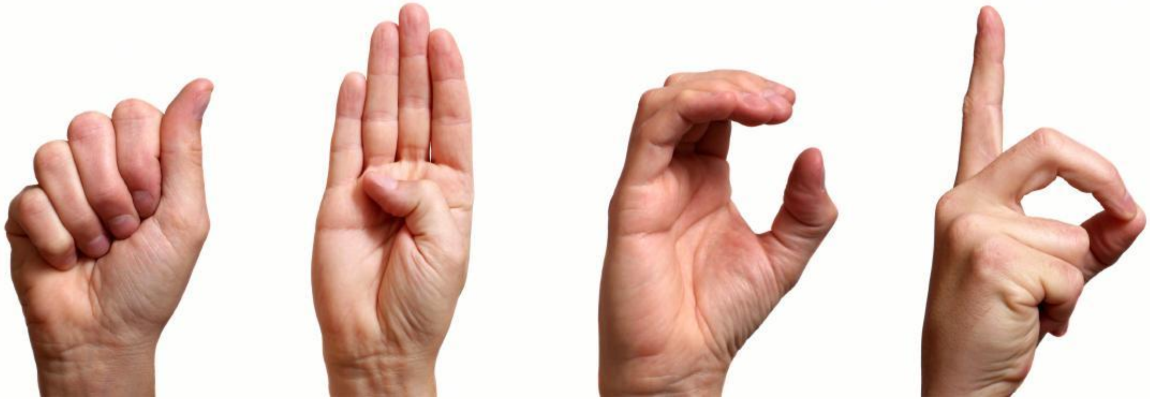
Sign Language Interpreter Glove



Group 24

Christopher Delgado — Electrical Engineering
Emmanuel Hernandez — Electrical Engineering
Jason Balog — Electrical Engineering
Ramon Santana — Electrical Engineering

Thank You to our generous sponsors!!!

Motivation

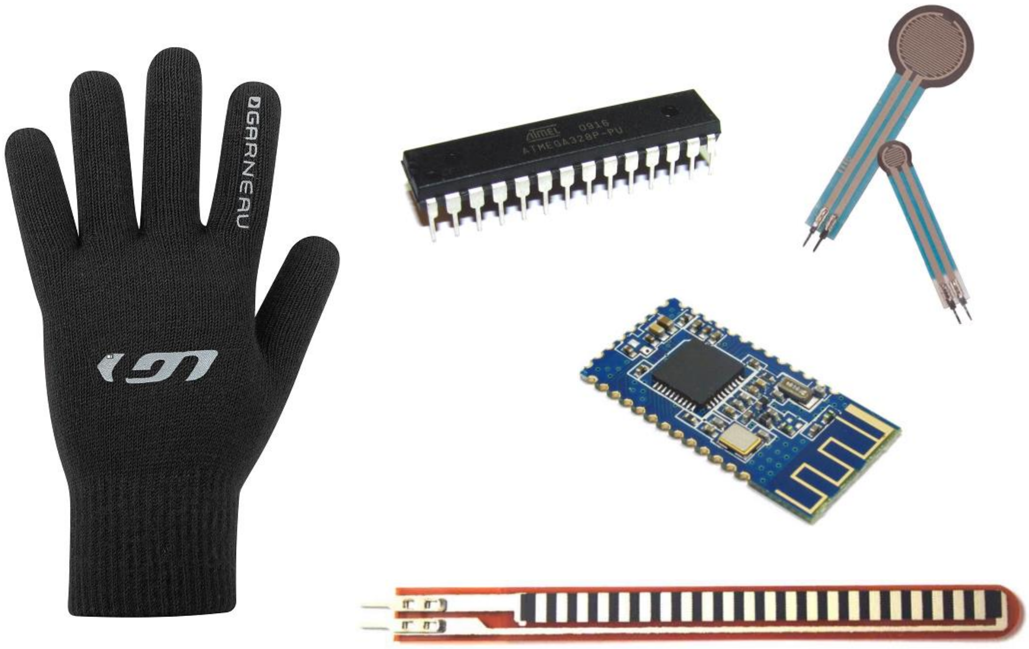
- The Survey of Income and Program Participation (SIPP) – estimates that about 1,000,000 are functionally deaf in the United States.
- The World Health Organization estimates that over 5% of the world's population – 360 million people – has disabling hearing loss (328 million adults and **32 million children**).
- The original motivation to pursue this project comes from one of our team members who has experienced the difficulty of communicating with his speech-impaired sister.

Goals and Objectives

- Our objective is to establish communication between a sign language speaker and a non-sign language speaker. Any letter the user signs will be displayed through a user interface where the non ASL-speaker can read the letter.

