Remote-Controlled DJ

ECE445 Design Document

Spring 2018

Team 12

Yifei Teng (teng9)

Jie Du (jiedu3)

Ningkai Wu (nwu10)

TA: *Jacob Bryan*

1. Introduction

1.1 Objective

The objective of this project is to deliver a remote-controlled DJ package. The problem we address is that by moving the wrist, the DJ will be able to remotely change the song and manipulate it with various DSP functions. We want to make the resulting music interesting. Our design will comprise a hardware component of wrist-strapped gesture sensor and a software component of motion analysis algorithms and signal processing algorithms. And all these interactions will happen over a Bluetooth connection, feeing the DJ from his station.

1.2 Background

In a party, the DJ is the guy responsible for supplying everyone an endless stream of entertaining music. But we all know he deeply wants to join the party! So we will build a remote gesture controlled DJ console that every DJ can take into the action. Due to the incorporation of physical turntables and hardware implementation of signal processing functions, DJ stations often cost upwards of $1000 dollars [7], and are very heavy and cumbersome to transport. We want to miniaturize the DJ for the average user, and at the same time add creative ways to interact with the sound. The primary uses of DJ has been to stream through prepared playlists. Therefore having a phone app provides the reasonable convenience of accessing the user's existing playlists. Through the gesture control offered by our wrist-strapped device, we achieve multiple dimensions of interaction that will be more intuitive and direct than the myriad of buttons and sliders on a DJ station.

1.3 High-level requirements

- Requirement 1: The device will detect the orientation of the hand with an accuracy of +/- 10 degrees.
- Requirement 2: The device will detect "one-shot" gesture events e.g. skipping songs by horizontally throwing the wrist outwards with a success rate greater than 90% and less than two false positives in ten minutes.
- Requirement 3: The device will measure the relative change in height with an accuracy of +/- 8 centimeters.

2. Design

Our system comprises two parts:

1. **A compact device** that straps to one's wrist and collects gesture information. The gesture can be used to navigate a playlist, change various effects, manipulate voice recorded from a microphone etc. It will also sample the onboard inertial measurement units at a rate of 500Hz, estimate orientation from the measurements, and send over the processed result at a lower rate of 20Hz.

2. **A phone app** that implements the various signal processing functions and outputs the music. The app is driven by gesture data from the embedded device.

Details

To send gesture data from the device to the app, we use the Bluetooth Low Energy protocol. The embedded device will contain a power module, a bluetooth module, sensors (IMU and barometer), a microcontroller and a few buttons and LEDs. The microcontroller mounted on the device fuses sensor data (using Kalman filters) to estimate the pose of the hand, in the form of orientation and change in height. We will define a custom protocol to stream these events along with continuously changing gesture data to the phone, which will make use of these data to perform signal processing tasks. In addition, the phone will record the user's voice through a microphone and mix it into the final audio. The microchip on the embedded device will need to be reasonably powerful to perform sensor fusion.

Specifically, we have three modes to control the phone app. Different hand motions are mapped to different musical effects. The mapping is as follows:

All modes: Turning of wrist maps to texture change. (Like turning the texture knob on a guitar amp). Slicing your hand horizontally outwards through air moves to the next song. Slicing inwards goes to the previous song.

Mode 1: Height maps to pitch. Up/down rotation of palm maps to reverb.

Mode 2: Sudden upwards rotation of palm maps to activating looping, sudden downwards rotation maps to stopping the looping.

Mode 3: Height maps to wah-wah effect.

We'll use buttons for switching between modes, and for on/off.

