Babel Glove

Ann Dang, Andrew Glisson, Madison Manley, and Shreya Mistry

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

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

***Keywords* — ASL, Low Energy Bluetooth, machine learning, microcontroller, sensors**

# Introduction

The Babelglove is a glove, embedded with sensors, that can translate the ASL hand gestures of the wearer to english text, using a computer program. It is lightweight and comfortable for the user to wear. The glove is fitted with flex sensors, a gyroscope and an accelerometer to capture the user’s hand and finger movements. A microcontroller and a bluetooth module are used to communicate between the sensors and the computer and the trained machine learning model take the data and produce an output on a desktop application. The entire system will be designed to be user friendly for all learning levels.

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



Fig. 1. Reference image of the English alphabet to American Sign Language.

# HARDWARE OVERVIEW

The Babel glove is employed with multiple flex sensors, an accelerometer, and gyroscope which is then connected to the MCU. The MCU sends the data from the sensors to the computer using a Bluetooth module.

**

Fig. 2. The Babel Glove Hardware Block Diagram, showing the main components that run the ASL translation.

1. *Flex Sensors*

The flex sensor is the primary element to collect data from the glove. A flex sensor is a device that relates how much the element bends to a change in resistance. One flex sensor is placed on each finger component of the glove and as the user bends their fingers, the flex sensor reads the change in resistance and sends this data to the microcontroller.  The greater the bend, the higher the resistance value. The different combinations in the change of resistance is used to identify each letter from the alphabet the user is signing.

For this project, the team decided to use the 4.5 inch flex sensor by Spectra Symbol. This flex sensor was chosen due to its reputation as a reliable part and also that it matched the specifications needed for this project. This sensor will be long enough to the fingers on the glove and has a good range to detect the motions as the fingers move in various ways. If needed, modifications can be made to the length of the flex sensor to better fit each finger placement. An adhesive will be used to secure the sensors in place on the glove. The different voltages will be measured and recorded. This will be used to help differentiate between the different letters and phases.

1. *Accelerometer and Gyroscope*

The accelerometer is used to help improve the motions that are being sensed using acceleration. It will detect other motion, such as moving the hand at different angles, by measuring the change in acceleration. A gyroscope is a device that measures or maintains rotational motion, known as angular velocity. A triple axis gyroscope measures rotation on the x, y, and z axis.  These are small, inexpensive devices that, when combined with the accelerometer, will help improve the measurements. The data collected from these two sensors will be more accurate and help provide more information to distinguish each letter or phrase. The figure below helps illustrate why an accelerometer and gyroscope are needed for the Babel glove.



Fig. 3. Reference images of Basic sign language gestures, many gestures are primarily based on movement so extra sensors were needed instead of just the flex sensors.

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

1. *Microcontroller*

When choosing an appropriate microcontroller for the Babel Glove, there are many basic considerations that should be accounted for. Because this project is using a mix of flex sensors and accelerometers, the MCU needs to at least have six analog input pins that each have an analog to digital converter (ADC). The power consumption of the processor should also be considered as it is desirable for the glove to employ only one battery to power the whole system. It would then be desirable to choose a unit that doesn’t consume a large amount of power. In the end, the group decided to use the ATMega328P. This MCU has six analog input channels and twenty digital channels. It has 2.5 KB of RAM, 32 KB of storage, and has an operating speed of 16 MHz. By using the ATMega328P, the group could easily program it using an Arduino Uno development board.

1. *Wireless Communication System*

The Babel Glove will be converting the hand gestures of the user into text, which will be shown on an external display. As the group is aiming for the glove to be as comfortable as possible for the user, the group wants to avoid having to use long cables and wires that are not only uncomfortable for the wearer, but also downright unaesthetic. Therefore, a wireless communication system is needed to help make the glove lightweight and portable.

The group has decided that Bluetooth technology should be used for the Babel glove. The user of the glove is expected to always be within a few meters of the computer; therefore, it is unnecessary to use Wi-Fi which is designed for greater distances and Near Field Communication technology, which requires the user to be within a few inches.

As there are multiple types of Bluetooth, the group decided to use BLE as it has a significantly lower power consumption compared to Classic Bluetooth. To communicate and send the sensor data from the glove to the computer for processing, we used a Bluetooth Low Energy module connected to the glove. By using Bluetooth Low Energy, we could maintain portability of the glove as well as reduce the power consumption versus using standard Bluetooth. Bluetooth Low Energy operates upon the Generic Attribute Profile (GATT) which provides common operations and a framework to be stored by the Attribute Protocol.