- **Hardware**
  - Flex sensors
  - Pressure sensors
  - Accelerometer
  - Gyroscope
  - MCU: ATMega

- **Software**
  - Android Mobile Application



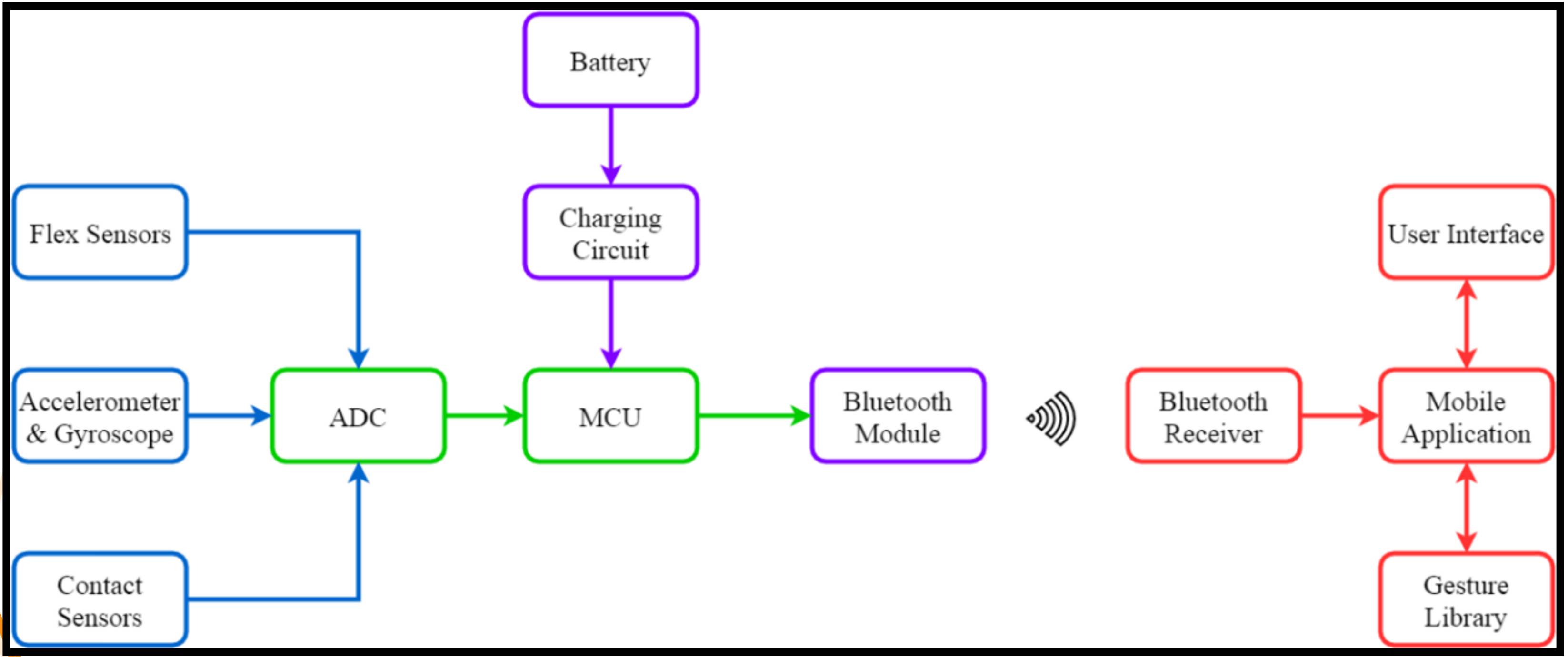
Specifications and Requirements

- Lightweight
- Portable
- Energy Efficient

Color	Very hard decision ☹
Wearing Style	Right handed glove
Dimensions	151mm x 196mm x 33mm
Glove Weight	1.9 lb
Connectivity	Bluetooth Low Energy
Wireless Transmission Range	10 Meters
Operating Temperature	-40 deg. Celsius ~ +85 deg. Celsius
Operating Voltage	+3.3 V and +5.0 V
Battery Type	Lithium-ion
Device Battery Life	24 hours

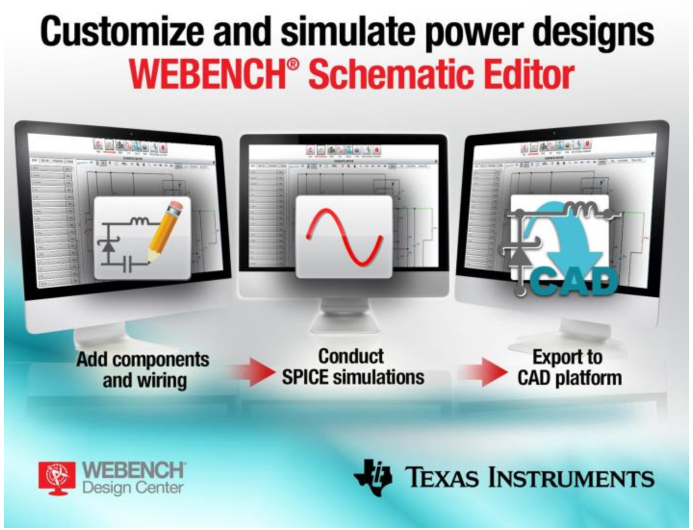
Specifications Table

Block Diagram



TI – Webench Schematic Editor

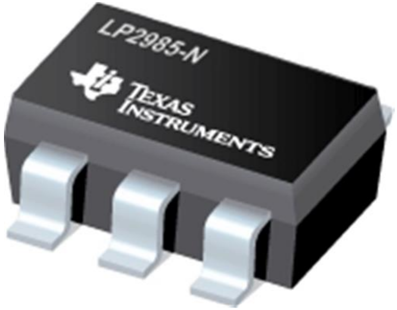
- You can use WEBENCH to create customized power supplies or DC-DC converters for your circuits
- This environment gives you end-to-end power supply designs and prototyping tools
- This alleviates the time and trouble associated with traditional power supply



Significant Hardware Decisions - Battery & Regulators

- **Polymer Lithium Ion Battery - 2000mAh**
  - 3.7V at 2000mAh
  - Built-in protection against over voltage and over current
  - Self-discharge rate <8% per month

