American Sign Language Alphabet Interpreter

Team 3 — Timothy Wong, Nicholas DeNardo, and Michael Genovese
ECE 445 Design Document — Spring 2017
TA: Jacob Bryan

Table of Contents

American Sign Language Alphabet Interpreter 1

Table of Contents 2

1 Introduction 3

- 1.1 Objective 3
- 1.2 Background 3
- 1.3 High-Level Requirements 4

2 Design 4

- 2.1 Physical Design 5
- 2.2 Sensors 5
  - 2.2.1 Flex Sensors Module 6
  - 2.2.2 Continuity Sensors 13
  - 2.2.3 Accelerometers 15
- 2.3 Microcontroller 16
- 2.4 Power 16
  - 2.4.1 Battery 17
  - 2.4.2 Power Converter 17
  - 2.4.3 Power Consumption 19
- 2.5 Output 19
  - 2.5.1 Speaker 20
  - 2.5.2 Status LEDs 20
- 2.6 Recognition Algorithm 21
- 2.7 Tolerance Analysis 22

3 Ethics and Safety 24

4 Requirements and Verifications 25

5 Schedule 25

6 Cost Analysis 29

References 30

Appendix 32

- A.1 Flex Sensor States 32
- A.2 Continuity Sensor States 33
- A.3 Accelerometer States 34

1 Introduction

1.1 Objective

As it stands, there exists a communication barrier between those who can use spoken language and those who cannot use spoken language. Many of those in the latter group who reside in North America rely on American Sign Language (ASL) to communicate due to their inability to effectively produce or process audible language. Yet, relatively few members of the hearing community possess the ability to interpret American Sign Language [1]. Acknowledgement of this inability enlightens us to the fact that an effective sign language translation system could allow hearing and non-hearing individuals to communicate effectively with each other.

We endeavor to take the first step toward effortless communication between the hearing and the deaf by developing a wearable device that audibly translates the ASL alphabet into the English alphabet. While ASL contains signs for many words, there are many instances in which fingerspelling, the process of spelling a word using the twenty-six letters of the ASL alphabet, must be employed to convey meaning. Furthermore, because fingerspelling relies on only twenty-six signs that involve only the right hand, constructing a system that is capable of interpreting only the ASL alphabet is must more tractable than constructing a system that is capable of interpreting all of American Sign Language. Yet, by developing a wearable device capable of audibly translating the ASL alphabet into the English alphabet, we lay the foundation for a practical American Sign Language translation system.

1.2 Background

Many consider communication to be a person's most critical skill [3]. Unfortunately, 360 million people worldwide suffer from a disabling hearing loss that makes verbal communication prohibitively difficult [4]. To overcome this disability, between five hundred thousand and two million North Americans use American Sign Language to communicate [1]. ASL is a complete language that allows its users to communicate by gesturing their hands, adjusting their facial expressions, and modifying their posture [5]. While ASL offers deaf and hearing-impaired individuals the opportunity to communicate among themselves and with a minority of North Americans, it does not allow them to communicate easily with most North Americans who cannot interpret American Sign Language. Naturally, this inability to communicate can lead to social isolation for the deaf and hearing impaired. Moreover, Amatzia Weisel informs us in her book Issues Unresolved: New Perspectives on Language and Deaf Education that the negative effect that the segregation of the deaf has on both the hearing and the deaf can be quantified in both academic and social settings [7].

Despite the magnitude and severity of this issue, relatively little progress has been made toward developing a wearable sign language interpretation system. The most noteworthy effort comes from two University of Washington students who developed SignAloud, a glove-based system that translates American Sign Language into audible English words [9]. Despite this product's award-winning functionality and performance, it fails to abolish the communication barrier between hearing non-hearing individuals due to its lack of portability. Because this system relies on a nearby computer to perform the interpretation process, this product cannot be widely deployed for everyday use. A truly practical system would provide the accurate interpretation of SignAloud without the aid of external, cumbersome hardware.

1.3 High-Level Requirements

- The system must audibly translate all twenty-six letters of the American Sign Language alphabet into the English alphabet with ninety-five percent correctness.
- The system must be able to function properly for four continuous hours without having its batteries charged or replaced.
- The form factor of the system must not significantly impede the user's ability to sign.

2 Design

The interpreter system can be broken up into the four main parts as shown in the block diagram of Figure 1. Paramount to overall system function is the sensor module which will contain the hardware needed to describe the state of the user's hand. Data from this module will be sent to the microcontroller which will process the data and provide a serial output for the speaker module. The speaker module will serve as the output of the overall system and will announce the letter being signed. All subsystems will receive power from the power module, and the microcontroller will also illuminate LEDs to inform the user that the system is operational.

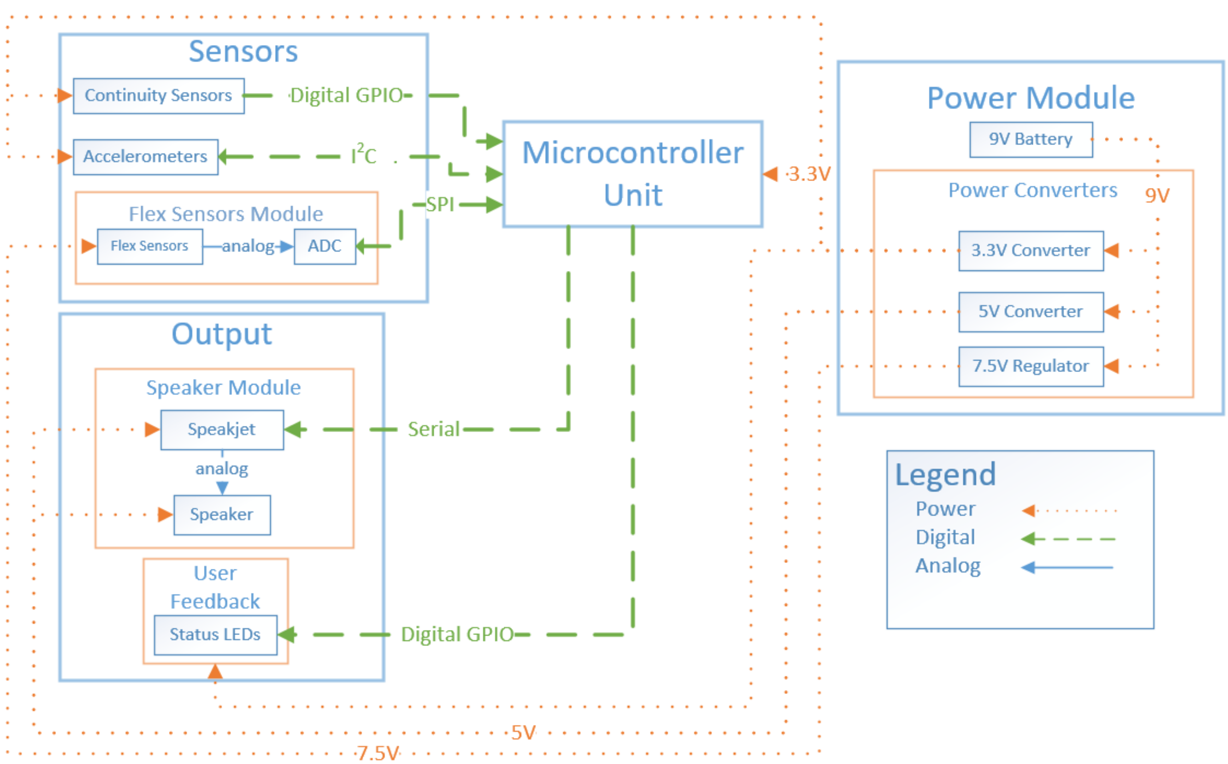

Figure 1. Block Diagram

2.1 Physical Design

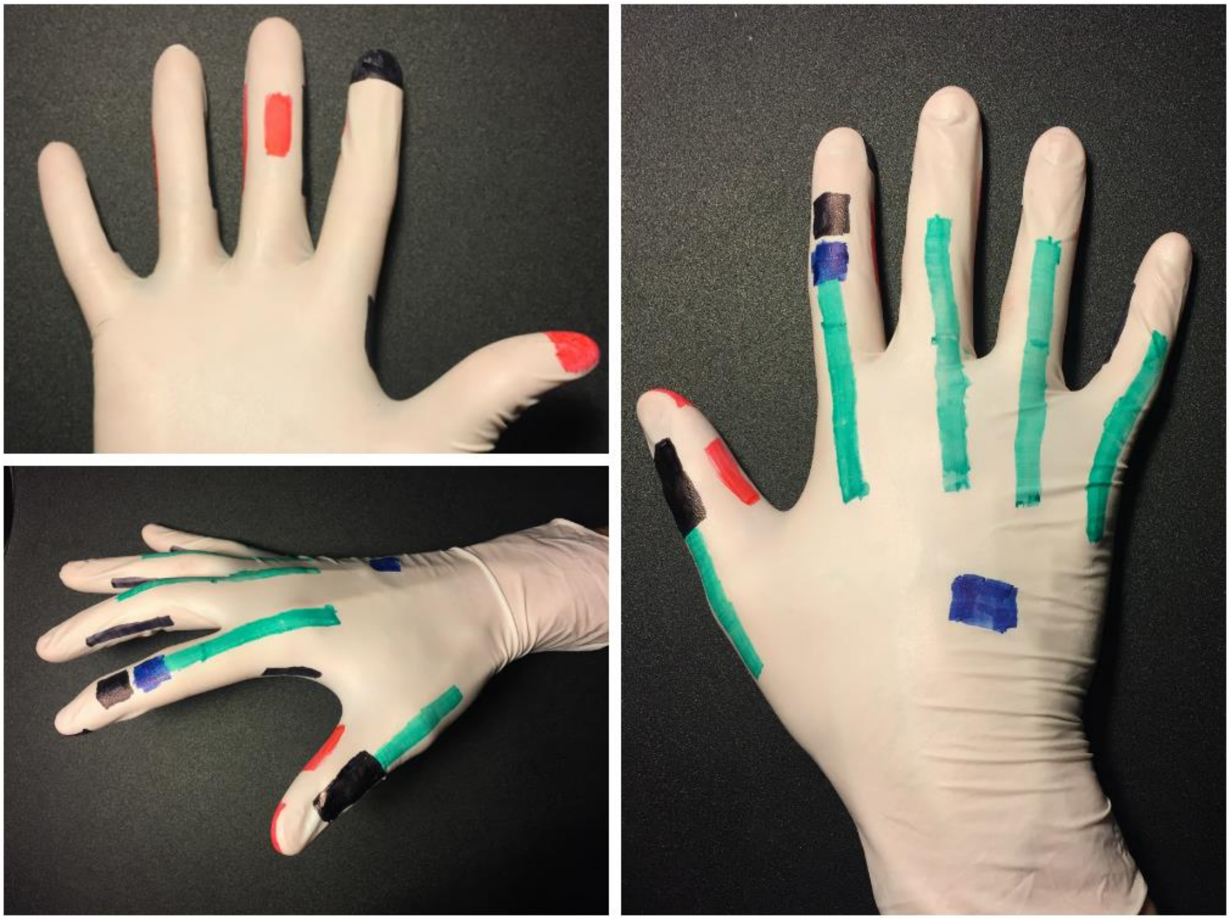

Figure 2. Proposed Sensor Configuration. The green stripes represent flex sensors, the blue squares denote accelerometers, and the red and black rectangles represent the high and low sides of continuity sensors, respectively.

The variety and placement of sensors in the system will be critical to our success in distinguishing characters. Reliable character recognition will be achieved through the system's ability to consistently measure finger articulation, the orientation of the hand and the index finger, and the grouping of fingers. To fully capture the range of motion in a user's hand, a combination of flex sensors, continuity sensors, and accelerometers will be used. Our proposed configuration of these sensors is show in Figure 2.

