# Sign Language Glove Teaching Device

Design Review

Reebbhaa Mehta – Daniel Fong – Mayapati Tiwari

TA: Igor Fedorov

ECE 445 – Senior Design

February 25, 2013

# Table of Contents

# 1.0. Introduction

## 1.1. Statement of Purpose

This project was chosen because presently the gloves that are used for sign language are the ones that convert sign language gestures into vocalized speech, however there are no gloves available that can help a person to learn sign language on their own. The goal of this project is to create a glove device that detects sign language gestures for letters used in American Sign Language, and inputs them into a computer, where a computer program checks the gesture. If the character is wrong the program will indicate, with the help of LED's and haptic feedback as to what was wrong with the gesture. It can be a good learning tool for sign language and the idea can be built on in the future. Our focus is to make the device easy to handle, and user friendly.

## 1.2. Objectives

1.2.1. Goals

- Build a glove device to detect sign language.
- Develop a database of the gestures of sign language alphabets for the computer program to use as reference while checking input gestures.
- Develop a computer program that checks the input gestures.
- Develop a vibration feedback circuit to make the user aware of any mistake
- Bluetooth communication between computer and glove device.

1.2.2. Functions

- Flex Sensors, accelerometers and gyroscopes to detect sign language gestures.
- Bluetooth communication between device and computer.
- PSoC 5 microcontroller program checks each gesture.
- Vibration feedback and LED's to help the user correct any errors.
- Glove device powered by batteries to avoid any risk of shocking users.

1.2.3. Benefits

- Vibration feedback and LED alerts help to correct any errors with gestures.
- Convenient to use as the device is wirelessly connected to the computer.
- Help users to become more adept with sign language.

1.2.4. Features

- Tri-axis accelerometer and Tri-axis gyroscope in one glove.
- Bluetooth transmission between computer and glove.
- Vibration feedback and LED response for error correction.
- One-directional flex sensors on each finger for reference.
- Mathematical Kalman filtering

