller and less costly. In the synchronous modes, the device can either run using the master or the slave clock source depending on which is selected, this allows the synchronous modes to run faster than the asynchronous modes. The USART0 on the ATmega328P can run with serial frames of varying sizes from five bits to nine bits depending on the initialization of the USART. This is all handled through the programming of the microcontroller, which is elaborated on in the datasheet.[69]

8 Wireless Communication

The wireless communication as discussed in the research section before is made up of many different choices. Some of these choices include the option of choosing between Wi-Fi or Bluetooth and also in the possibility of using xBee which does not fall in any of those categories. After doing the required research based on our

---

[67] See Appendix A – 70
[68] See Appendix A – 71
[69] See Appendix A – 72

project design constraints and limitations. We came to the conclusion that only three devices were suitable for our project. Therefore, we narrowed it down to three devices that were closely related but still had some features that differentiated them. Below will be a description of these devices and how they work along with some diagrams describing their key aspects.

8.1 HC-05 Bluetooth Module

This Bluetooth module is a very common and useful way to communicate with an Arduino board with is the most desired and favorite for our project in terms of programming and data processing. This module uses what is called a Serial port protocol or SPP. It also uses a transparent wireless communication connection setup. The serial port protocol mentioned before for this module is also part of the Bluetooth v2.0+EDR or enhanced Data Rate which offers a speed of 3 Mbps. Other important features are that it operates in the module baseband of 2.4 MHz spectral. Features with a lot of beneficial effects like that it offers an Adaptive Frequency Hopping technology along with a CMOS technology that uses CSR Bluecore-04 external Bluetooth chip. The size of this Bluetooth module or what is known as the footprint is of 12.7mm x 27mm, which is incredibly small as it offers some versatile possibilities of developing a design that is suitable for many cycles.

The hardware features that it offers are simple but very useful. These features are an integrated antenna, an edge connector, uses UART interface that can be programmable, also it has what is called pin input and output PIO control. The power that it uses is of about 1.8 Volts and to about 3.6 Volts input or output I/O. This incredibly small power classifies it as power Bluetooth module. This module possesses the classical -80 dB sensitivity.[70]

Some software features of this Bluetooth module include to auto reconnect 30 min after is has been disconnected as a result of being further than the range of connection. To pair, it is allowed to connect automatically as a default and the pairing code is '0000' as a default as well. It also has the feature auto reconnect to the last device that was powered on as default. To show that the device is connected or disconnected, the port PIO1 can be used for this; when is low, it is disconnected, and when it is high it is connected. Also, it is important to mention that the HC-05 has the option to be able to work as Master or Slave. Where another family module called the HC-06 can only work on slave mode. Another part of the software side of this module is that it has two modes, Data mode which allows to

[70] See appendix A – 73

send and receive data from another Bluetooth module, and a Command mode where the user can send AT commands.

When comparing these two modules that are on the same family, they are much alike. Basically they look the same, the physical aspect is the same but their pins are designed for different functions. As mentioned before one of the main differences is that the HC-06 is an only slave type device and the HC-05 does both master and slave functions. For example, the gloves being the master and connected to another Bluetooth module on slave mode which in our case will be the computer. In the case of using the HC-06 Bluetooth module which is only designed to be used on slave mode, so for example, the robot with the glove with the HC-06 Bluetooth module will only be set on slave mode to communicate with a computer that is Bluetooth enabled as a master. For simple cases like a wireless serial bridge, using a HC-06 module is okay, but in our case where we need to send signals both ways, it would not be a good thing to only have a slave only device. The other main difference is that the HC-05 uses different software firmware from the HC-06 firmware used. What this means is that the different firmware uses different pins in both modules even though they are same. Since we find the implementation of the HC-05 simpler and more useful for our project design, we have decided to narrow it down to this single module. Below is a simple representation of the most important components of the HC-05 Bluetooth module.

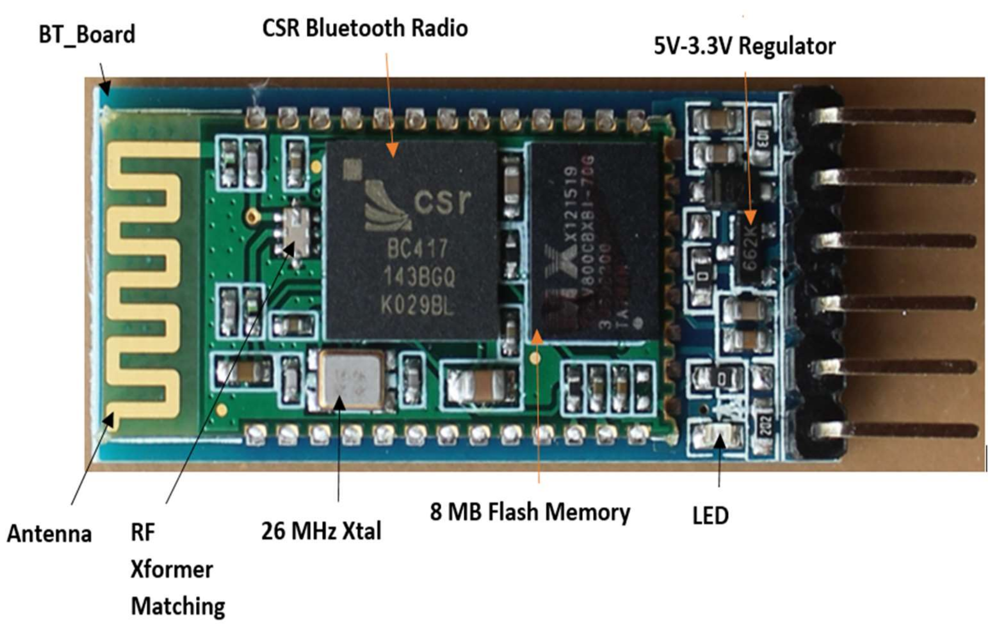


Figure 8.1 – 1: Representation of the HC-05 Module[71]

[71] See appendix A – 74

8.2 BlueSMiRF Gold Bluetooth Module

This is a new Bluetooth module from the SparkFun Electronics. As mentioned before this serves as a way to replace long wires that could get on the way of our project, and in this case in the way of our hand gestures. This new Bluetooth module technology comes in two ways of the same family. One of them is the BlueSMiRF which is the preferred one, and the other being the Bluetooth Mate. Both of products are capable of transmitting and receiving data which basically means that they are both equipped with a transceiver. These Bluetooth modules can reach up to 100 meters of communication. Also, they are designed in a new way to make them simpler to use. This simply means that there will not be any implementation to deal with regarding the protocol stack. The way to replace this protocol stack implementation is to just use transmission of data with a serial interface. This data is then capable of being piped or redirected to whatever Bluetooth module the other device is using.

Below will be an explanation of the different options of this Bluetooth technologies which are Silver, Gold, Mate, and SMiRF. When differentiating between the Gold and the Silver versions, it is important to note that the Gold versions uses an RN-41 Bluetooth Module.[72] The Silver one uses an RN-42 Bluetooth module. What this difference does is a difference between the range and the transmission power. Therefore, the RN-41 is considered a class 1 Bluetooth module. The class 1 Bluetooth module cover up to 100 meters regarding the range, and regarding the power it has a small battery life use it transmits a high power. The class 2 Bluetooth module only has a very limited range and consequently does not use that much power giving it a longer battery life.

The other part that differentiates between this Bluetooth modules is whether it is Mate or SMiRF. The only reason these are different is because they have different pin layout. In the Mate version, it is design to match products like the Arduino Pro which is FTDI Basic and also this version is design for FTDI Cable. The reason why it can easily be plugged into the Arduino Pro is because it has a standardized pin layout for serial interface communication and as well as the power supply pin layout. Another device it can be easily plugged into, the serial header that is, because of the standardized pin layout is the Arduino Pro Minis.

[72] See appendix A – 75

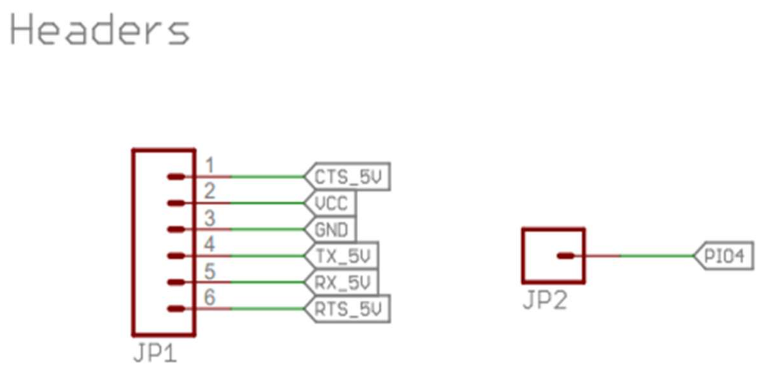


Figure 8.2 – 1: Serial Header for BlueSMiRF

The design for these Bluetooth modules is simple and general in the most part. The design works for both the RN-41 and RN-42 because they both have the same pin layout or that they are basically pin compatible. In general, the circuit schematic for these Bluetooth module is basically the same and since we will be using the BlueSMiRF Gold Bluetooth module it will work for that purpose. The above is a representation of the general pin layout header for the different.

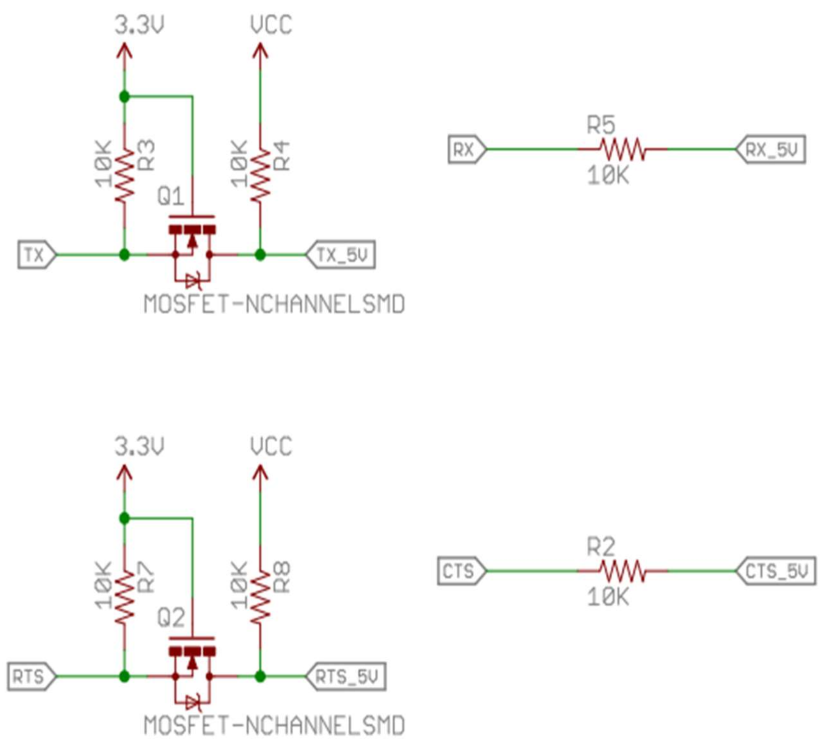


Figure 8.2 – 2: Level shifting Schematic BlueSMiRF Gold

These are basically the most important part of the design. The above schematic represents the key part of the design which is level shifting. This piece of circuit is located between the RN-41, which is the gold module, and the output header.

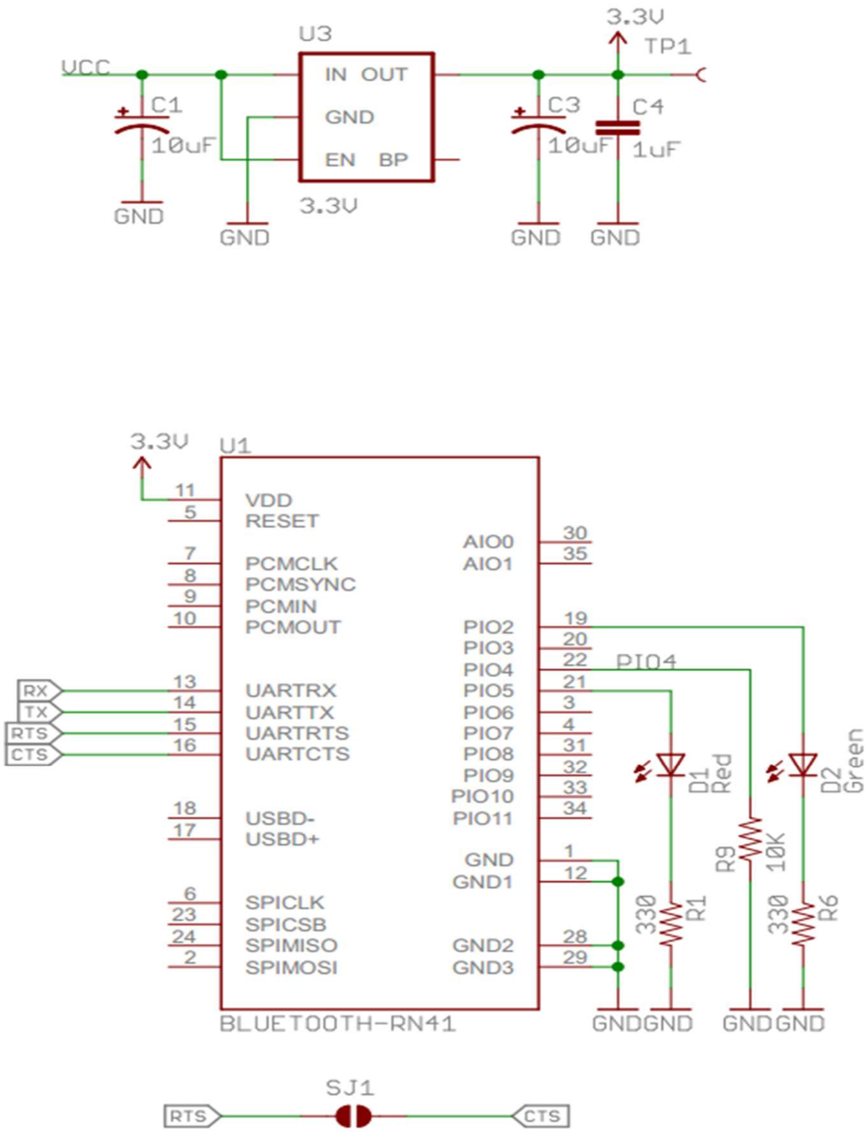


Figure 8.2 – 3: Main Schematic for the BlueSMiRF Gold Bluetooth[73]

This is the full schematic for the Bluetooth module we will be using which is the gold BlueSMiRF Bluetooth module. What this schematic basically does and shows is the operating voltage working at 3.3 Volts of the Roving Networks which are used to power the Arduino which works at an operating voltage of 5 Volts. This in turn allows the communication to be guaranteed between the Bluetooth devices

[73] See appendix A – 76

that are operating in the local area network LAN for the desire project. Another important fact is the linear regulator of this circuit design which allows to work with voltages between 3.3 Volts to 6 Volts to power the Bluetooth module. Also, the schematic describes the two LEDs that this module comes equipped with which serve two different purposes. The green LED is designed for 'Connect' and the red LED is designed for 'Stat'. What this does is to be able to know what status the Bluetooth module is currently on.

Some important features of this Bluetooth module are the antenna which is located on the end of the chip. Of course, we have to be careful with this part of the device to make sure nothing is blocking its path since this is how the data will be collected and sent over. We must have to make sure we do not place this part of the Bluetooth module close to any metal device or even to not have it so hidden that the antenna will be blocked.