2.2 Sensors

The purpose of the sensors subsystem is to capture enough information about the state of the user's hand to determine if the user is signing a letter of the ASL alphabet and if so which letter. The information that this system seeks to capture is the extent to which each of the user's fingers are bent, how the fingers are making contact with one another, and the orientation of both the middle digit of the user's index finger and the palm of the user's hand with respect to the surface of the earth. This subsystem receives power from the power supply module and transmits the data it collects to the microcontroller for processing and interpretation according the interface of each individual sensor as detailed below. Accurate and precise data collection and transmission by the sensor subsystem is critical to the success of the overall system because the requirement for the overall system to translate all twenty-six letters of the American Sign Language alphabet into the English alphabet with ninety-five percent accuracy cannot be achieved without reliable data on the state of the user's hand.

2.2.1 Flex Sensors Module

The flex sensors subsystem provides the microprocessor with data on the extent to which each of the user's fingers and thumb are bent. This information is collected by measuring the output voltages of voltage divider circuits formed between flex sensors and resistors using an Analog-to-Digital Converter (ADC). Each output voltage is described by Equation 1 where V_{OUT} is the output voltage, $V_{DD,FLEX}$ is the input voltage to the circuit, R_{Flex} is the resistance of the flex sensor, and R_d is the resistance of the dividing resistor. The schematic for this circuit is shown in Figure 3.

$$V_{OUT} = \frac{R_{Flex}}{R_d + R_{Flex}} V_{DD,FLEX} \quad (1)$$

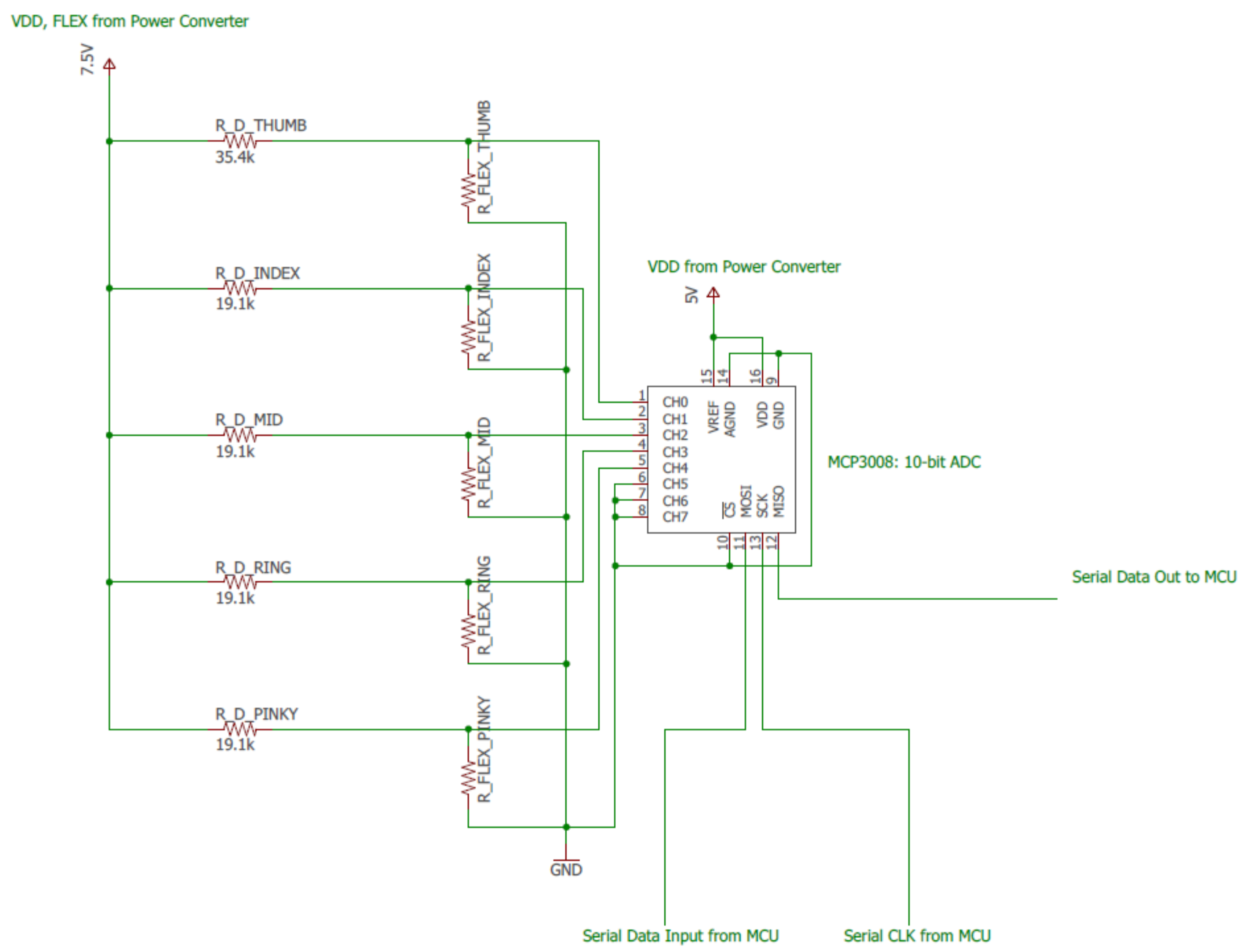

Figure 3. Circuit Schematic for Flex Sensors Module.

The flex sensors provide the core functionality of the circuit by varying in resistance as they bend with a digit. Specifically, the resistance in a flex sensor adhered to a digit increases as that digit becomes more jointed. Figures 4 and 5 show how the resistances in a set of seven 4.5-inch and six 2.2-inch flex sensors vary when bent as they might be when adhered to a finger or thumb, respectively. The information that this subsystem provides will aid the microprocessor in determining which letter is being signed by signifying whether the flex sensors on the index, middle, ring, and pinky fingers are in the low, moderate, or high resistance state and whether the flex sensor on the thumb is in the low or high resistance state. Clustering of index finger and thumb resistance states are shown in Figures 6 and 7, respectively.

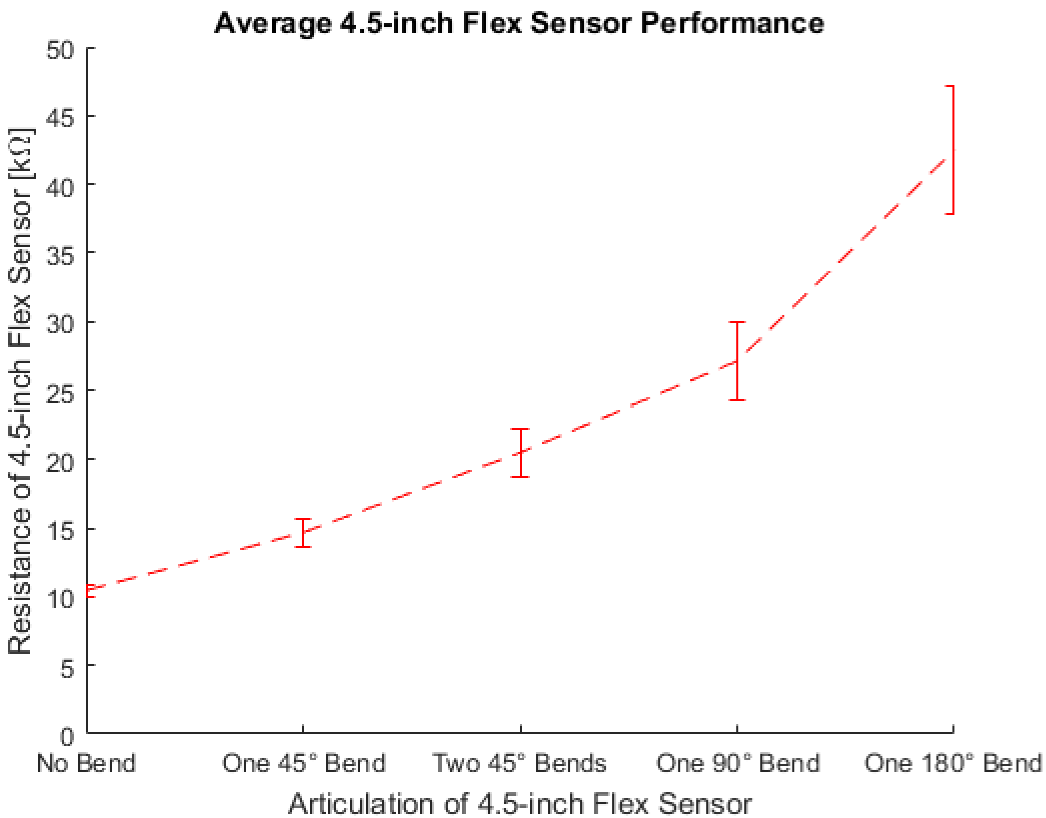

Figure 4. Average resistance of seven 4in. flex sensor at various degrees of bending. The error bars denote the standard deviation in the resistance at each level of bending.

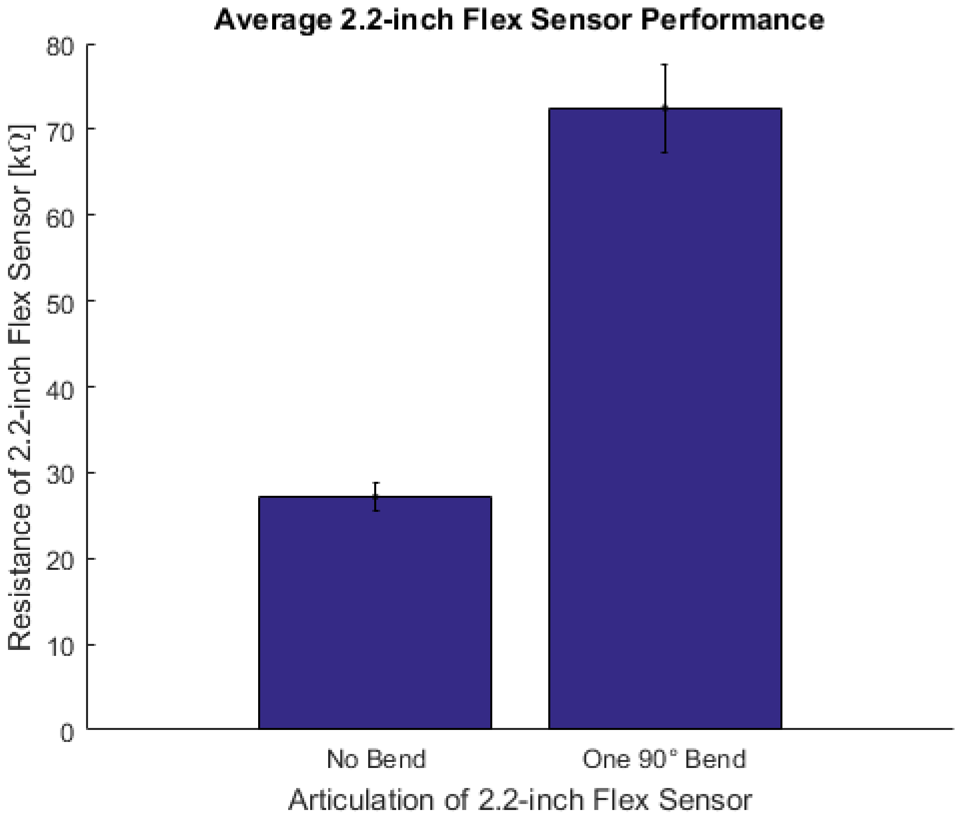

Figure 5. Average resistance of six 2in. flex sensor at various degrees of bending. The error bars denote the standard deviation in the resistance at each level of bending.