- **Regulators - TI LP2985**
  - The purpose of voltage regulators is to keep a constant voltage level
  - Operating range: 1.8V – 10V

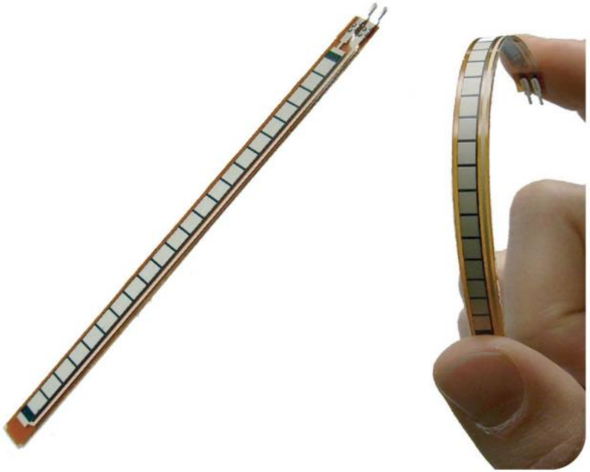


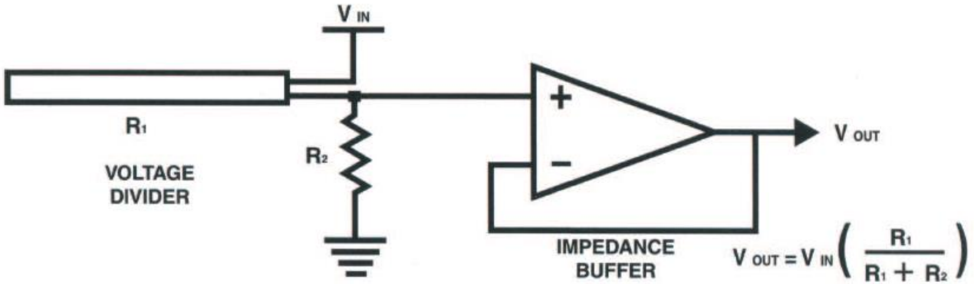
Significant Hardware Decisions - Wireless Communication

WiFi	**Bluetooth Low Energy**	NFC
Pros: - Security (WPA2) - Range 150-300 feet	**Pros:** - Range 20-120 feet - Consumes less power than Wi-Fi	**Pros:** - Consumes less power than Bluetooth Low Energy
Cons: - Needs Router - Consumes lots of Power - Not portable friendly	**Cons:** - No cons for SLIG ☺	**Cons:** - Limited Range (about 0-4 inches)

Significant Hardware Decisions – Flex Sensors

- Flex sensors will be the primary sensors employed in this projects.
- Flex sensors will be used to detect the degree to which each finger is bent on the hand performing the sign language gesture.
- The combination of different degrees of flex for each finger will be the identifying mark for most of the letters.

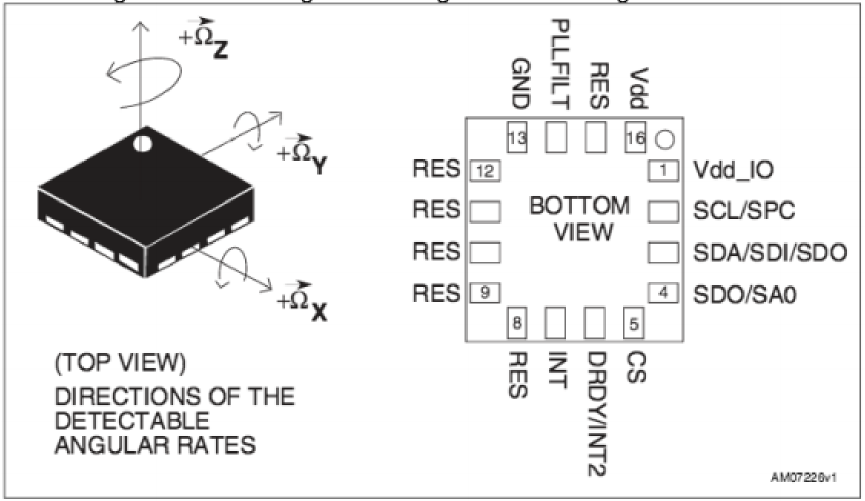




Significant Hardware Decisions – Gyroscope and Accelerometer

- The flex sensors can only detect how much bending they are experiencing and so they neglect factors like tilting the hand back and forth at different angles or the orientation of the hand.
- Instead, accelerometers and gyroscopes can be used to measure these type of parameters, which are crucial in identifying certain sign language letters.
- Examples include "j" and "z" or distinguishing between "g" and "q".

Figure 3.3: Pin Diagram showing measurable angular velocities



Significant Hardware Decisions – Contact Sensors

- Due to the fact there are a few pairs or groups of sign language letters that are not distinguishable by the flex alone, we will need to implement contact sensors.

- These sensors will be helpful if not crucial in telling apart the following pairs of sign language: **R**, **U**, and **V**, **S** and **T** and **M** and **N**.