## 2.0. Design

### 2.1. Block Diagrams

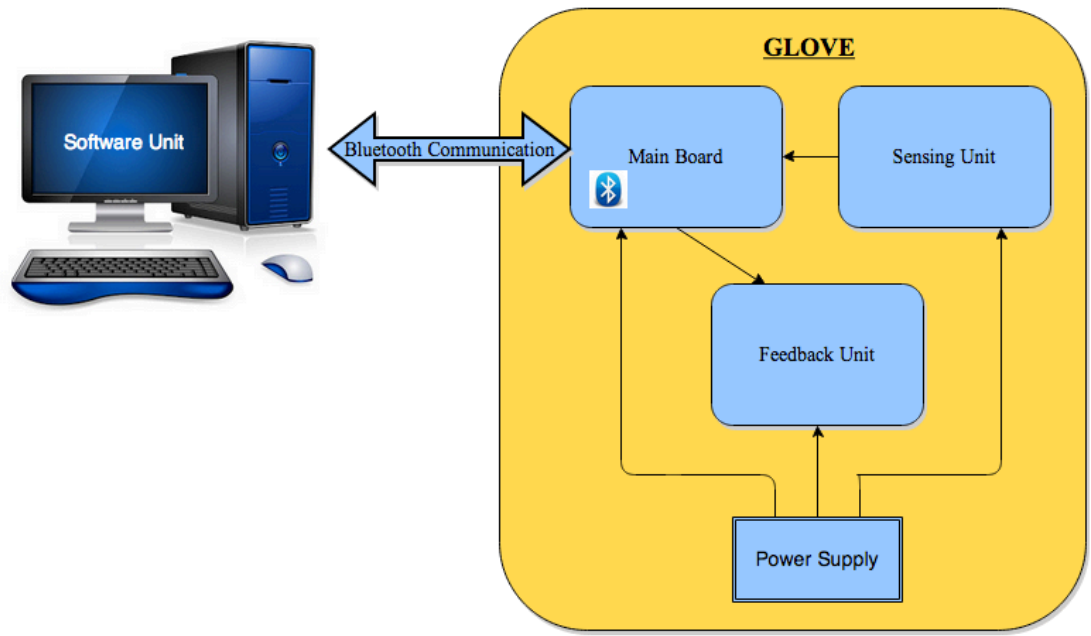


Figure 1: Top Level Block Diagram

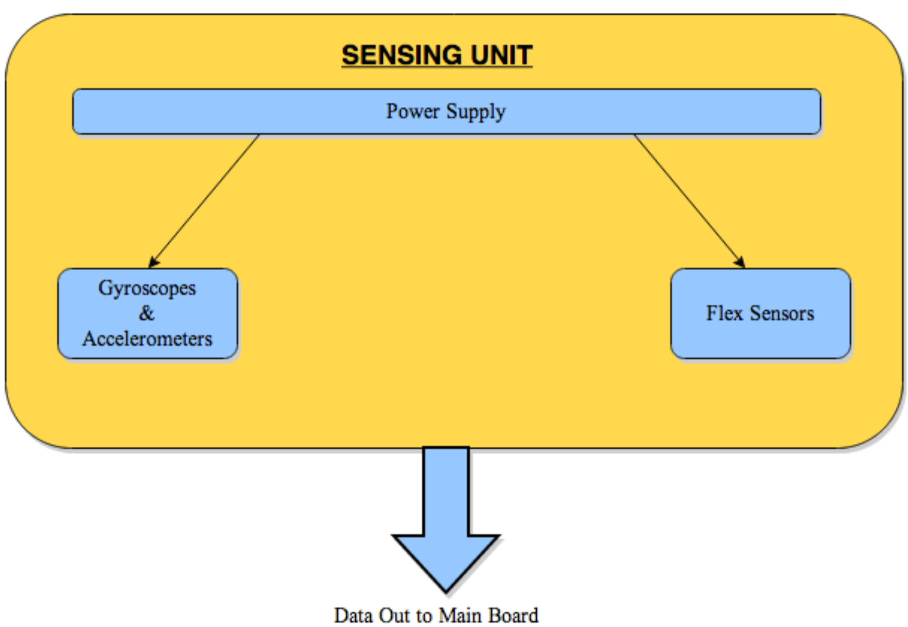


Figure 2: Sensing Unit Block Diagram

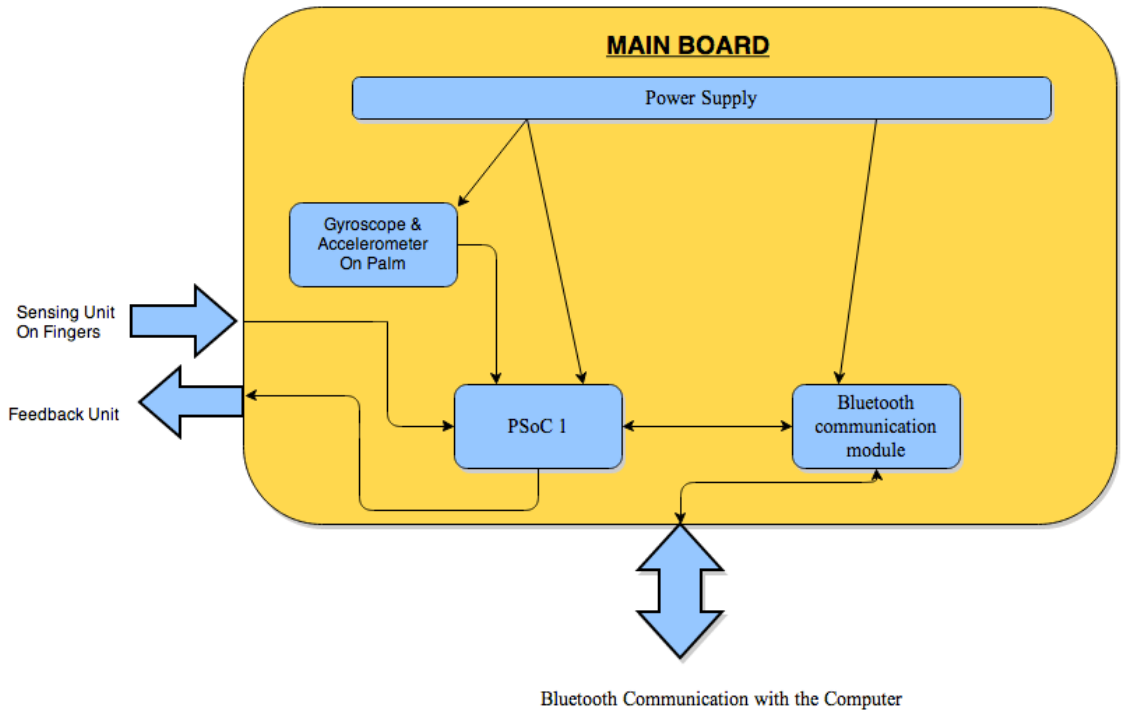


**Figure 3: Main Board Block Diagram**

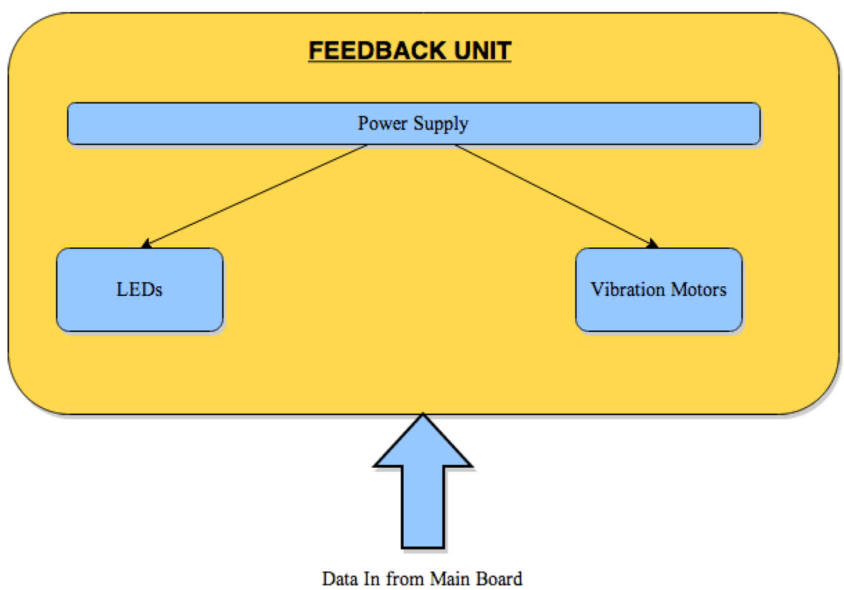


**Figure 3: Feedback Unit Block Diagram**

## 2.2. Block Diagram Description

### 2.2.1. Top Level Block Diagram

The glove device consists of a main board, sensing unit and a feedback unit. The sensing unit detects the position of the hand and fingers for each gesture and sends the data from the three components of the sensing unit (gyroscopes, accelerometers and flex sensors) to the main board, mainly consisting of microcontroller and Bluetooth module. On the main board the data from the sensing unit initially goes to the microcontroller where it is filtered, using Kalman Filtering. The filtered data is then compared with the database with the help of Perceptron Algorithm (working shown later) to check if the users gesture was correct. After the check is performed the valid feedback is sent to the feedback unit to notify the user the final result of the gesture. Once the user has been notified the gesture data is sent to the computer via Bluetooth and stored on the computer for future reference.

### 2.2.2. Main Board

*Overall Summary:*
The main board consists of the power supply, Bluetooth module, microcontroller hip and the Voltage Divider circuit placed on the PCB board. The power supply powers all the other units/components with the appropriate voltages, using voltage divider where needed. The microcontroller filters out the noise, using Kalman Filtering (working shown later), to extract reliable data for each gesture, which is than compared to the database from the computer for that particular gesture. Once the input data has been checked the microcontroller sends out the feedback signals, making the user aware whether the gesture was right or wrong. The Bluetooth chip provides the communication between the

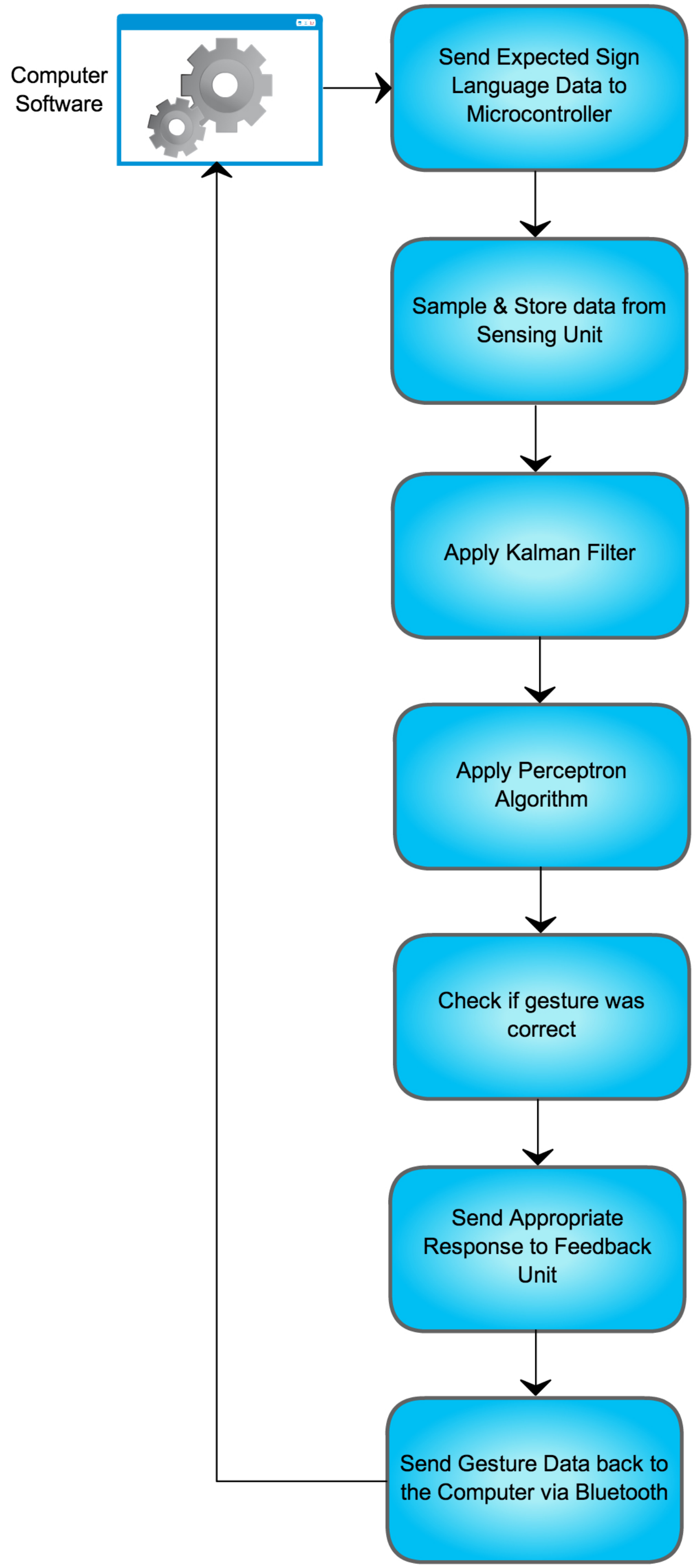


Figure 4: Overall Data Flow

microcontroller and the computer.

The computer software's purpose is to allow users to become adept at signing. For every letter they sign, the users data is recorded and checked and once the feedback has been sent to the user the users input gesture is sent to the computer via Bluetooth for future reference so that users can monitor their progress, and also learn from their mistakes.

*Power Supply:*

The power supply will be a lithium-ion battery that provides 5V (+/- 0.5V). A voltage regulator will step down the voltage to 3V (+/- 0.2V) to power the microcontroller, gyroscopes, flex sensors, accelerometers, vibration motors, LED's and Bluetooth chip. The schematic for power supply can be seen in Appendix Fig. 11.

*Bluetooth:*

The Bluetooth transmitter and receiver will be Texas Instrument CC2540. It was chosen due to low cost, low power and specifications. The Bluetooth transmitter will receive the location and movement specifications of each sign language action through Bluetooth and transfer to the microcontroller. It will also receive the calculation data from the microcontroller through the PCB and transfer the sensor and computational data to the computer through Bluetooth for Matlab debugging. The schematic for Bluetooth can be seen in Appendix Fig. 12 [2].

**Input –** The data to the Bluetooth module comes from two sources. First it gets data from the computer database, which it sends to the microcontroller for comparison. Secondly, after the microcontroller has processed and compared the data, the Bluetooth gets the data of users input gesture, which it sends to be stored in the computer for future needs and references.

**Output –** The Bluetooth transmits the data received from the database of the computer to the microcontroller for comparison. Secondly it sends the users input gesture data to the computer to be stored in memory.

*Microcontroller:*

The microcontroller will be CY8C52xxx (PSoC 5) by Cypress. The Microcontroller will collect all the information from the sensing unit. Thereafter it will filter out the noise to provide more reliable data by using Kalam Filtering. The microcontroller than uses the filter data and compares it to the computer database for that particular gesture using Perceptron Algorithm. After the gesture has been checked the microcontroller helps to make the user aware whether he was right or wrong. Thereafter, the users input gesture is sent to the computer via Bluetooth to be stored in memory for future reference. The schematic for Microcontroller can be seen in Appendix Fig. 10[1].

**Input –** Microcontroller receives data from the sensing unit, which it later filters out to reduce noise using Kalman Filter

**Output –** The microcontroller has two outputs. Firstly, it sends a signal to the Feedback unit to alert the user of the result after it has compared the users input to the vector database. Secondly, the microcontroller sends the users input gesture data to the Bluetooth module to be stored on the computer.