Some of the most important features of this BlueSMiRF will be described below, these are the factors which allowed us to make an appropriate final decision between the HC-05 and this Bluetooth module. This seemed like a better fit for our project based on the following features:[74]

- v6.15 Firmware
- FCC Approved Class 1 Bluetooth****Radio Modem
- Extremely small radio - 0.15x0.6x1.9"
- Very robust link both in integrity and transmission distance (100m)
- Low power consumption : 25mA average
- Hardy frequency hopping scheme - operates in harsh RF environments like Wi-Fi, 802.11g, and Zigbee
- Encrypted connection
- Frequency: 2.402~2.480 GHz
- Operating Voltage: 3.3V-6V
- Serial communications: 2400-115200bps
- Operating Temperature: -40 ~ +70C
- Built-in antenna

These features were all useful in the decision making of our Bluetooth module. Some of the most important features that were key were the low power consumption which allows an average of 25mA of current. Also, the allowance to work in difficult environments like the ones xBee and other Wi-Fi specifications. Also we needed to simplify our design to buy the least amount of components, therefore; having a built in antenna was optimal for our design. Since this Bluetooth

[74] See appendix A – 77

module had one along with the other key features, it was best choice for our project. The temperature of the device was also important since it is going to be located so close to the human tissue. This means that the Bluetooth module cannot heat up to the point it is going to hurt the wrist of the person wearing the gloves. Therefore, an optimal temperature of this feature is fine with our design.

8.3 xBee

This is a radio module that allows for wireless communications in a form to point to multipoint communication. This technology is actually part of the IEEE 802.15.4 standard in 2003. This standard is generally based on the fact of end-to-point and also star communications over-the-air at a baud rate of 250 Kbps. Also, xBee is a brand name or part of the form factor families so it is compatible with these. Form factor in computer terms is the size and dimensions, physical arrangement or in other words the power supply and other physical aspects like mounting holes on a motherboard.

At first when it was just being developed, only two technologies were introduced at the same time. The first one being the one with 1 mW of power called xBee, and the second one having a higher power of 60 mW named xBee-PRO. The regular xBee module is a low power and low cost solution so in other terms it would consume less battery and in applications scenarios it is definitely way more efficient. So any device using regular xBee can use a simple battery power method to power the whole system. When talking about the range of this module, in an outside setting it has a range of about a 100 meters with some variation based on the protocol is implementing. In indoor environments the range would go down to almost 30 meters, this would be perfect for our translator glove project since we do not need much range to communicate from the gloves to the computer.

In cases where this range is not optimal for the desired project and where more range is needed. Then we would use an xBee-PRO. An xBee-PRO module would be added to increase range capability and it would require far more power. Also the size of the xBee-PRO is somewhat larger to the normal xBee but they are relative similar. Although, it is important to mention that the output power will vary from product to product. Another important fact is that is has greater sensitivity for receiving purposes. Regarding the range just like the original module, its outdoor range varies but it is close to a mile range. In an indoor environment the xBee-PRO also has a drop in range but it is close to 100 meters. The cost of the improvement in performance come down to that more current consumption is used. So when selecting an output source you have to be careful with the relative proportion of the xBee module.

Both of these modules are relatively close in size and therefore they share the same pinout in a board. This simply means that they can be easily replaced with each other from a board by just popping them off the board. So based on the need of the design, one can effectively replace one another and with no further problems to adjust to the pinout. These two modules have been added and are now part of all the new xBee radios that are now sold and advertised by the Digi brand company.

These xBees modules come in two forms factors, in surface mount form SMT and in through-hole form. Other volume higher applications only use SMT to reduce production cost. Also, there is different antenna options for xBees, these include PCB embedded antennas, wire, U.FL and RPSMA. The xBees can function in either transparent data mode or in the mode of Application Programming Interface. The transparent data mode is basically based on the fact that the Data in ports DIN or pins transmits the data directly over the air to the desired radio receiver without any modification of the information. The information that the packets transmitted can either be transmitted point-to-point or broadcasted in a star type of mode where it is transmitted to multiple points. Therefore, this method is used in stances where the protocol does not allowed to the data to be modified or altered to the original format. For this instances AT commands are used to control the format of the radio's settings. In the second mode, the application programming interface API, the packets or information are put together in one packet form which allows for addressing modification, setting different parameters, and for the packet to send feedback. The kind of feedback it can send could be like the management of digital input and output I/O and as well as for the analog input ports.

The xBee can be used along with the normal Arduino board to communicate wirelessly with the desired component. The way it can communicate with the Arduino board is through the use of Zigbee. Zigbee is known as a simple way to communicate wirelessly with other devices. This new technology proves to be less inexpensive than other wireless technologies such as Bluetooth and Wi-Fi. Useful examples scenarios that use Zigbee are light switches, wireless traffic signals, basically short range scenarios that would require wireless communication like electrical meters with home displays. Also, it uses the concept that only a low rate will be desired, so this is important to take into account when making the final decision. These limitations include the low power consumption that limits the range to only 100 meters from a line sight of view. The way these devices transmit data is through the use of what is called a mesh network that has devices between each other that serve as intermediate ones in a way that it can reach further distances. The specific xBee module is created by MaxStream.[75] What this module does, is that the shield separates the xBees components into a through-hole mount pad. Also, it uses female pins for digital signals which are pins 2 to 7. This shield also uses jumper settings for communication between the boards. These jumpers have removable plastic sleeves and they determine how the communication will be between the microcontroller and the FTDI USB-to-serial chip on the Arduino board or in the case of the ATmega8. This allows for the information or data to be sent to the microcontroller through the use of the xBee wirelessly. The shield can be used with not only one xBee module but with the different types of xBees modules. Below is a normal connection diagram between an xBee and an Arduino board. This allows communication with other xBees modules. In the case of our project, this module would work with other modules like the second glove.

Above is a relatively normal hook up between an xBee and an Arduino. The Digital output pin DOUT of the xBee which is the port number two is hooked up to the Receiver port Rx of the Arduino which is port number zero on the Arduino board itself. Next, the Digital input port of the xBee which is port number three on the xBee has to be hooked up to port number 1 on the Arduino board which is the transmitter TX port and on the board it is port number 1. Also, the grounds are hooked together, this means on the xBee port number 10 will be hooked up with the ground port on the Arduino board located on the power pins.

The procedure for this is originally started by connecting these components to a breakout board designed specifically for the xBee module. Therefore, first you hook up the ports with the Arduino, and then it is preceded by inserting the xBee module on the breakout board. Then the computer has to be connected to the Arduino also in order for data transfer that already has been processed by the Arduino. This data is then put into a database in the computer for purposes of updating an existing library. This library basically classifies every single piece of data collected from the Arduino board. In case of a second glove to fully implement the American Sign Language translation to the computer, then a second xBee module with hooked up with another microprocessor will be used. These two will communicate with the computer in order to collect the full amount of data from the sensors in each glove to then update the computer database again. What is left is how to communicate the xBee module with the computer, this is easily done since it is open source and the code can be pasted to Arduino IDE. Then, there are two ways of communicating with the computer; these are by either using an xBee explorer

USB or by simply just using an xBee explorer dongle. These two ways are equally effective

9 Software

The software will be used to convert the recorded hand gestures from the gloves into text then speech. This section describes the programming languages, environment we have chosen along with the types of algorithm we considered and chose. Along with a quick mention of the graphic user interface and text to speech library we will use.

9.1 Programming Language

Python was chosen for the prototype development for its ease of use and simple syntax. It was also chosen because the Computer Engineering student has programmed in python for a while and is most familiar with it compared to the other languages. Programming in python comes at a price of speed due to it being an interpreted language compared to a compiled language like C++, the extent of which will not be known until the full prototype has been completed.

C++ was chosen for the final software for its portability along with speed over Python. Switching from a scripting language (Python) to a compiled language (C++) gives the user an advantage in the fact that they do not need to install outside programs, such like the python interpreter, and have the need to make sure they install the correct version on their machine. Not only that, the user will be able to use the compiled program (.exe) on any machine they wish, so it will be easy to go from one computer to another, as long as the computer has the proper hardware (the wireless module we will be using). It was also chosen for the microcontroller programming because of how the online community set up their open source examples and code. Being able to use an Object Oriented Programming language with the microcontroller can be beneficial with making objects for each of the sensors to collect data from and to access the data to send over the wireless to the computer program.

9.2 Programming Environment

Needing a proper Integrated Development Environment (IDE) while developing code is needed. An IDE that can debug as well as compile and run the code is a bonus. There are multiple IDEs at our disposal for use, and picking the right one from the start will be beneficial to define a project standard from the beginning.

The Python IDE we have decided to use is called PyCharm, developed by JetBrains. We will be using the free community edition. List of features that PyCharm offers are as follows[76]:

- Inline debugger
- Run/Debug configurations
- Interactive console
- Built-In terminal
- Unit testing support
- Plugin support

The C++ IDE we have decided to use is called Code::Blocks, an open-source IDE developed by the Code::Blocks team. The compiler used will be the MinGW compiler. List of features offered by Code::Blocks are as follows[77]:

- Inline debugger
- Disassembly debugging
- Viewing CPU registers
- Class browser
- Built-In terminal
- Plugin support
- Multiple supported compilers (standard installed is MinGW/GCC)

For programming the microcontroller chosen (Atmega328P) there are two IDEs in consideration. The official Atmel Studio IDE, and the Arduino IDE. The Atmel Studio IDE can program the chosen microcontroller without any external setup on the microcontroller end. Whereas the Arduino IDE requires that the microcontroller is bootloaded with the Arduino bootloader in order to be programmed with the Arduino IDE. We will be using the Arduino IDE during the development phases of the project, and if we need to debug down to the register level or to save memory on the microcontroller for future possible features, we will switch over to the Atmel Studio IDE. List of features offered by Arduino IDE are as follows:[78]

- Built-In serial terminal
- Third-Party hardware support
- Burning the Arduino bootloader to non-bootloaded microcontrollers

[76] See Appendix A – 80

[77] See Appendix A – 81

[78] See Appendix A – 82

- Ability to program multiple different Arduino and non-Arduino branded development boards

List of features offered by the Atmel Studio IDE are as follows:[79]

- Ability to program 300+ Atmel AVR and Atmel SMART ARM-based devices
- Plugin support through their official online apps store (Atmel Gallery)
- Inline debugger
- Interrupt monitoring
- Real-time variable tracking
- Full chip simulation

9.2.1 Machine Learning

As it stands we will need a supervised learning to teach it the correct words and phrases, a classification method seems it would be ideal but while researching other projects with the same topic and direction some have seemed to use a regression model to run the learning and recognition portion of the software. We will be comparing two well-known machine learning algorithms, support vector machines (classification), and neural networks (classification and regression).

9.2.1.1 Support Vector Machines

As said before, this method is a supervised learning model.[80] It is given a training set that is assigned to one category and the training algorithm builds a model that is going to be able to sort new data into the correct category. This makes it a non-probabilistic binary linear classifier. Not only it can be a linear classifier, it can perform nonlinear classification, mapping the inputs into high-dimensional spaces. Given a training data set of n points in the form

$$(\vec{x}_1, y_1),\ \ldots,\ (\vec{x}_n, y_n)$$

y_i is either 1 or -1 depending on what class x_i belongs to. Each x_i is a vector with p dimensions. Support vector machines then divides the group of points of x_i for which y_i is equal to 1 from the group of points that their y_i is equal to -1. If the training data is linearly separable we would use a hard-margin approach, which selects two parallel hyperplanes that separate the two classes of data.

---

[79] See Appendix A – 83

[80] See Appendix A – 84

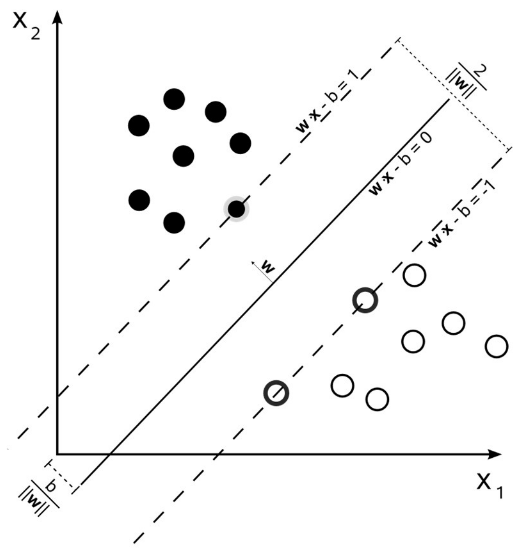


Figure 9.3.2.1 - 1: Hard-Margin example

It maximizes the distance between the closest data points. If the data is not linearly separable we can use a soft-margin, which introduces a hinge loss function:

$$\max\left(0, 1 - y_i(\vec{w} \cdot \vec{x_i} - b)\right).$$

Where w is the normal vector hyperplane. This is to minimize the error of the line fitting and making sure that most of the data is properly separated. For data on the wrong side of the margin, if the distance is small enough, the support vector machine will behave identical to a hard-margin.

Nonlinear classification is possible with a method called the "kernel trick", this method uses a nonlinear kernel function instead of a dot product in the linear classification. While this is better for classification of data that cannot be separated by a linear equation, this increases the generalization error of support vector machines.

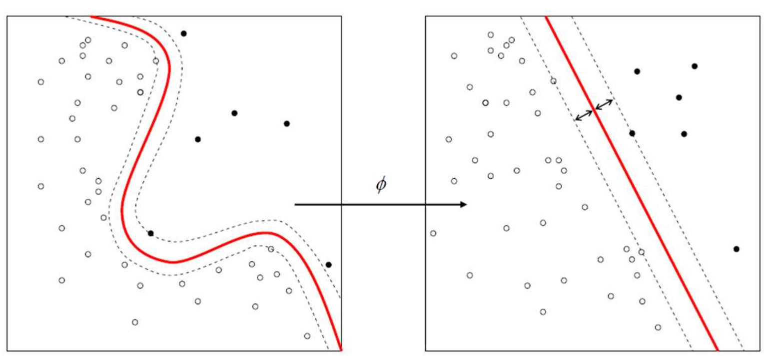


Figure 9.3.2.1 - 2: Kernel trick example

9.2.1.2 Neural Networks

Artificial neural networks are inspired by the nervous system of animals in the brain. They are used to estimate or approximate functions that can have a large number of inputs.[81]

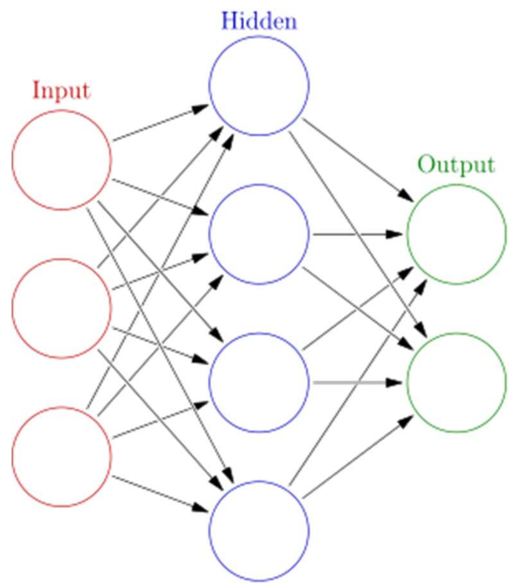


Figure 9.3.2 - 1: An artificial neural network representation

The input goes through a hidden layer of nodes called neurons through a weighted connection. The neuron then transforms and weighs the input to be transported to another neuron until an output neuron is reached. This kind of algorithm is used in computer vision and speech recognition. An artificial neural network is defined by three parameters:

1. The interconnection pattern between the different layers of neurons
2. The learning process for updating the weights of the interconnections
3. The activation function that converts a neuron's weighted input to its output activation

Neural networks have the ability to learn with different methods, supervised, unsupervised, and reinforcement. This is what makes them so liked, the ability to fit the learning method that works for your specific project. In supervised learning the software is given a set of example pairs and it tries to find a function that the input can be mapped. This could be used for pattern recognition (classification)

[81] See Appendix A – 85

and function approximation (regression), the pattern recognition would be best for our needs which is for sequential data (gesture recognition).

9.2.2 Text to Speech

Text-to-speech is one of the fastest things to implement for the software. It will just be importing a well-known text-to-speech library and using the built in functions.

9.2.2.1 SAPI: Microsoft Speech API

This is the official Microsoft Text to Speech API, it allows the use of speech recognition and speech synthesis within Windows. It's intended for use in C++. SAPI 5.4 is the most updated Speech API that started to ship with Windows 7. Major features of the API include:[82]

- Voice command
- Voice dictation
- Voice text
- Direct speech recognition
- Direct Text-To-Speech

Speech generation is as simple as sending a string to a created object interfacing with SAPI.