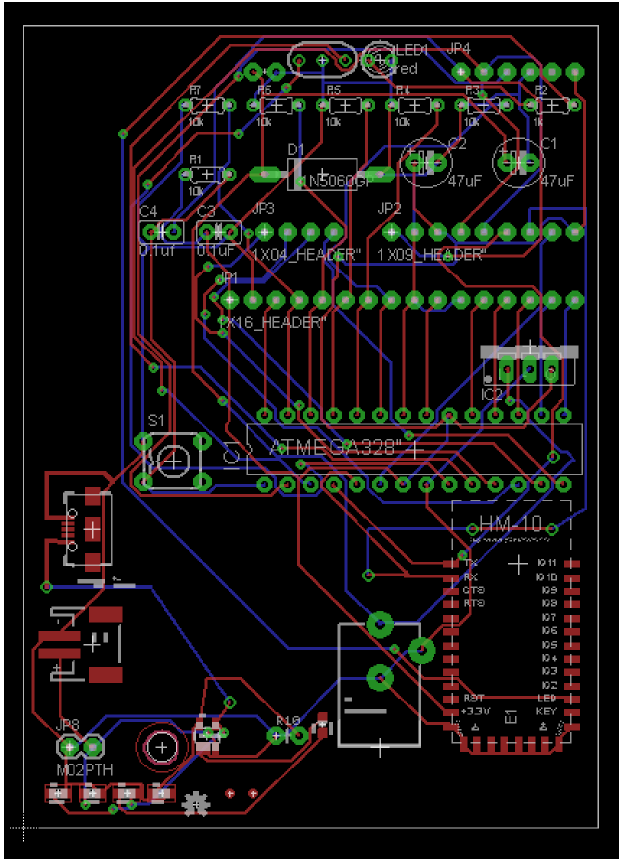
- The contact sensors were made with strips of copper braids that were connected to the power supply through wire leads.

Significant Hardware Decisions – PCB

- The PCB was designed using Eagle CAD software and most parts were sourced from Digikey.com

- The PCB for the SLIG will integrate all the components of the glove including:
  -the MCU
  -the Bluetooth module
  -the various sensors,
  - the power supply circuit.

- It was designed as two layer board measuring 2.5” by 3.5” and should be worn on the forearm.



Significant Hardware Decisions – Microcontroller

- There are many options available to us when choosing an appropriate microcontroller unit for the SLIG.

- The unit that is chosen will have to employ at least 6 analog input pins (with included adc units), have digital i/o pins, and also be able to establish a line of serial communication.

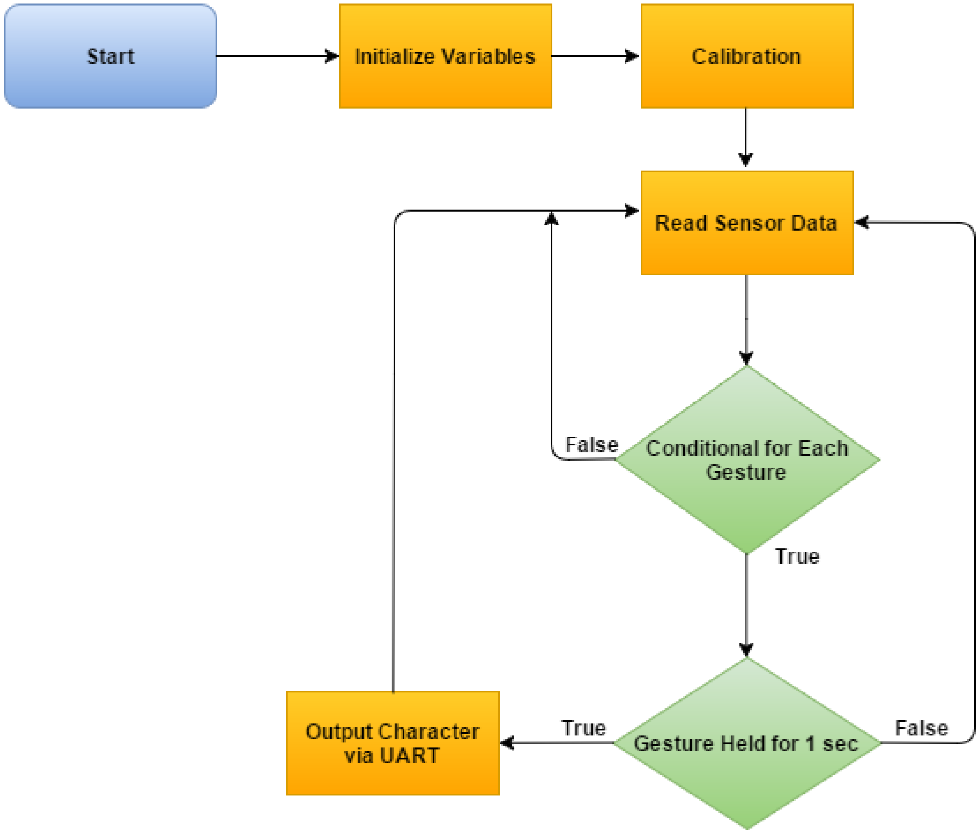
- When considering the microcontroller unit, it is also important to note that the two units come with their own native programming environments.