### 2.2.3. Sensing Unit

*Overall Summary:*

The sensing unit consists of six gyroscopes and accelerometers (as one unit) and six flex sensors to accurately detect the position of the hand and each finger for each gesture. The data from the sensing components goes to the microcontroller where it is processed.

*Tri-axis Gyroscopes and Tri-axis Accelerometers:*

For our design instead of using a separate gyroscope and accelerometer, we plan to use MPU-6050, which has an integrated 3-axis gyroscope and 3-axis accelerometer. This way we will be able to use the space on the glove more efficiently and avoid crowding of devices at the fingertips. The layout for MPU-6050 or mini fingerboards can be seen in Appendix Fig. 13. The schematic for MPU-6050 can be seen in Fig. 14 [(3)] of the Appendix.

Gyroscopes will detect the angular velocity in three-axis, which will help us in calculating the angle of the glove in each direction that will help us to know the orientation of the glove. Different hand gestures can be differentiated this way.

Accelerometers will be used for tilt sensing. It helps us to know how fast something is moving. Along with gyroscopes, accelerometers will also help us to know the orientation of the glove and help us to differentiate hand gestures. Accelerometers will be very useful for two particular sign language gestures, i.e, ‘J’ and ‘Z’.

**Output –** MPU 6050 chips send data of the position of each finger and palm, when the user performs a gesture, to the microcontroller.

*Flex Sensors:*

One directional flex, FLX-03, sensors are needed to detect how much each finger is bent in order to check that each hand gesture is within the given tolerance level. Flex sensors as of now are just an added feature in our design, which will help us to know more accurately the positions of each finger and hand.

Basic Flex Sensor Circuit is just a voltage divider circuit (please refer to Appendix Fig. 15 [(4)]) with Equation 1.1, where $R_2$ is the flex sensor resistance.

$$V_{out} = V_{in}\left(\frac{R_1}{R_1 + R_2}\right) \quad \dots (1.1)$$

The theoretical output voltage range will be between 1 – 2.5 V.

The impedance buffer in the schematic is a single sided operational amplifier, used with these sensors because the low bias current of the op amp reduces error due to source impedance of the flex sensor as voltage divider. Op amps that can be used are LM358 or LM324.

**Output –** All the data from the flex sensors is sent to the microcontroller so that it can know exactly how much each finger is flexed for a particular gesture.

Results & Simulation:

The following data was taken for a flex sensor when it is flexed at different degrees. The initial value of the flex sensor is 9.48 kΩ when it is unflexed. On the other hand 22.7 kΩ is the resistance of the sensor when it is completely flexed. The resistance values were measured using a multimeter and the corresponding voltages were measured using an oscilloscope. The Figure 5 shows the results of Voltage vs. Resistance for a flex sensor. The y-axis is voltage and the x-axis is resistance of the flex sensor.

**Table 1:Resistance and Voltage Data for Flex Sensor**

| $R_2$ (Flex Sensor) | $V_{out}$ |
|---|---|
| 9.48 kΩ | 1.78 V |
| 15.3 kΩ | 1.69 V |
| 17.0 kΩ | 1.59 V |
| 21.2kΩ | 1.36 V |
| 22.7 kΩ | 1.32 V |

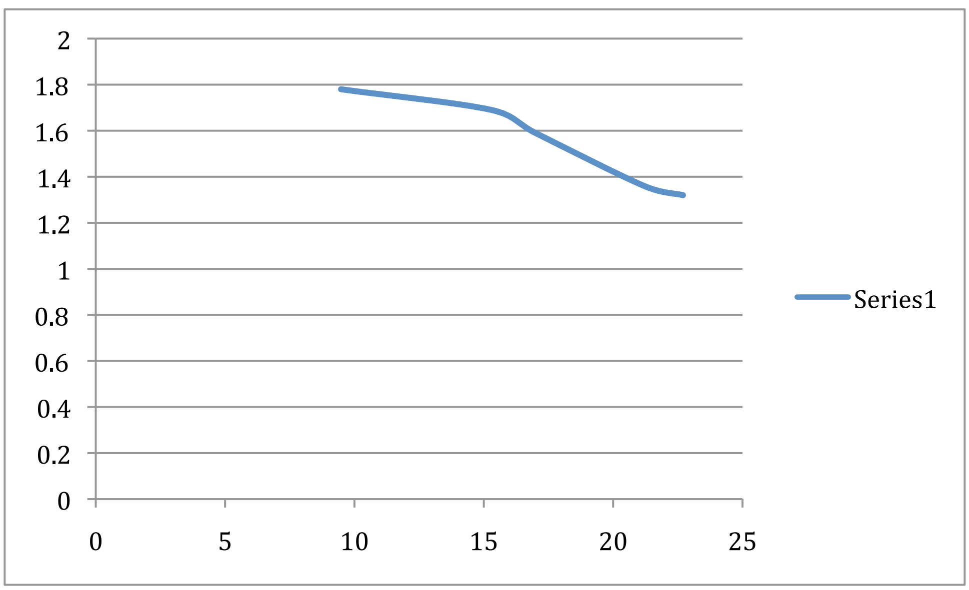


**Figure 5: Voltage vs. Resistance for Flex Sensor**

### 2.2.4. Feedback Unit:

*Overall Summary:*

The feedback unit consists of five vibration motors and five LEDs, placed on each fingertip. The feedback unit is used to notify the user whether the input gesture was correct or not. Providing feedback with the help of vibration motors and LED's helps a user to better grasp the gestures for each sign, thus helping him to learn faster.

*Vibration Motors:*

Vibration motors are placed at the fingertips to alert the user when a hand gesture is incorrect by vibrating only for the fingers that are incorrectly placed. Motors are controlled by PSoC 1, which after checking the input gesture alerts the motors when a gesture is wrong. For our design we plan to use a Pico Vibe 310-103 10mm Vibration Motor – 2.7mm Type. The schematic for vibration motor feedback can be seen in Appendix Fig. 16[6]

**Input –** Meaningful data is sent from the microcontroller to the vibration motors, once the microcontroller is done checking the gesture, to notify the user of the result. For a wrong gesture the fingers, which were wrongly placed, will vibrate making the user aware of what was wrong with the gesture.

*LEDs:*

Bicolor LEDs (with red and green colors), TLUV5300, are used with vibration motors to show more accurately which finger is in an incorrect position for a particular gesture. If a finger is in an incorrect position the corresponding LED will be red, else it will be green. PSoC also controls LED's in the same way as the vibration motors.

**Input –** The microcontroller notify the LEDs of which fingers were positioned correctly and which weren't. If a finger was correctly placed the LED flashes green else it is red.

## 2.3. Software & Computations

### 1. Computer Software

The computer software is tutorial software that helps to teach Sign Language. Initially a database is created while the learning algorithm takes in data and creates a vector model that it uses to match the gyroscope and accelerometer data points to the correct gesture. The computer software sends the data for the current gesture to the PSoc and receives data from the gyroscope and accelerometer along with the output of the perceptron algorithm, this way it tracks the user's progress. The computer software keeps a record of the user's learning curve, by recording data from numerous training sessions.

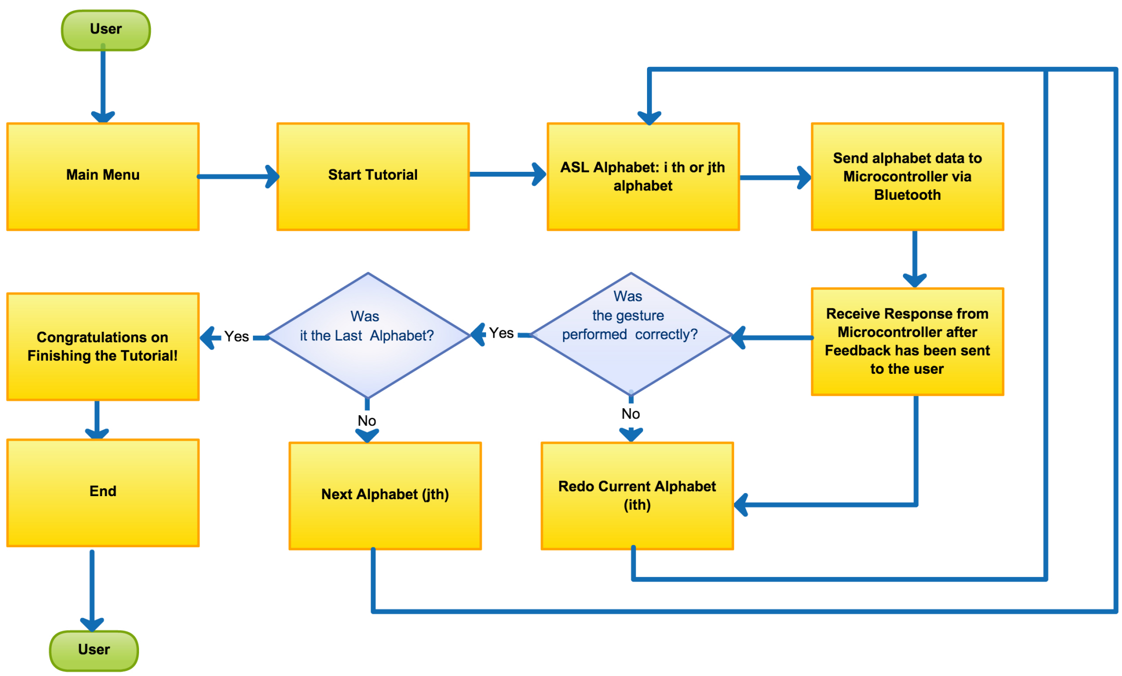


Figure 6: Computer Software Data Flowchart

### 2. Voltage Divider

Voltage divider circuits will be used to power the all the components from the battery as the components work at a lower voltage than the power supply. The voltage for LEDs will be 2V while for all other components will be 3V. The voltage dividers will work as shown in the appendix in schematic. The equation for voltage divider is as follows:

$$V_{out} = V_{in}(\frac{R_1}{R_1 + R_2})$$

### 3. Kalman Filter

Kalman Filtering is used to reduce the noise received from the sensor unit data providing a more stable and suitable result. The filtering is performed by the microcontroller before it compares the sensing unit data using the matching algorithm.