2.1 Block Diagram

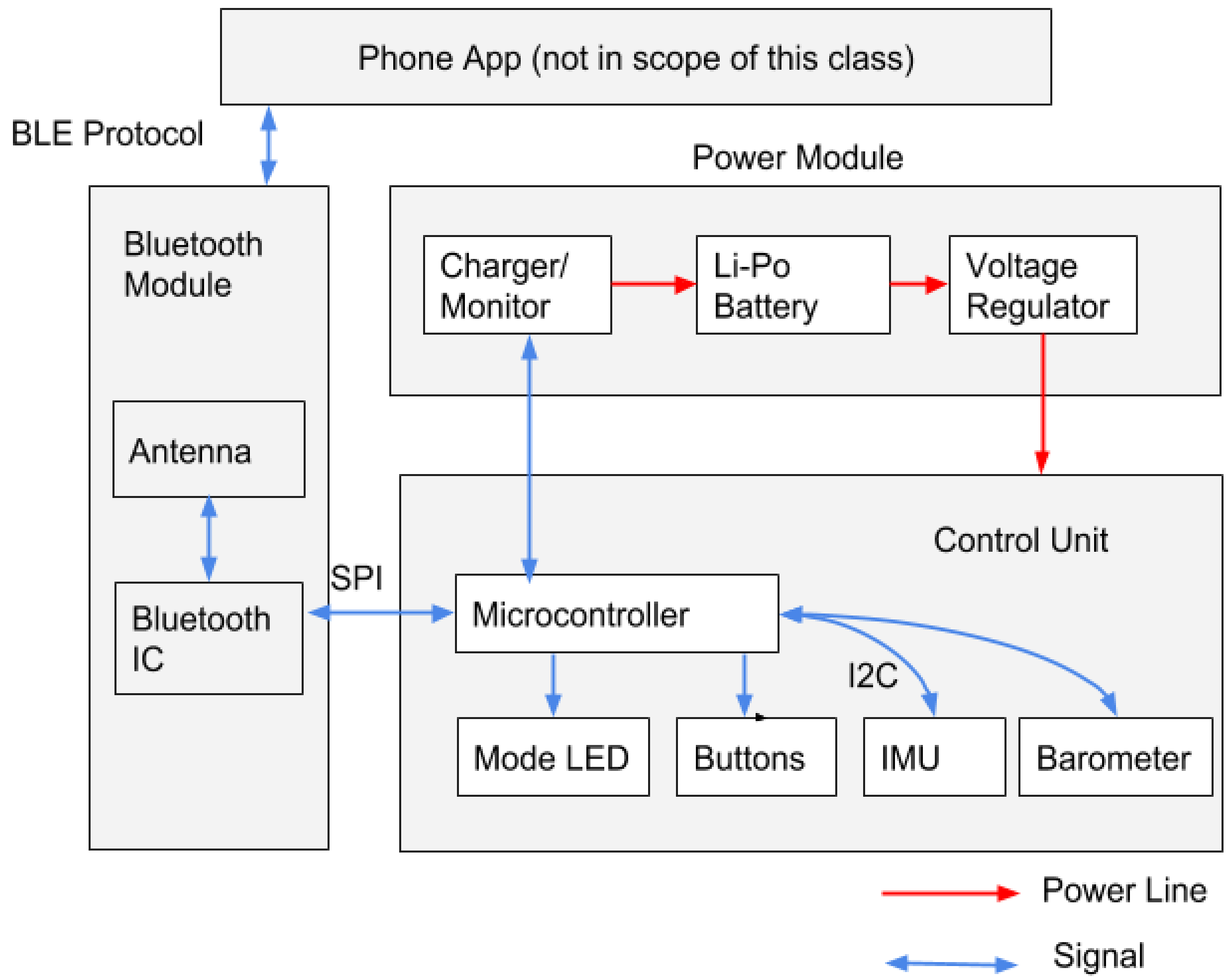

Fig 1. Block Diagram

2.2 Physical Design

The device should comfortably strap onto the user's wrist. We will design a casing for our electronics that partially matches the shape of an iPod nano, so it fits in one of the off-the-shelf exercise phone holder wristbands (Fig. 2). The design consists of two halves, each of which will be 3D printed and pieced together to form the casing.

Referring to Fig. 3, we plan to place the battery in between the two M3 nut holders, glued to the bottom part of the casing. The PCB will be mounted on top of the M3 holders using screws. Holes will be carved out of the front panel as needed to expose buttons and LEDs.