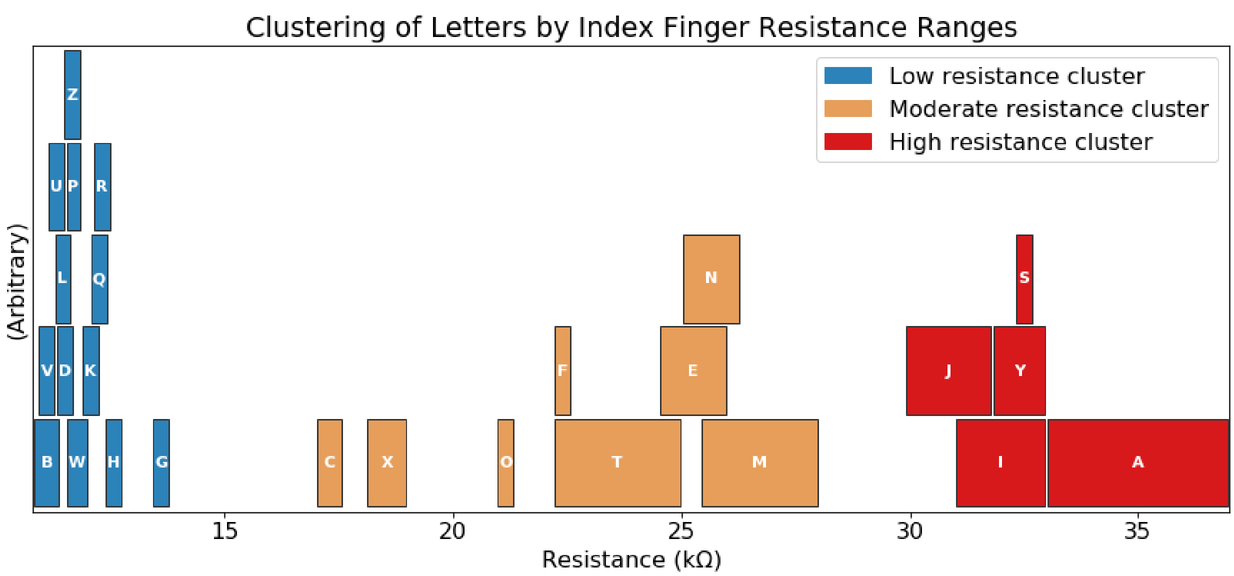

Figure 6. Clustering of letters by index finger flex sensor resistance

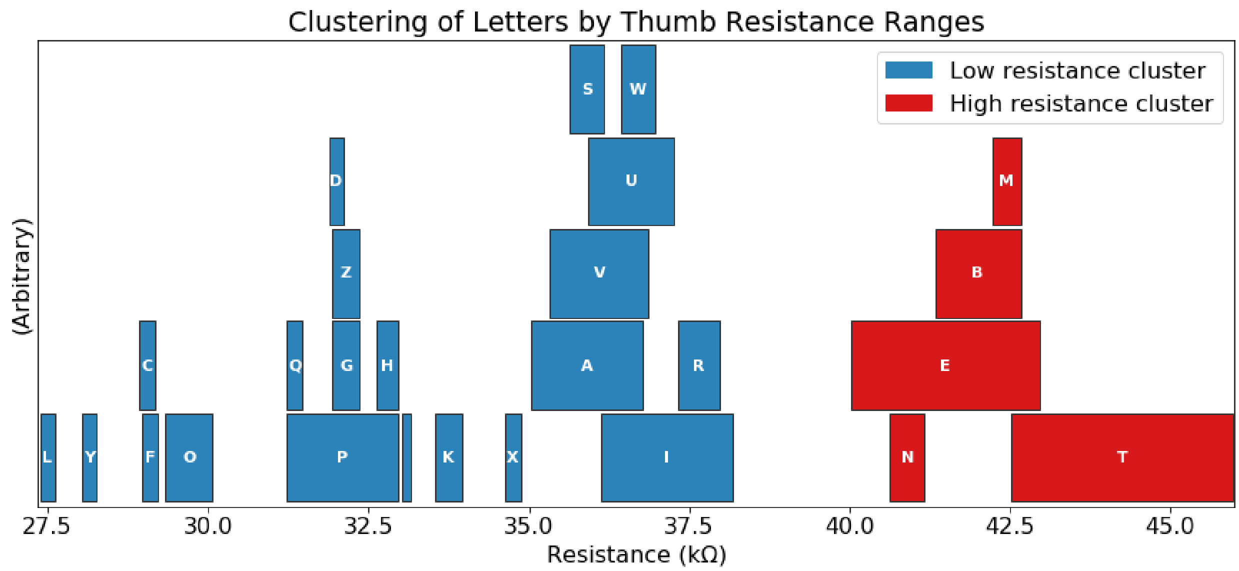

Figure 7. Clustering of letters by thumb flex sensor resistance.

Furthermore, this clustering enable us to define threshold resistances between each resistance state. It should first be noted that defining suitable threshold resistances is possible with is clustering because the distribution was selected to maximize the resistance gap between each grouping. Secondly, in selecting threshold resistances we seek to provide reasonable resistance ranges for each cluster. To accomplish this goal, we define the threshold resistances to be halfway between the edges of each cluster. This decision places the resistance thresholds at 15.38kΩ and 28.95kΩ for the 4.5-inch flex sensors and 39.10kΩ for the 2.2-inch flex sensors. Here, we note that the threshold resistance for each finger's flex sensor will be optimized to that finger's letter clustering in the final implementation. Yet, we approximate the middle, ring, and pinky finger thresholds with those of the index finger for design purposes due to a presently insufficient amount of clustering data.

Having established the boundaries of the flex sensors' resistance states, we needed a device that produced a voltage output representing the flex sensor's resistance. Our solution was a voltage divider. While other topologies exist for producing voltage measurements of resistances, alternatives such as common source amplifiers would increase the cost and design complexity of the flex sensor subsystem without providing substantial improvements in performance.

Following our decision to use a voltage divider, we made the decision to use the flex sensor as the output resistor rather than as the dividing resistor. In explaining this decision, let us consider an exemplary voltage divider circuit where the output resistor is R_2, the dividing resistor is R_1, and the output voltage is given by Equation 2. We can determine how the output voltage varies with a change in the value of each resistor by computing the partial derivatives of V_{OUT} with respect to R_1 and R_2. As Equations 3 and 4 show, the magnitude of the rate of change of V_{OUT} with respect to the value of either resistor is given by the value of the other resistor multiplied by the input voltage and divided by the square of the sum of the two resistances. Because these magnitudes are equal for a fixed value of the other resistor, we realize that neither position offers increased output voltage gain. For this reason, we arbitrarily placed the flex sensor in the position of R_2 so that the output voltage would increase as the flex sensor resistance increased.

$$V_{OUT} = \frac{R_2}{R_1+R_2} V_{IN} \tag{2}$$

$$\frac{\partial V_{OUT}}{\partial R_1} = \frac{-R_2}{(R_1+R_2)^2} V_{IN} \tag{3}$$

$$\frac{\partial V_{OUT}}{\partial R_2} = \frac{R_1}{(R_1+R_2)^2} V_{IN} \tag{4}$$

Now we faced the critical decision of selecting values for the dividing resistors. In doing so, we sought to maximize the ranges of voltages corresponding to the low and high resistance ranges to increase the system's noise immunity. Specifically, we wanted to maximize the difference between the maximum output voltage and the upper threshold voltage and the difference between the lower threshold voltage and the minimum output voltage. Figure 8 shows how the normalized maximum, minimum, and threshold output voltages vary with the dividing resistance for the 4.5-inch flex sensor. Figure 9 shows how the sum of the output voltage ranges for the high and low resistance states varies with the dividing resistance. By selecting the resistance that maximized this sum, we optimized the 4.5-inch flex sensor voltage divider circuits for noise immunity. After repeating this process for the 2.2-inch flex sensor circuits, we arrived at the values in Table 1.

2.2-inch Flex Sensor Dividing Resistance	35.4kΩ
4.5-inch Flex Sensor Dividing Resistance	19.1kΩ

Table 1. Dividing Resistor Values.

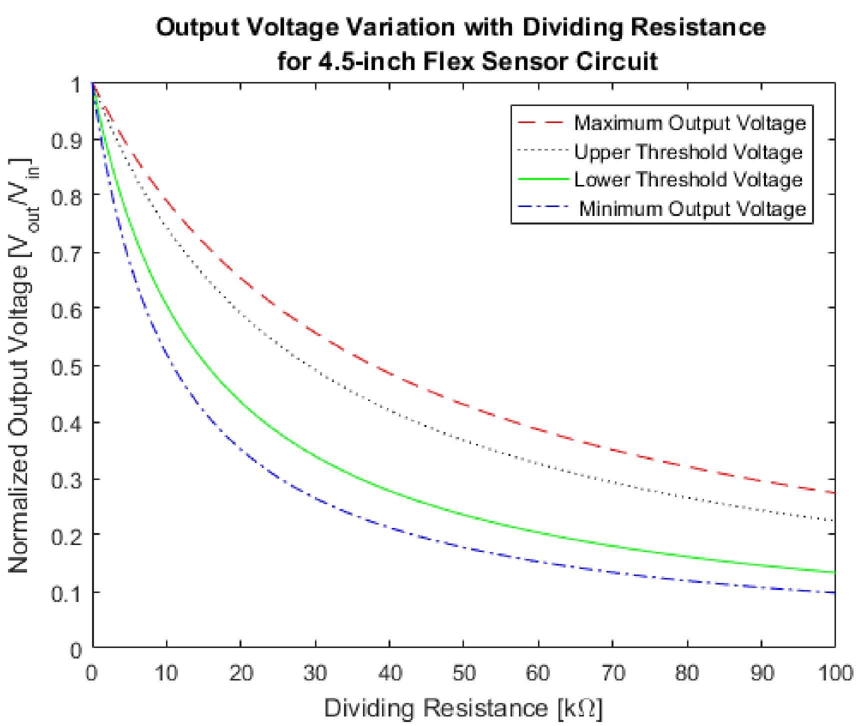

Figure 8. Normalized output voltage variation with dividing resistance.

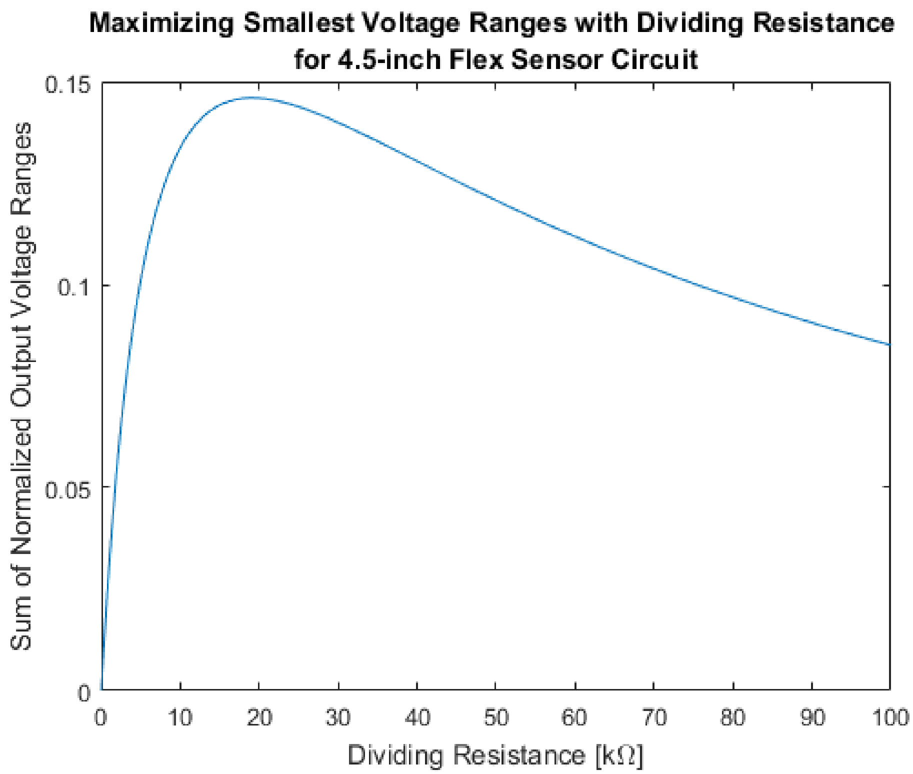

Figure 9. Sum of voltage ranges for high and low resistance states for 4in. Flex Sensors.

Additionally, Equation 1 relates that the voltage ranges for each resistance state scale with the input voltage. This fact leads us to desire a large input voltage. However, if the selected input voltage leads to an output voltage larger than 5V, additional circuitry or a significantly more expensive ADC would need to be used to handle the large output voltage. To avoid the significant increase in cost and design complexity, we opted to set the input voltage at 7.5V, nearly the highest input voltage that kept the output voltage below 5V. With this input voltage, we expect to see the output characteristic for the voltage divider circuits shown in Figures 10 and 11.