Table 3.2: MSP430 vs. ATMega32 comparison

Feature	MSP430	ATMega32
Analog Input Pins	8	12
Digital Input Pins	8	20
Random Access Memory	512 Bytes	2.5 Kilobytes
Data Bus	16 bits	8 bits
Speed	16 MHz	16 MHz
Cost/Vendor	$9.99 → mouser.com	$24.95 → ebay.com

Gesture Recognition Software

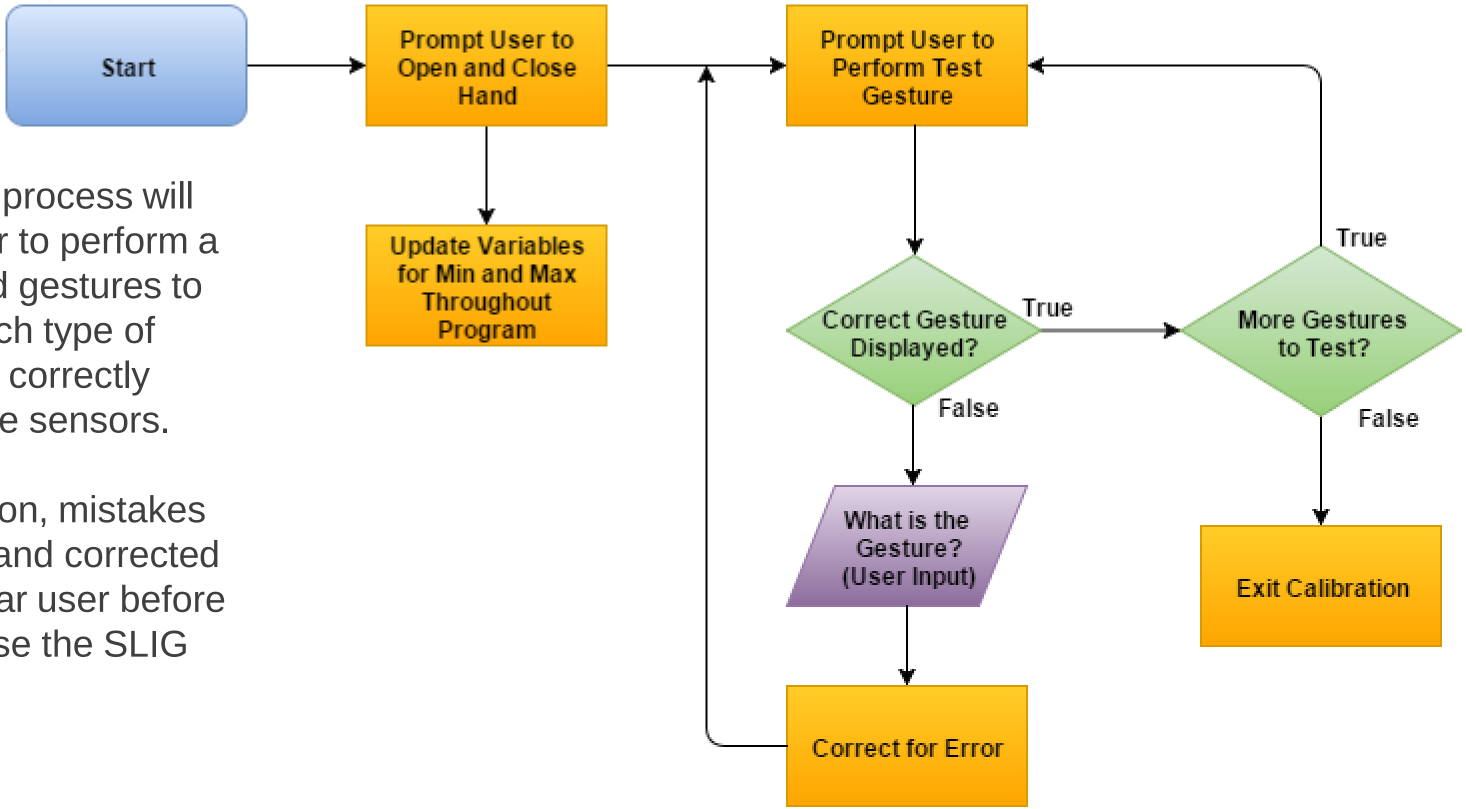
- The gesture recognition system will use thresholds to determine what hand gesture is being performed.
- Through experimentation, the group will determine the range of values that are included in every type of hand gesture and set minimum and maximum limits to the conditional statements accordingly.



Calibration Process

- The calibration process will consist of several tests that occur during the ‘setup’ process of the program, before the microcontroller unit goes into its infinite loop and begins to constantly read the sensor data.
- To keep the SLIG as easy-to-use as possible for the user, the group will try to use only a few gestures during the calibration process that capture all of the different possible hand positions. If this proves ineffective and allows for errors, then the calibration process will be altered to require all possible hand gestures be performed during calibration.
- Calibration will test hand gesture recognition as well as test for orientation/hand position.
- If an error is detected during the calibration process, then corrective measures will be taken to adjust the values for certain sensors from that particular user.