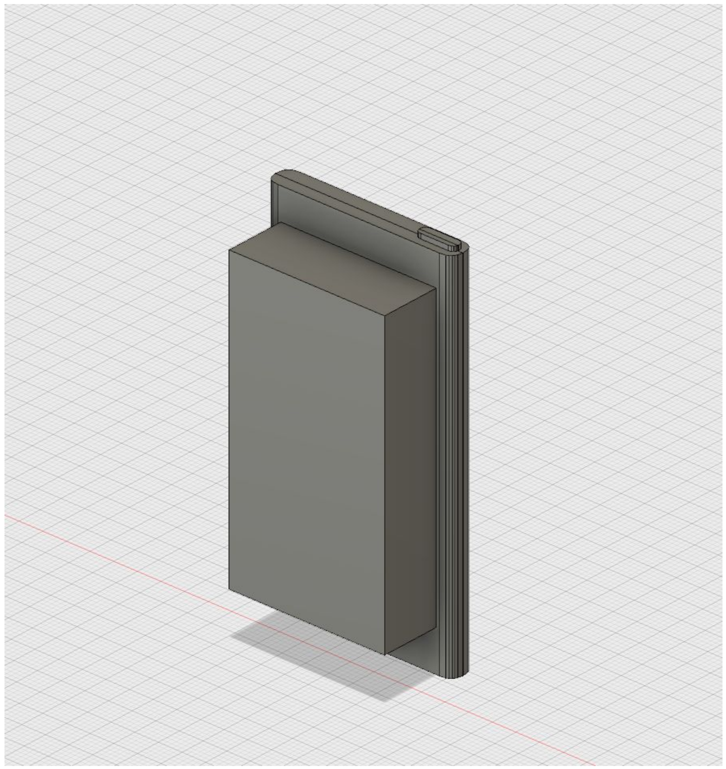

Fig 2. 3D External View of Physical Design

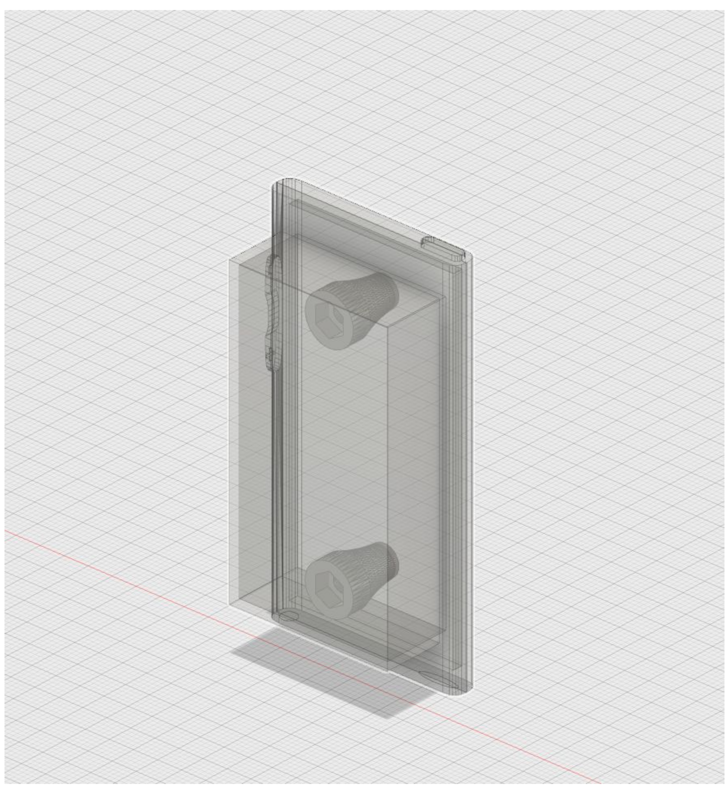

Fig 3. 3D Internal View of Physical Design

Fig 4. Side View **Fig 5. Front View**

We intend to place the battery pack and the charging module inside the iPod-shaped body, while placing the main PCB in the rectangular extrusion.

Now we will discuss the functional overviews and block requirements of each element in our block diagram.

2.3 Power Module

Power module includes battery charger and monitor, Li-Po battery, and a voltage regulator.
Li-Po battery will be used as a power source to keep the communication network up continually, and it is boosted to 5V by a boost converter to power the microcontroller. And the regulated voltage 3.3V will be used to power all control unit except microcontroller. We can charge Li-Po battery from a USB port(5V) and monitor the voltage of battery continuously while charging and discharging.

2.3.1 Charger/Monitor

We plan to use MCP73831 charge management controller to charge the battery with supply voltage 5V from a USB port, and a voltage divider circuit that communicates with microcontroller to monitor the voltage of our Li-Po Battery.

Requirement	Verification
1: When the battery is fully charged from a USB port(5V), the voltage of battery should be between 4.15V and 4.2V.	1: After battery is fully charged(LED of MCP73831 lights up), measure the voltage of battery by voltmeter and ensure it is within our desired range.
2: Monitored voltage by microcontroller lies +/- 2% from the actual voltage of battery.	2:When we charge and discharge the battery, we can read voltage from microcontroller and measure the voltage by voltmeter at the same time. Compare these two values and make sure percentage difference between these two values is less than 2%.

2.3.2 Li-Po Battery

The lithium-Polymer battery is used to power control unit..

Requirement	Verification
1: The battery must be able to store enough charge to provide at 3.3-4.2V for current draw 150mA at least five hours continuously.	1: A: Fully charging the battery. B: Discharging the battery at 150mA for five hours. C. Measure the voltage of battery by voltmeter and make sure it is above 3.3V.

2.3.3 Voltage Regulator

This block composes of a 5V boost converter to power microcontroller and a voltage regulator that supplies 3.3V to other elements in the control unit. These two chips must be able to handle the peak input from the battery (4.2V) at the peak current draw (150mA).

3.3V Voltage Regulator	
Requirement	**Verification**

1: *The voltage regulator must provide 3.3V +/- 5% from a 3.3-4.2V source, under any current load up to 150mA.*	1: A: Connect battery and voltage regulator, and supply 150mA current. B: Measure the output voltage of regulator and make sure percentage difference between measured value and 3.3V is less than 5%.
2: *Must maintain thermal stability below 125°C at a peak current draw of 150mA.*	2:Measure the temperature of voltage regulator by IR thermometer with 150mA current.

5V Boost Converter	
Requirement	**Verification**
1: *The boost converter must provide 5V +/- 5% from a 3.3-4.2V source, under any current load up to 150mA.*	1: A: Connect battery and boost converter, and supply 150mA current. B: Measure the output voltage of regulator and make sure percentage difference between measured value and 3.3V is less than 5%.
2: *Must maintain thermal stability below 125°C at a peak current draw of 150mA.*	2: Measure the temperature of boost converter by IR thermometer with 150mA current.

2.4 Control Unit

The control unit is responsible of reading accelerometer, gyroscope, and magnetometer values from the Inertial Measurement Unit. It performs sensor fusion to estimate the orientation of the hand.