9.2.2.2 PyTTS: Python Text To Speech

PyTTS is a python wrapper for SAPI, which requires a third party extension and SAPI 5.0+. Since it uses SAPI it has all the features as it but with the added benefit of it being able to be used in python.[83]

9.3 Testing

Extensive testing and reinforcement learning is needed to validate the functionality of the software. The software will contain a feature which allows the user to reinforce the given machine learning model created after learning all of the letters. This will give the user the opportunity to translate the sign they are giving the program and let the program know if it is correct or incorrect. By doing this we give the machine learning model more data points it then can use to classify each

[82] See Appendix A – 86

[83] See Appendix A – 87

sign correctly. We then save the data into a CSV file which then the program will use to re-create the machine learning model upon startup.

10 Power Systems

This section will cover the parts that will be able to power the sign language glove. After going through research about the different systems, these were the components that were selectively chosen.

10.1 Battery

The Li-Po battery (PRT-08483) is a popular choice for an application such as the ASLA. This battery packs 2000 mAh while also being very small (54 x 60 mm) and lightweight. It boasts the highest energy density out of all currently available batteries of this scope at 440 Wh/l and 160 Wh/kg. One of the main advantages the battery offers is its high level of protection, having integrated protection from over voltage and over current by using carbon material as opposed to metals or alloys. If left alone, it has an incredible discharge rate at less than 8% a month. Its temperature functionality is also impressive, being able to withstand temperatures of -25°C to 60°C.

Li-Po batteries usually have an output of 3.7 volts. However, most of the components used in the overall model design uses 5 volts. In order to adjust to account for this change, a switching voltage regulator will need to be used in conjunction with the Li-Po battery to bring the output from 3.7 volts up to 5 volts. When using a switching voltage regulator, the current in the battery will drop. This is why the 2000 mAh level battery was initially chosen over the 1000 mAh battery.[84]

10.2 Charger/Voltage Regulator

Since Li-Po batteries have an output of 3.7 volts and most of the components used in the overall model design uses 5 volts, there has to be an adjustment to account for this change, a switching voltage regulator will need to be needed to bring the output from 3.7 volts up to 5 volts.

The switching voltage regulator that will be used is the SparkFun Power Cell LiPo Charger/Booster. The main reason for choosing this model is that it also has a micro-USB charger built in for charging of the lithium battery in conjunction with the switching voltage regulator itself. It is a small, convenient 2-in-1 package deal

[84] See Appendix A – 88

(although it can be used exclusively for just one or the other). As expected, the model is equipped with two physical inputs/outputs: 1 for the micro-USB charger and the other to connect to the battery itself. It is 1.03x0.95 inches in dimension, making it very convenient having two main components of the overall system embedded into 1 single unit. It is conveniently equipped with an LED light that indicates when the battery is charging and when it is full and also has a power saving mode for lower usage. Another added protection that comes with the unit is a 2.6 volt under voltage lockout that can be adjusted. This prevents the battery voltage from dropping too low and potentially damaging the battery.[85]

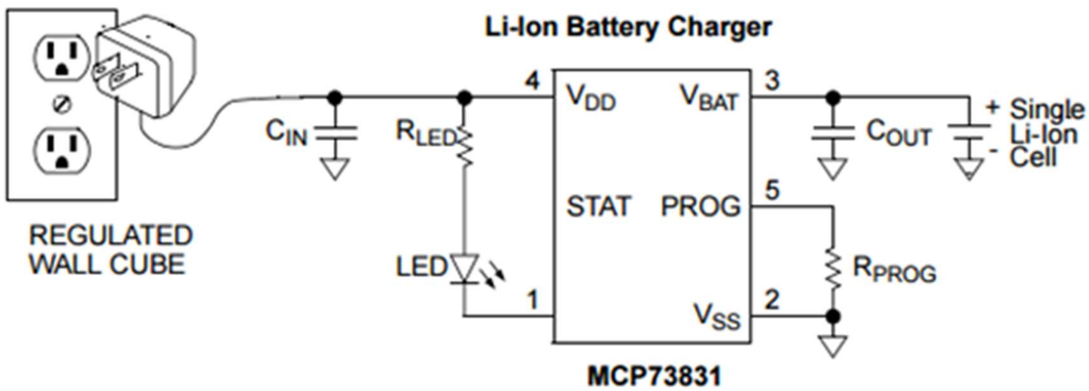


Figure 10.2-1: Li-ion Battery Charger Schematic[86]

The SparkFun Power Cell LiPo Charger/Booster gives users the option of regulating the battery voltage output to 3.3 volts at 200 mAh max or 5 volts at 600 mAh max (5 volts in this case). It is convenient in that it can regulate any types of batteries whether they be lithium or nickel cadmium or any other battery with different specs. The only inconvenient aspect of this model is that soldering must be done

[85] See Appendix A – 90

[86] See Appendix A – 91

on certain pins and connections to use the switching voltage regulator.[87]

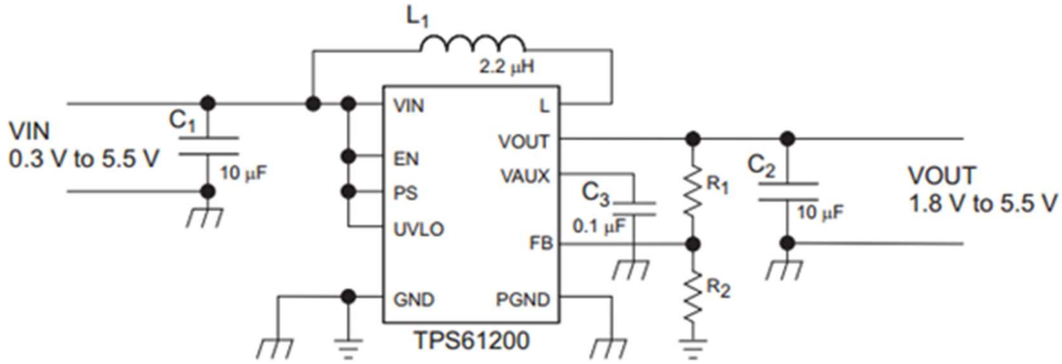


Figure 10.2-2: Booster Diagram[88]

11 Schematic Design and Printed Circuit Board

Making sure that all the individual components researched and chosen for the project design are able to connect properly and fit within a size that is suitable for placement in a glove is a very important aspect of the design process. Choosing a software that will be easy to use yet still be able to handle the large number of components and connections necessary for the design is also another important topic of the section. The schematics will show the all the necessary components and connections in an easy to read sense, while the printed circuit board will compile the schematic design into a more compact physical design with files that can be sent to companies for printing.

11.1 Electronic Design Automation (EDA)

EDAs, also known as electronic computer-aided design (ECAD), are a category of software tools with the purpose of aiding in the realization of printed circuit boards and integrated circuits.[89] The focus for our project will be on software's that focus in the aid for development of schematic designs and printed circuit boards. The process behind the ECAD laying out the printed circuit boards begins by developing a schematic that can later be interpreted by a schematic capture process. Most CAD software's have free versions with the ability to do schematic

[87] See Appendix A – 92

[88] See Appendix A – 93

[89] See Appendix A – 94

capture and translate the schematic to a board layout editor, from which files for the circuit board fabrication are provided.

11.1.1 CadSoft EAGLE

Easily Applicable Graphical Layout Editor (EAGLE) is an EDA where the user can implement their schematic design using EAGLEs schematic editor, then export their schematic to a board layout editor where they can route their printed circuit board to their liking and export Gerber files for printing. EAGLE was designed with the principles of simplicity, flexibility, and cost efficiency. They allow for the user to expand the functionality of the program by importing self-defined commands through User Language Programs (ULPs), as well as import libraries of self-defined and self-generated components and devices.[90] Within the layout editor you can change the width, length, and spacing parameters of the copper traces; you may also lay a copper ground pour, and a have up to sixteen layers for the board.

All the parts used in the project can be found on Adafruit and SparkFun where you can download EAGLE files for all the components, which includes all the device modules for schematic design, and pre-assembled schematics for reference. With the ability to download all the necessary EAGLE files we will be able to generate a schematic to represent all the components and the connections necessary for our project and produce the printed circuit board that will be needed to implement the project.

With the latest free version of EAGLE the increased the number of active schematic sheets that can be active on the same design. The previous free versions only allowed one schematic sheet to be active for one design, now the newest version allows two sheets which allows for easier implementation of designs that require more than a one layer printed circuit board.[91] The tools of the schematic editor make it easy to drag and drop objects in the workspace, as well as drawing up the interconnections of devices. The EDA also has the ability to check that the consistency of the layout by running their Electrical Rule Check tool, which will notify the user if there are any miss connections, discontinuities, or any connections that seem like they should not be made.[92]

The layout editor allows the user to realize the schematic they made into a real board design and rearrange the elements into the area of the board where it should be placed and then anchor them still. Part descriptions can also be written

[90] See Appendix A – 95
[91] See Appendix A – 96
[92] See Appendix A – 96

onto the board and after the fabrication process the label should be visible, this help with the assembly process of the board to know which device goes where. As mentioned previously, a copper pour may be carried out in order to ensure that the board is properly grounded. The size and angles of the traces may also be controlled. To ease the process of verifying that the wires do not cross, EAGLE has developed a few tools that may be used like rats nest and auto route. The rats nest tool helps clean up the wires to minimize any crossings that are not allowed, while the auto router tool helps layout the actual traces of copper in the best possible way by running a spacing algorithm. Even though the auto route tool should be able to make the most efficient designs, it does not always work to make all connections and sometimes the connections have to be cleaned up and rerouted by hand.[93]

11.1.2 Altium Designer

Altium Designer is another EDA software tool package for printed circuit board, and integrated circuit designs. Altium has a lot of the same features as EAGLE, and some even more precise and more efficient to use, but the big drawback to Altium is there is no free version. This EDA is very powerful and can actually handle much more complex designs than EAGLE, but it is really meant for the top of the line printed circuit board designs at an industry level, and it comes with industry pricing. Altium Designer 14 pricing starts at $7,245 as October 2013.[94] Also, because of the pricing and limited access to the EDA there is very few online and open source materials that help in the design process when using Altium Designer.

11.1.3 ExpressPCB

ExpressPCB is another free EDA software tool set that allows for simple schematic design and board layout. Like EAGLE, you can have a library of components that you can drag and drop across the workspace and you have the ability to connect the components like you would in other EDAs designing environments.[95] One of the fallbacks is that is not as widely accepted online as EAGLE and this comes with less extensive libraries of parts and less design assist tools in the EDA itself. ExpressPCB does allow you to link your schematic to your printed circuit board, so you can develop the printed circuit board from the schematic capture of your schematic, but it is a little less user friendly than the EAGLE user interface.

[93] See Appendix A – 98
[94] See Appendix A – 99
[95] See Appendix A – 100

11.2 Schematic Design

Schematic designs showcase in a symbolic and simplified manner of how the individual components at a low level will connect in an open environment including the scale of the elements and their relative spacing. The components chosen for the project can all be found on two open-source sites, Adafruit and SparkFun that provide access to sets of files including EAGLE libraries of parts, schematics, and board layouts.

11.2.1 ATmega328P - PDIP Schematic

This schematic layout seen in Figure 11.2.1-1 is of the ATmega328P with a PDIP packing type. It shows the 23 input/output pins plus the other pins for input voltages, reference voltages, and grounds. This specific schematic shows a part of the Arduino UNO and some of the connections seen on the Arduino board. There are other components on the Arduino board that are not shown in this schematic, but they are not necessary at this point for making the proper connections to the other devices in the project.

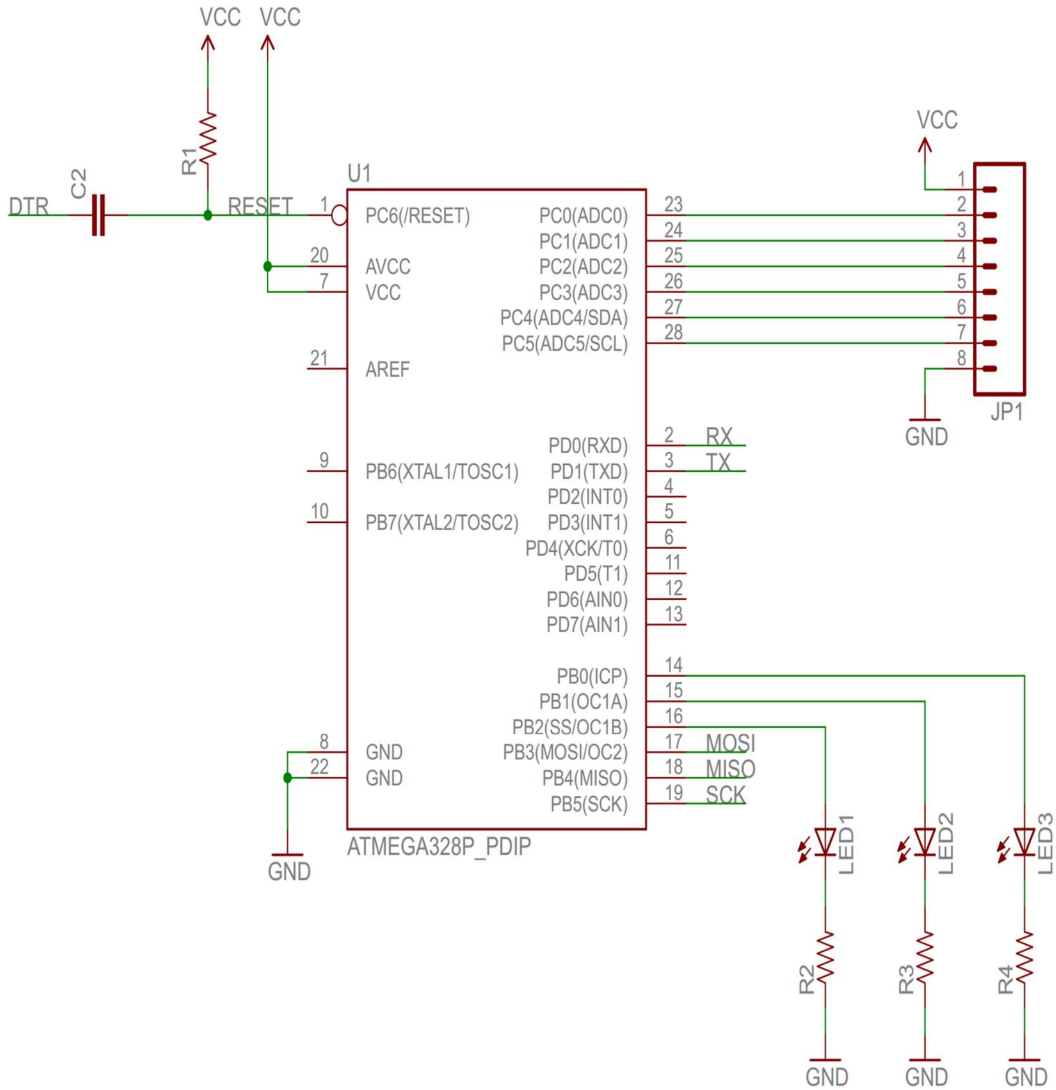


Figure 11.2.1 – 1: ATmega328P with PDIP Packaging[96]

11.2.2 Inertial Measurement Unit Schematic

The inertial measurement unit (IMU) is broken up into two different schematics; one for the main body where the gyroscope/accelerometer housing and connections are, and another for a pull-up resistor configuration that is used for interfacing the IMU with the microcontroller when using the I^2C protocol. Figure 11.2.2-1 shows the IMU body and the pins for the voltage supplies, grounds, and the input/output functions. The SDA/SDI labeled pin is used for the bi-directional transmission of the sensor data for the I^2C interface. The SCL labeled pin is the serial clock for the device that is used for both the I^2C and SPI interface set-ups. SDO/SA0 labeled pin is used to determine the least significant

[96] See Appendix A – 101

for the address. The CS pin is disconnected for I^{2}C configuration. The Int1 and Int2 pins are used as programmable interrupts for the accelerometer and gyroscope pair. The other pins are for the SPI configuration and auxiliary clocks and data.[97]