The Kalman filter works by recursively processing set of measurements containing noise and combining the measurements to estimate the actual state of the unknown variables. The Filter first makes a prediction of the different states. Then, it does a weighted average based upon the certainty of the prediction between the predicted state and the newest measurement to produce the estimate of the underlying states. The Kalman Filter assumes that the error and measurements have a gaussian distribution.

We will use the Kalman filter on the combined data from the gyroscopes as well as the accelerometers, thus giving us the best solution and most accurate data. Since this way we can also eliminate the error due to gyroscope drift.

[7] Kalman Filter algorithm is as follows:

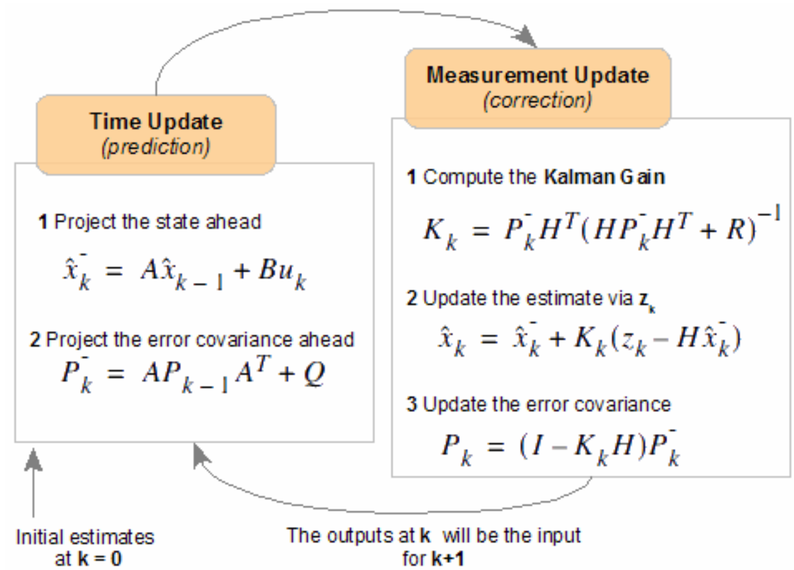


**Figure 7: Kalman Filter Algorithm**

Kalman Filter was simulated using MATLAB. As can be seen from figures we expect the filter to reduce the noise level from the data thereby providing a more accurate data which lead to positive results.

*Simulation Notes -* The original signal was an object moving with a constant acceleration of 1m/s^2 with initial position at 0. Gaussian noise of mean 0 and standard deviation .3 was added to the position measurement. Gaussian noise of mean 0 and standard deviation .03 was added to the position measurement. The Kalman Filter was applied to the acceleration and position measurements to predict the original values based upon the noisy measurement inputs.

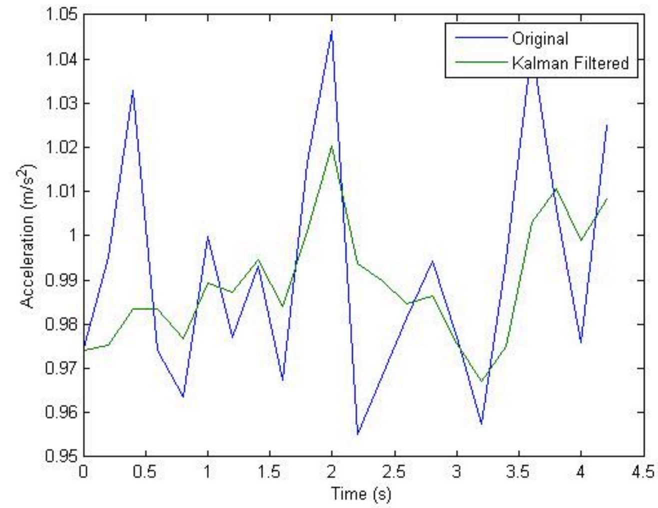


**Figure 8: Karman Filter Acceleration**

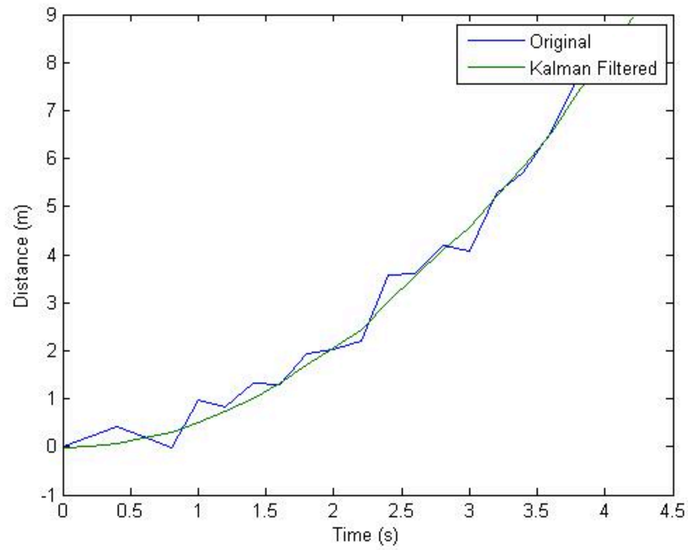


**Figure 9: Karman Filter Distance**

## 4. Perceptron Algorithm

Perceptron algorithm is a matching algorithm. It matches the vector model value to the input value to check if the input is correct. We decided to use the perceptron learning algorithm as our matching software, we picked this foe a few reasons. It is perfect for an n-dimensional linear functions and using this we can map any n-dimensional linear separator that (in our case) passes through the origin to an n-dimensional weight vector such that $W_x > 0$ for positive points and $W_x < 0$ for negative points.

The algorithm consists of two parts, first the algorithm "learns" via supervised learning, it is during this time that the algorithm comes up with a model that fits the one at hand and assigns weights.

After about 100 data sets of each symbol the algorithm should be able to correctly match the gestures performed by the user up to 98% accuracy.

[5] The perceptron algorithm is as follows:

PERCEPTRON ALGORITHM
Initialize $\mathbf{w_1} = \mathbf{0}$
At each round $t \in \{1, 2, \cdots\}$

- Receive input $\mathbf{x_t}$
- If $\mathbf{w_t} \cdot \mathbf{x_t} \geq 0$, predict $+1$, else predict $-1$
- If there is a mistake (i.e., if $y_t(\mathbf{w_t} \cdot \mathbf{x_t}) < 0$), set $\mathbf{w}_{t+1} \leftarrow \mathbf{w_t} + y_t \cdot \mathbf{x_t}$, else set $\mathbf{w}_{t+1} \leftarrow \mathbf{w_t}$.