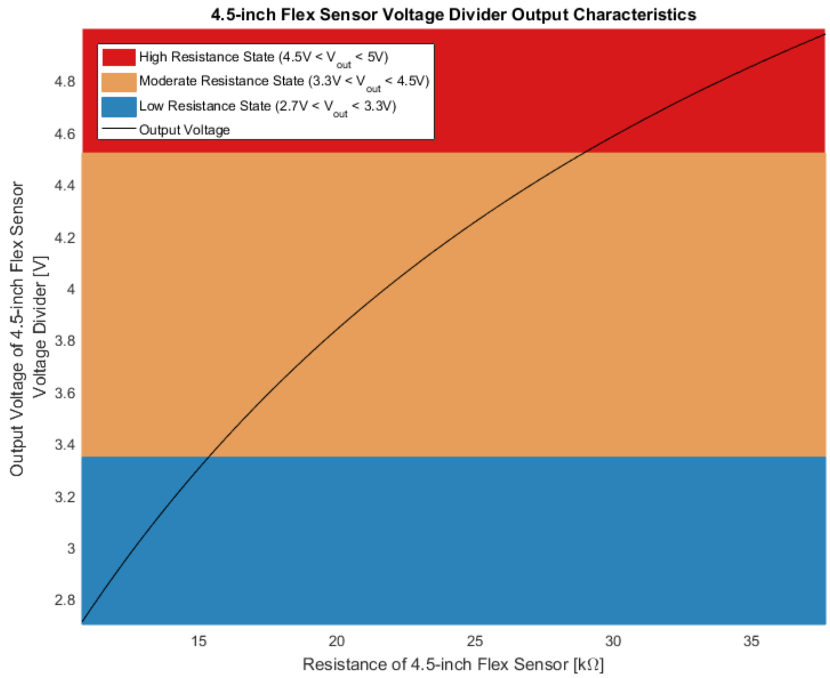

Figure 10. Relationship between output voltage and flex sensor resistance for 4.5-inch flex sensor voltage divider.

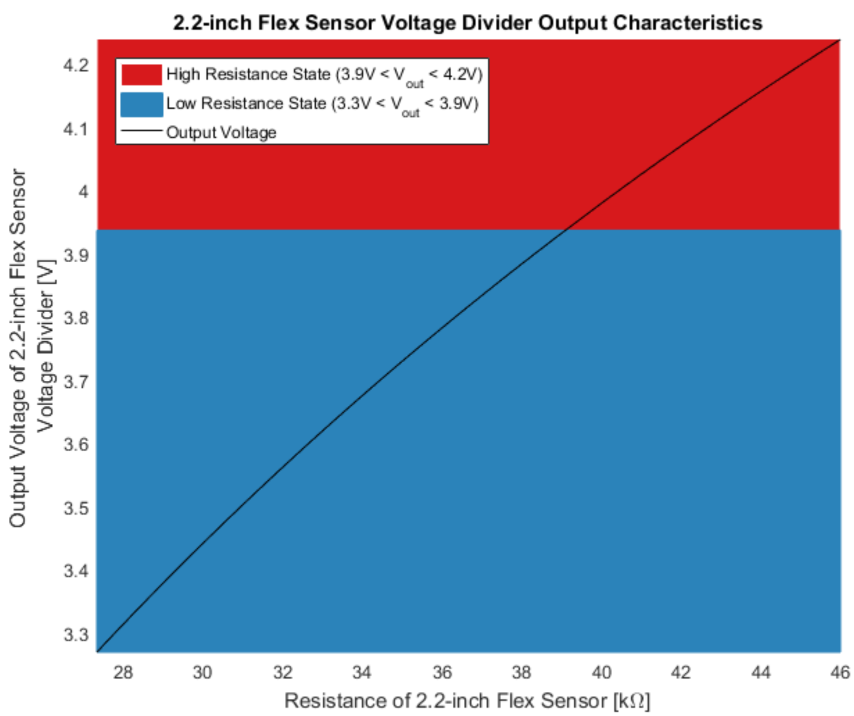

Figure 11. Relationship between output voltage and flex sensor resistance for 4.5-inch flex sensor voltage divider.

Figures 10 and 11 show that we have relatively narrow voltage ranges on the order of hundreds of millivolts. For this reason, we need an ADC capable of detecting voltage variations on the order of tens of millivolts. Equation 5 defines the voltage resolution, V_{RES}, of an ADC where the reference voltage, V_{REF}, will be $V_{DD,FLEX} = 7.5V$. When we solve this equation for N as in Equation 6, the number of bits output by the ADC, we find that we need an ADC that represents voltages with at least ten bits. The MCP3008 from Microchip provides us with the ten bits of resolution that we need as well as a multiplexing capability to support all five flex sensor voltage divider circuits. Additionally, the MCP3008 has an SPI serial interface which is supported by our microprocessor. For these reasons, we determined that the MCP3008 was a suitable choose for the ADC of the flex sensor module.

$$\frac{V_{REF}}{2^N} = V_{RES} \tag{5}$$

$$N = \log_2 \frac{V_{REF}}{V_{RES}} = \log_2 \frac{7.5\text{V}}{0.010V} = 9.55 < 10 \tag{6}$$

As a final, practical step, we determined suitable physical implementations for the resistors. The input voltage we selected allows us to calculate a worst-case power dissipation below 1.2mW for all dividing resistors. Additionally, Figure 9 shows that the voltage ranges do not change significantly with small changes in the value of the dividing resistor. For these reasons, we see that we can implement the 4.5-inch flex sensor dividing resistors using Panasonic 19.1kΩ thick film resistors and the 2.2-inch flex sensor dividing resistors using KOA Speer 34.4kΩ thin film resistors. Both types of resistors have 1% tolerances, 250mW power ratings, and 1206 surface mount packages, so their specifications satisfy our requirements [14], [15].

In the end, this module receives 5V and 7.5V voltage supplies from the power converter module and communicates how bent the user's fingers are to the microprocessor over an SPI interface. This subsystem is critical to the success of the overall system because without reliable data on the extent to which each of the user's fingers are bent each letter of the ASL alphabet will not be able to be uniquely identified with ninety-five percent correctness.

2.2.2 Continuity Sensors

To enable the system to determine how the user's fingers are grouped together, we have strategically placed continuity sensors across the glove as can be seen in the physical diagram of Figure 2. These sensors accomplish several specific functions. Firstly, these sensors determine if adjacent fingers are in lateral contact with one another. Secondly, these sensors determine if the tip of the thumb is in contact with the inside of the pinky finger, with the inside of the ring finger, or with the tip of the index finger. Finally, these sensors determine if the underside of the middle finger is in contact with the top of the index finger or the top of the thumb. In doing so, the continuity sensors serve a critical role by providing the information that allows the system to differentiate between similar signs. Without these sensors and the information they provide, the system would not be able to translate the letters of the ASL alphabet into the English alphabet with ninety-five percent correctness. Furthermore, these sensors receive power from the power supply module and transmit the information they collect to the microprocessor across a digital input pin.

Each continuity sensor will be constructed by sewing patches of conductive fabric at the locations shown in Figure 2. The patches represented by red rectangles in Figure 2 will be connected to 3.3V from the power supply module through a 200kΩ resistor. These patches will serve as the high sides of the continuity sensors. The patches represented by black rectangles in Figure 2 will be connected to ground. These patches will be referred to as the low sides of the continuity sensors. Figure 12 shows the circuit model for each continuity sensor. As shown in Figure 12, we represent contact between the two sides of the continuity sensor by a single pole single throw (SPST) switch. When contact is made between the two sides of a continuity sensor, the switch is closed and the output is connected to ground. Conversely, when the high side of a continuity sensor is not in contact with a patch connected to ground, the circuit is effectively open. As a result, the voltage at the output node will be near 3.3V.

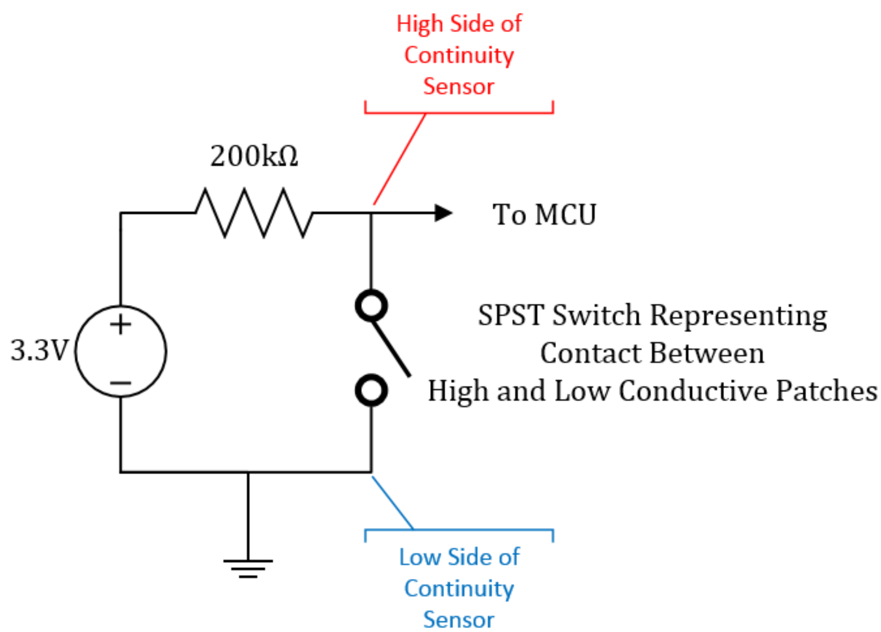

Figure 12. Continuity Sensor Circuit Model.

In reality, opening the switch in Figure 12 will not open the circuit because the input impedance of the microprocessor pins is not infinite. According to the data sheet for the ARM LPC1114 microprocessor, the input impedance of the digital pins is 2.5MΩ. Thus, the output of the continuity sensor is the output of a voltage divider with the 200kΩ resistor as the dividing resistor and the input pin as the output resistor. Additionally, the data sheet for the microprocessor also states the logical high inputs are those inputs whose voltages rise above 3V[16]. From these facts, we knew that we needed to select the resistor between the 3.3V supply and the high side of the continuity sensor to force the output above 3V when the switch in Figure 12 is open. By applying Equation 2, we found that we needed a resistance below 250kΩ. Realizing this constraint, we sought to maximize the resistance to conserve power, provide a reasonable resistance margin to ensure correct functionality, and minimize the component cost. In response, we decided to implement the resistor using a 3₵ Panasonic 200kΩ thick film resistor with a 1% tolerance and a 250mW power rating in a 1206 surface mount package [17].

2.2.3 Accelerometers

The purpose of the accelerometers is to detect both the orientation of the palm and the orientation of the middle digit of the index finger with respect to the surface of the earth. This information allows the system to differentiate between signs which are rotations of one another and to determine when the hand is in motion. Without these accelerometers, the system would not be able to differentiate between several sets of letters. Thus, the system would be incapable of meeting the requirement that it translate all twenty-six letters of the ASL alphabet into the English alphabet with ninety-five percent correctness.

To implement the accelerometers, we have selected the MPU6050 3-axis accelerometer from InvenSense. The MPU6050 offers the competitive cost of many other accelerometers with the added benefit of a 3-axis gyroscope that can be used to filter the accelerometer data and provide a more accurate description of the orientation of the hand. Additionally, the MPU6050 also contains an onboard ADC that allows it to communicate with the microcontroller over an I^2C interface. This feature simplifies the implementation of the accelerometers module and prevents us from using all the analog inputs of the microcontroller and having to purchase additional ADCs. Furthermore, the MPU6050 also allows for a programmatically configured gyroscope and accelerometer range that can be used to optimize the resolution of the measurements to our application [18].

Because our classification algorithm relies on determining the orientation of the hand and the middle digit of the index finger with respect to the surface of the earth, our application does not require measuring a large range of accelerations. For this reason, we can configure the MPU6050 to only measure a reduced range of accelerations. By doing so, we can increase the resolution of the measurements. Specifically, by selecting the measurable acceleration range to be ±2.0g, we can use all 16-bits of the MPU6050's onboard ADC to quantize the measurement. This yields a sensitivity of 0.000061g per quantization level. By making these design choices, we reduce the signal-to-noise ratio (SNR) of the accelerometer data and allow the system to receive more reliable data.