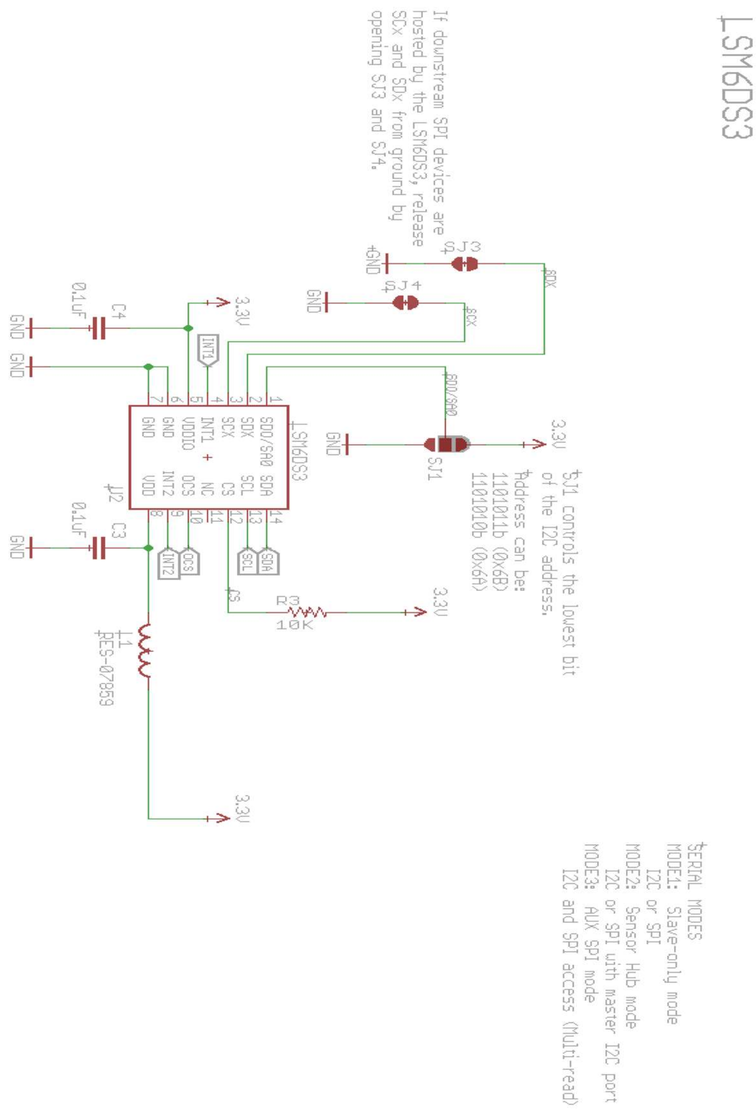


Figure 11.2.2 – 1: IMU Body and Connections

The schematic in Figure 11.2.2-2 depicts a pull-up resistor that is enabled in order for the I^{2}C operation of the IMU to function properly. The serial data line is a bi-directional interface for sending and receiving data and the pull-up resistor is used to connect the two lines to the Vdd_IO, and allow for clear communication; when the data bus is free, both lines are high.

---

97 See Appendix A – 102

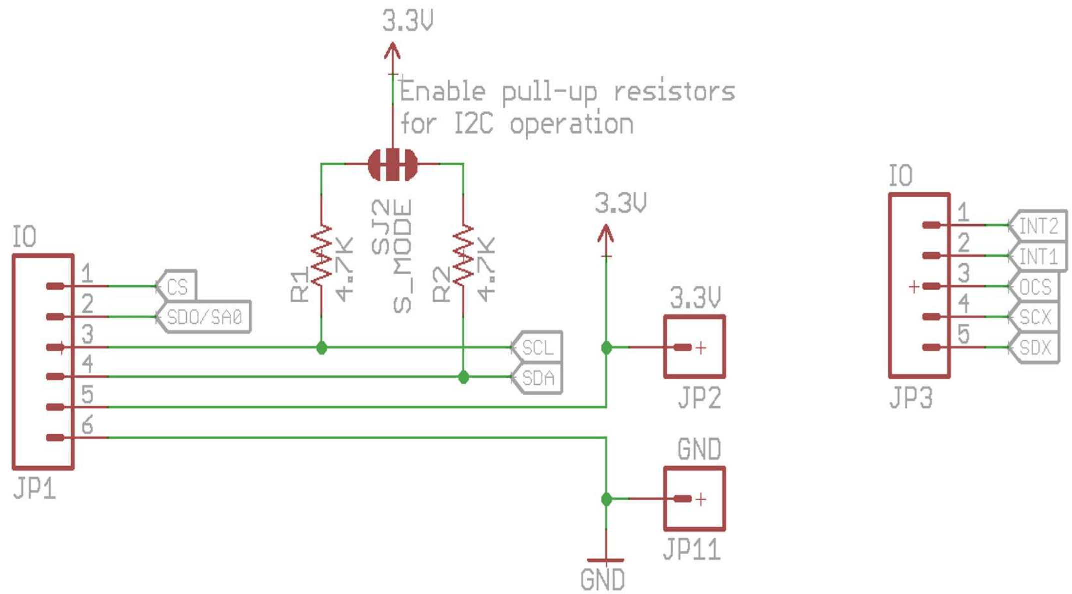


Figure 11.2.2 – 2: IMU Pull-Up Resistor for I^2C Interface

11.2.3 Battery Charger Schematic

The battery changing element is represented by the schematic in Figure 11.2.3-1 where the USB interface and other connections are visible. The way the project plans on charging the battery is using the USB interface, although the JP9 in represent an alternate way to power the charging station using another battery, but using a battery of too high a voltage can cause damage to the internal components. The battery being charged will be connected to the VBAT (JP1/12) port.[98]

[98] See Appendix A – 103

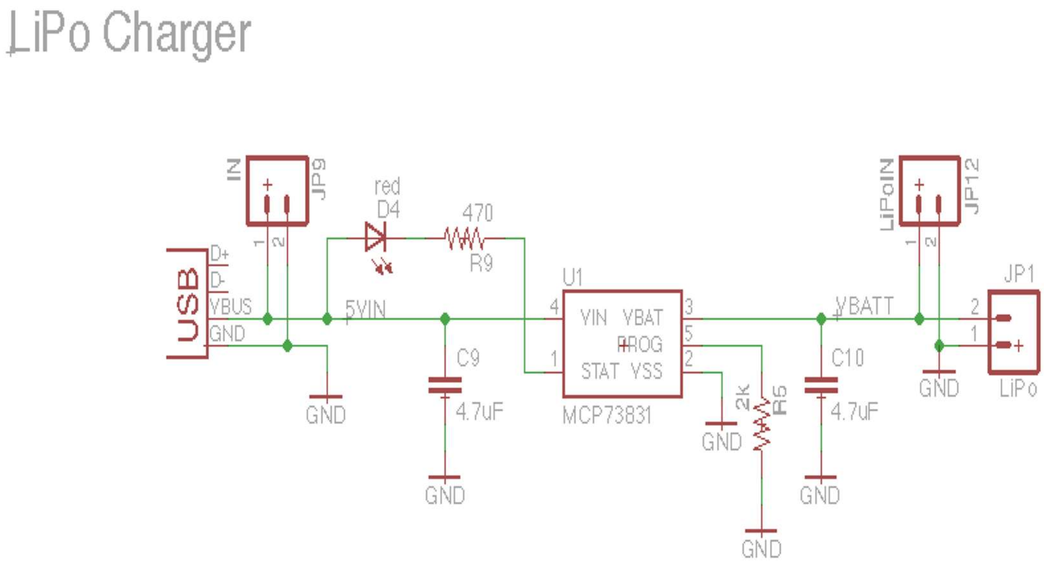


Figure 11.2.3 – 1: LiPo Charger with USB Interface and Connections

The switching regulator portion of the battery charger is used to boost the incoming voltage for the microcontroller. The JP2 out port represented in the schematic is present on the actual board as a solder pad where headers or pins are needed to be soldered in order for the microcontroller to be properly supplied.[99]

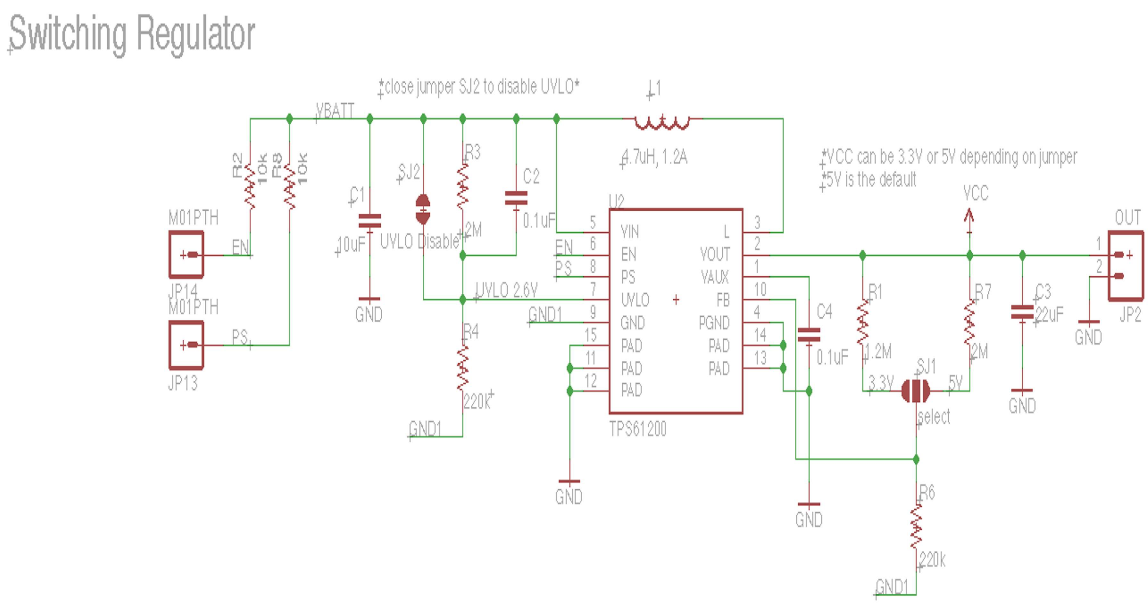


Figure 11.2.3 – 2: Switching Regulator for Voltage Step-Up

---

99 See Appendix A – 104

11.2.4 Voltage/Fuel Gauge Schematic

PIN		NAME	FUNCTION
UCSP	TDFN		
A1	8	SDA	Serial Data Input/Output. Open-drain 2-wire data line. Connect this pin to the DATA signal of the 2-wire interface. This pin has a 0.2μA typical pulldown to sense disconnection.
A2	7	SCL	Serial Clock Input. Input only 2-wire clock line. Connect this pin to the CLOCK signal of the 2-wire interface. This pin has a 0.2μA typical pulldown to sense disconnection.
A3	1	CTG	Connect to Ground. Connect to VSS during normal operation.
B1	6	QSTRT	Quick-Start Input. Allows reset of the device through hardware. Connect to GND if not used.
B2		N.C.	No connect. Do not connect.
B3	2	CELL	Battery Voltage Input. The voltage of the cell pack is measured through this pin.
C1	5	$\overline{\text{ALRT}}$	Alert Output. Active-low interrupt signaling low state of charge. Connect to interrupt input of the system microprocessor.
C2	3	V_{DD}	Power-Supply Input. 2.5V to 4.5V input range. Connect to system power through a decoupling network. Connect a 10nF typical decoupling capacitor close to pin.
C3	4	GND	Ground. Connect to the negative power rail of the system.
—	—	EP	Exposed Pad (TDFN only). Connect to ground.

Table 11.2.4 – 1: Pinout for MAX1704X[100]

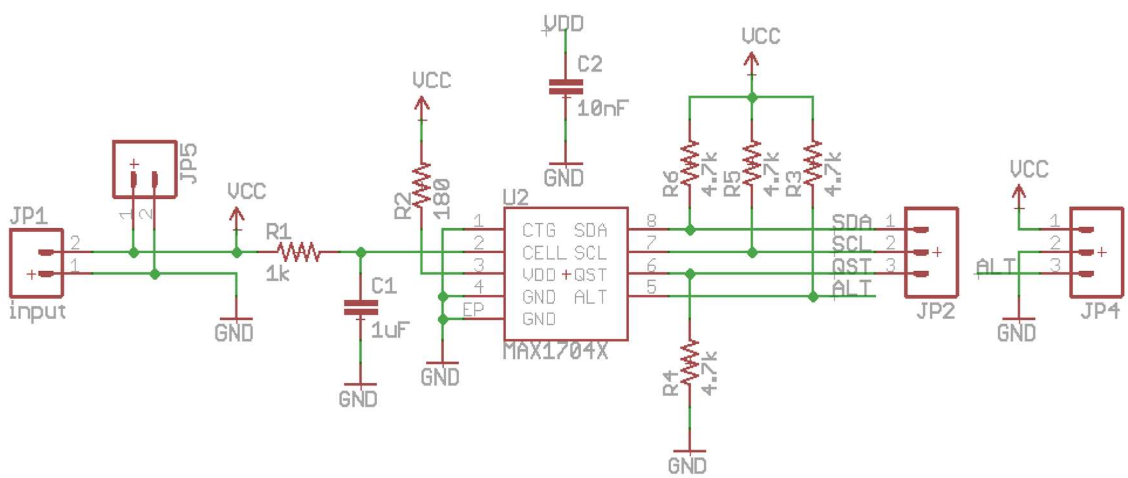


Figure 11.2.4 – 1: Voltage/Fuel Gauge and Ports

11.3 Printed Circuit Board Layout

The printed circuit board of the project is one that is very important for the design to succeed, but is quite far off in the distance. It is one of the final step of the project once all of the prototyping is done and all the individual components have been acquired and verified that they are working. The printed circuit board creates all the physical connections using a conductive tracks, pads, and sheets to make

100 See Appendix A – 105

sure all the electrical components of the design are able to communicate properly. These conductive materials are separated by a non-conductive substrate so that the electrical components only make connections with the other devices they are supposed to be but no others.

Printed circuit boards may also have multiple layers, each separated by a layer of non-conductive material where each surface may have their own set of connections that could not be made on the previous. When using more than one layer, if a connection is made for a device on one side of the board to a device on the other side of the board the connection is made by a metal connection that goes through the board called a via. When using more than two layers you can have connections called partial vias that go only go to internal layers but not all the way from one outer surface to the other surface. Our project will not have to deal with partial vias since we plan on only having to use a printed circuit board that requires one or two layers. Since the board design had not yet been drafted the exact number of layers is not yet known, but with the number of components being used it can be assumed a bi-layer board will work.

The way the printed circuit board is generated starts with creating a schematic that makes all the proper connections, and through schematic capture software a preliminary board may be laid. When the board is initially generated the dimension of length and width may be established. In EAGLE you can also control the width and depth of the signal traces as well, this is useful for establishing the proper dimensions of traces for power routes that may require larger traces and smaller traces for transmission of data signals. Since this is the case when our group is able to finish the printed circuit board design we have to make sure to account for the trace spacing and size when laying them. In EAGLE there is a control window where we can set the exact spacing and sizing parameters we want and there is a tool we can run that checks to see if the traces lid out actually match the parameters set forth. Another physical parameter that, usually in larger boards, may become an issue is heat dissipation, in which case the use of heat sinks may need to be considered. EAGLE also allows the user to do a copper pour to create a ground plane, which is an important feature of circuit boards that is sometimes overlooked when being designed.

The last step to printed circuit board design is the generation of the Gerber files which are generated once the design is completed. The EDA software has the ability to compile the design in these files, and it is from these files that the printed circuit board manufacturer will be able to realize the design in a physical manner.

For our project we plan on having the electronics to be portable and fit on a glove, so a reasonable design size that should work for both the electronics and the size of glove should be around a 2.25” by 5” board size. These size restraints would make the board about the width of the average wrist but keep it to a relatively short length on the forearm. It should also be enough room to make sure that all the components can be connected together properly.