Calibration Function Flowchart



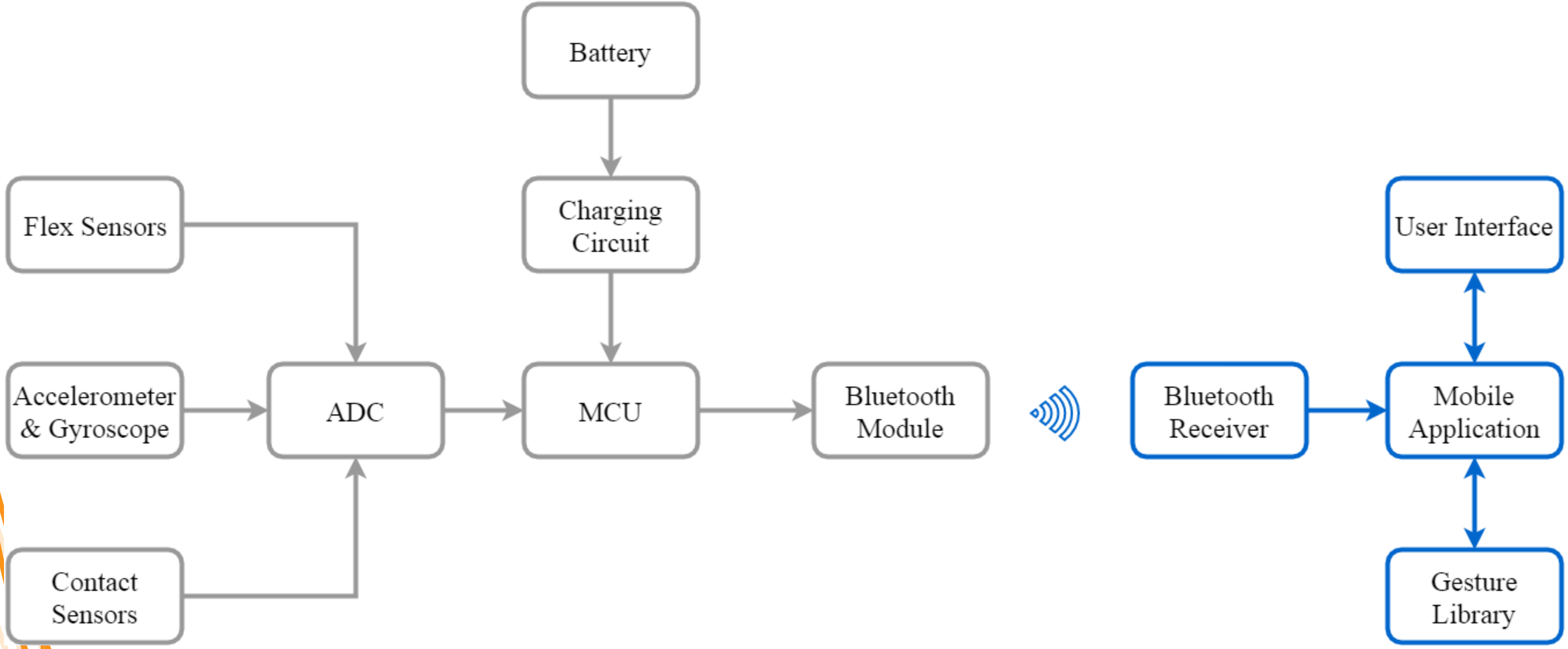
- The calibration process will prompt the user to perform a number of hand gestures to confirm that each type of motion is being correctly picked up by the sensors.
- During calibration, mistakes can be caught and corrected for that particular user before they begin to use the SLIG

Conditional Statements for Gesture Recognition

- Group will experiment with a “mapping” function that takes in the minimum and maximum values provided by the current user during calibration and maps them into a predetermined number of discrete values. For example, bending one finger through its entire range of motion would yield an output of 1-4, with each one of the four “levels” representing one “zone” of that finger’s motion, regardless of the specific values read in by the sensors.

- If mapping function proves ineffective for whatever reason, the group will use the raw numbers collected for the minimum and maximum values during calibration in the conditional statements to determine the hand gesture.

Software Components: Mobile Application



Mobile Application: Overview

- The mobile application is an important feature of this project that will serve as the user interface for the sign language glove.
- The mobile application is responsible for wirelessly displaying hand gestures performed by the glove onto a mobile phone screen as text.
- In order for the mobile application to be successful, the design of the app should consider the user, be simple and elegant, and meet all design requirements.