In the end, the group has decided to use the HM-10 4.0 BLE module from DSD Tech. The HM-10 uses the CC2541 chip by Texas Instruments. Newer versions of HM were considered, but they use newer versions of Bluetooth such as 4.2 and 5.0, which not all computers support.

1. *Power Source*

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

The group has decided to use a lithium ion polymer battery for the Babel glove because they are light, thin, and powerful. These batteries are low-maintenance and have no memory and do not need a full discharge to remain in good shape. They have a high specific energy and high load capabilities with Power cells. They also have a long cycle and shelf-life and low self-discharge and take about 2-3 hours to be fully charged and have a 99% charge efficiency.

A 3.7V lithium ion polymer battery – 2500 mAh from adafruit.com was chosen to power the Babel Glove. The battery comes pre-attached with a 2-pin JST-PH connector and includes the necessary protection circuitry. This keeps the battery voltage from overcharging or over-use and will cut-out when the battery is dead at around 2.8V. A Li-ion/Li-Poly constant-voltage/constant-current charger is recommended to charge the battery at a rate of 1200 mA or less. The battery model is LIPO785060 weights 43g and is 47mm x 61mm x 6.7mm in size.

1. *Charging*

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

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

1. *Voltage Regulators*

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

# SOFTWARE DETAIL

Overall software for our glove is split into three sections: the embedded software on the MCU, the machine learning algorithm, and the translation output. The embedded software on the MCU was written using the Arduino IDE and was designed to send the data from the sensors to the PC for translation output. The machine learning algorithm and translation output are hosted on an external computer running on Python 3.8.3.



Fig. 4. Overview of the software implemented into the Babel Glove, including the embedded programming, machine learning and the text and speech software.

1. *Embedded Software*

 This program mainly sends the data from the flex sensors, accelerometer, and gyroscope to the PC for processing. The gyroscope and accelerometer were connected to the MCU using I2C lines, SDA and SCL, which use two analog channels on the ATMEGA328P. Four of the five flex sensors were connected to ADCs on the MCU and the fifth flex sensor was connected to a digital pin. To simulate an analog value for the flex sensor connected to a digital pin, the circuit seen in the figure below was used with a 1 uF capacitor and 550Ω resistor. The RCtime function was used to give a value that would simulate the output from an analog pin.



**Fig 5: Circuit to connect flex sensor to digital pin**

Once these connections were made, the Bluetooth module was connected to send the sensor’s data. To do this, the TX and RX line of the Bluetooth was connected to the RX and TX pin of the MCU, respectfully. The data from all the sensors was then programmed to be sent to the PC via Bluetooth.

1. *Machine Learning*

This section goes into the detailed analysis of the design of the chosen machine learning algorithm used in our glove system. Our team researched several different and varied machine learning algorithms for our projects, we narrowed the type down to classification, and then chose support vector machines (SVM) for the machine learning algorithm to translate our user’s ASL signs. [4] Support vector classification was chosen because it excels at high dimensional spaces and datasets with a large number of attributes, which helps our problem. SVM processes all of our attributes, the flex sensors, accelerometer and gyroscope, and finds the best hyperplane in the space. As shown in the figure below, the algorithm calculates the optimal hyperplane with the maximum margin between multiple classes or categories. In terms of our project, the SVM learned to separate the different ASL signs we fed it, finding the largest margin between the signs and then given the user’s input, predicts the appropriate hand sign.

To train our developed machine learning program, our team created a dataset of the ASL signs that we wanted to translate to demonstrate our glove’s translation ability. The dataset contains our component values and labels for the entire alphabet and the phrases: hello, goodbye, please, thank you, yes, and no. Our team created cases for each ASL sign using multiple techniques to help the variation of the values and in turn the accuracy of the model. We repeated the same hand sign over and over and collected the values of the flex sensors and accelerometer and gyroscope to help expand the case size in the support vector machine classification algorithm. We also used different hand sizes for each case to help train the model to accurately predict the correct ASL sign when the user has a smaller or larger hand size. We followed the typical community standard for datasets in machine learning, [1] and that is splitting the dataset into training data and testing data. 80% of the dataset is dedicated to training and will be imputed into the algorithm and the other 20% is dedicated to test and evaluate the training algorithm.



Fig 6. Pipeline of the software that connects to our glove, including the machine learning prediction, and text and speech output.