11.4 Quotes and Edits

At this point it is important to pick a manufacturer that will not overcharge for the process of printing the circuit board, as well as one that will complete the process without error and in a reasonable time. There are two companies that seem to be reasonably priced and come with good reviews, OSH Park and 4PCB.

For pricing, OSH Park does it by price per square inch. For the bi-layer boards which is the focus for our project, OSH Park prices their boards at $5 per square inch, a pricing that includes the production of three copies of the board. If a four layer board becomes necessary the price is double. 4PCB has a student deal that has a baseline price for their two layer boards, $33 each no matter the size. Like OSH Park, if a four layer board is needed the price is doubled. So as far as the pricing of the boards is concerned OSH Park seems to have the better deal since their board pricing includes three copies of the original board, unless the board becomes too large then it may become reasonable to buy 4PCB but just buy one or two boards. [101] [102]

When it comes to timely shipping it seems like 4PCB wins by a little. They have a guaranteed shipping time for their standard printed circuit boards to start three days from the time of the order. This means from the time the order is placed and the manufacturing process is finished and the product is in the mail it should only take three days, but then the mail speed becomes the limiting factor. OSH Park brings free shipping and say the boards ship in under twelve calendar days from the time of ordering, but have no guarantee. So if it comes down to a time crunch 4PCB may be a better option.[103]

After reviewing the two companies and comparing their prices and their shipping times it seems like for our project, with a rather small planned board size and the hope that the board is bought ahead of time, the company we plan on using for the manufacturing of the board is OSH Park. Now since the board has still not been

[101] See Appendix A – 106
[102] See Appendix A – 107
[103] See Appendix A – 108

fully developed and the exact size of the board has not been established, this choice is still not the absolute decision.

11.5 Testing

The board development usually takes place after the prototyping of the project has taken place, and the operation and connectivity of all the components is known. The prototyping should allow us to see what the right connections to be made are and how these connections should behave when active. Development of the project on a breadboard should behave the same as when developed on a printed circuit board, but this is not always the case and is why printed circuit board testing is so important. There are many factors that play into the performance of the printed circuit board and the implementation of the design on it. One of the more common issues with the performance of on the printed circuit board is the assumption that everything is ideal, where in reality the traces have a limited capacity for current, they also have a resistance that is not ideal, and if there are vias in the board there is the possibility for a capacitance and inductance generated that have a great effect on devices operating at high frequencies.[104]

Another major issue that is common when assembling the board is with the connectivity at soldered joints. When the solder is set on the board to create a fluid metallic connection/contact with connecting electrical elements, the solder can sometimes become non-functional and cause a discontinuity in an electrical signal. There are many things that can cause this issue and must be carefully watched and tested when in the assembly stage. A cold joint is one of the most common issues with hand soldering and is usually the result of overheating the solder directly with the soldering iron. The soldering iron is meant to heat the element the solder is going to be placed on, yet sometimes the solder itself is heated by the iron and this can ruin some of the solder's metallic properties causing a connectivity issue. This issue can either render the part to not conduct at all, or just make it unstable and have the part work intermittently. Another really common issue that arises with hand soldering is a dry joint, which occurs from the movement of a joint as the solder cools and hardens. This can be hard to see since visually the joint may not move much, but the connection to the metallic pad or connector of the board may have been lost creating discontinuity. Both of these common issues must be monitored as the assembly process advances because issues like this are very fixable but can be hard to find sometimes. One way to make sure that this issue is minimized is to use measurement tools, like a digital multimeter, to do intermittent

[104] See Appendix A – 109

continuity checks and make sure that the joints and connections being soldered are made properly.[105]

As components are being placed and soldered to the board it is important to check their individual continuities to the board and to the other components they are supposed to making connections with. Once a system that can operate on its own and tested is placed on the board it should also be tested to make sure the whole board is not assembled just to find a system not operating properly. For example, once the microcontroller is in place, along with the power, it should be tested to make sure that it can operate with the board design. Then other components may be added and tested as well. This should continue until the final component is assembled and tested. Once the assembly of the board is complete it should be fully operational and be able to connect to all the sensors and operate the way the glove is intended.

12 Design and Prototype

In development of our prototype, we know the microcontroller takes in sensor data through in and out pins and converts analog signals to digital information. Once digital data is received by the MCU, it will then be transmitted wirelessly from the Bluetooth device to a computer where data processing will take place. After discussing all the components that are going to be used in our sign language translator glove. We will explain how all the components are going to be tested throughout this section and also how everything is going to integrate to work together to form a prototype.

12.1 Testing Environment

One thing to consider when testing the glove is the testing environment in which it will take place. There are certain weather conditions that could affect the functionality of the device. For example, most electronic parts should not be exposed to extreme temperatures. The sensors, especially, could be affected if used in a humid environment. It could in fact decrease performance if put under these circumstances. Dust is also a problem that should be considered. If dust gets into an IC, it could damage the device and stop it from working. Our glove is not designed to be water resistant or waterproof in any way. Water/rain is definitely something that could damage all the electronics embedded onto the glove. It is best use and test the sign language glove when under ideal conditions, such as inside a room with air conditioning.

---

[105] See Appendix A – 110

12.1.1 Temperature

Temperature is important to consider when looking at all the parts that will be used in our design for the glove. Many electronic components function in different temperatures than others. So it best to compare the temperature ranges of all the different components in order operate the glove in temperatures that are ideal. Since the efficiency of each component is dependent on the temperature that the glove will be operated in. Below you can see a table stating all the temperature ranges of each component. The temperature range is used and stated to make sure the components perform without any complications or errors, as long as the temperatures stay on the range specified.

Component	Temperature (C)
Flex Sensor	-35 to 80
IMU	-40 to 85
MCU	-40 to 85
Bluetooth	-40 to 70
Batteries	-25 to 60
Charger/Voltage Regulator	-40 to 85
Fuel Gage	-20 to 70

Table 12.1.1-1: Temperature Ranges for the components

12.2 Hardware Prototype

The next section will be a description of every single hardware piece that will be used in this project. These multiple parts of hardware will work together in order to facilitate the translation of the hand gestures using a glove. All of these hardware parts will be attached to both gloves and will aid in the translation process. The hardware parts for this project include flex sensor, inertial measurement unit or IMU, the microprocessor MCU, and the Bluetooth module.

12.2.1 IMU

There were not enough analog pins to connect all the sensors together all at once. So we have to use another type of interface in order to connect everything properly, such as the I²C interface. This is easiest and best method to connect the board to. We can test the IMU making sure the device is able to successfully connect to the Bluetooth module. All the data recorded by the IMU should be able to transmit from the IMU to the MCU, which then transmits the signal from the Bluetooth then to the computer. All the data should be able to be seen through the computer.

The accelerometer in the IMU can read data up to 6.7 kilosamples/sec, which is much greater than the gyroscope. The gyroscope can read data at 1.7 kilosamples/sec. It can operate at about 1.25mA at 1.7ksps. The IMU has 11 pins that can be utilized in our design. On the left side of the breakout board are the power and I^2C pins. This is where all the communication for the interface will take place. Below is a diagram of the breakout IMU board. The first pin in the top left hand side, GND, is ground and can be used to connect to more parts. The power supply pins are labeled as 3.3V, this is where the microcontroller will supply voltage to the breakout board. Speaking of power supply, since there aren't enough analog pins for all the sensors, the I^2C interface serial data will be connected through the SDA/SDI pins. The address for the I^2C interface will be connected through the SDO/SA0 pin. The serial clock pin is labeled as SCL and will connect to the MCU.

On the right side of the breakout IMU board are the pins that will deal with the accelerometer and gyroscope connections and interrupt outputs. The pins labeled INT1 and INT2 are the accelerometer and gyroscope interrupt pins. They can be used interchangeably, so it doesn't matter which pin is used for the two sensors. These two pins will connect the IMU to the analog pins of the microcontroller which is how the IMU will be able to communicate with the MCU.[106]

[106] See Appendix A – 111

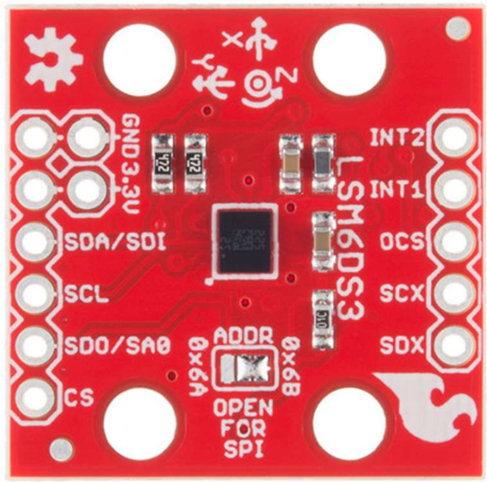


Figure 12.2.1-1: IMU Board

Originally in the design, we specified that we needed two IMU's, one in the PCB and one in the index finger. The second IMU was needed for word recognition. Since time was a problem, we decided to just use one IMU for letter recognition. A header for an extra IMU were added to the Printed Circuit Board for the possibility of word recognition.

Figure 12.2.1-2: LSM6DS3 Chip

12.2.5 Glove

The type of glove we decided to use for this project will be running gloves. They can be found at a relatively low cost. The size of the glove will be picked out according to one of our team members hand size. The gloves are made up of Nike's Dri-FIT fabric material and also includes breathable mesh, which can be seen in the figure below. Running gloves are ideal because it provides a lightweight and comfortable alternative to most other gloves. The glove will for sure be put under a lot of heat because of all the electronic devices that will be mounted on top of it. One benefit in running gloves is that it allows for heat and moisture to leave the glove so the user doesn't feel too much heat inside the glove. The main reason this glove was chosen is because it provides moisture and temperature control.

Figure 12.2.6-1: Running Glove Back and Front[107]

12.2.6 Flex Sensors

One way to test the flex sensors to see if it is running the way it is supposed to be. You have to make sure the resistance is correct, according to what it was specified in the datasheet for the flex sensor. In order to measure the resistance, we need to connect the flex sensor to a multimeter. The flex sensors has 2 pins in the bottom which can be connected to a multimeter. Once it is connected, we can bend the flex sensors to measure the resistance. The 4.5'' flex sensor should have a resistance of about 10k ohms when flat but when max bent, it should have a resistance of about 20k ohms. For the 3'' the resistance should read about 25k ohms when at rest and it should read about 125k ohms at max bend. Resistances may have up to 30% error according to the datasheet for the flex sensors.

[107] See Appendix A – 112

The prototype for this project will use the hand measurements of one of our team members, with three 4.5″ sensors and two 3″ sensors. The three 4.5′′ flex sensors will be used for the ring finger, middle finger and for the index finger. The two 3′′ flex sensors will be placed on the thumb finger and on the pinky. In the figure below, we can see how all the components will be laid out in each hand.

There will be a total of 10 flex sensors that will be used in our glove since there will be one flex sensor per finger. All flex sensors are going to be in the back of the hand, as well as most of the other components. We decided to put one IMU in each hand. The location of the IMU is on the back of the index finger. Since there are too many sign language gestures that look alike except for small differences, we will need contact sensors in order to differentiate them correctly. There will be two sets of contact sensors in each hand. One set that consist in between the middle finger and the index finger. Another set of contact sensors will be located in the front of the middle finger and in the back on the index finger. All of these electronic devices will be embedded into the back of the running glove.

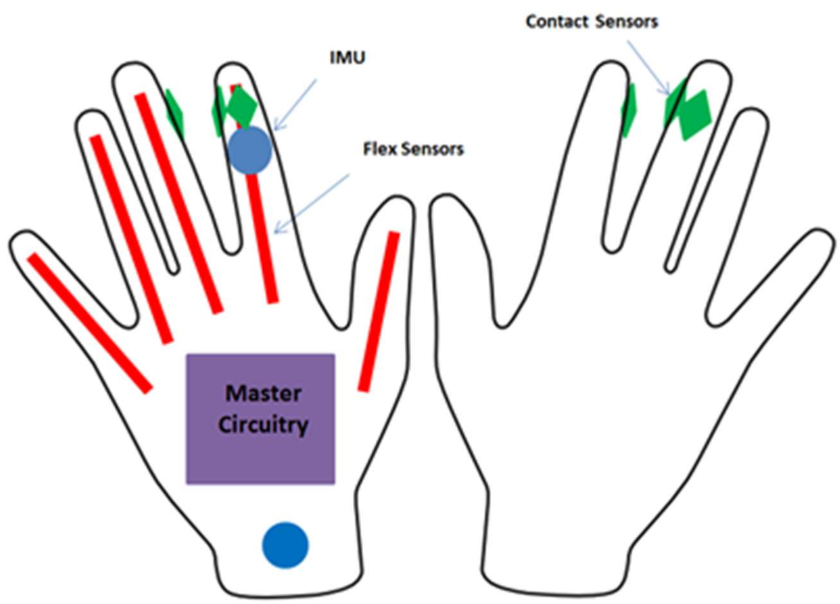


Figure 12.2.6-2: Glove Components Back and Front

12.2.7 Contact Sensors

Multiple contact sensors will be constructed by using copper tape and wires. It is much easier to build contact sensors than actually purchasing them through a retailer. The sensor part will be made up of 0.5′′ square copper tape. The location of the contact sensors can be seen in the figure above. The copper squares will be connected to wire that will be connect to the MCU through a digital pin. This is how the contact sensor it will communicate to the system. It will act as a switch. Another wire connected to the contact sensor will act as a ground.

The contact sensor used in the glove is connected to a digital pin. The glove will use 2 contact sensors which have 3 terminals. One terminal is located in the index finger which is connected to power, while the other two terminals are ground terminals that connect in parallel with a pull-down resistor to digital pins. Once the power terminal touches either of the other terminals, the closed connection drives the corresponding digital pin to higher levels, where the MCU will recognize the change. This acts like a digital switch. This data will be combined with other sensor data and be relayed forward.

12.2.8 Batteries/Charger

In order to charge the batteries that will power the system, we need to look at how it all connects together. When it comes time to charge the Li-Po batteries that are embedded into the sign language glove, all you need to do is plug the micro-USB cable to the port that is provided in the board. The figure below shows how this can be achieved. The charge current is about 500mA, this is how much it can be supplied to the battery. There is an LED that will turn ON when charging and it will also turn OFF when the batteries are fully charged. This is helpful in preventing overcharge from occurring.[108]

We decided to use a 2000mAh Lithium Ion Polymer Battery. The size of the battery (54 x 60mm) is small enough where it won't make the glove bulky. The battery was chosen based on estimated power consumption. It has an estimated 6.5 hours of runtime. Its temperature functionality is also impressive, being able to withstand temperatures of -25°C to 60°C. There will be an LED in the PCB that will turn on when the battery is fully charged, which is helpful so the battery doesn't overcharge and get damaged over time.

12.2.9 Voltage Regulator

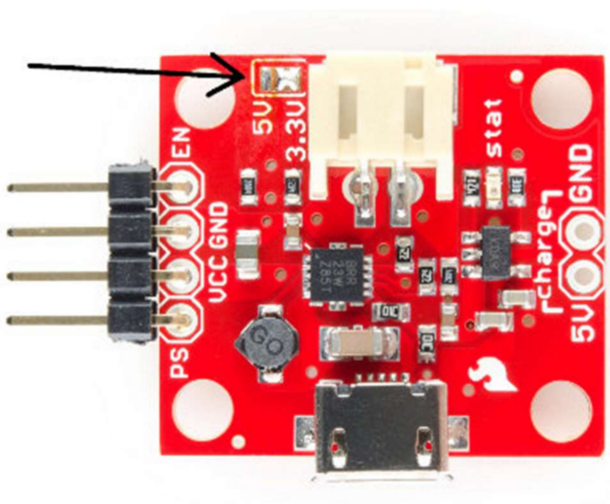
The voltage regulator is actually embedded into the charging board that was discussed in the previous section. We will use the same board to discuss this component. This voltage regulator has the ability to step-up and step down the voltage. From the figure below, you can see that the board allows the us to use the 5V and 3.3V as the output voltage. The max current you can use for the 5V is 600mA

[108] See Appendix A – 113

and for the 3.3V is 200mA. Next we will discuss the 4 pins in the top of the board including:

- PS - Power Saver
- VCC - Input Voltage
- GND - Ground
- EN - Enable

The power saver and enable pins are used to shut down the regulator when not in use in order to conserve the power so it lasts longer. Putting the EN pin to low will shut down the regulator. By making the PS and EN pin low, it lowers the drain current of the Li-Po battery. The VCC pin is the input voltage that will come from the microcontroller. The GND pin will be connected as well to the MCU ground



pin.[109]

Figure 12.2.9-1: Battery Charging method/Voltage Regulator[110]

12.2.10 Battery Voltage Sensor

This sensor helps our group when determining if the battery is running low. It is useful because we wouldn't want out glove to die at an important time where we are in the middle of use. As you can see from the figure below, this fuel gage consists of 8 pins. The first pin we will discuss is the VCC pin which is located on the top right hand side. This is where the input voltage will come in, which is distributed by the MCU to this board. It has a minimum input voltage of 2.5V and a max input voltage of 4.5V. The SDA pin will be used as the serial data I/O and it

109 See Appendix A – 115

110 See Appendix A – 116

will connect to the data signal of the I^2C interface. Clock input can be connect to the SCL pin if needed. This pin also connect to the two wire interface.[111]

The battery voltage sensor will be connected directly to the microcontroller unit (MCU) to measure the battery levels. The Bluetooth connected to the MCU will transfer the relevant data to the computer display, making it easy for the user to read the battery levels while using the ASLA glove.