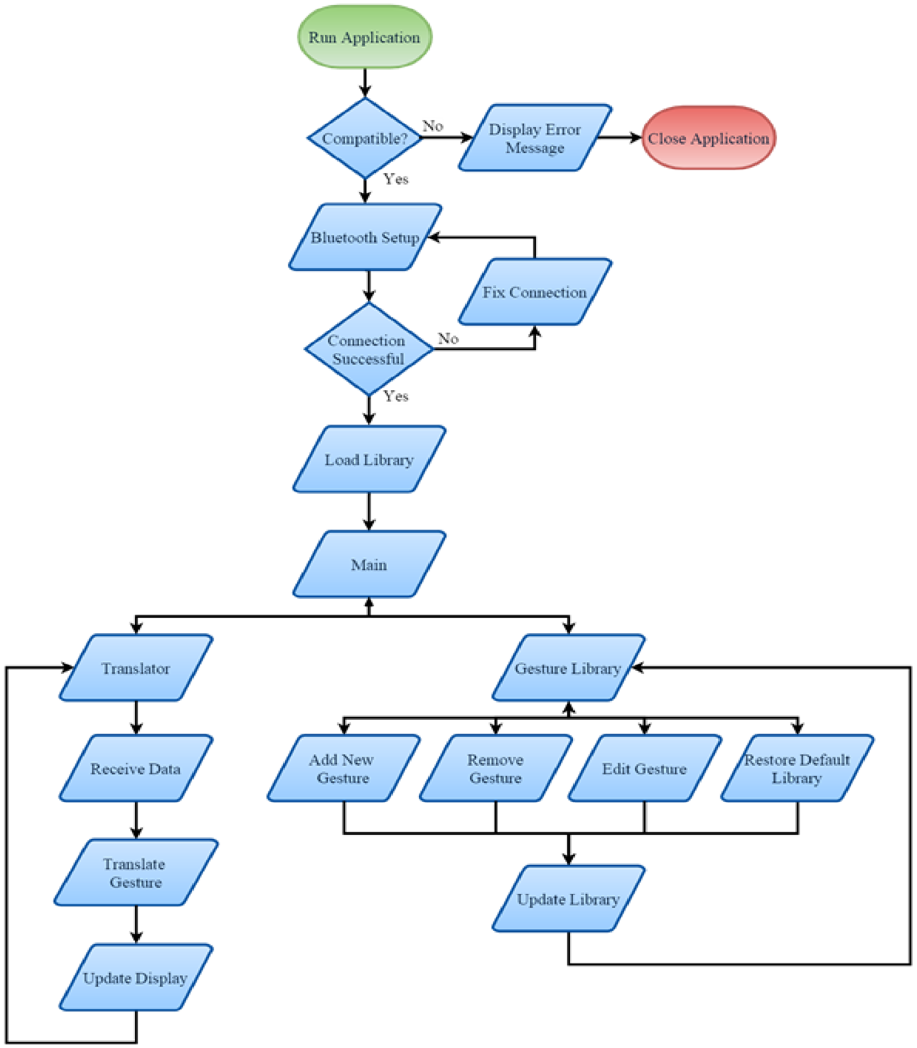
Potential Mobile Platforms

- **Android**
  - Programming language: Java
  - IDE: Android Studio
    - Open source and cross-platform compatible
  - BLE compatible
  - Everyone in our group owns one!
- **iOS**
  - Programming Languages: Swift or Objective-C
  - IDE: Xcode
    - Only Mac OS compatible
  - BLE compatible
- **Windows**
  - Programming Languages: C++, C#, and Visual Basic
  - IDE: Visual Studio
  - BLE compatible

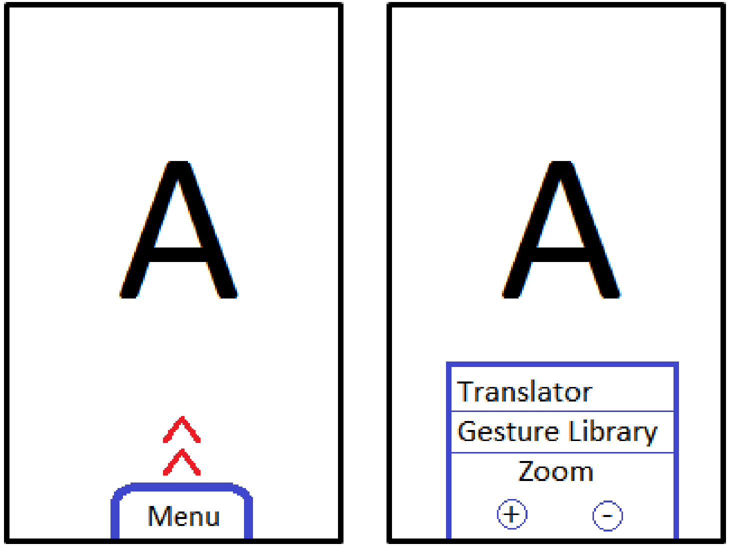
Android

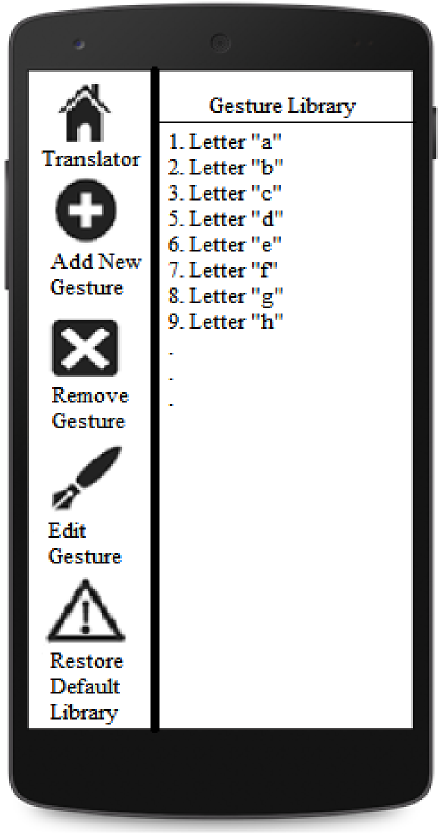
Minimum Requirements	
Device Type	Android
Mobile Platform	Smartphone
Bluetooth Version	Bluetooth Low Energy v4.0
Platform Version	Android 4.3
Codename	Jelly Bean-MR2
Android API Level	18

Mobile App Flowchart



User Interface





Expected Difficulties

- Building a mobile application for the first time.
- Anyone that uses the glove will do the gestures slightly differently, even if it's the same person. This means that the machine and the software both need to be dynamic enough to be able to make the correct command.
- There is so much data that we can collect out of the sensors, but making sure that the software and the machine both understand the data we are sending to them will be our greatest challenge.