2.4.1 Inertial Measurement Unit

We plan to design a similar circuit to [8], which internally uses the MPU-9250 chip.

Requirement	Verification
1: *The device will detect the acceleration of the hand with an accuracy of +/- 8 mg-rms.*	1. A: Set up IMU and microcontroller on a breadboard. B: Hang the breadboard with a string in a windless environment and hold still. C: Read out the acceleration value from microcontroller directly and make sure the magnitude is within 9.8g +/- 8 mg rms.

2.4.2 Altimeter/Barometer

We plan to use the MS5611-01BA barometer [9], which claims to have a high-resolution module of 10 cm and fast conversion down to 1 ms.

Requirement	Verification
1: *The device will measure the relative change in height with an accuracy of +/- 20 centimeters.*	1: A: Set up barometer and microcontroller on a breadboard. B: Fix the breadboard on hand and read the initial height. C: Move hand 1 meter up and read out the value from microcontroller. D: Make sure this value is within 15 cm from the initial height. E: Repeat B to D for 1 meter down.

2.4.3 Microcontroller

The microcontroller we plan to use is NXP1768. It is a Cortex-M3 processor powerful enough to perform sensor fusion while handling Bluetooth data transmission. We plan to implement a Kalman filter on the IMU data to update orientation continuously. Since the MPU-9250 is capable of sampling at a maximum rate of 500 Hz, we plan to run the algorithm at this rate.

Requirement	**Verification**

1: *The orientation estimation algorithm must be accurate within 5 degrees after stabilizing for 10 seconds.*	1: A: Set up IMU and microcontroller on a breadboard. B: Fix the breadboard on hand and rotate our hand in x-axis (world axis) with a degree of 45. C: After 10 seconds, get the output value in yall, pitch and roll form from microcontroller and convert them into degrees. D: Make sure the difference in each axis is within 5 degrees. E: Continue B-D with a rotation in y, z axis with a degree of 45. F: Try out a degree of -45 in x, y, z axis and make sure the difference in each axis is still within 5 degrees.
2: *The drift in orientation must be less than 15 degrees after 5 minutes of active use.*	2: A: Set up IMU and microcontroller on a breadboard. B: Fix the breadboard on hand and after stabilizing, randomly rotate hand in x, y, z axis (world axis) for 5 minutes. C: Rotate hand to the original stabilized position. Read out the yall, pitch and roll values from microcontroller and convert them to degrees. D: Make sure the the rotation in x, y, z axis is within 15 degrees.
3. The microcontroller must detect gestures as indicated by a particular pattern in the change in orientation, and report to the phone app with a latency less than 1 second.	3: A: Set up IMU, microcontroller and bluetooth module on a breadboard. B: For “next song” function, fix the breadboard on hand and slice hand horizontally outward. C: Make sure the phone can recognize the gesture in 1 second by outputting the time difference between the time of sliding and the time phone receives it. D: Repeat the same process for all gestures mentioned in mode functions.

4. The microcontroller must detect changes in height with an accuracy better than 15cm.	4: A: Set up IMU, barometer and microcontroller on a breadboard. B: Fix the breadboard on hand and read the initial height. C: Move hand 1 meter up and read out the value from microcontroller. D: Make sure this value is within 15 cm from the initial height. E: Repeat B to D for 1 meter down.

2.4.4 Buttons

We need four buttons in total. Three of them are used for mode switch and the fourth one is used for on/off.

Requirement	**Verification**
1: *The buttons are required to be robust and persistent with a success rate greater than 90%. Success means our microcontroller is be able to detect it and change modes accordingly if any button is pressed.*	1: A: Connect three buttons with our microcontroller. Make sure the microcontroller is able to receive individual input from any button. B: For each button, press the button 100 times and make sure the true positive detections are more than 90 times.

2.4.5 Mode LED

We need three yellow LEDs to represents three different modes. Each LED is responsible for one mode. If our device is off, all LEDs are supposed to be turned off.

Requirement	**Verification**
1: *The LEDs are required to be able to last for at least five hours.*	1: A: Connect our microcontroller with a single LED and 330 Ohms resistor. Make sure the LED is able to be visible for at least five hours continuously.

2: *The LEDs are required to be visible to indicate which mode is currently on.*	1: A: Connect our microcontroller with three LEDs, three 330 Ohms resistors, and a button for each LED. Make sure the LED is visible once its button is pressed and we can tell the difference between the ON LED and OFF LED. B: Make sure at most one LED is able to be on at one time.

2.5 Bluetooth Module

The Bluetooth module we plan to use is Adafruit Bluefruit LE SPI Friend - Bluetooth Low Energy (BLE). Data from the control module is sent to the module via data bus and then received by phone app via BLE protocol.