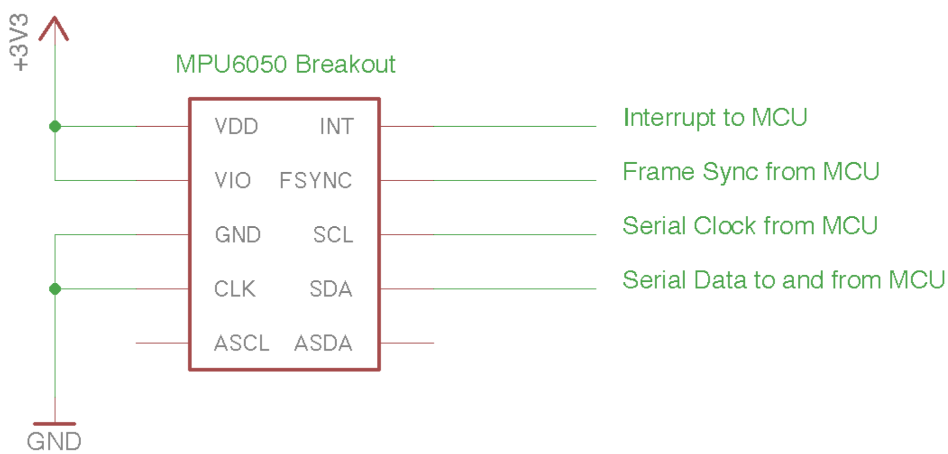

Figure 13. Schematic of MPU6050 breakout board.

2.3 Microcontroller

The LPC1114 microcontroller takes in a 3.3V input at the VDD pin. Analog inputs to the microcontroller are I^2C communication with the two accelerometers and serial communication with the flex sensor ADC. Digital inputs are given from the continuity sensors. The outputs of the microcontroller are the status LEDs and serial communication with the speech synthesis module. The microcontroller will read data from the sensors and attempt to interpret the sensor data to yield a map between the hand sign to a letter. This characterization will be done several times during a specified interval to insure accuracy of interpreted gesture. Once there is sufficient confidence in the accuracy of the interpretation, the microcontroller will communicate over serial with the speech synthesis module.

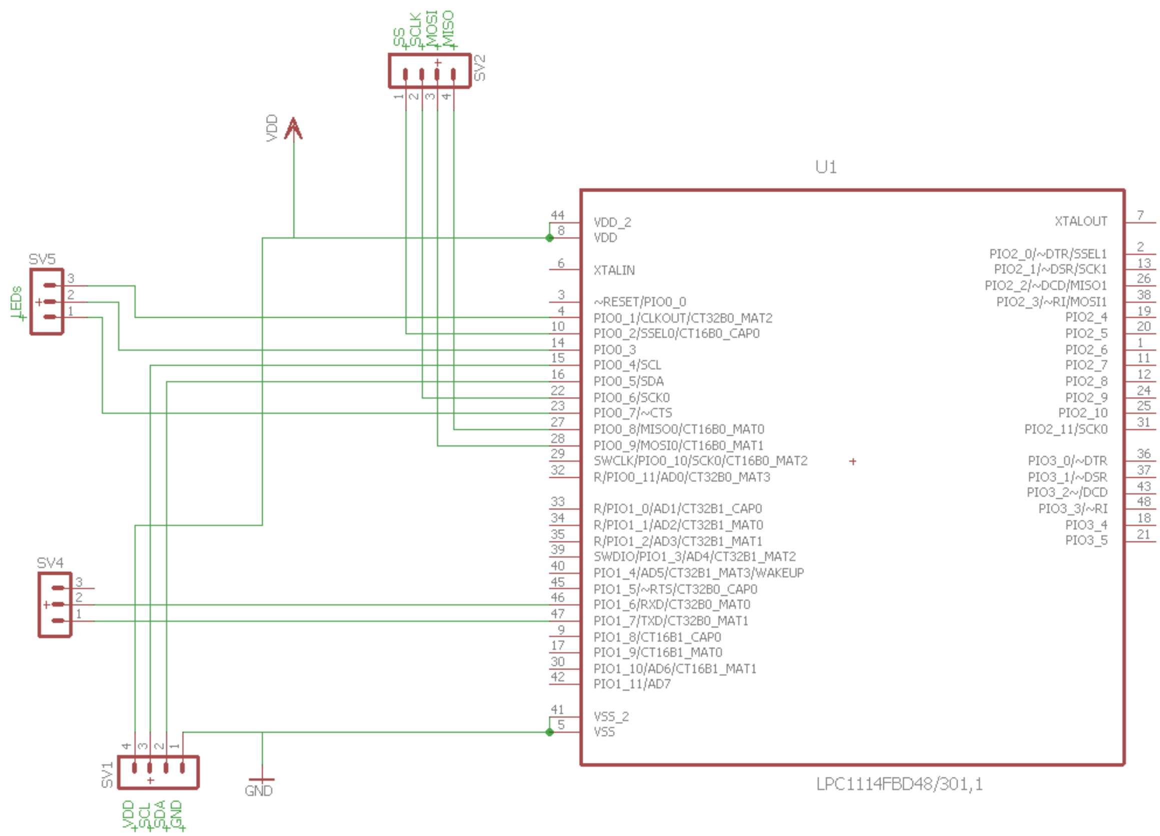

Figure 14. Schematic for microcontroller.

2.4 Power

The power module will provide power to the rest of the modules. Voltage from the battery will be converted to the voltage levels necessary for the microcontroller, sensors, and output modules.

2.4.1 Battery

The power for the microcontroller, sensors, and the output modules will be powered by a single 9V alkaline battery. Based on preliminary research, it appears that a single 9V battery will provide six hours of battery life.

2.4.2 Power Converter

The voltage supplied by the 9V battery will be stepped down to the 3.3V, 5V and the 7.5V required by the sensors, the microcontroller, and the speaker. Two buck converters will be used to step down from the battery voltage to 3.3V and 5V and we require that these converters have minimal voltage ripple in order to avoid supplying the sensors and the microcontroller with voltages above their maximally allowed voltage. Integrated switches were used to minimize the space taken by the power converter and minimize switching noise. Figure 18 shows the expected output voltage from the 3.3V and 5V power converters respectively. The simulations show the voltage ripple to be well within the ±0.1V tolerance. Passive components used in the 3.3V and 5V power converter were based on the manufacturer's application circuit in the datasheet [19]. Linear regulator was chosen to supply a fixed voltage to the flex sensors due to the sensitivity of the flex sensor data to input voltage and to maximize efficiency by avoiding the switching losses associated with buck converters.

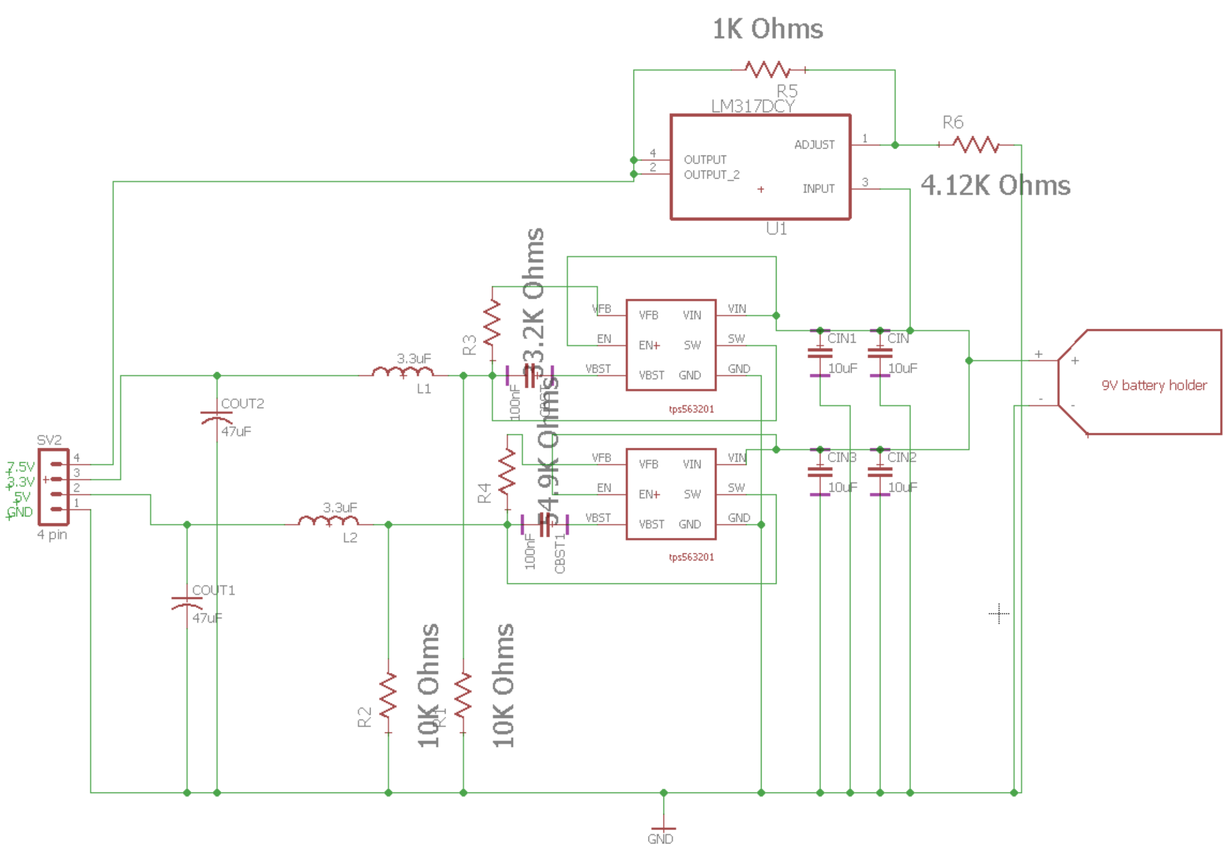

Figure 15. 3.3V, 5V, and 7.5V power supply schematic.

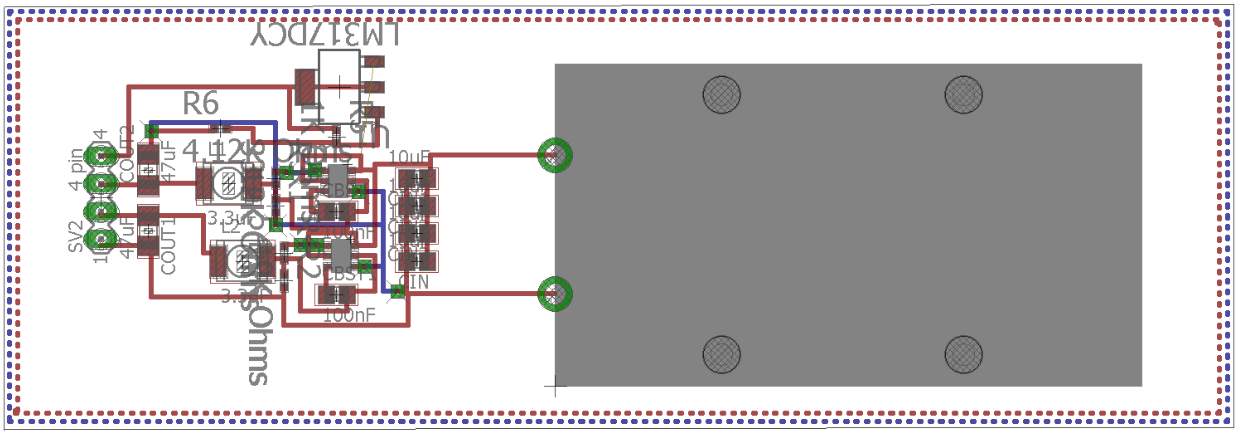

Figure 16. Power supply PCB design.