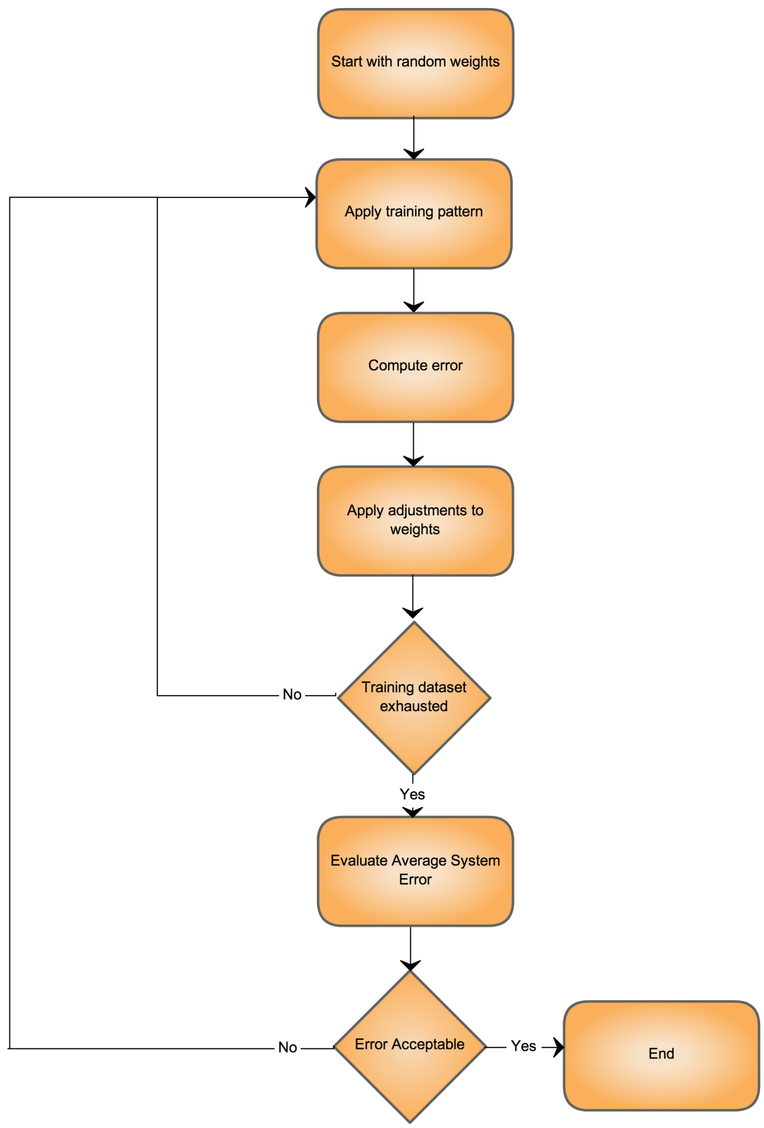


**Figure 10: Perceptron Algorithm Flowchart**

Simulations for Perceptron Algorithm:

The simulations were achieved by implementing single neuron Perceptron in 2 dimensions on Matlab, the simulations are based on two sets of random data divided by the line x=y. We can see that the algorithm has been able to successfully differentiate between the two data sets. In the first case the training set is much larger compared to the test data, but in the second case our training set is more comparable to our test data in size, this goes to show that even with relatively small training data sets this algorithm performs well and that it is suitable for our implementation. We will be using this to differentiate between linearly separable vectors in a three dimensional space.

Training Data Size = 400, Test Data Size = 84, Test Errors = 0

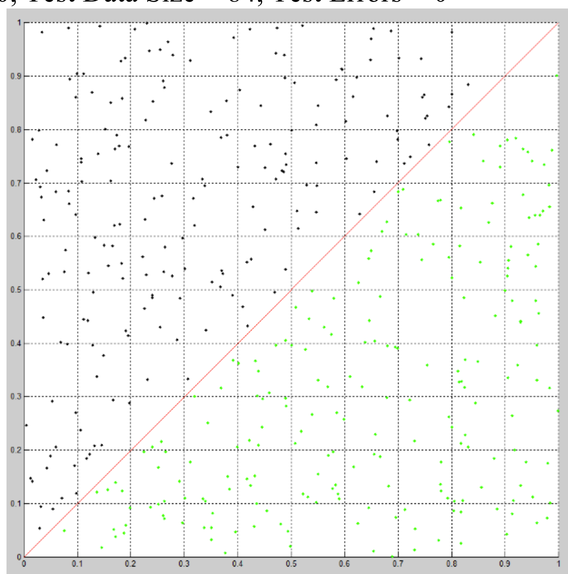

Figure 11: Simulation 1 for Perceptron

Training Data Size = 500, Test Data Size = 469, Test Errors = 1

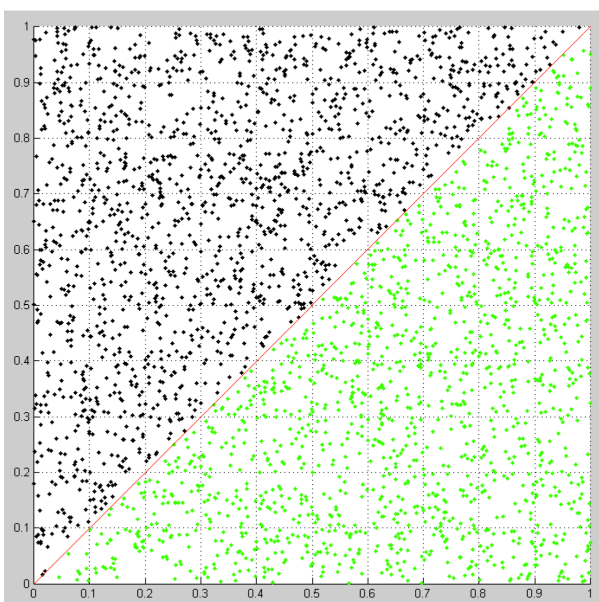


Figure 12: Simulation 2 for Perceptron

# 3.0 Verification

## 3.1. Performance Requirements And Testing Procedures

### 3.1.1 Sensing Unit

| Performance Requirements | Verification |
|---|---|
| **Gyroscope & Accelerometer (MPU 6050)** | |
| 1. Accelerometer and Gyroscope data should be consistent. | 1. The following steps will be taken to verify the consistency:<br>a) Place MPU 6050 on a flat surface, the measurements should be 0°/s (+/- 2°/s) for gyroscope data.<br>b) Tilt the surface and record the data from the gyroscopes and accelerometers.<br>c) Calculate 'θ' from accelerometer data from its x-axis and z-axis. $\theta = \arctan(-\frac{z}{x})$<br>d) Calculate 'γ' from gyroscopes z-axis data which gives 'ω' – $\omega = \gamma \times t$<br>e) The value of θ and γ should be close. |
| **Flex Sensors** | |
| 1. Check resistance range from 10kΩ - 40kΩ (+/- 1kΩ)<br>f) Output 10k ohms when unflexed (when fingers are straight)<br>g) Output 40kΩ when completely flexed (hand is curled to make a fist) | 1. Probe flex sensors with digital multimeter to check for resistance<br>a) When the sensor is unflexed the output resistance can be checked with DMM which should display a resistance of 10kΩ (+/- 1kΩ)<br>b) When the sensor is completely flexed the output resistance can be checked with DMM which should display a maximum resistance of 40kΩ (+/- 1kΩ) |

### 3.1.2 Main Board

| Performance Requirements | Verification |
|---|---|
| **Power Supply** | |
| 1. The power supply must be able to provide at least 5V (+/-0.5V)<br><br>2. Voltage regulators to flex sensors, accelerometers & gyroscopes, Bluetooth module, microcontroller, LED's and vibration motors should be able to output 3V (+/- 0.2V) | 1. The power supply will be connected to a 2Ω load. Using a multimeter we should measure the voltage to be 5V (+/-0.5V). The same procedure can be performed with a bigger load.<br>2. Black multimeter lead is plugged into the "COM" socket of each regulator and red lead is plugged to the "V" socket. The multimeter should measure a voltage of 3V (+/- 0.2V) |
| **Microcontroller** | |
| 1. Successfully receive valid data from sensing unit<br>a) Create sample data mimicking data from gyroscopes & accelerometers, that is, 500°/s for gyros and +/-2g for accelerometers.<br>b) Successfully receive sample analog signal mimicking flex sensors in the range of 0-2.5V, with a voltage above 1.25V considered a high.<br>2. Communicate with Bluetooth module by generating a 2.4GHz signal<br><br>3. Successfully filter noise from environment using Kalman Filter<br><br>4. Successfully implement Perceptron Algorithm that will give us 98% accuracy to match data.<br><br>5. Alert feedback unit with correct results | 1. Test the signals from two sources and compare to test the data<br>a) Compare the data received by microcontroller with the sample data used for testing.<br>b) Check the values with those received from microcontroller against the sample data<br><br>2. Code the microcontroller to send a signal of 2.4 GHz every second and graph the output with an oscilloscope. After the signal to frequency domain the frequency can be verified.<br>3. Check the results of the Kalman filtering in the microcontroller to the results of our Matlab simulation for Kalman Filter<br>4. After applying the training program to the algorithm, we will check to see if it is capable of matching data correctly to the accuracy specified.<br>5. Check that the microcontroller is capable of sending a feedback signal to 6 sample loads. |
| **Bluetooth** | |
| 1. Check if it is transmitting the correct data | 1. Compare the data sent by PSoC microcontroller with the data received by computer, if it's a match than the Bluetooth module is transmitting the correct data. |

### 1.1.3 Feedback Unit

| Performance Requirements | Verification |
|---|---|
| **LEDs** | |
| 1. Operating voltage of a maximum 3V Typical operation for 2.4V | 1. Connect LED into a breadboard and input a voltage of 3V. If the LED is on then it is working. Also the LED should be off for very low voltage inputs. |
| **Vibration Motors** | |
| 1. Operating voltage of 3V (+/- 0.1V)<br><br>2. Basic functionality | 1. Ensure the vibration motor is getting an input voltage of 3V by probing using a multimeter<br>2. Write a program to keep the motors to vibrate for 1s every 10s to test the basic functionality of the motors. |