Requirement	**Verification**
1: *The Bluetooth module is required to be able to send data to phone within 10 meters.*	1: A: First connect bluetooth to phone within a short range, e.g. 1m. B: Increase the distance from the initial distance up to 10m and make sure the connection is stable and the phone can still receive complete data sent by microcontroller.
2: *The Bluetooth module is required to be able to send data to phone within 0.5 second.*	1: A: Connect bluetooth to phone within a short range, e.g. 1m. B: Output time1 from microcontroller before it sends fused data to bluetooth. Output time2 from phone after it receives fused data from bluetooth. Make sure the time difference is smaller than 0.5 second. C: Increase the connection range from the initial distance up to 10m and make sure the time difference is still within 0.5 second range.

2.6 Schematics

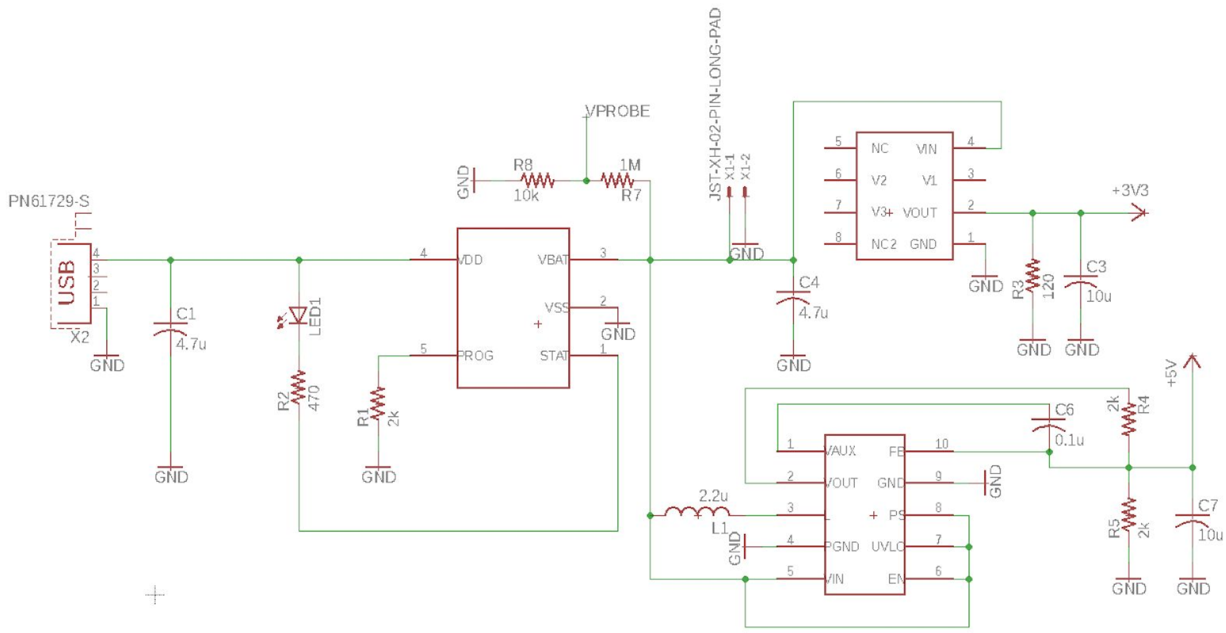

Fig 6: Power Module

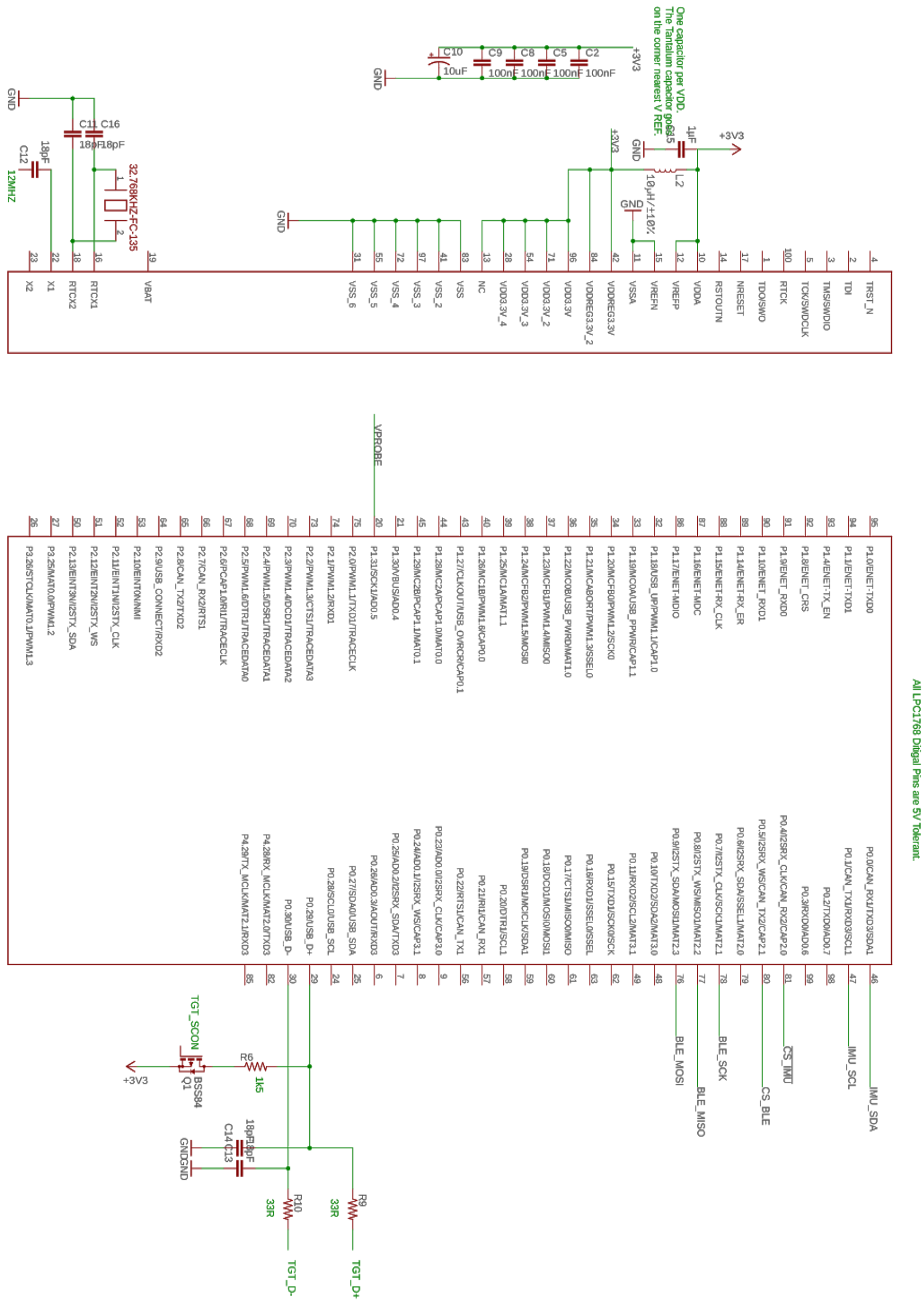

Fig 7: Microcontroller Module

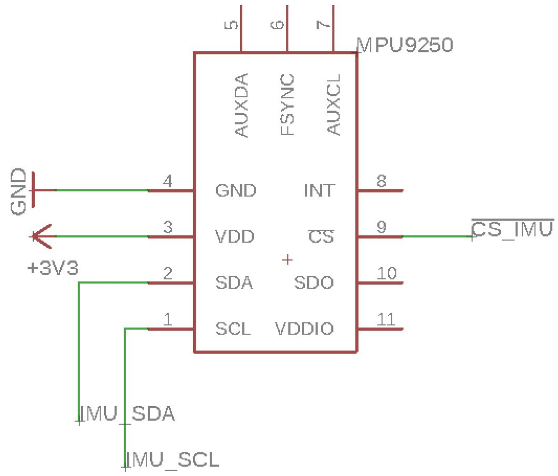

Fig 8: Inertial Measurement Unit

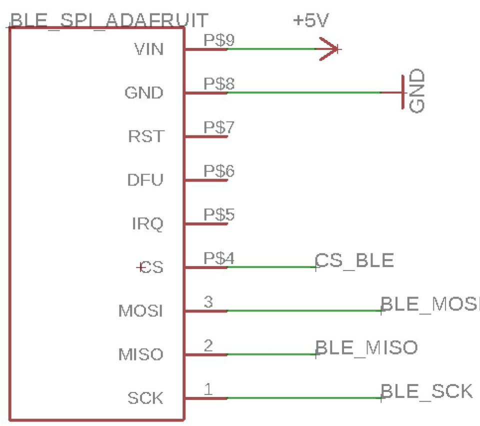

Fig 9: Bluetooth Unit

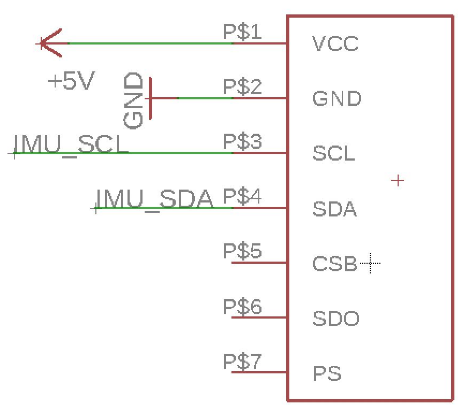

Fig10: Barometer Unit

2.7 Risk Analysis

Our project integrates both IMU and barometer/altimeter sensor values to determine the movement of one hand. Therefore, the greatest risk in our project is whether our sensors are able to provide precise values, e.g. hand orientation and its change in height. At the meantime, these sensors need to respond fast enough if user moves hand fast. If the movement is determined correctly, then the rest of design is straightforward.