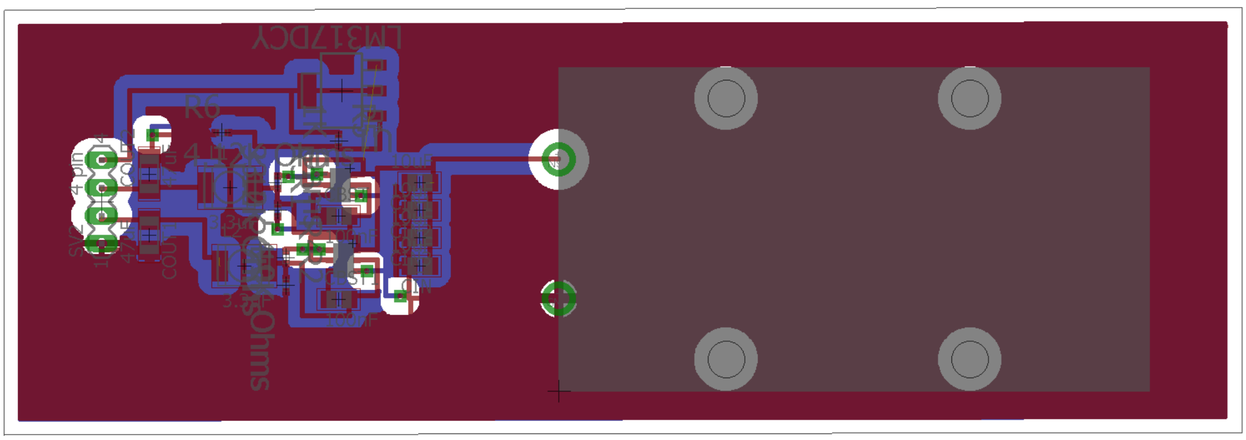

Figure 17. Power supply PCB design with ground planes.

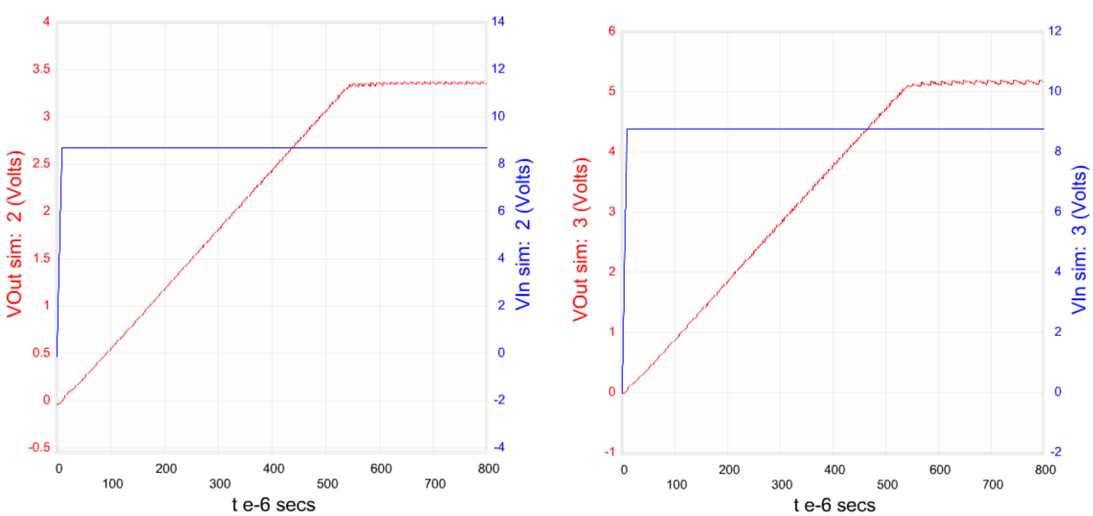

Figure 18. 3.3V and 5V power supply simulations.

2.4.3 Power Consumption

- The LPC1114 operates at 3.3V and draws 9mA under normal operating conditions. The input voltage range is 1.8V to 3.6V. The expected power draw is 29mW.
- The continuity sensors will operate at a 3.3V voltage and draw 2mA. The expected power consumption is 7mW.
- The output module will contain three status LEDs. The LEDs and current limiting resistor will be supplied power from the 3.3 V supply, and each draw 260μA. The total power drawn should be 2.6mW.
- The MPU6050 draws 3.9 mA at 5V, thus the expected power consumption is 20mW.
- The speakjet speech synthesizer module operates at 5V and draws 25mA. The expected maximum power draw is 125mW.
- The expected speaker power draw is .5W. The speaker will operate at a voltage of 5V, thus maximum current raw should be .1A
- The flex sensors will operate from the 7.5V power supply. Since the expected maximum resistance as seen by the power supply is roughly 12kΩ when all fingers are curled, the expected power draw is 5mW.
- The 3.3 V power supply is expected to operate at 60% efficiency based on datasheet efficiency graphs.
- The 5V power supply is expected to operate at 85%-90% efficiency based on datasheet efficiency graphs.
- The theoretical efficiency of the 7.5V power supply is 83%.
- The total expected power draw when the speakers are operating is .821W and .24W when the speakers are idle.

A typical 9V battery has a rated capacity of capacity 500mAh when the current draw is 100mA or 600mAh when the current draw is 25mA. Assuming the speaker is continuously drawing maximum power, the expected battery life is expected to be 2.6 hours, under normal operating conditions, the expected battery life is 8.2 hours.

2.5 Output

The output module will audibly announce the letter being signed in English and will provide the user with feedback on the interpreter's operation.

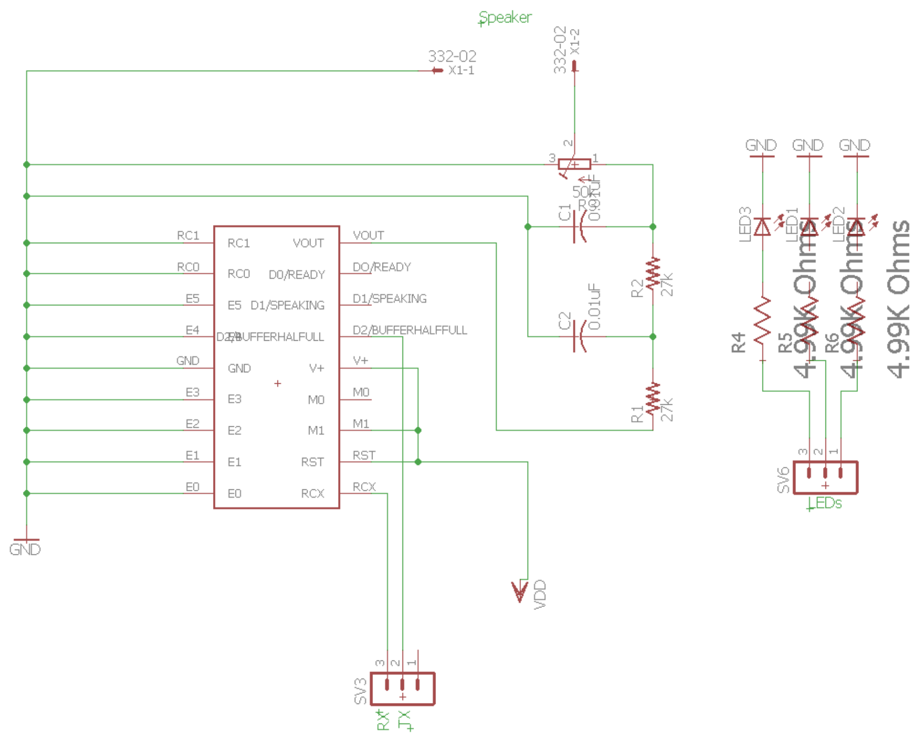

Figure 19. Output module schematic.

2.5.1 Speaker

Once the sign is interpreted by the microcontroller, the microcontroller will communicate with the speaker module. We anticipate using the speakjet chip to convert the ASCII string data generated by the microprocessor to an analog voice signal read by the speaker. The microcontroller will communicate with the speakjet chip via serial communication. The output of the speakjet will go to the speaker which must be able to output a volume of at least 50db for normal room conversation. The schematic in Figure 19 is based on the application circuit in the manufacturer's datasheet [20].

2.5.2 Status LEDs

The microcontroller will also use its digital outputs to switch on status LEDs for power and hand sign reading. The status LEDs will inform the user of the current operations of the microcontroller such as whether the device is on or if there was an error in interpreting the hand sign.

2.6 Recognition Algorithm

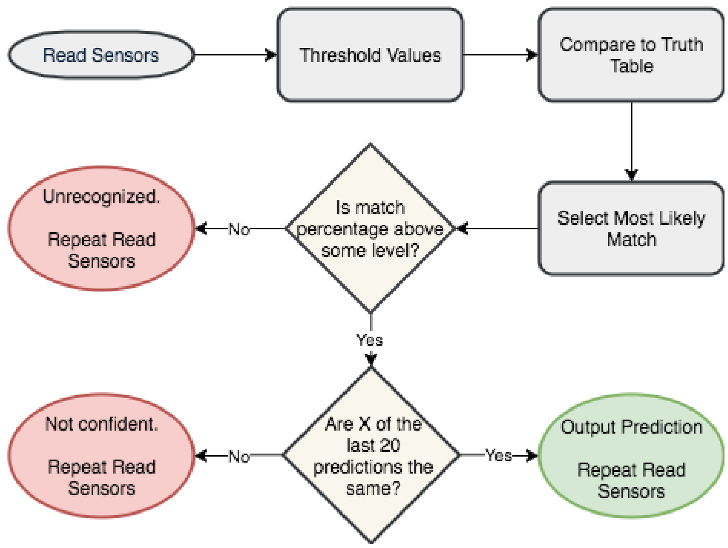

Figure 20. High level flow of recognition algorithm.

The core of our proposed character recognition algorithm is the comparison of sensor inputs to a predetermined truth table. The truth table includes discrete values for all onboard sensors. Truth table values will be precomputed and determined based on data collected from our initial prototype. As we refine the physical implementation of our design, truth table values will be updated the reflect the most accurate state. A preliminary truth table based on our initial sensor experiments can be found in Appendix A1, A2, and A3.

The recognition loop begins by thresholding the current raw inputs of each sensor. The thresholding steps converts the quantized data of the flex and accelerometer sensors into just a few discrete states. For example, in the case of the index finger flex sensor, the reading will be grouped into one of three clusters. As previously discussed, Figure 21 illustrates the clustering of our preliminary sample data.

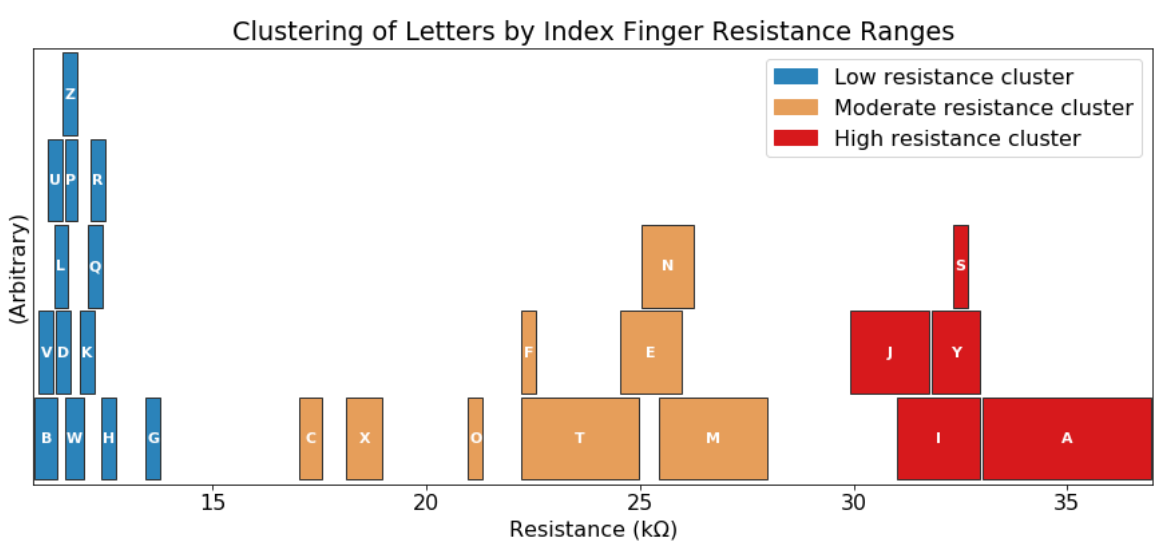

Figure 21. Clustering of letters by index finger flex sensor resistance. (Same as Figure 6. Repeated for clarity)

After thresholding, the input sensor vector can be compared to the truth table. The most likely character will be selected based on the Manhattan distance between the input vector and the known character vectors. Our preliminary truth table has been designed such that any combination of characters has a hamming distance of 2. Thus, we set a maximum distance for a character to be recognized. Input vectors exceeding the maximum distance will be considered unrecognized and the process recognition algorithm repeats.