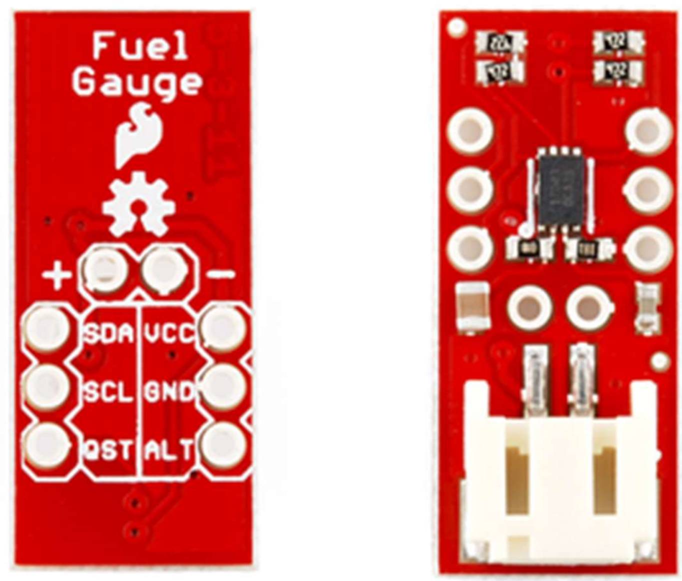


Figure 12.2.9-1: Battery Voltage Sensor Back/Front[112]

12.2.11 Bluetooth

The Bluetooth implementation is basically based on the different signals that the Bluetooth module must receive and transmit to the other Bluetooth module or the computer in the case of our project. In our project we will be implementing the BlueSMiRF Gold Bluetooth module that can be found and purchased on the SparkFun website. This was the best choice for our project design since it has a very low power feature and also a suitable operating voltage that works well with our selected Arduino board.

One of the good things about purchasing these Bluetooth modules is the way they are designed which give the user a practical and easy way to implement these boards with whatever design that you are working with. This simply means that they basically come ready to just plug them in. Although, these Bluetooth modules

---

[111] See Appendix A – 117

[112] See Appendix A – 118

are not basically ready to be used because the fact that the pin headers must be solder before in order to make an electrical connection. These pins header are the six through hole place that is displayed in the back of the Bluetooth device depicted below:

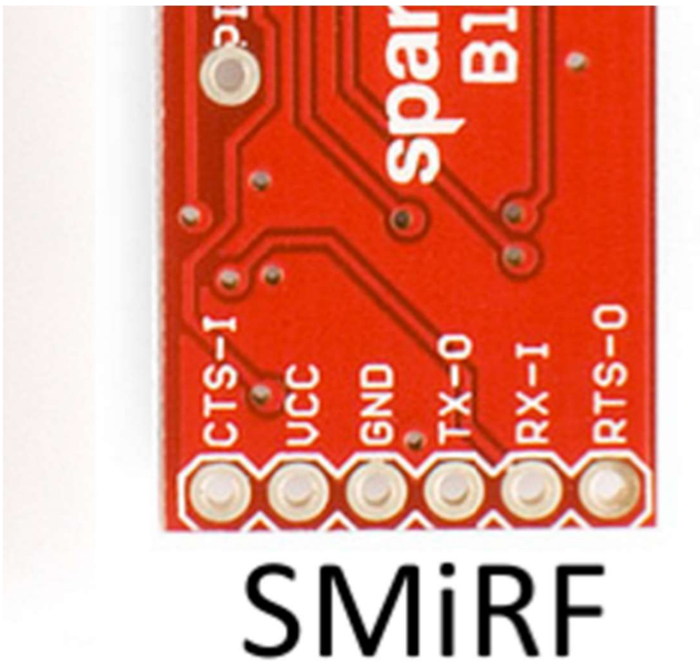


Figure 12.2.11 – 1: BlueSMiRF Six-plated through-holes[113]

The above figure represents the six-plated through-holes that must be solder before to make an electrical connection so it can be easily plugged into an Arduino board. To do this the use of a female header. A female header allows to connect the six-plated through-hole of the Bluetooth module with a series of pins that can be solder to the board easily. This allows to simply hook up the Bluetooth module with the Arduino board.

Another option to connect the Bluetooth to the board is to simply use wires to directly connect these pins to the board. The way to do this is by simply solder the wires to the Bluetooth module and then to connect these respectively to the Arduino board. This method might be needed in case where the connections are crossed and a direct connection using a female header is not recommended.

To set up the connection this case with the Arduino board, it is important to know this Bluetooth device will be transmitting a series of signals. These signals are also known as TTL-level signals. These TTL-level sigs are not to be confused with the

[113] See appendix A – 119

RS-32. Also, in regards to the voltage it is important to distinguish between the different voltages allowed for functioning which are 3.3 Volts and 5 Volts for this Bluetooth module. There is an option of not directly connecting the Bluetooth modem to the Arduino board which is using SoftwareSerial. This terminology of SoftwareSerial allows for the communication to other different digital pins by making the software replicate the different functions of these pins. These digital pins of the Arduino board can basically be programmed to serve other functions. On the Arduino board the speed of the Receiver Rx are of 57600 bps. In general, the SoftwareSerial helps avoid spurious bus connections. Also, this method is to also avoid any random spurious data from conflicting with the initial upload. The diagram of an Arduino Board with our selected Bluetooth module the BlueSMiRF. The connection is simple, the grounds are connected to each other, the VCC connected to the 5 Volt port of the Arduino board. The transmitter port TX-0 of the Bluetooth is connected to the Digital pin number two of the Arduino board or D2. The receiver port RX-I of the Bluetooth is connected to the digital port of the Arduino board which is D3. It is important to highlight that the digital pins of the Arduino board are designed to be capable of supporting the TX and RX connections the Bluetooth module which just means that the Bluetooth modules TX and RX ports can be connected to any of the digital pins of the Arduino board, and not only to the labeled ones.

Label	Function	I/O	Description
CTS-I	Clear to send	Input	Serial flow control signal. Not required
GND	Ground	Power In	Common to Bluetooth devices as 0V
RTS-O	Request to send	Output	Hardware flow control. Not critical for normal serial communication
RX-I	Serial Receive	Input	Receives data should be connected to the TX of the device
TX-O	Serial Transmit	Output	Sends data to other device. Should be connected to the RX of the device
VCC	Voltage Supply	Power In	Signal routed through a 3.3 voltage regulator, then sent to the Bluetooth module 3.3V-6V.

Table 12.2.11-1: Pinouts and their respective functions[114]

[114] See appendix A – 120

At this point of the Bluetooth connection, the full connection is almost there. The next step is to link up the connection between the two Bluetooth devices which will be the computer in this case since both of the gloves will be communicating back to the computer data base. In order to do this we need to work with the Bluetooth firmware. What this really means is that the communication between the Bluetooth modules will be done by the serial interface which takes control of these. This would allow to control the data transmissions. The way this interface works is in a pipeline type of way which basically consists in the following manner. First, the data is received through the RX-I pin of the Bluetooth module and is then sent through the Bluetooth connection. Next, the data received from the Bluetooth device is passed through the Bluetooth device and directed through the TX-O transmitter serial port connected to the Arduino board. A simple but useful schematic of the pipeline diagram is presented on the next page:

Figure 12.2.11-2: Bluetooth Data Pipeline[115]

Above is a perfect example of the communication that will be taken place in our project. The serial data connections will be hooked up to the Arduino board to transfer all the data collected from the sensors on the glove. Then the data received from the Bluetooth module connected to the glove will be passed through the Bluetooth module and directed to the antenna of the module to be then sent to the computer to update the database or to just compare the data from the glove with the one on the already existing database.

After long periods prototyping and testing, we designed a complete system on a PCB which houses the Bluetooth chip used in the Bluetooth module the BlueSMIRF Gold. This chip is the RN-41 manufactured by Roving Networks. As shown in the picture below, our group placed this chip on the PCB so that the Antenna would not have any problems with the ground copper plane of the PCB as recommended

[115] See appendix A – 121

in the Data Sheet. This is why only the antenna part of the RN-41 chip is off the board. The rest of the connections are the same as shown in the prototype using the digital pins through the software implementation of UART. Also, we included two LEDs, one to be able to tell when the Bluetooth is powered on and the other LED is to indicate that the Bluetooth is connected.

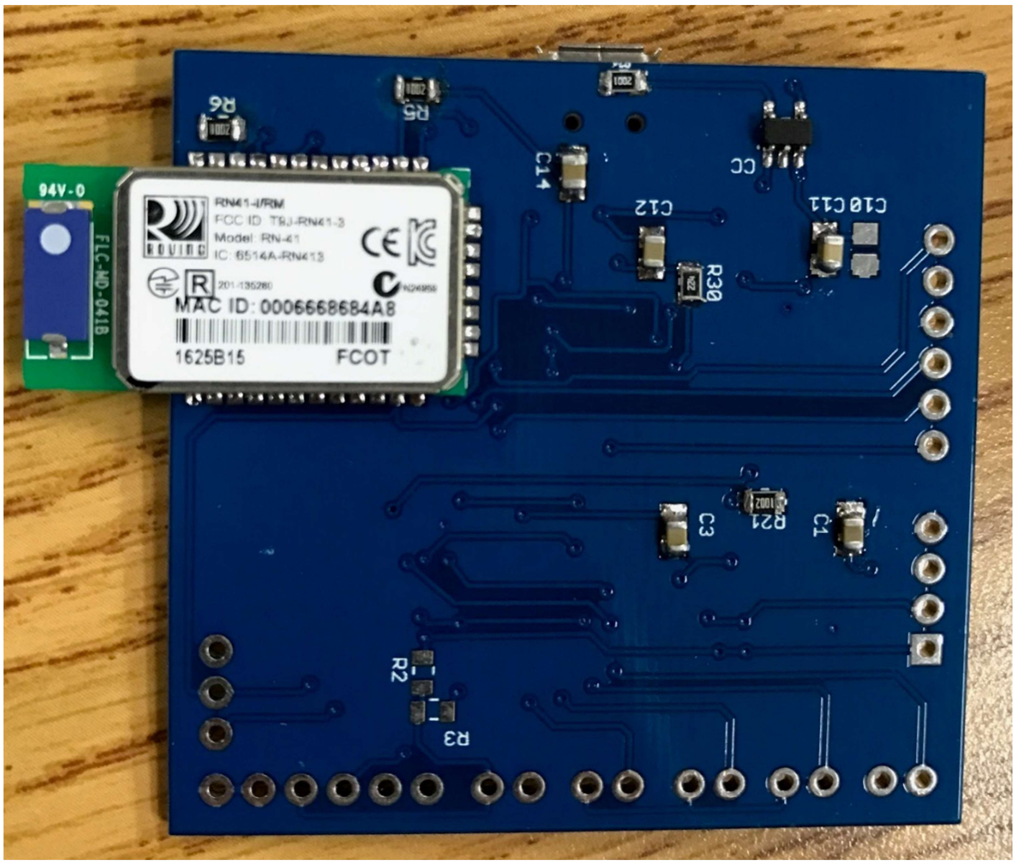


Figure 12.2.11-2.1: RN-41 Bluetooth Chip on the final PCB.

12.2.12 Microcontroller

The microcontroller will be the center of the portable system and the way all the sensors will be able to communicate their data to the computer for the analysis. It is important to note when connecting the microcontroller to the sensors and the battery of voltage levels necessary to power the devices, the device limits, and the specific interfaces necessary for the devices to communicate through. If these aspects of the design are not paid enough attention to overpowering a component can ruin it and be both time consuming and costly to fix. Also the note about making sure the proper interface is used will shorten any troubleshooting errors we run into, because if the wrong connection type is made it could cause the device to malfunction and in the worst case, signals could drive high a pin that should not be and potentially cause device failure.

The microcontroller is going to be prototyped using one of Arduinos development boards, the Arduino UNO. For the purposes of powering and programming the Arduino, a USB cable is most likely going to be used. The battery may also be connected through the VIN pin of the development board to test that the system works with the planned power source. The VIN pin on the Arduino can be powered with up to 12V and be regulated to 5V, taking note that if a voltage less than 7V is supplied a 5V output may not be supplied. The external power sources may be connected directly to the 5V and the 3.3V pins but there is no internal regulator on the development board for those connections so the damage may ensue. The 3V3 pin on the board may be used for a 3V regulated input. The GND pin is for grounding the board and making sure that the device operates properly and safely.[116]

The bootloader pre-loaded onto the controller by Arduino takes up 0.5 KB of the ATmega328's 32 KB memory. The ATmega also has another 2 KB of SRAM and 1 KB of EEPROM. All fourteen of the digital pins the development board provides access to are all configured for input-output functionality and operate at a 5V power level. The pins are also rated to provide and receive 20 mA. The development board also gives access to the other peripherals contained within the microcontroller, like the SPI, TWI, and UART interfaces. The dev board utilizes its 10, 11, 12, and 13 pins for the single SPI interface, the A4 and A5 pins for the TWI (I^2C) communication, and the PD0 and PD1 pins for the UART (TTL) serial communication interface. There are also six 10-bit resolution analog input pins (two of which are reserved for SPI) that will all be utilized for sensor input.[62]

The board also gives the user the ability to use external interrupts through the PD2 and PD3 pins which can be written to using the attachInterrupt function. Pulse width modulation is also available pins 3, 5,6,10, and 11 to generate an 8-bit output using analogWrite function. There is also a built-in LED that can be controlled as well.[122]

The Arduino UNO development board will be programmed such that all the necessary data from the sensors is taken in and then transmitted through the Bluetooth properly. This will be done be hooking up the IMUs, voltage gauge sensor, and possibly a flex sensor to the TWI. The TWI is designed such that it can take in data from multiple sources and keep their integrity. The contact sensors will

[116] See Appendix A – 122

be attached to the digital pins where their data will be taken in. The final device being attached to the microcontroller board would be the Bluetooth wireless module, which would connect via the two TTL pins. These are the pins for the USART interface and are compatible with the Bluetooth module chosen.

Once all the electrical elements are connected to the Arduino properly it becomes an issue of making sure the controller is programmed properly and that the pins are configured appropriately for their desired functionality. We want all the inputs to be configured so, and for the more intricate interfaces, such as USART, the datasheet has specific code instructions and descriptions for the programing. Once the data is transmitted through the Bluetooth to the PC it becomes a pure software compilation of the data to determine the proper word or expression being expressed by the sign.

After verifying the prototype of the project to work, a final PCB was made to realize all the components necessary for the project into one board that would fit onto the back of the hand. The board itself is small and lightweight, so the final microcontroller that was found to work for both the projects analog needs and size constraint was the ATmega328P-TQFP. This provided the correct number of analog and digital pins, while supplying enough power at a fast enough clock rate to perform the project's needs, while staying compact on the back of the hand.