To train the Support vector machine model, we first used a python class from the machine learning framework sklearn, called GirdSearchCV. It tests the training data against different parameters and kernels and finds the optimal combination for the imputed training data. The optimal parameters the program found were kernel rbf at C = 1000, gamma = 0.001. To finally fit and train the machine learning model, we used sklearn’s C-Support Vector Classification class with the parameters found via the grid search, we were able to achieve a 85% accuracy.

When the user is operating the glove, we use the trained SVM model to predict what ASL sign the user is making. As shown in Figure 2, our software pulls the glove data values and parses the data to input into the model. The pre-trained SVM model then predicts the sign and then outputs it via the translation output software into text and speech for the user.

1. *User Interface*

To display the result to the user, we will be using a graphical user interface displayed on a web page. The main purpose of the application will be to take the translation and display the translated gesture in the interface. The front-end side of the application will be built in HTML and JavaScript using the Node.js framework. We decided to use a JavaScript-based framework to develop the application because JavaScript had documented Bluetooth platform compatibility, and a mature and actively maintained framework.





Fig 7. Pipeline and Layout of the Graphical User Interface

The interface will take the translated gestures and display them on the screen for the user to see. From the web page, the user will connect to the glove using web Bluetooth which will start sending messages for the interface to output. The application utilizes Chrome’s Web Bluetooth API which is used to find the glove and connect to it. After the application finds and connects to the glove, the application receives the translated gestures from the glove’s MCU. The data is received as Generic Attribute Profile (GATT) Notifications, which is used by Bluetooth Low Energy devices. Once the application receives the data, the data is then displayed in a text area for the user to see.

Our team decided to create a web-based application for its portability between mobile and desktop devices. The application will utilize web Bluetooth libraries so that the application will be able to receive data from the glove for processing.

The layout for our application has been kept very minimal to reduce clutter and to provide a better ease of use for the user. The user interface elements on the page are the connect and disconnect buttons which are used to connect and disconnect from the glove as well as the text area which displays the translated gesture to the user.

1. *Software Changes*

Unfortunately, the group had issues connecting the user interface with the machine learning program, therefore gesture recognition had to be done with the MCU. To do this, before the program will determine which letter is being used, the user will need to calibrate the glove for five seconds. During this time, the user will fully bend the glove then fully stretch their fingers to get the minimum and maximum bend of the flex sensors for the user specifically. These values will then be normalized using the map function in Arduino. This is done because each person will not have the same exact flex sensor values as each hand is different. Doing this allows the glove to be used by multiple people instead of being designed specifically for one person. Once this is done, the sensors will continuous collect data every second and using multiple conditional statements, the output for each gesture will be determined. The program will then send the output via Bluetooth to the user interface to be printed.

****

Fig. 8: Software diagram for gesture recognition

# HARDWARE DESIGN

1. *The Glove*

The glove will be the main component that the hardware and other electronics besides the PCB will be attached to. The glove will need to be made from a comfortable and flexible material. It will also need to be effective against moisture. For the size of the glove, it will fit the average hand that is approximately 7.3 inches. This should be big enough to hold all the sensors and the electronics with the exception of the PCB. The glove should also be lightweight. With all the electronics and hardware attached to it, it should not weigh more than three pounds. Also, the glove needs to have five full individual finger slots that are not attached together. It is important that part of the finger slots should not be cut off as this will affect the sensor placement and that none of the fingers should be attached to one another as this would restrict movement. The back of the glove should be empty with no strap of velcro to secure the glove. This will impact the sensor placement.

1. *PCB*

A printed circuit board (PCB) will be designed to include the microcontroller components and power supply components. The group decided to design a basic two layer PCB that could be designed as small as possible. Because of very limited experience in soldering, through hole mounting (THM) was used for the majority of components as the group members don’t have much experience with soldering in general and through hold components are easier to replace, making testing a prototype easier compared to surface mounted components.

Eagle CAD software was used to design the schematic and layout for the PCB. The Gerber files were also generated using this software. Eagle was used since it is a free software for students and the members had some familiarity with the program due to the University of Central Florida’s EEL 3926L Junior Design course.

Designing the PCB required the use of additional libraries for specific parts. One of the most important libraries used was the Sparkfun library which included many components such as the ATMega328P MCU and crystal resonator. This made ordering parts much easier. After designing the schematic, the connections were manually routed.



Fig 9. Diagram of our PCB layout

Bay Area Circuits was decided as the PCB manufacturer due to their reliable reputation and quick shipping. Four boards were ordered through a special offer for around $90. Most parts were manually searched for and ordered through Digikey, Sparkfun, and Amazon.