To prevent premature recognition of characters, the last 20 predictions are saved for comparison. A prediction is only output if some percentage of the previous 20 predictions agree. Specific values will be determined experimentally when the final prototype is built. This final check serves as a prediction debounce, filtering potentially erroneous characters.

To increase the robustness of our system, we will implement a calibration procedure. Differences in hand size, shape and individual 'accent' cannot be accounted for entirely in our hardware and circuit design. Primarily, the flex sensors will produce inconsistent measurements between different users. However, because of our thresholding process, we do not care about the raw measurement, instead just the discrete state after thresholding. We propose to calibrate the threshold cutoffs so that letters are clustered similarly across all users.

Using the output module, we will audibly instruct the user to fingerspell specific letters of the alphabet. By choosing letters that exist on the boundaries of each cluster, we can tune the threshold cutoffs to each user. Again, consider Figure 21, by prompting the user to fingerspell 'G', 'C', 'M', and 'J', we can establish the appropriate threshold boundaries between each cluster on the index finger. As we collect more data, we will refine the set of boundary letters to most accurately calibrate the system.

2.7 Tolerance Analysis

For this system to be fully functional, reliable data on the extent to which the user's fingers and thumb are bent must be gathered from the flex sensors module. Specifically, this data must be precise enough to allow the microprocessor to discern whether each flex sensor is in the high, moderate, or low resistance state. To ensure that this is the case, we require that the flex sensor circuits produce normalized voltages that differ from the normalized threshold voltages by more than the normalized resolution of the ADC when the flex sensors are in the high, moderate, and low resistance states. Here, we refer to normalization by the input voltage, $V_{DD,FLEX}$. It is important to note that it is the normalized voltage that is important in this case because any change in output voltage due to a change in the supply voltage will be compensated for by the ADC, which receives the supply voltage as its reference voltage. Analysis of Figures 3, 6, and 7 and Equation 1 reveals the parameters that determine whether the flex sensors module meets this requirement.

Specifically, Equation 1 shows that meeting this requirement depends on having a sufficiently low tolerance in the dividing resistor, R_d. Should Rd be too much higher than expected, a high or moderate state resistance may be interpreted as a moderate or low state resistance, respectively. Similarly, if R_d is too much lower than expected, a low or moderate state resistance may be interpreted as a moderate or high state resistance, respectively. To ensure that these misclassifications will not occur, we must observe that the worst-case resistance in each resistance state does not cause the normalized output voltage to cross the normalized threshold voltage when R_d goes from its nominal value to its worst-case values.

To evaluate the integrity of the normalized thresholds, we can evaluate Equation 7 for the values of R_{Flex} shown in Figures 6 and 7 closest to the thresholds when R_d takes its worst-case values. The tolerance of the dividing resistors in the flex sensors module is 1%, so we must evaluate R_d at 99% and 101% of it's nominal value. The results are shown in Tables 2 and 3. By comparing the results in Tables 2 and 3 to the normalized threshold voltages in Table 4, we see that the normalized output voltages do not cross the normalized threshold values when the dividing resistances goes from its nominal value to its worst-case values. Additionally, we also note that the difference between the normalized threshold voltages and the normalized output voltages is larger than the normalized resolution of the ADC, 0.001. For this reason, we can conclude that we have selected dividing resistors with small enough tolerances to allow our system to function in worst case conditions.

$$\frac{V_{OUT}}{V_{DD,FLEX}} = \frac{R_{Flex}}{R_{Flex}+R_d} \tag{7}$$

Flex Sensor Resistance, R_{Flex}	Nominal, Normalized Output Voltage	Worst Case, Normalized Output Voltage
Least Resistance in High Resistance State	0.610	0.608
Greatest Resistance in Moderate Resistance State	0.594	0.597
Least Resistance in Moderate Resistance State	0.471	0.468
Greatest Resistance in Low Resistance State	0.419	0.421

Table 2. 4.5-inch Flex Sensor Normalized Output Voltages.

Flex Sensor Resistance, R_{Flex}	Nominal, Normalized Output Voltage	Worst Case, Normalized Output Voltage
Least Resistance in High Resistance State	0.531	0.528
Greatest Resistance in Low Resistance State	0.519	0.522

Table 3. 2.2-inch Flex Sensor Normalized Output Voltages.

4.5-inch Flex Sensor Normalized Upper Threshold Voltage	0.602
4.5-inch Flex Sensor Normalized Lower Threshold Voltage	0.446
2.2-inch Flex Sensor Normalized Threshold Voltage	0.525

Table 4. Normalized Threshold Voltages.

Furthermore, this tolerance analysis offers insights into the limiting factors in our project. Fundamental to our analysis is the idea that having more precise dividing resistors would create a larger voltage margin between the output voltage when the flex sensor is at the edge of a resistance state and the threshold voltage. Of course, implementing more precise resistors would increase the cost of the system. Secondly, we also see that we could increase our system's ability to detect small differences between output voltage and a threshold voltage by using a higher resolution ADC. This, of course, would also come at a higher cost.

Thirdly, our analysis reveals that our system has voltage differentials on the order of only 100mV between letters at the respective high and low ends of adjacent resistance states. This means that our system is very susceptible to noise. To remedy this, we could increase the supply voltage for the flex sensor module. This increase would come at the cost of increasing power consumption, cost, and design complexity. Specifically, if the supply voltage to the flex sensor module was increased, we would have to add additional hardware to keep the output voltage below 5V or procure a more expensive ADC capable of handling large input voltages.

Finally, our analysis also reveals that our design would benefit from empirical optimization. For instance, if a dividing resistor is found to have a larger or smaller resistance than expected, the threshold voltages could be adjusted to maintain maximal voltage margins between output voltages at the edge of resistance states and threshold voltages. This optimization could be performed manually during development by hardcoding adjusted threshold values, or it could be implemented in software by adjusting the threshold voltage during the calibration process.

3 Ethics and Safety

The greatest safety concern in our proposed design is the battery. The relatively large energy density of the batteries makes the battery pack susceptible to various thermal and electrical hazards. Sustained skin exposure to temperatures greater than 48˚C can result in third degree burns within 5 minutes [10]. As thermal buildup is an inevitable product of energy storage and discharge, we have designed our battery pack to promote air flow and convective heat dissipation to the environment. To mitigate the risks involved with short circuits, our design includes current limiting circuitry around the battery. Further, we have selected alkaline batteries over more hazardous chemistries like lithium-ion batteries.

As with many wearable technology applications, there is an inherent requirement that numerous electrical components be coupled closely with the user's body. The effects of long term exposure of electronics positioned closely along the body is still an area of open research. In a similar vein to the potential negative effects of prolonged cell phone radiation [11], there is the potential that wearable electronics may be shown to have negative effects for long term users. In accordance with #9 of the IEEE Code of Ethics [12], to avoid potential harm to users of our product, it is important the we maintain a cautionary position on any potential adverse or harmful effects of wearable technology.

In addition to a concern for our user's safety, we also consider the safety of development a main priority. As we move through the development process, there are several potential hazards that may not be present in the final product. For example, while in development, a battery will not be used to power the device. This exposes all those working with our prototypes to a larger potential source of voltage and current. Developer and early user safety is an equally important concern to that of end user safety.

As stated in 1.7 of the ACM Code of Ethics [13], "…communication technology enables the collection and exchange of personal information on a scale unprecedented in the history…", therefore it is of notable concern to fully protect the user's privacy. A potential compromise in the security of the device could allow a malicious actor to obtain a user's entire conversation. It is critical that we preempt such a breach of privacy. As such, our design has prioritized the need for the device to be entirely self-contained. Performing all necessary data processing onboard the device and without maintaining any logs, the potential for a data leak is greatly reduced.

We strive to fulfil guidelines set forth by #3 and #7 of the IEEE Code of Ethics by representing all our technical claims honestly and willingly accepting criticism of our work. We will meaningfully credit all those who make contributions to our project.

4 Requirements and Verifications

Power Module

Requirement	Verification	Points
Battery module must be able to power the device for 4 hours of continuous use.	1. Attach ammeter between battery module and power module 2. Fingerspell the alphabet for 15 minutes 3. Verify the battery's mAh rating is 16 times larger than the average measured current	3
9.0V to 7.5V DC buck converter must supply a DC voltage of 7.5V ±0.1V with a maximum current of 100mA.	1. Attach 75Ω resistor across the converter output as load 2. Attach multimeter across load 3. Ensure the voltage it is within the range 4.9V-5.1V.	3
9.0V to 5.0V DC buck converter must supply a DC voltage of 5.0V ±0.1V with a maximum current of 200mA.	1. Attach 24Ω resistor across the converter output as load 2. Attach multimeter across load 3. Ensure the voltage it is within the range 4.9V-5.1V.	3
9.0V to 3.3V DC buck converter must supply a DC voltage of 3.0V ±0.1V with a maximum current of 100mA.	1. Attach 33Ω resistor across the converter output as load 2. Attach multimeter across load 3. Ensure it is within the range 3.2V-3.4V.	3
		12

Controller Module

Requirement	Verification	Points
Sensor input should deterministically map sensor inputs to character outputs.	1. Set the microcontroller to listen to serial in for data 2. Send a random feature vector over serial 3. Verify output of system is the same on all attempts 4. Repeat for five additional random sensor inputs	1
Controller module recognition algorithm makes character predictions at 15Hz.	1. Monitor the serial out port of the microcontroller 2. While wearing the glove, fingerspell the alphabet 3. Using the microcontroller serial output, verify predictions are made at a minimum of 15Hz	3
Character must be recognized within 1 second of completing the entire gesture.	1. While wearing the glove, fingerspell the entire alphabet 2. Using a timer, measure the latency between gesture and output 3. Ensure the latency never exceeds 1 second	3

Repeated characters must be recognized within 3 seconds of the previous character's identification	1. While wearing the glove, fingerspell the entire alphabet 2. Immediately after recognition, fingerspell the character again 3. Ensure the latency never exceeds 3 second	3
Entire controller module should draw less than 40mW.	1. Attach ammeter between power module and controller module 2. Fingerspell the alphabet 3. Ensure no more than 12mA is drawn on average	2
Classification of characters should, in total, be 95% accurate.	1. While wearing the glove, fingerspell the entire alphabet five times 2. Ensure there are no more than six misidentifications.	5
		17

Sensor Module

Requirement	Verification	Points
The sensor subsystem must produce a unique feature vector after quantization for each letter of the ASL alphabet.	1. Monitor the serial out port of the microcontroller 2. While wearing the glove, fingerspell the entire alphabet 3. Verify every letter produces a unique feature vector	5
The worst case power dissipation in each continuity sensor must be less than 1mW	1. Attach an ammeter to the input of the continuity sensor 2. Touch the positive side to the negative side 3. Ensure no more than 0.33mA flows through the circuit 4.Repeat for the five remaining continuity sensors.	3
Each flex sensor circuit must produce voltages that differ from the threshold voltages by more than the resolution of the ADC when the flex sensors are in the high, moderate, and low resistance states.	1. Attach a multimeter to the output of the flex sensor circuit 2. Sign the letter that produces a flex sensor resistance nearest the threshold resistance in the high resistance state. 3. Measure the voltage. 4. Repeat for the moderate and low resistance states. 5. Ensure there is minimum of 0.01V difference between each measured voltage and the threshold voltages. 6. Repeat for the four remaining flex sensors.	3
Entire sensor module should draw less than 70mW.	1. Attach ammeter between power module and the flex sensor array 2. Fingerspell the alphabet 3. Repeat for continuity and accelerometer sensors 4. Verfiy total power consumption is less than 70mW	3