2.8 Tolerance Analysis

There are many factors which affect the operation of our system. The most critical one is the accuracy of sensor fusion algorithm responsible for calculating the orientation of the arm. We will analyze backwards from the design requirement to obtain a first order estimate that can serve as minimum tolerances for the sensors. Referring to high level requirement 3, we need to estimate the absolute height of the device using a noisy barometer and a noisy accelerometer. By investigating the Kalman filtering behavior, we will work out the desired accelerometer and barometer accuracy. The state of the system can be characterized as:

$$\hat{s} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix}$$

Where $\widehat{s}$ is the estimate of state and x is the vertical location.

Given

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

$$B = \begin{pmatrix} \frac{1}{2}(\Delta t)^2 \\ \Delta t \end{pmatrix}$$

The state-space system mode is given by

$$s_k = \begin{pmatrix} x_k \\ \dot{x}_k \end{pmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} s_{k-1} + \begin{pmatrix} \frac{1}{2}(\Delta t)^2 \\ \Delta t \end{pmatrix} \ddot{m}$$

$$s_k = As_{k-1} + B\ddot{m}$$

Where $\ddot{m}$ is the accelerometer reading.

The state estimate error covariance is

$$P_k = AP_{k-1}A^T + Q_k$$

Where Q_k is the process noise covariance matrix, modelling any external noise in the measurements.

Assuming the accelerometer reading is perturbed by some additive gaussian noise w_k such that $\ddot{m}_k = a_k + w_k$, and the barometer reading is perturbed by some additive gaussian noise n_k such that $m_k = h_k + n_k$, it can be shown that

$$Q_k = BB^T {\sigma_{a,k}}^2 + \frac{{\sigma_{x,k}}^2}{2}$$

Where $\sigma^2_{a,k}$ is the variance in the accelerometer reading, and $\sigma^2_{x,k}$ is the variance in the barometer reading.

We would like to know the variance in height, assuming the system is able to converge to a steady state, and that would be

$$\sigma_{\hat{x}}{}^2 = \frac{\Delta t^4 {\sigma_{a,k}}^2}{4} + \frac{{\sigma_{x,k}}^2}{2}$$

Assume we want the height estimate error to be less than 8 cm with 99% certainty, that requires $\sigma_{\hat{x}}{}^2$ to be less than $(8/2.4)^2$ = 11.1. Under a short enough sample interval, we observe that the primary contributing factor to the variance is the barometer measurement, and it needs to be less than 22.2. The barometer needs to have a reported accuracy of +/-11 cm, which is above our design accuracy of +/-10 cm.

3. Cost and Schedule

3.1 Labor Cost

Assume the labor cost is 45$/hr for each person, and total hours for 3 people to complete the project is calculated below :

Circuit/ PCB Design	Circuit Test	PCB soldering	Data Fusion code	Phone App Design	Integration	Debug	Total Hour
60hr	20hr	24hr	50hr	50hr	40hr	40hr	284hr

Therefore, Total Cost = 284hr * 45$/hr =12780$

3.2 Parts

#	Description	Part Number	Manufacturer	Quantity	Cost($)
1	Bluefruit LE SPI Friend - Bluetooth Low Energy (BLE)	2633	Adafruit	1	17.5
2	3.3V Voltage Regulator	LD1117D33CTR	Mouser	5	2.4
3	Lithium Ion Polymer Battery - 3.7v 500mAh	LP503035	Adafruit	3	7.95
4	3.7V to 5.0V Voltage Booster	TPS61200DRCR	Mouser	3	7.02
5	Battery Management	MCP73831T-2ATI/OT	Mouser	5	3.4
6	Microcontroller	NXP1768	Sparkfun	1	54.95
7	IMU	MPU9250	Sparkfun	1	14.95
8	Altimeter Sensor	MS5611	Amazon	1	13.99
9	Surface Push Button	None	ECE Shop	5	0.4
10	Yellow LED	None	ECE Shop	5	0.2
11	iPod Case	None	Amazon	1	8.59

12	3D Printing	None	Shapeways	1	16.07
Total Cost: $147.42					

GRAND TOTAL = LABOR + PARTS = $12780 + $147.42 = $12927.42

3.3 Schedule

Week	Jie	Ningkai	Yifei
2/19	1: Order parts for power module 2: Work on schematics and calculations	1: Order parts for bluetooth and microcontroller module 2. Microcontroller schematics	1: Design Physical Housing
2/26	1: Design PCB layout, 2: Doing power module circuit tests and modify the schematics if needed	1: Unit testing barometer 2: Write code for bluetooth module	1: Prototype iOS App 2: Unit testing inertial measurement devices
3/5	1: Order first PCB 2: Help debugging bluetooth code	1: Unit testing bluetooth module	1: Validate sensor fusion design on breadboard
3/12	Spring Break	Spring Break	Spring Break
3/19	1: PCB soldering 2: Reorder parts that are needed 3: Test our circuits	1: PCB soldering	1: Validate iPhone integration on breadboard 2: Refine physical design and order second iteration
3/26	1: Make some modifications on first PCB layout and order second PCB iteration	1: Unit test LED and button 2: Make modifications on schematics in needed	1: Integration testing on PCB
4/2	1: Work on gesture detection Mode 1 code	1: Work on gesture detection Mode 2 code	1: Work on gesture detection Mode 3 code
4/9	1: Second round soldering	1: Combine three modes with other functions	1: Fine-tune gesture detection
4/16	1: Final testing and debugging 2: Adding more functions of	1: Unit test microcontroller	1: Finalize product

	our device if we have time		
4/23	1: Prepare for final presentation and demo	1: Rehearse final demo sequences	1: Rehearse final demo sequences
4/30	1: Work on final report	1: Work on final report	1: Work on final report