The PCB was assembled using a soldering kit ordered through Amazon. The Diligent Analog Discovery 2 was used to help troubleshoot any problems with the board. To test the board, the sensors were sewed onto the glove and connected by cables to the board. Unfortunately, the PCB did not work correctly; therefore, the group had to use a solderable PC breadboard due to time constraints caused by Covid-19.

V. CONCLUSION

The Babel Glove is a glove that was created to translate American Sign Language into text. The purpose of the glove is to assist in communicating with ASL. The Babel Glove consists of five flex sensors which act like variable resistors and change resistance as each finger is bent. The data from the flex sensors in combination with the data gathered from the accelerometer and gyroscope helps determine which gesture is being performed.

The data from the sensor is sent to the microcontroller unit, the ATMega328P. This MCU has six analog input channels and twenty digital channels which will be used to connect the sensors and Bluetooth. This unit will collect the data and send it to the computer via Bluetooth.

The data sent through the Bluetooth will be used to  convert the hand gestures of the user into text, which will be shown on an external display. The software component of the glove is split into three sections: the embedded software on the MCU, the machine learning algorithm, and the translation output. The embedded software on the MCU was written using the Arduino IDE and was designed to send the data from the sensors to the PC for translation output. The machine learning algorithm and translation output are hosted on an external computer running on Python 3.8.3.

BIOGRAPHY

**Ann Dang**

**Computer Engineering**

Ann Dang is a first-generation undergraduate Computer Engineering student at the University of Central Florida. In the summer of 2014, she participated in the High School Research & Engineering Apprenticeship Program at UCF’s Center for Research in Computer Vision learning about different projects and papers that applied computer vision. Ann is a current Systems Engineering intern at Lockheed Martin supporting the Digital Sensor Systems team in modeling and simulation. She has been supporting the program since June 2019 starting as a College Work Experience Program student. In this project, Ann was responsible for creating the software for the user interface as well as helping to integrate the hardware and software together. After UCF, Ann will continue working at Lockheed Martin as a Systems Engineer Associate.

**Andrew Glisson**

**Computer Engineering**

Andrew Glisson is an undergraduate student at the University of Central Florida, majoring in Computer Engineering. He has been taking classes at the university for four years under the Electrical and Computer Engineering Department. Andrew’s focus on the project is building the machine learning algorithm used to decode what the user’s hand gestures translate to in English. He has taken circuit analysis and software engineering classes but feels most comfortable with the software side of this project.

**Shreya Mistry**

**Electrical Engineering**

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

**Madison Manley**

**Electrical Engineering**

 Madison Manley is an electrical engineering undergraduate student at the University of Central Florida. She has done undergraduate research for all four years in nanoelectronics and has been involved in various programs such as the Research and Mentoring Program (RAMP), Girls Excelling in Math and Science (GEMS), and the Student Undergraduate Research Council (SURC). In the summer semesters, Madison had the opportunity to do research at the University of California, Berkeley and Columbia University. Although her background and interests are in nanoelectronics, UCF has given her the knowledge and skills to contribute to the hardware design of the glove. After UCF, Madison will be an incoming electrical engineering Ph.D. student at Georgia Institute of Technology.

ACKNOWLEDGEMENTS

 The authors wish to acknowledge the assistance and support of Dr. Samuel Richie, Dr Lei Wei for their time, support, and understanding during this unprecedented time.

 We also want to thank our committee members for taking the time out of their day to be a part of our committee and being understanding of our situation.

REFERENCES

[1 ] Ng, Andrew. Structuring Your Machine Learning Project. 2020, cs230.stanford.edu/blog/split/.

[2] Gyroscope. 2020, <https://learn.sparkfun.com/tutorials/gyroscope/all>.

[3] Alapati, Sreejan. Yeole, Shivraj. A Review on Applications of Flex Sensors. 2020, <https://www.researchgate.net/publication/318850816_A_Review_on_Applications_of_Flex_Sensors>.

[4] Gandhi, Rohith. Support Vector Machine - Introduction to Machine Learning Algorithms. 5 July 2018, towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

[5] BU-201: Basics You Should Know – Battery University. April 2020. <https://batteryuniversity.com/learn/article/lead_based_batteries>

[6] Woodford, Chris. Bluetooth – Explain That Stuff. April 2020. <https://www.explainthatstuff.com/howbluetoothworks.html>

[7] RCtime – Arduino. July 2020. <https://www.arduino.cc/en/Tutorial/RCtime>