SignalMe

UIUC ECE 445 - Spring 2019 - Project 21 Proposal

Team Members

Derrick Miller

Paloma Contreras Porras

Wonyong Lee

TA

Anthony Caton

1 Introduction

1.1 Objectives

Micromobility is a term referring to all forms of lightweight personal transportation, including, but not limited to, bicycles, skateboards, scooters, and rollerblades, along with electric variants. To be safe for use on the road, all of these modes of transportation require intuitive and standardized signaling and lighting. Unfortunately, this is currently not the case. While regulations for bicycles require certain forms of forward and rear illumination, signaling currently takes the form of arm based signals. Other modes of transport do not have similar regulations. As such, current micromobility signaling as a whole is fragmented and non standardized compared to equivalent systems on motor vehicles.

We propose to solve this with a universal signaling vest usable for all forms of micromobility. With movement based sensors and explicit button controls, the vest will give users intuitive and unified signaling with the ease of use found in the signal systems for motor vehicles. At the same time, the lighting used will be highly visible and understandable for others on the road using common signal conventions in the form of forward illumination on the chest, brake lighting on the back, and left and right turn signals around the shoulders. As an additional feature only possible with a motion aware wearable, the vest will also provide a high visibility flashing hazard operation that activates when a user has any incidents resulting in them falling to the ground.

1.2 Background

According to the NHTSA, “In 2016, there were 840 bicyclists killed in traffic crashes in the United States” with most deaths occurring during low light night hours [7]. Furthermore, according to the 2012 National Survey on Bicyclist and Pedestrian Attitudes and Behaviors [15]. Visibility and safe riding practices play a large role in reducing these kinds of incidents. To ride safely, riders should use cumbersome hand signals that many motorists should also be aware of. Even though the number of cyclists that use signal lighting in the US has not been identified, a case study carried out in London [16] showed that the percentage of cyclists riding with both front and rear lights lit was of 25% in a minor road and 58% in a major road. We can assume that the statistics for the United States will be similar to those in the UK. During a time where many people are beginning to use ride-share programs such as Veoride, Bird, and Lime, the development of a safe intuitive signal system can be part of an effort to reduce the number of these incidents despite increasing utilization rates.

Currently, on the market, there do exist electronic signal systems for those on bikes in the form of backpacks, gloves, saddlebags, and more. These systems allow for signal initiation at the push of a button. However, the current solutions are in no sense of the word “smart.” The turning off of a signal is based either on timers or manual user input which either risk a signal turning off in mid-turn, or requiring more effort than existing hand signals. Additionally, none of these systems are usable on modes of mobility other than bicycles. SignalMe aims to bring motion tracking to the area of micromobility signaling to make the act more like signaling in a car or other motor vehicle.

1.3 High-Level Requirements

1. **Requirement 1:** The vest must automate road signalling to the level of a car for both brake lights and turn signals. Remotely initiated turn signals should be turned off after the user maintains a straight path for ten (10) feet after a right angle turn or lane change. A brake light should also turn on for at least a second at some point during a rider stop at the 50 feet before an intersection.

2. **Requirement 2:** The vest must provide street legal night lighting for all signals. All lighting should be visible from 500 or more feet away in clear night conditions.

3. **Requirement 3:** The vest must have a reasonable amount of battery life. Under night conditions the vest should remain fully functional for at least two (2) hours off of a single full battery charge.