12.2.13 Printed Circuit Board (PCB)

A schematic was designed that included connections for power, sensors, Bluetooth and MCU all in one. It was important to label nets with appropriate names so when errors arose debugging became much easier. Also keeping different functional groups of the overall system separated, as seen in Figure below, made it much easier to see that all the components and connections were made properly.

Trying to keep the components on a single side of the board was desired, but it was determined that some of the components needed to be placed on the bottom layer in order for the design to fit properly. This gave us more room to make simpler traces. A ground plane was used on both sides of the board to ground all the components further eliminating crossing traces and cleaning up the board design. The ground plain also helps reduce signal interference. Vertical traces were mostly used in the top layer of the PCB, while Horizontal traces were mostly used in the bottom layer. The PCB design can be seen down below in Fig. F. This method of laying traces was recommended to minimize routing interference.

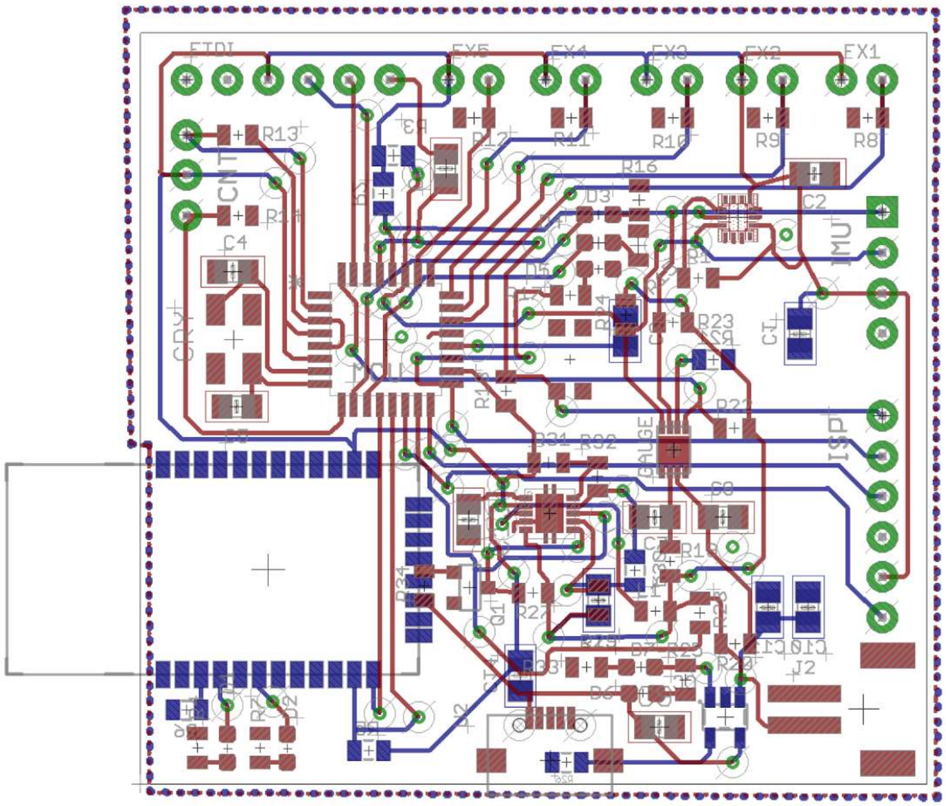


Figure 12.2.13-1: ASLA's 2 layer PCB Design

The flex sensors pins are located on the top of the board to make the shorter connections to the flex sensors. The Bluetooth was placed on the bottom layer because it took up a lot of space and was not necessary to keep on the top. The antenna part of Bluetooth projects out from the board because it cannot have any copper pour or traces underneath it due to potential interference. The JST connector used for the battery was specifically placed in the bottom right corner to allow for easier connection of the battery. The final PCB dimensions came out to be 1.8 x 1.9 in, which is an ideal size to fit in the back of the glove.

The groups PCB needs were met by the board layout company PCBWay. The decision to use this company came once it was heard that other groups were able to get working boards in a cheap and timely manner from them. The boards that were received from them were printed properly, electrically sound, and shipped in a relatively short amount of time. After the final board was populated and tested, all connections worked properly, and the project was functioning without error.

12.3 Software Prototype

Having a software prototype is important to have, probably even more important than a hardware prototype for early in the development stages. This is so we can develop stable software and give the programmers enough time to learn the theory behind machine learning and be able to apply it to the project. As machine learning is a very math intensive and complex subject the team is planning on developing the prototype during the off semester of summer. The software prototype must, at the very least, include the main part of the machine learning algorithm, not finely tuned but developed enough that it will be able to recognize a majority of the alphabet with only one glove. The software prototype is going to be split into a couple major components and standalone prototypes and will be integrated together as each component has reached a mature stage in its development. List of major components are:

- Embedded software glove sensor data acquisition
- Main software glove sensor data recording
- Machine learning
    - Introducing new data set to learn
    - Deciding on appropriate word or phrase based on captured data
- Graphic user interface
- Text-To-Speech

Each component will be rigorously tested and when deemed stable it will undergo integration with the rest of the software.

12.3.1 Embedded Software

The embedded software that goes into the chosen microcontroller will be one of the first prototypes developed with the hardware prototype. The embedded software prototyping milestones are:

1. Reading of analog sensor data
2. Reading of binary sensor data
3. Reading of I^2C sensor data (IMU and/or possible additional analog-to-digital IC)
4. Sending of data through the Bluetooth module
5. (Optional) Integrating a checksum if we notice data packets are missing over the wireless communication

Figure 12.3.1.1 shows the flow diagram of how the embedded software works.

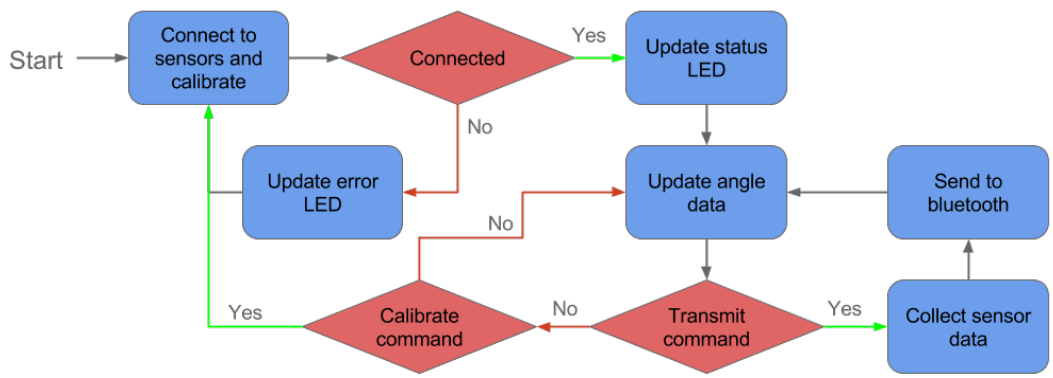


Figure 12.3.1.1: Embedded Software Flow Diagram

To test the reading of sensor data we will have the development board for the chosen microcontroller hooked up to the computer with a debugging interface running. This is not only to make sure that all the sensors are being read, but to make sure the microcontroller can handle the transmission of all of the data first through USB. If the data being received is within acceptable range we will continue to testing over wireless. If the data being received is not within the acceptable range we will have to test different data transmission rates or calibrate each sensor multiple times depending on the nature of the issue of the data being received. This will then be repeated again with the Bluetooth module. If the loss of packets is apparent with the Bluetooth module and is hindering the performance and accuracy of the conversion software running on the host computer we will incorporate a checksum to guarantee the validity of the data packets.

12.3.1.1 Complimentary filter

The accuracy of the accelerometer and gyroscope by themselves is questionable but when merged together they have been proved quite reliable. To combine the two sensor data we must first come up with a control system which governs them. The accelerometer data we obtain is reliable over time, while stuttering during the beginning of movement. The gyroscope is the opposite, it can detect rapid movement with ease while causing drift over time. We must use a low-pass filter for the accelerometer while using a high-pass filter with the gyroscope. Before we pass the accelerometer and gyroscope data through these filters we must first pre-process it. With the accelerometer we must first find the arctangent of the two axis we want to observe. The gyroscope we must integrate the last axis, as that's the plane we are observing, with respect to time. After we pass this data through the high-pass and low-pass filter we then add the two angles we get and this is our final data with respect to the horizontal of that plane. The equations for this are shown below.

$$\theta_{accel} = \arctan(\frac{accel_Z}{accel_y})$$

$$\theta_{new} = \alpha\left(\theta_{old} + \dot{\theta}_{gyro} * dT\right) + (1 - \alpha)(\theta_{accel})$$

12.3.2 Data Recording

This is part of the main software that is going to run on the host computer. The function of this is to use the Bluetooth module in order to read the incoming data packets sent from the microcontroller. The data, depending if we are recording a new word/phrase or using the real time capture for sign-to-speech portion of our project will either be manipulated and stored or just manipulated. For the scope of the prototype we are going to be focusing on the storing of the data. The file format we will be storing the data is going to be in the file format known as Comma Separated Values or CSV for short. The purpose of this file format is to separate the data with commas, the data must be stored in a particular predetermined order. Each data entry will be separated by a new line. This will be our raw data to be saved and to be used when training a new word or phrase.

12.3.3 Machine Learning

The machine learning prototype will be split into two concurrent developing prototypes, learning a new word/phrase prototype and the sign-to-speech prototype. Each prototype will handle half of the machine learning algorithm and when integrated will be the full machine learning algorithm that will be used by our project. Below is the translating flow diagram that the program will have:

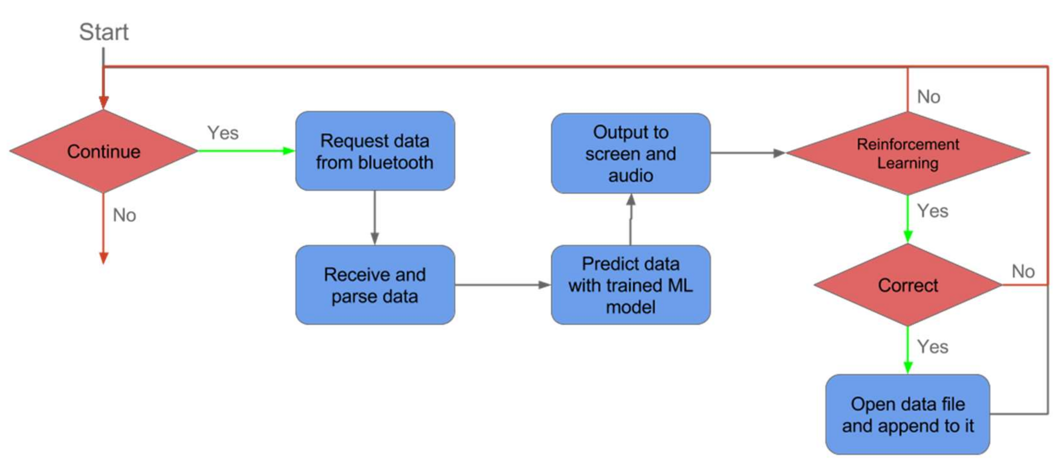


Figure 12.3.3.1: Translating Flow Diagram

12.3.3.1 Learning letters

This will be the "supervised" part of supervised learning with the algorithm we chose. Teaching the software with the sample of data over time and telling the software what is the expected output. We will train the software multiple times, like that of registering a new fingerprint or a new vocal phrase that our phones/computers will respond to. This trained phrase will be saved to the host's computer with the file format CSV. The format of which will be predetermined before prototyping to ensure a known standard while developing the software. This may change as time progresses to enhance accuracy and validity of the trained file. The user will also be able to look at the current library of expressions linked to each word or phrase and the ability to retrain the expression, delete it, or test it to see if it captured it correctly (this portion of the software will be developed in the sign-to-text section). The software will then record five data samples for each letter as a base line for the machine learning model. The flow diagram for learning the letters is shown below.

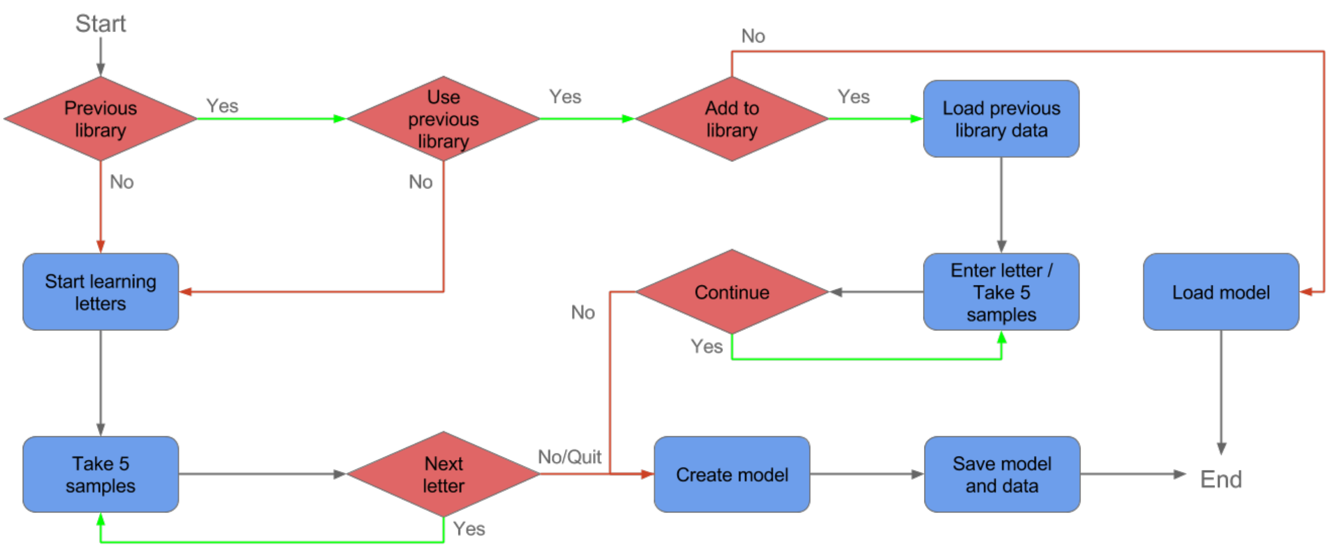


Figure 12.3.3.1.1: Learning Flow Diagram

12.3.3.1 Expansion of Library

The user can expand the beyond the original twenty-six letters of the alphabet. They can include numbers and any other sign that does not require movement. By adding this the glove is customizable to the user based on his or her needs. This can then be easily used with words and phrases when it is implemented down the line as stated in section 9.6.

12.3.3.3 Words and Phrases

The project can implement words and phrases with more research into how to create a time series data model prediction. We have thought about this in a number of ways but due to time failed to implement it. One way we can implement this is to have a data sample for a fixed amount of time (around three seconds) as soon as the glove detects movement. This can give us a rough but rather unreliable data sample. If the sign is shorter than the fixed amount of time the user then would have to hold the last position until the glove is done capturing data.

12.3.3.4 Sign-To-Text

This is the decision making section of the whole machine learning algorithm. Based on the data inputted it will decide the best output from the trained set of words and phrases. As mentioned in the previous section there will be a testing area for new phrases and words. This will be the base of the program to ensure it is working at the unit level, being able to distinguish correctly the intended output from a real time input. After this has been complete we will continue with having the software be able to detect multiple signs over a relatively short period of time. This is the part that will take time integrating and fine tuning, making sure that the response time is good along with high accuracy and possibly even some prediction. This is why developing a prototype as early into the summer is favorable, to ensure the software has enough time to be developed and finished.

12.3.4 Text-To-Speech

Going from text to speech is a relatively simple module as it only needs to implement an already built API based on the programming language of choice. Which for the prototyping stage is python. Testing this would also be simple for the fact all we have to do is give it a string input and listen to it to make sure it is outputting the right word through the speakers.