The standard deviation of each output of the accelerometer is less than 0.5G	1. Place the accelerometer flat on a table 2. Monitor the serial out port on the microcontroller 3. Print out all quantized values of the accelerometer over serial 4. Record data for 1 minute 5. Ensure the standard deviation of each output is less than 0.5G	1
		15

Output Module

Requirement	Verification	Points
The speaker must be able to output 50db of audio from a distance of one meter.	1. Place an audio noise meter capable of measuring decibels 1 meter from the glove 2. Fingerspell a letter 3. Verify at least 50db was measured	2
Entire output module should draw less than 700mW.	1. Attach ammeter between power module and output module 2. Fingerspell the alphabet 3. Ensure no more than 140mA is drawn on average	2
The status LEDs must be bright enough to be comfortably seen from a meter away	1. Stand one meter from glove in a well light room 2. Verify all status LEDs are clearly visible.	2
		6
	Total Points	**50**

5 Schedule

Week	Nick	Tim	Mike
6-Feb	- Select and order parts - Write proposal	- Select and order parts - Write proposal	- Select and order parts - Write proposal
13-Feb	- Finalize sensor schematic - Preliminary flex sensor resistance analysis	- Finalize power schematic - Order output module parts	- Preliminary flex sensor thresholding analysis
20-Feb	- Begin construction of working prototype - Design sensor module PCB	- Test power module - Experiment with output module	- Begin construction of working prototype - Test continuity sensors
27-Feb	- Organize collection of initial data set	- Finalize power and output module design	- Refine truth table
6-Mar	- Design sensor module v2	- Test basic microcontroller functionality - Design power module v2	- Analysis of initial dataset
13-Mar	- Test and debug sensor module	- Design output module v2 - Order final PCBs	- Begin recognition algorithm
20-Mar	- Spring Break	- Spring Break	- Spring Break
27-Mar	- Test power efficiency of system	- Test and debug power and output module	- Prototype wrist mount
3-Apr	- Test character classification in real world use cases	- Debug recognition algorithm	- Finalize wrist mount - Debug recognition algorithm
10-Apr	- Test and debug entire system	- Test and debug entire system	- Test and debug entire system
17-Apr	- Write final paper	- Write final paper	- Write final paper
24-Apr	- Present project	- Present project	- Present project
1-May	- Checkout of lab	- Checkout of lab	- Checkout of lab

6 Cost Analysis

PARTS				
Part Name	**Distributor**	**Unit Cost**	**Quantity**	**Total**
Buck Converter	Mouser	$3.17	2	$6.34
9v Alkaline Battery	Mouser	$2.37	1	$2.37
Various Passive Elements	DigiKey	$5.00	1	$5.00
PCBs	PCB Way	$10.00	3	$30.00
			Power Module	**$43.71**
Microcontroller	Chip1Stop	$4.74	1	$4.74
Various Passive Elements	DigiKey	$5.00	1	$5.00
PCBs	PCB Way	$10.00	3	$30.00
			Controller Module	**$39.74**
Speaker	Sparkfun	$1.95	1	$1.95
SpeakJet	Sparkfun	$24.95	1	$24.95
Various Passive Elements	DigiKey	$5.00	1	$5.00
PCBs	PCB Way	$10.00	3	$30.00
			Output Module	**$61.90**
Flex Sensor (4.5")	Sparkfun	$12.95	4	$51.80
Flex Sensor (2.5")	Sparkfun	$7.95	1	$7.95
Accelerometer	Sparkfun	$9.95	2	$19.90
Conductive Thread	Sparkfun	$3.95	1	$3.95
10-bit ADC	Adafruit	$3.75	1	$3.75
Various Passive Elements	DigiKey	$5.00	1	$5.00
PCBs	PCB Way	$10.00	3	$30.00
			Sensors Module	**$122.35**
Glove	Amazon	$12.99	1	$12.99
			Miscellaneous	**$12.99**
			PARTS TOTAL	**$267.70**

LABOR				
Team Member	**Hourly Rate**	**Total Hours**	**Expense Multiplier**	**Total Cost**
Nick DeNardo	$33.50	160	2.5	$13,400.00
Tim Wong	$33.50	160	2.5	$13,400.00
Mike Genovese	$33.50	160	2.5	$13,400.00
			LABOR TOTAL	**$40,200.00**
			GRAND TOTAL	**$40,467.70**

References

[1] Lane, Hoffmeister, and Bahan. *A journey into the deaf-world.* San Diego, CA.: DawnSignPress, 1996.

[2] T.W. Deacon. *The symbolic species: The co-evolution of language and the brain*. New York, NY: W.W. Norton, 1997.

[3] G. Satell. (2015). *Why Communication Is Today's Most Important Skill* [Online]. Available FTP: www.forbes.com Directory: sites/gregsatell/2015/02/06/why-communication-is-todays-most-important-skill/#24cccaad3638

[4] (2015). *Deafness and hearing loss* [Online]. Available FTP: www.who.int Directory: mediacentre/factsheets/fs300/en/

[5] (2015). *American Sign Language* [Online]. Available FTP: www.nidcd.nih.gov Directory: health/american-sign-language#1

[6] J. Holt-Lunstad, T.B. Smith, J.B. Layton, “Social Relationships and Mortality Risk: A Meta-analytic Review”, *PLoS Med.,* 7(7): e1000316. doi:10.1371/journal.pmed.1000316

[7] Weisel, Amanda. *Issues Unresolved: New Perspectives on Language and Deaf Education.* Washington, DC: Gallaudet University Press, 1998.

[9] (2016). *UW undergraduate team wins $10,000 Lemelson-MIT Student Prize for gloves that translate sign language* [Online]. Available FTP www.slate.com Directory: blogs/future_tense/2015/01/06/tencent_s_wechat_worldwide_internet_users_are_voluntarily_submitting_to.html

[10] *Cabinet for Health and Family Services. Commonwealth of Kentucky.*, 2013. [Online]. Available: http://chfs.ky.gov/NR/rdonlyres/2CFE26ED-59A5-4F60-91A7-E2B5C0E9D000/0/BurnsintheElderly091213.pdf. Accessed: Feb. 3, 2017.

[11] "Cell phones and cancer risk," National Cancer Institute, 2016. [Online]. Available: https://www.cancer.gov/about-cancer/causes-prevention/risk/radiation/cell-phones-fact-sheet. Accessed: Feb. 6, 2017.

[12] "IEEE Code of Ethics," IEEE, 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. Accessed: Feb. 6, 2017.

[13] "Code of Ethics," ACM, 2016. [Online]. Available: http://ethics.acm.org/code-of-ethics. Accessed: Feb. 6, 2017.

[14] "ERJ-8ENF1912V Panasonic," Mouser Electronics, 2017. [Online]. Available: http://www.mouser.com/ProductDetail/Panasonic/ERJ-8ENF1912V/?qs=sGAEpiMZZMtlubZbdhIBINbsJb0hdM3zdtaLmoHvH7s%3d. Accessed: Feb. 22, 2017.

[15] "RN73H2BTTD3442F100 KOA Speer," Mouser Electronics, 2017. [Online]. Available: http://www.mouser.com/ProductDetail/KOA-Speer/RN73H2BTTD3442F100/?qs=sGAEpiMZZMu61qfTUdNhG50pnceRs4VswL4ih6WxkFU%3d. Accessed: Feb. 22, 2017.

[16] NXP, "LPC1110/11/12/13/14/15," LPC1114 datasheet, March 2014.

[17] "ERJ-8ENF2003V Panasonic | Mouser Europe," in *Mouser*, Mouser Electronics, 2017. [Online]. Available: http://www.mouser.com/ProductDetail/Panasonic/ERJ-8ENF2003V/?qs=sGAEpiMZZMtlubZbdhIBINbsJb0hdM3zoIr%2fb1JCZdI%3d. Accessed: Feb. 24, 2017.

[18] "MPU-6050 Datasheet," InvenSense. [Online]. Available: https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf. Accessed: Feb. 25, 2017.

[19] "TPS563208," Texas Instruments. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps563201.pdf. Accessed: Feb. 25, 2017.

[20] "speakjet Natural Speech & Complex Sound Synthesizer User's Manual," Sparkfun. [Online]. Available: https://www.sparkfun.com/datasheets/Components/General/speakjet-usermanual.pdf. Accessed: Feb. 25, 2017.

Appendix

A.1 Flex Sensor States

Flex Sensor States					
Letter	**Thumb**	**Index**	**Middle**	**Ring**	**Pinky**
A	0	2	2	2	2
B	1	0	0	0	0
C	0	1	1	1	1
D	0	0	1	1	1
E	1	1	1	1	1
F	0	1	0	0	0
G	0	1	2	2	2
H	0	0	0	2	2
I	1	2	2	2	0
J	1	2	2	2	0
K	0	0	1	2	2
L	0	0	2	2	2
M	1	2	2	2	2
N	1	2	2	2	2
O	0	1	1	1	1
P	0	0	1	2	2
Q	0	0	2	2	2
R	1	0	0	2	2
S	1	2	2	2	2
T	0	2	2	2	2
U	1	0	0	2	2
V	1	0	0	2	2
W	0	0	0	0	1
X	1	1	2	2	2
Y	0	2	2	2	0
Z	1	0	2	2	2

Table 5. Distribution of discrete flex sensor states. Organized by finger. Listed values are after thresholding raw readings. The '0', '1', and '2' states represent the low, moderate, and high resistance states, respectively.

A.2 Continuity Sensor States

Continuity Sensor States						
Letter	**Index**	**Middle**	**Ring**	**Thumb**	**Thumb Tip**	**Back Middle**
A	1	1	1	1	0	0
B	1	1	1	0	0	0
C	1	1	1	0	0	0
D	0	1	1	0	0	0
E	1	1	1	1	0	0
F	0	1	1	0	1	0
G	1	1	1	0	0/1	0
H	1	0	1	0	0/1	0
I	1	1	0	0	1	0
J	1	1	0	0	1	0
K	0	0	1	0	1	0
L	0	1	1	0	0	0
M	1	1	0	0	1	1
N	1	0	1	0	1	1
O	1	1	1	0	1	0
P	0	0	1	0	1	0
Q	1	1	1	0	0/1	0
R	0	0	0/1	0	1	1
S	1	1	1	0	1	0
T	0	1	1	0/1	1	0
U	1	0	1	0	1	0
V	0	0	1	0	1	0
W	0	0	0	0	1	0
X	0	1	1	0	0	0
Y	1	1	0	0	0	0
Z	0	1	1	0	0	0

Table 6. Distribution of continuity sensor states. Organized by finger. Values which depend heavily on user's form or 'accent' are denoted by 0/1. The ‘1’ state is when contact is being made between the two sides of a continuity sensor. The ‘0’ state is when contact is not being made.

A.3 Accelerometer States

Accelerometer States						
Letter	**Index (X)**	**Index (Y)**	**Index (Z)**	**Hand (X)**	**Hand (Y)**	**Hand (Z)**
A	0	-1	0	0	1	0
B	0	1	0	0	1	0
C	0	0	1	0	1	0
D	0	1	0	0	1	0
E	0	0	1	0	1	0
F	0	0	1	0	1	0
G	-1	0	0	-1	0	0
H	-1	0	0	-1	0	0
I	0	-1	0	0	1	0
J	-1	0	0	-1	0	0
K	0	1	0	0	1	0
L	0	1	0	0	1	0
M	0	-1	0	0	1	0
N	0	-1	0	0	1	0
O	0	-1	1	0	1	0
P	0	0	1	0	-1	0
Q	0	-1	0	0	-1	0
R	0	1	0	0	1	0
S	0	-1	0	0	1	0
T	0	-1	1	0	1	0
U	0	1	0	0	1	0
V	0	1	0	0	1	0
W	0	1	0	0	1	0
X	0	0	1	0	1	0
Y	0	-1	0	0	1	0
Z	0	0	1	0	0	1

Table 7. Distribution of accelerometer sensor states. Organized by accelerometer placement and axis. The '-1', '0', and '1' states signify measurements of negative acceleration, nearly no acceleration, and positive acceleration.