4. Safety and Ethics

For the safety concerns, we will follow safe battery usage listed in the course website since our project utilizes Li-Po battery and a charge for it. Our charger is an industry made IC device and it shuts charge controller off when charging input is higher than required voltage range, which will reduce the likelihood of hazards while charging. We will also check our circuit before connecting it to battery in order to prevent short circuit which may lead to electric shock. After studying safe battery usage, we learn many potential hazards of over charging, over discharging, and over current. We will check datasheet of our selected Li-Po battery carefully and take caution when we deal with it.

Moreover, in order to make device comfortably strap onto the user's wrist and doesn't fall off when user moves, we will use 3D printing to make a mold to hold our device tightly, and then use armband from the market to attach device to user's wrist.

For the ethical issues, we will be guided by IEEE and ACM code of ethics. We may encounter many problems in the project. When the problems occur, we will not try to pretend the problems are not there and move on to the next task. Based on #5 of IEEE Code of Ethics, "to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors" [1]. Therefore, when we encounter hard problems, we will first discuss in the group and think about the strategies to solve the problem. If we can't solve the problem ourselves, we will talk with our TA for the solution.

We will do a lot of research on the project, and there might be some brilliant ideas from a paper or our classmates that help us with the project. We need to respect ideas of others and cite them in our project report. Based on #1.6 of AMC Code of Ethics, it is very important to" Give proper credit for intellectual property" [2].

Many circuit and performance tests will be made for the project. When we have the test results, we will be honest and write notes and make corrections when we have the incorrect result. Based on #3 of IEEE Code of Ethics,'' to be honest and realistic in stating claims or estimates based on available data''[1]. We will not use unreal data to make our project seem working well.

5. Citations

[1] Ieee.org. (2018). *IEEE IEEE Code of Ethics*. [online] Available at: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed 9 Feb. 2018].
[2] Acm.org. (2018). *ACM Code of Ethics and Professional Conduct*. [online] Available at: https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct [Accessed 9 Feb. 2018].
[3] Manager, S., 1Ah, L., 110mAh, L., 850mAh, L. and 400mAh, P. (2018). *Battery Babysitter Hookup Guide - learn.sparkfun.com*. [online] Learn.sparkfun.com. Available at: https://learn.sparkfun.com/tutorials/battery-babysitter-hookup-guide?_ga=2.93141789.1476657923.1517777758-793387797.1516935255#lipo-fuel-gauge [Accessed 9 Feb. 2018].
[4] MPU-9250, S. (2018). *MPU-9250 Hookup Guide - learn.sparkfun.com*. [online] Learn.sparkfun.com. Available at: https://learn.sparkfun.com/tutorials/mpu-9250-hookup-guide?_ga=2.53729448.1893352856.1518141768-793387797.1516935255 [Accessed 9 Feb. 2018].
[5] Sensor, M. and Sensor, B. (2018). *Barometric Pressure Sensor - TE Connectivity* | *DigiKey*. [online] Digikey.com. Available at: https://www.digikey.com/en/product-highlight/t/te-connectivity-measurement-specialties/ms5611-01ba-barometric-pressure-sensor [Accessed 9 Feb. 2018].
[6] Industries, A. (2018). *Adafruit Bluefruit LE SPI Friend - Bluetooth Low Energy (BLE)*. [online] Adafruit.com. Available at: https://www.adafruit.com/product/2633 [Accessed 9 Feb. 2018].
[7] American Musical. *Pioneer DDJSX2 Professional DJ Controller.* [online] AmericanMusical.com. Available at: https://www.americanmusical.com/Item--i-PIO-DDJSX2?src=Y0802G00SRCHCAPN&adpos=1o1&scid=scplpPIO+DDJSX2&sc_intid=PIO+DDJSX2&gclid=Cj0KCQiAh_DTBRCTARIsABlT9MaC9EYL5T6RwqgdbdXrabY-VJQHX-LJpFrHlEd2KquUM5-4JevxkTUaAjNIEALw_wcB [Retrieved Feb 8, 2018].
[8] SparkFun IMU Breakout - MPU-9250. Available at: https://www.sparkfun.com/products/13762.
[9] Digikey MS5611-01BA Barometric Pressure Sensor. Available at: https://www.digikey.com/en/product-highlight/t/te-connectivity-measurement-specialties/ms5611-01ba-barometric-pressure-sensor.
[10] Adafruit Bluefruit LE SPI Friend - Bluetooth Low Energy (BLE). Available at: https://www.adafruit.com/product/2633