## 3.2. Tolerance Analysis

| Block Name | Focus | Acceptable Ranges |
|---|---|---|
| Microcontroller<br>1. The data sent to the Bluetooth chip<br>2. Data sent to feedback unit | The acceptable ranges for these two parameters will be decided by Kalman Filter and Perceptron Algorithm (matching algorithm) | Depending on the performance of the Kalman filter and Perceptron algorithm the range should be 98 – 100 %. |
| Bluetooth<br>1. Response time to send data and receive data from the computer | Focus lies on the speed at which data can be transferred while insuring the integrity of data. | Response time acceptable upto 2second.<br>Accuracy of data transferred should be 98 – 100% |
| Power Supply<br>1. Supply power to the rest of the system as per the need of each component | It is important to ensure that each component is supplied with the right amount of power so that it can operate comfortably in its operational range. | The voltage supply is 2V for the LEDs and 3V for all other components with a maximum 50mA current. |
| | | |
| Gyroscopes & Accelerometers<br>1. Supply microcontroller with accurate data for each finger position and the position of palm | Focus lies on the accuracy and integrity of data supplied | There is no reason as to why the integrity of data should be compromised. So the results expected are 100% accurate |
| Flex Sensors<br>1. This data is an enhancement to the | The resistance should always in the range of 10k – 40k Ohms | Since flex sensor data is not as important as gyroscope data and is not as accurate, |

| | | |
|---|---|---|
| data provided by gyroscopes and accelerometers. | | accuracy of 95% is acceptable |
| | | |
| Vibration Motors<br>1. The vibrations should not exceed 0.9g<br>2. Correct motors should be on during feedback | Focus is that the correct motors are on at the right time so that meaningful feedback can be provided to the user. Low vibration feedback is better so that the user is comfortable while using the device | Any vibration range between 0.4 – 0.9 g is good for this purpose. |

## 4.0. Ethics

We will uphold IEEE code of Ethics and Academic Honesty.

We will make sure our device is safe to use. We will take the necessary precautions to make sure that no one is harmed while using the device.

## 5.0. Safety Analysis

Safety is one of the highest priorities for the glove device. The device should be carefully made to not expose the user to any current from the circuit.

We also need to worry about the device components getting too hot as they are close to the skin and may cause harm.

During the development process safety precautions must be taken while soldering and using the hot glue gun.

# 6.0. Cost Analysis and Schedule

## 6.1. Cost Analysis

6.1.1. Labor

| Name | Hourly Rate | Time Invested (Hours) | Total = Hourly Rate x 2.5 x Time invested |
|---|---|---|---|
| Reebbhaa Mehta | $35.00 | 180 | $15,750 |
| Daniel Fong | $35.00 | 180 | $15,750 |
| Mayapati Tiwari | $35.00 | 180 | $15,750 |
| **Total** | | 540 | **$47,250** |

6.1.2. Parts

| Item | Part Number | Quantity | Company | Cost ($) |
|---|---|---|---|---|
| Glove | - | 1 | Pearl iZUmi | 10.00 |
| Flex Sensors | FLX – 03 | 5 | Images Scientific Instruments | 70.00 |
| Accelerometers & Gyroscopes | MPU 6050 | 6 | Invensense | 105.00 |
| Vibrating Motors | Pico Vibe 310-103 | 5 | Precision Microdrives | 35.00 |
| Vibrating Motor Driver | DRV2603 | 5 | Texas Instruments | 5.85 |
| LEDs | TLUV5300 | 5 | Mouser Electronics | 1.00 |
| Microcontroller | CY8C52xxx | 1 | Cypress Perform | 100.00 |
| Bluetooth | CC2540 | 1 | Texas Instruments | 6.00 |
| Battery | Lithium-ion – 5V | 1 | Panasonic | 5.00 |
| Resistors, Capacitors, Inductors | - | | University | 5.00 |
| Voltage Regulator | LP2951ACP | 8 | Jameco Electronics | 5.00 |
| PCB | - | 6 | University | 50.00 |
| **Total** | | | | **397.85** |

6.1.3. Grand Total

| Section | Total |
|---|---|
| Labor | $47,250.00 |
| Parts | $397.85 |
| **Grand Total** | **$47,647.85** |

## 6.2. Schedule

| Week | Task | Responsibility |
|---|---|---|
| 2/25 | Prepare Design Review | All |
| | Enable Bluetooth communication | Reebbhaa |
| | Implement Kalman Filter on PSoc | Daniel |
| | Design PCB V2 | Reebbhaa |
| | Order Revised Parts | Reebbhaa |
| 3/4 | Configure gyroscopes and accelerometers | Mayapati |
| | Test Kalman Filter | Daniel |
| | Solder PCB/Connect PCB to Sensor Unit | Reebbhaa |
| | Create Computer Software | Reebbhaa |
| | Implement Perceptron Algorithm on Psoc | Mayapati |
| 3/11 | Apply Training Algorithm and Test Perceptron Algorithm | Reebbhaa |
| | Test glove without feedback Unit | Daniel |
| | Debug glove without feedback Unit | Reebbhaa |
| | Assemble Feedback Mechanism | Mayapati |
| | Implement Feedback Mechanism Software | Daniel |
| 3/25 | Prepare Mock-up Presentation | Mayapati |
| | Test & debug Feedback Mechanism | Reebbhaa |
| 4/1 | Tolerance Analysis | Reebbhaa |
| | Test entire device | Daniel |
| | Verification of Specifications | Mayapati |
| 4/8 | Testing and Debugging | Mayapati<br>Daniel<br>Reebbhaa |
| 4/15 | Complete Project | All |
| 4/22 | Prepare Demo | Daniel |
| | Prepare Presentation | Reebbhaa |
| | Prepare Final Paper | Mayapati |
| 4/29 | Demo | All |
| | Presentation | All |
| | Final Paper | All |
| | Check Supplies | Mayapati |

## 7.0. Appendix

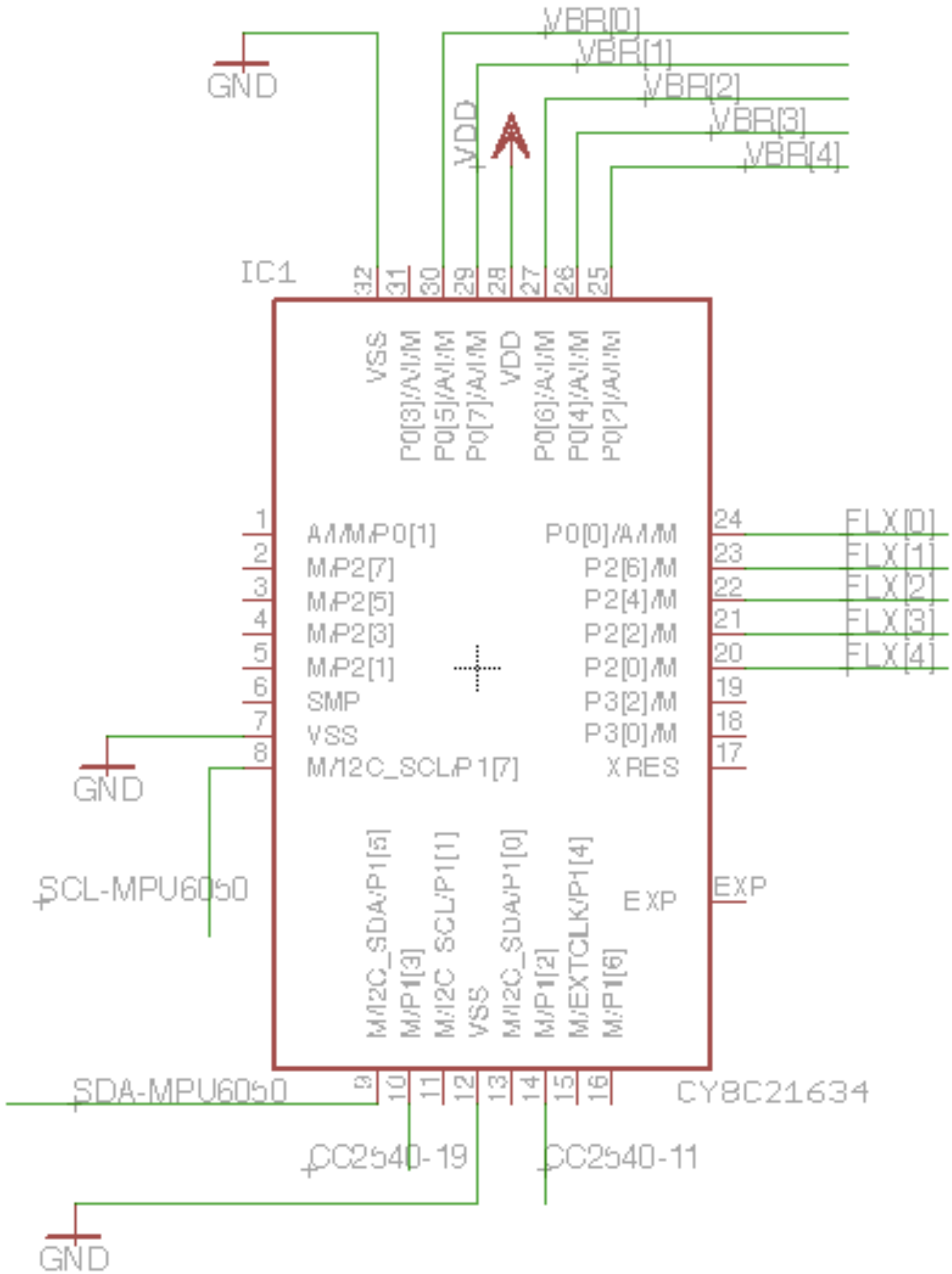


Figure 13: Microcontroller Schematic [1]