12.4 Bill of Materials

After putting together all of the group efforts, we have come to some final agreements regarding the different components needed to put together for our design. As mentioned in previous sections the American Sign Language has a few amount of components. Although, there is not a high amount of components for this design, a careful selection of the components made for variety of factors. One of the factors includes that at the current moment, our group does not have any funding meaning that we will be putting money aside for this project. Another factor to take into account is the fact that for prototyping purposes, we are going to design one glove to begin with. After the full development of the first glove, then we will proceed to add the second one for the full implementation of sign language which includes full sentences instead of just one glove which is only capable of implementing letters. For this reason, the cost of our design will fluctuate. Therefore, our bill of materials (BOM) will have to take into account the implementation of two gloves not just one. This means that we have to double the cost. The following is a comprehensive table for the bill of materials. This table includes the part name, part number, the quantity required, the cost of each part, and the total cost of the whole project or basically of all the parts needed:

Part Name	Part Number	Quantity	Cost
BlueSMiRF Gold Bluetooth	WRL-12582	2	69.90
Nike Run Glove		2	14.99
SparkFun 6 Degrees of Freedom Breakout- IMU	SEN-13339	2	79.80
SparkFun Power Cell - LiPo Charger/Booster	PRT-11231	2	39.00
Polymer Lithium Ion Battery	PRT-08483	2	25.90
SparkFun LiPo Fuel Gauge	TOL-10617	2	19.90
4.5'' Flex Sensor	SEN-08606	6	77.70
3'' Flex Sensor	SEN-10264	4	31.80
Copper Tape		1	5.95
Atmel MCU	Atmega328P-PDIP	2	8.60
Printed Circuit Board		4	56.25
		Total	429.79

Table 12.4 – 1: Bill of materials

13 User Manual

This is a simple description of the basic functionalities of our project. With this manual, the user will have some basic understanding which include the hazards and the charging methods for this project.

13.1 Glove Operations

This Glove operates off a rechargeable battery that must be uniquely recognized by the user to know when it is on or when it is off. Also, with the operations there must be some basic understanding of how and what could be some major hazards for our translator glove. These points will be explained below.

13.1.1 Charging the Glove

Since the battery will be not be easily removable, we will discuss how to charge the batteries. The battery will charge using a Li-Po charger. The charger system has a Micro-USB port that will be used to charge the batteries, which plugs into the wall outlet.

1. If an LED light in the microcontroller blinks red, this mean the battery is close to running out of power.
2. Plug the Micro-USB cable into the Micro-USB port in the charging mechanism.
3. If the LED in the charging system turn ON, that means the battery is fully charged.

13.1.2 Hazards

Keep the glove away from general electric hazards. Such like

- Extreme temperatures
- Fires
- Water and humid areas
- Large falls
- Electricity including static

Any of these may damage the system beyond repair. Make sure to remove gloves after operation to not damage the contact sensors.

13.2 Software Operations

When starting up the software the user is greeted by the main menu. With the ability to either train a new word or phrase, or to start the sign-to-speech program. The base program will have the whole alphabet along with some commonly used words and phrases.

13.2.1 Training Words and Phrases

Here a library of included words along with the ability to record a new word/phrase, re-record a word/phrase, test a word/phrase, or just delete an entry.

13.3.1.1 Record a New Word or Phrase

Recording a new word is easy, just select the option to do so and on screen prompts should help you. You will be instructed to type the word or phrase you would like to train. Then a countdown timer should start, once counted down the software will tell you to sign the word or phrase you want to teach it. This will happen multiple times to ensure accuracy. It then will ask you to check the validity of the recording by telling you to sign it again and to approve or reject the word it thinks it is.

13.3.1.2 Re-record a Word or Phrase

Re-recording a word or phrase would be the same steps as recording a new one with only one change. You would need to select that word or phrase in the shown list of known word or phrases and pressing "Re-record".

13.3.1.3 Test a Word or Phrase

First you select a word or phrases from the list of known word or phrases, and then you click "Test", this will launch the testing program and ask you to sign the word. It will tell you either it captured the right gesture or after a few seconds it will time out and ask you to re-record the word or phrase.

13.2.2 Sign-To-Speech

It will prompt you to connect the glove via Bluetooth if not done so. After of which a real time display of the currently signed word is shown on the screen, with a button to enable or disable the computer speech generation.

13.3 Connecting the Glove to the Host Computer

To connect the gloves to the computer is like any other Bluetooth device. You go to your Bluetooth settings on the host computer if it has built in or you are using a Bluetooth USB communicator. You pair the computer with "ASLA" depending if it is right or left hand. The pin requested will be the standard 0000. After you are done with this the gloves should be paired with the host computer. The software then will attempt to connect to it every time it is started. If it is found and connected it continues as normal, if not the software then terminates.

Appendices

Appendix A - URL Links and References

[1] https://github.com/dsarmiento/ASLA
[2] http://libguides.gallaudet.edu/content.php?pid=119476&sid=1029190
[3] https://en.wikipedia.org/wiki/Sign_language
[4] http://www.lifeprint.com/asl101/topics/wallpaper1.htm
[5] https://cdn-shop.adafruit.com/datasheets/SpectraFlex2inch.pdf
[6] http://www.spectrasymbol.com/flex-sensor
[7] http://www.sensorwiki.org/doku.php/sensors/flexion
[8] https://www.tekscan.com/product-group/embedded-sensing/force-sensors
[9] https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf
[10] http://www.dimensionengineering.com/info/accelerometers
[11] http://www.livescience.com/40102-accelerometers.html
[12] http://diyhacking.com/arduino-mpu-6050-imu-sensor-tutorial/
[13] http://www.livescience.com/40103-accelerometer-vs-gyroscope.html
[14] https://learn.sparkfun.com/tutorials/gyroscope/how-a-gyro-works
[15] http://www.arm.com/products/processors/cortex-a/index.php
[16] http://www.arm.com/products/processors/cortex-r/index.php
[17] http://www.arm.com/products/processors/cortex-m/index.php
[18] https://en.wikipedia.org/wiki/TI_MSP430
[19] http://www.ti.com/lsds/ti/microcontrollers_16-bit_32-bit/msp/low_power_performance/msp430f5x_msp430f6x/products.page#p2954=LQFP&!p1342=LCD;USB
[20] http://www.ti.com/lit/sg/slab034ad/slab034ad.pdf
[21] http://www.ti.com/lsds/ti/microcontrollers_16-bit_32-bit/msp/low_power_performance/msp432p4x/products.page
[22] http://www.atmel.com/products/microcontrollers/avr/megaavr.aspx
[23] http://www.atmel.com/products/microcontrollers/avr/tinyavr.aspx
[24] http://www.atmel.com/products/microcontrollers/avr/avr_xmega.aspx
[25] http://www.atmel.com/products/microcontrollers/avr/32-bitavruc3.aspx
[26] http://www.webopedia.com/TERM/W/Wi_Fi.html
[27] http://www.ampedwireless.com/learningcenter/default2.html
[28] http://cp.literature.agilent.com/litweb/pdf/5989-4204EN.pdf
[29] https://en.wikipedia.org/wiki/Bluetooth_low_energy
[30] http://www.ashwireless.com/story/what-Bluetooth%C2%AE-low-energy-and-how-it-different-%E2%80%98classic%E2%80%99-Bluetooth%C2%AE

[31] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5626224
[32] https://en.wikipedia.org/wiki/Machine_learning
[33] http://embedded-computing.com/articles/analytics-driven-embedded-systems-part-2-developing-analytics-and-prescriptive-controls/
[34] https://en.wikipedia.org/wiki/Statistical_classification
[35] https://www.activestate.com/blog/2015/10/python-gui-programming-wxpython-vs-tkinter
[36] https://en.wikipedia.org/wiki/Speech_synthesis
[37] http://batteryuniversity.com/learn/archive/whats_the_best_battery
[38] http://www.analog.com/library/analogdialogue/archives/48-08/battery_controller.html
[39] http://batteryuniversity.com/learn/article/charging_nickel_based_batteries
[40] http://aeguana.com/blog/step-up-boost-dc-dc-converters/
[41] https://learn.sparkfun.com/tutorials/i2c
[42] https://cdn.sparkfun.com/assets/3/d/1/b/6/51adfda8ce395f151b000000.png
[43] https://learn.sparkfun.com/tutorials/i2c
[44] https://cdn.sparkfun.com/assets/6/4/7/1/e/51ae0000ce395f645d000000.png
[45] https://learn.sparkfun.com/tutorials/i2c
[46] https://learn.sparkfun.com/tutorials/serial-communication
[47] https://cdn.sparkfun.com/assets/d/1/f/5/b/50e1cf30ce395fb227000000.png
[48] https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus
[49] https://cdn.sparkfun.com/assets/e/3/b/d/1/50e5d529ce395fd27b000001.png
[50] http://compnetworking.about.com/cs/wireless80211/a/aa80211standard.htm
[51] http://pocketnow.com/2014/01/23/5ghz-wifi
[52] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57853
[53] https://www.w3.org/XML/Core/#Publications
[54] http://www.metlabs.com/blog/battery/top-3-standards-for-lithium-battery-safety-testing/
[55] https://www.ucamco.com/files/downloads/file/81/the_gerber_file_format_specification.pdf?ac40b80482e8e1864842659f85fb1c9c
[56] https://www.sparkfun.com/products/8606
[57] http://www.spectrasymbol.com/wp-content/themes/spectra/images/datasheets/FlexSensor.pdf
[58] https://cdn-shop.adafruit.com/datasheets/SpectraFlex2inch.pdf

[59] https://www.sparkfun.com/products/13339
[60] https://cdn.sparkfun.com/datasheets/Sensors/IMU/LSM6DS3_Breakout_v10.pdf
[61] https://www.sparkfun.com/products/10736
[62] https://www.sparkfun.com/products/10617
[63] http://cdn.sparkfun.com/datasheets/Prototyping/MAX17043-MAX17044.pdf
[64] http://www.atmel.com/devices/ATMEGA328P.aspx?tab=parameters
[65] http://www.atmel.com/devices/ATMEGA2560.aspx?tab=parameters
[66] http://www.atmel.com/devices/ATMEGA328P.aspx?tab=parameters
[67] http://www.atmel.com/devices/ATMEGA2560.aspx?tab=parameters
[68] http://store-usa.arduino.cc/
[69] http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf
[70] http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf
[71] http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf
[72] http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf
[73] http://wiki.iteadstudio.com/Serial_Port_Bluetooth_Module_(Master/Slave)_:_HC-05
[74] http://84ace.com/?p=45
[75] https://learn.sparkfun.com/tutorials/using-the-bluesmirf?_ga=1.22217851.1767113737.1459231623
[76] https://learn.sparkfun.com/tutorials/using-the-bluesmirf
[77] https://www.sparkfun.com/products/12582
[78] https://en.wikipedia.org/wiki/XBee
[79] https://github.com/ibr-alg/wiselib/wiki/Example-MQTTSN-network-(Raspberry-Pi-and-Arduino)
[80] https://www.jetbrains.com/pycharm/features/
[81] http://www.codeblocks.org/features
[82] https://www.arduino.cc/en/Guide/Environment
[83] http://www.atmel.com/tools/atmelstudio.aspx
[84] https://en.wikipedia.org/wiki/Support_vector_machine
[85] https://en.wikipedia.org/wiki/Artificial_neural_network

[86] https://en.wikipedia.org/wiki/Microsoft_Speech_API#SAPI_5.4
[87] https://pypi.python.org/pypi/pyTTS/3.0
[88] http://cdn.sparkfun.com/datasheets/Prototyping/Lithium%20Ion%20Battery%20MSDS.pdf
[89] http://cdn.sparkfun.com/datasheets/Prototyping/Lithium%20Ion%20Battery%20MSDS.pdf
[90] https://www.sparkfun.com/products/11231
[91] http://cdn.sparkfun.com/datasheets/Components/General%20IC/33244_SPCN.pdf
[92] https://www.sparkfun.com/tutorials/379
[93] https://www.sparkfun.com/datasheets/Prototyping/tps61200.pdf
[94] https://en.wikipedia.org/wiki/Electronic_design_automation
[95] http://www.cadsoftusa.com/eagle-pcb-design-software/
[96] http://www.cadsoftusa.com/eagle-pcb-design-software/software-version-news/software-version-news-in-v7/
[97] http://www.cadsoftusa.com/eagle-pcb-design-software/schematic-editor/
[98] http://www.cadsoftusa.com/eagle-pcb-design-software/layout-editor/
[99] http://www.altium.com/company/newsroom/press-releases/altium-delivers-new-altium-designer-14
[100] https://www.expresspcb.com/
[101] https://github.com/sparkfun
[102] https://cdn.sparkfun.com/assets/learn_tutorials/4/1/6/DM00133076.pdf
[103] http://cdn.sparkfun.com/datasheets/Components/General%20IC/33244_SPCN.pdf
[104] https://www.sparkfun.com/datasheets/Prototyping/tps61200.pdf
[105] http://cdn.sparkfun.com/datasheets/Prototyping/MAX17043-MAX17044.pdf
[106] https://oshpark.com/
[107] http://www.4pcb.com/pcb-student-discount.html
[108] http://www.4pcb.com/pcb-order-policy.html
[109] https://en.wikipedia.org/wiki/Printed_circuit_board
[110] https://en.wikipedia.org/wiki/Soldering#Soldering_defects
[111] https://cdn.sparkfun.com/assets/learn_tutorials/4/1/6/DM00133076.pdf
[112] http://www.footlocker.com/product/model:225828/sku:NRGD3020/nike-dri-fit-tailwind-run-gloves-mens/black/anthracite/&SID=9114&inceptor=1&cm_mmc=SEM-_-PLA-_-Google-_-nrgd3020&gclid=Cj0KEQjwo_y4BRD0nMnfoqqnxtEBEiQAWdA127d6Q2iYchWwfY7cdJRC5UxLzq2KmepMu_iNztLX6OcaAnj88P8HAQ
[113] https://www.sparkfun.com/datasheets/Prototyping/tps61200.pdf

d

[114] https://www.sparkfun.com/tutorials/379
[115] http://cdn.sparkfun.com/datasheets/Prototyping/MAX17043-MAX17044.pdf
[116] https://www.sparkfun.com/tutorials/379
[117] https://learn.sparkfun.com/tutorials/using-the-bluesmirf
[118] https://www.sparkfun.com/products/10617
[119] https://learn.sparkfun.com/tutorials/using-the-bluesmirf/hardware-overview
[120] https://www.arduino.cc/en/main/arduinoBoardUno
[121] https://learn.sparkfun.com/tutorials/using-the-bluesmirf/hardware-overview
[122] https://www.arduino.cc/en/main/arduinoBoardUno

Appendix B - Emails Requesting Permission for Pictures

Image Permission

Charlie Munoz <cmunozag10@gmail.com>
to web.management

7:17 PM (1 hour ago)

web.management
web.management@digi.com
Show details

Hello,
I am a student at the University of Central Florida working on my senior design project and I would like to use one of the figures from your website in my design document. We are required to write and design a document for our project and it would be for educational purposes only. Would it be possible to obtain permission to use this figure?

Thank you for your time

...

Charlie Munoz

Permission to Use Image/Data

CF Colin Fox
Wednesday, April 27, 2016 at 8:14 PM
To: support@adafruit.com

To whom it may concern,

I am a student at the University of Central Florida working on my senior design project and I would like to use some of your figures and schematics on devices/components I plan on using from your website in my design document. We are required to write and design a document for our project and it would be for educational purposes only. Would it be possible to obtain permission to use these figures?

Thank you,
Colin Fox

Permission to Use Image/Data

CF Colin Fox
Wednesday, April 27, 2016 at 8:16 PM
To: customerservice@sparkfun.com

To whom it may concern,

I am a student at the University of Central Florida working on my senior design project and I would like to use some of your figures and schematics on devices/components I plan on using from your website in my design document. We are required to write and design a document for our project and it would be for educational purposes only. Would it be possible to obtain permission to use these figures?

Thank you,
Colin Fox

Permission to Use Image/Data

CF Colin Fox
Wednesday, April 27, 2016 at 8:25 PM
To: customerservice@atmel.com

To whom it may concern,

I am a student at the University of Central Florida working on my senior design project and I would like to use a schematic/package layout of a devices I plan on using from your website in my design document. We are required to write and design a document for our project and it would be for educational purposes only. Would it be possible to obtain permission to use this schematic/package layout?

Thank you,
Colin Fox