Milestones:

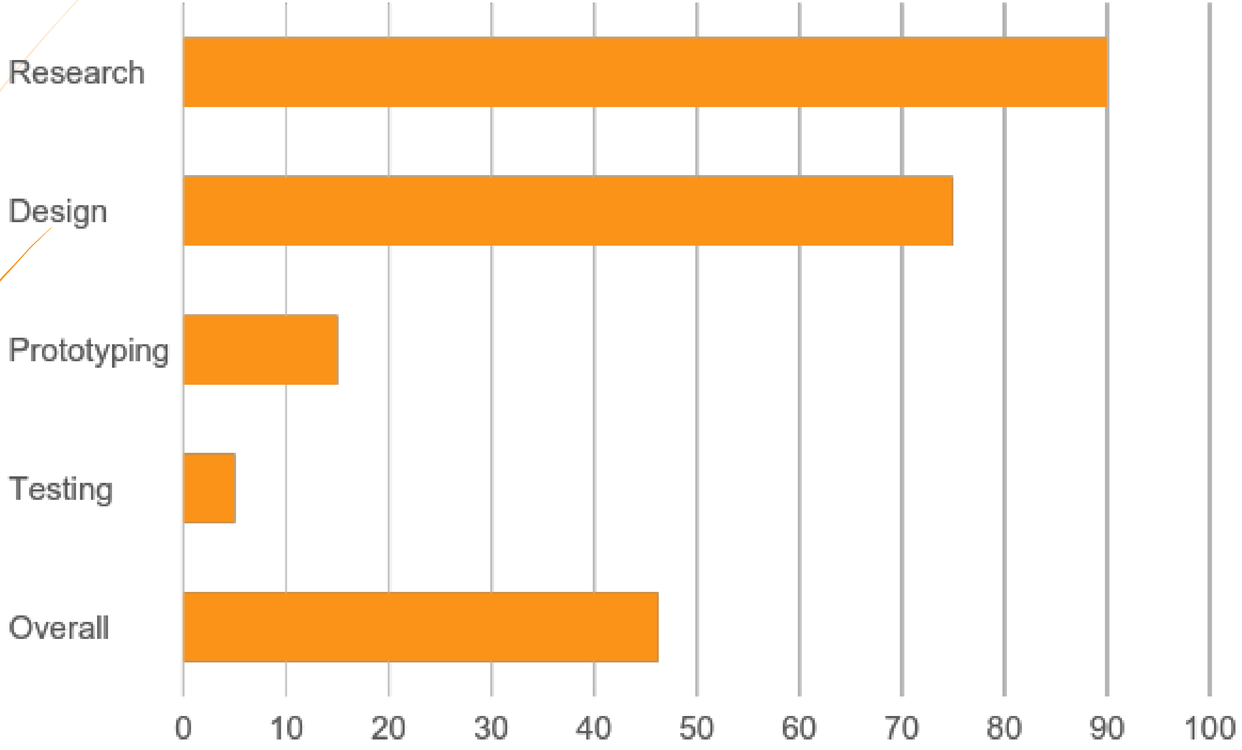
Number	Task	Start	End	Duration (weeks)	Responsible
Senior Design I					
1	**Brainstorming**	9/1/2015	9/8/2015	1	The Team
2	**Project Selection & Role Assignments**	9/8/2015	9/15/2015	1	The Team
	Project Report				
3	Initial Document - Divide & Conquer	9/8/2015	9/15/2015	1	The Team
4	First Draft	9/15/2015	11/3/2015	7	The Team
5	Final Document	11/3/2015	12/8/2015	5	The Team
	Research & Documentation				
6	Bluetooth	9/15/2015	10/5/2015	3	Ramon
7	Flex Sensors	9/15/2015	10/5/2015	3	Chris
8	Accelerometers	9/15/2015	10/5/2015	3	Chris
9	Software	9/15/2015	10/5/2015	3	Jason
10	Power Source	9/15/2015	10/5/2015	3	Jason
11	Microcontroller	9/15/2015	10/5/2015	3	Emanuel
	Design				
12	Bluetooth	10/6/2015	11/3/2015	4	Ramon
13	Flex Sensors	10/6/2015	11/3/2015	4	Chris
14	Accelerometers	10/6/2015	11/3/2015	4	Chris
15	Software	10/6/2015	11/3/2015	4	Jason
16	Power Source	10/6/2015	11/3/2015	4	Jason
17	Microcontroller	10/6/2015	11/3/2015	4	Emanuel
18	**Order & Test Parts**	11/3/2015	12/8/2015	5	The Team
Senior Design II					
19	**Build Prototype**	1/11/2016	3/1/2015	7	The Team
20	**Testing & Redesign**	3/1/2015	3/29/2015	4	The Team
21	**Finalize Prototype**	3/29/2015	4/15/2015	2	The Team
22	**Peer Presentation**	TBA	TBA		The Team
23	**Final Report**	TBA	TBA		The Team
24	**Final Presentation**	TBA	TBA		The Team

Budget And Financing

- **Sponsored by Boeing and Leidos**

Part Description	Price per Unit ($)	Quantity	Cost ($)
Battery	$12.95	1	$12.95
Charging Circuit	$21.95	1	$21.95
Microcontroller	$10	1	$10
Flex sensors	$12.95	10	$129.50
Accelerometer & Gyroscope	$39.95	1	$39.95
Glove	$21.99	1	$21.99
PCB	$74.64	1	$74.64
Bluetooth adaptor	$10.53	1	$10.53
Contact Sensors	$6.95	5	$34.75
Regulators	$0.55	5	$2.75
Miscellaneous parts	-	-	$100.00
Boeing & Leidos sponsorship	-	-	– $359.80
Total cost	**-**	**-**	**$99.21**

Current Progress



Immediate plans for the successful completion of the project

- Finish PCB design
- Order the rest of the parts
- Finish prototype & begin testing

Questions ?