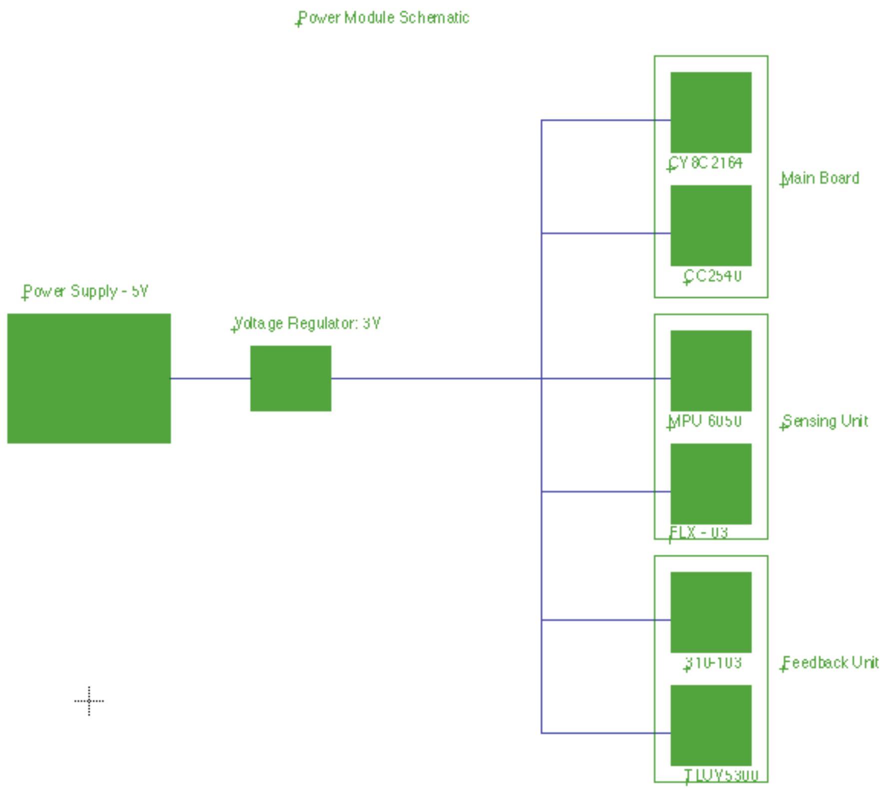


**Figure 14: Power Supply Schematic**

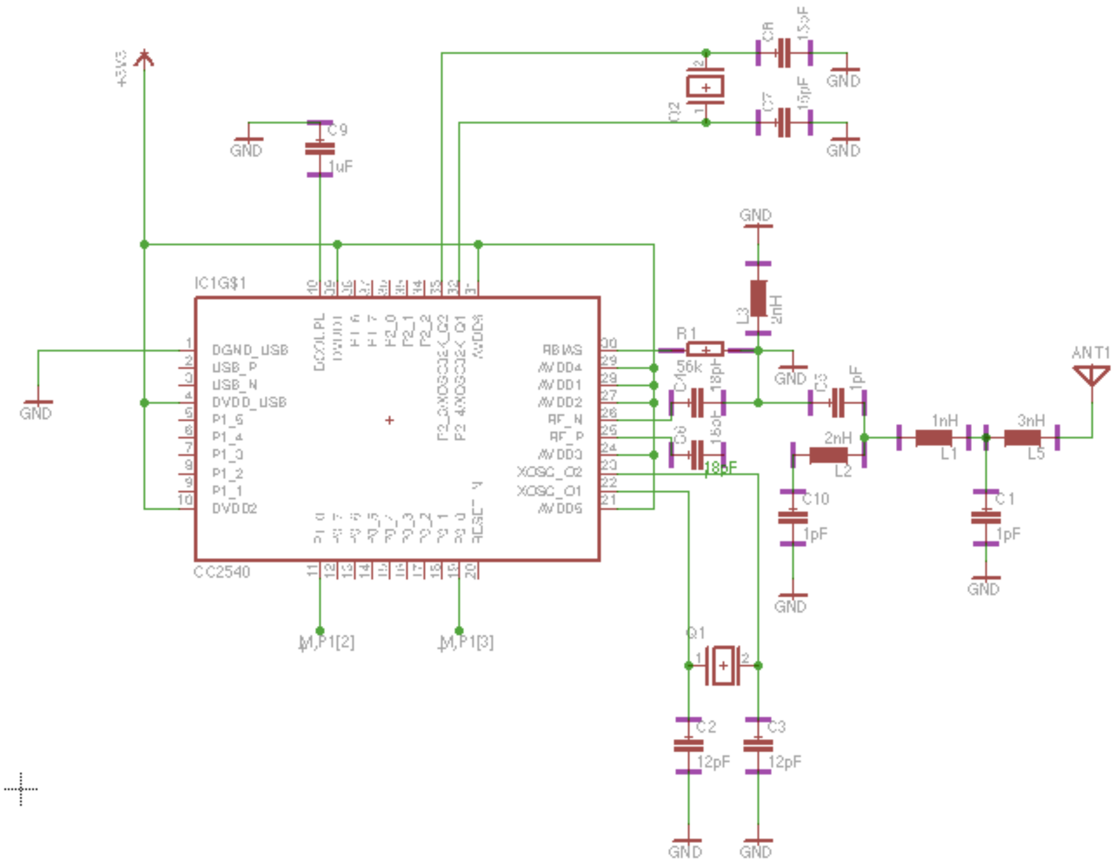


**Figure 15: Bluetooth Schematic (2)**

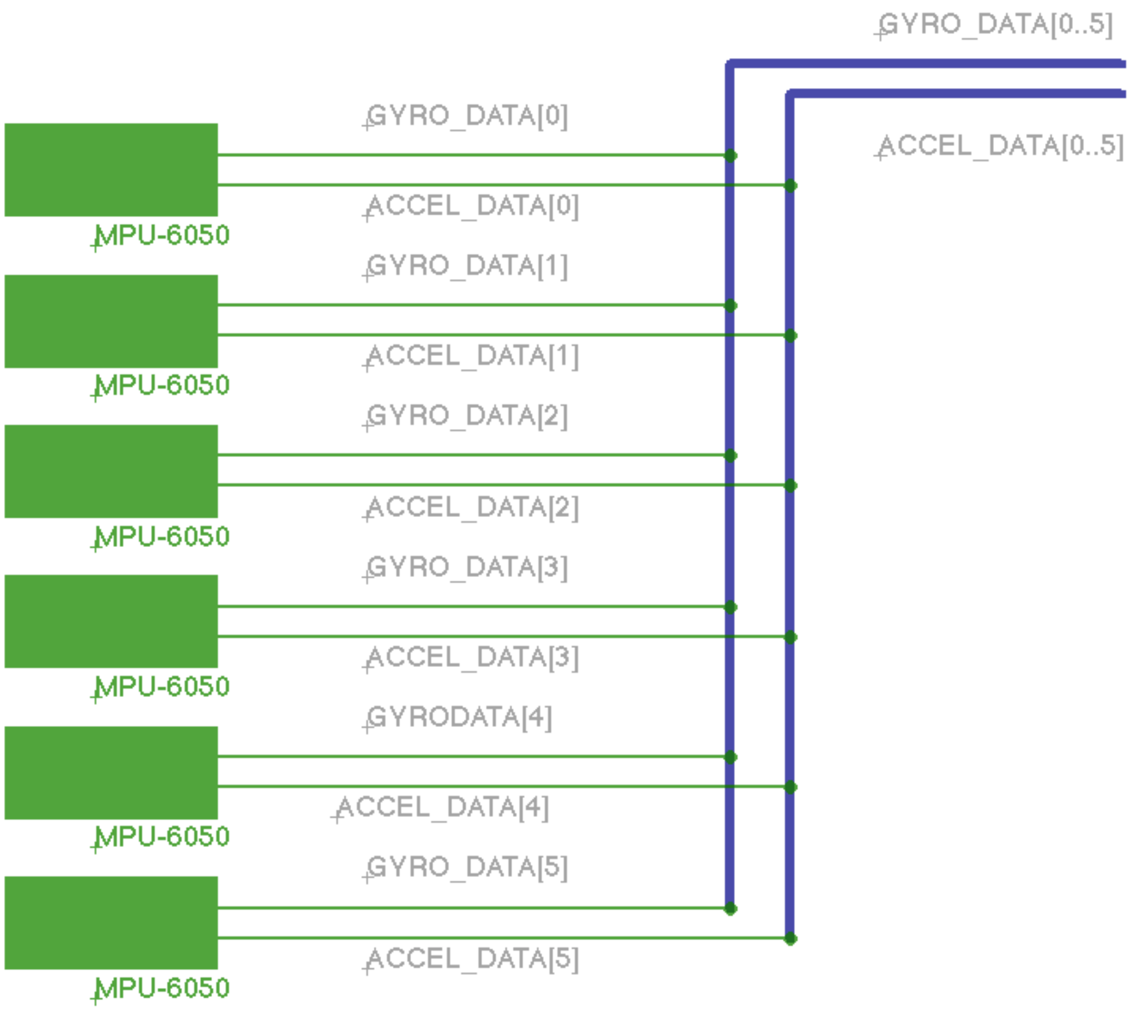


**Figure 16: MPU 6050 Layout (Finger PCB's)**

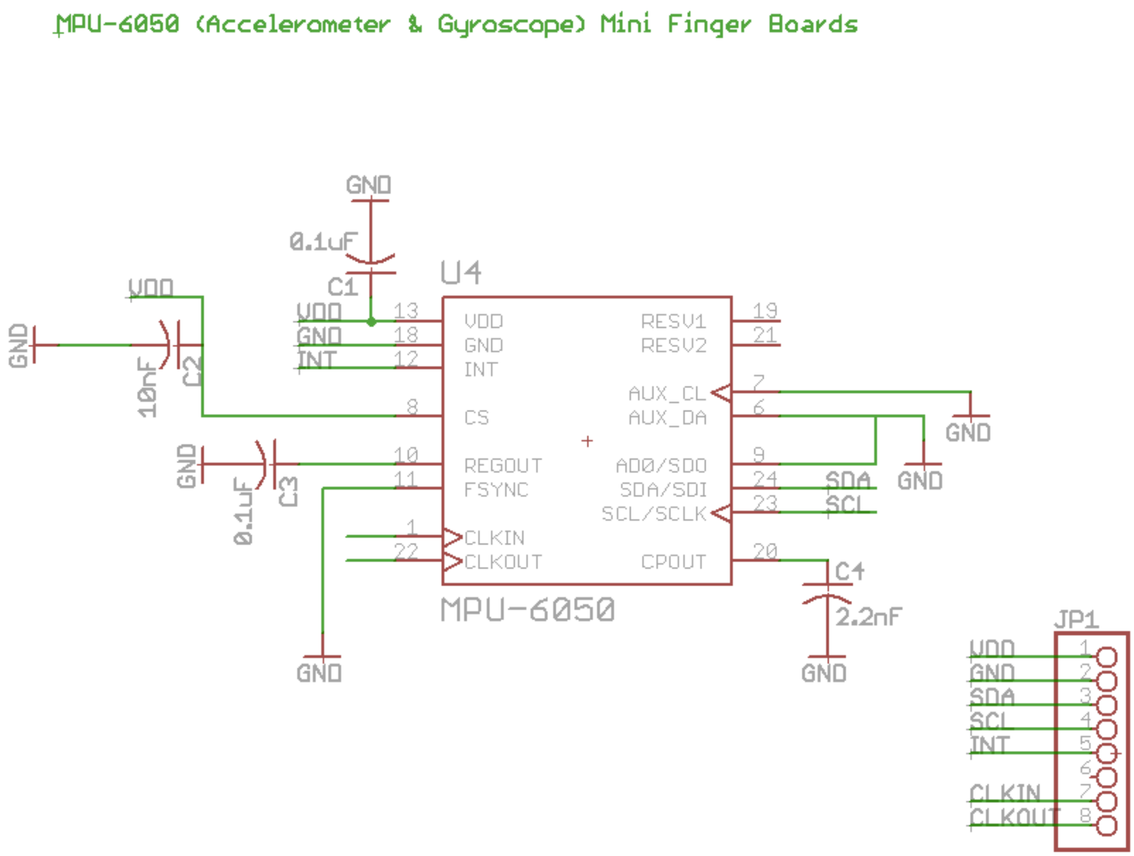


**Figure 17: MPU 6050 Schematic (3)**

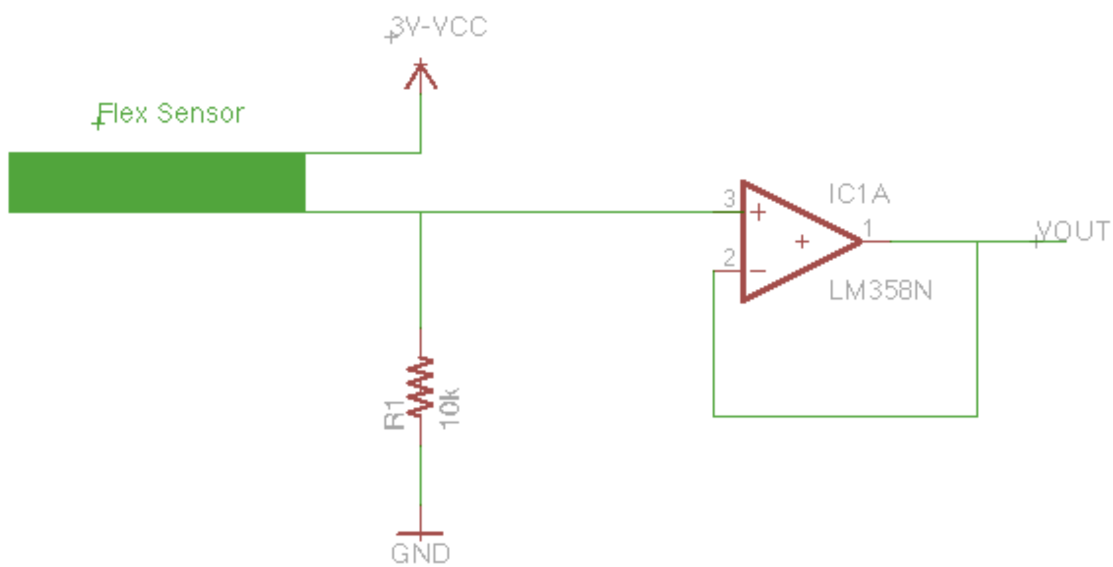


**Figure 18: Flex Sensor Schematic (4)**

**Functional Diagram**

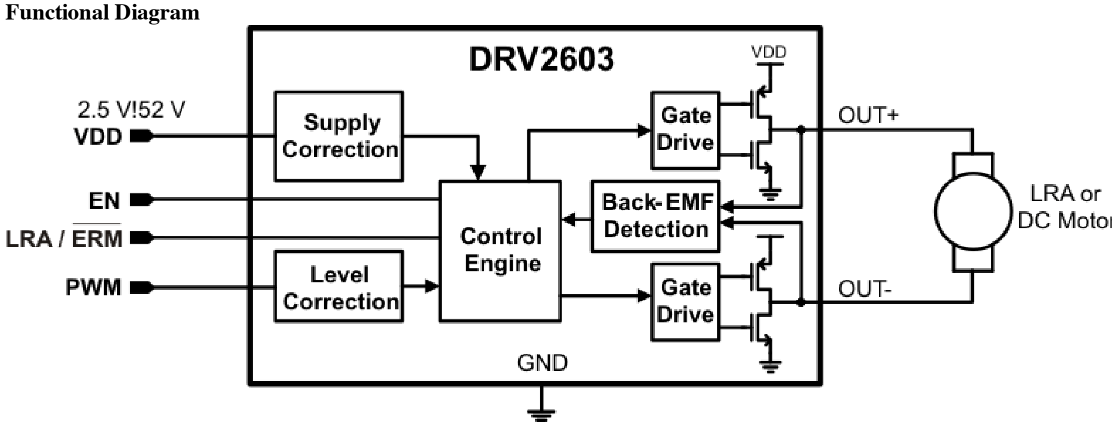


Figure 19: Feedback Unit Schematic (6)

## 8.0. Citations


For Schematics:

(2) http://www.ti.com/lit/ds/swrs084e/swrs084e.pdf (Page 24)

(3) http://invensense.com/mems/gyro/documents/PS-MPU-6000A.pdf

(4) http://www.sparkfun.com/datasheets/Sensors/Flex/FLEXSENSOR(REVA1).pdf (Page 2)

(6) http://www.ti.com/lit/ds/symlink/drv2603.pdf (Page 1)

Other Sources:

http://www.oksolar.com/led/led_color_chart.htm

(7) http://bilgin.esme.org/BitsBytes/KalmanFilterforDummies.aspx

(5) http://www.cs.ucla.edu/~jenn/courses/F11/lecture8.pdf

http://www.kpsec.freeuk.com/components/led.htm