2 Design

2.1 Block Diagram

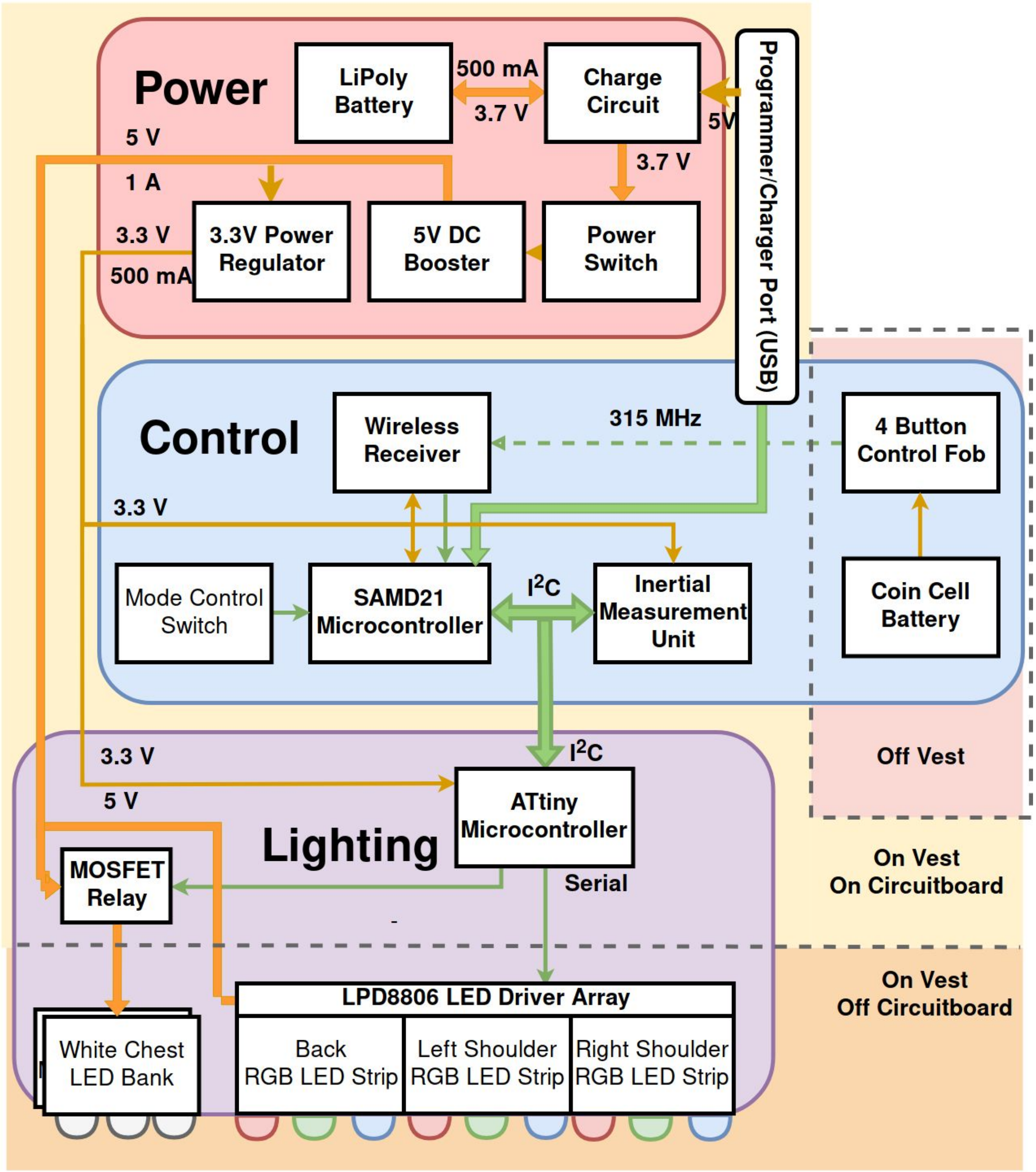


Fig 1. Block Diagram

2.2 Physical Design

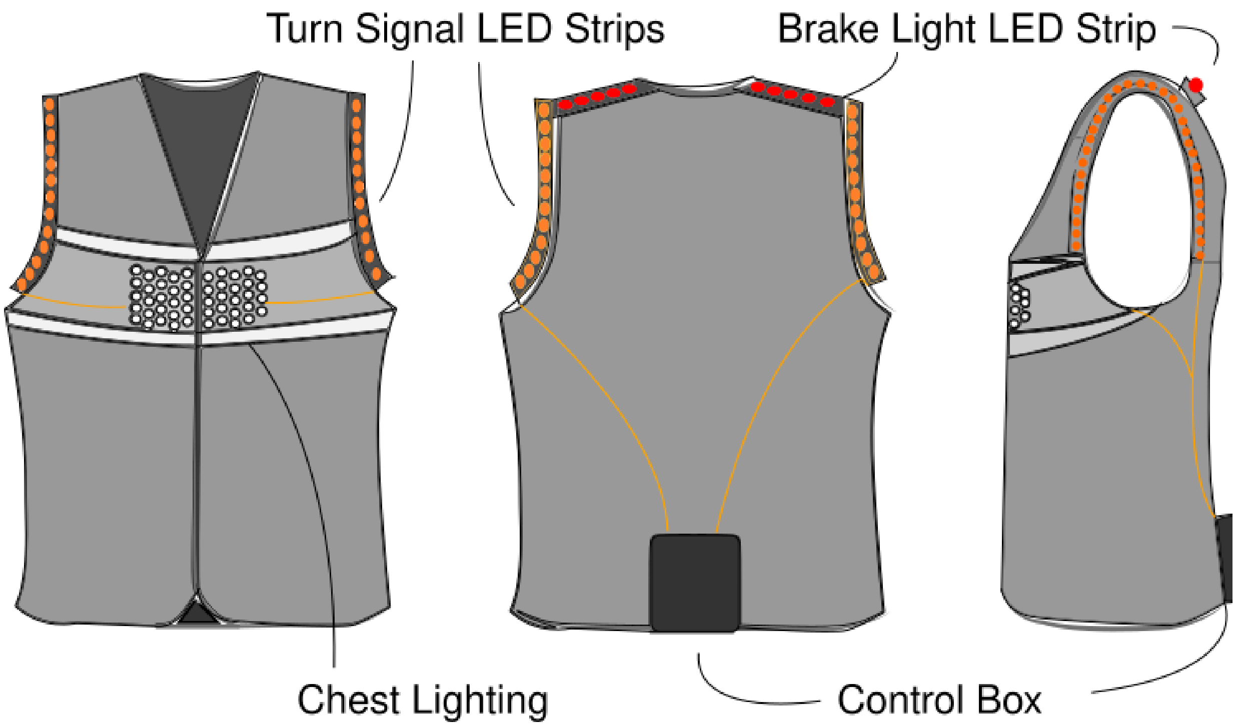


Fig 2. Physical Concept Diagram

The entire device is integrated into a one size fits all vest. Over the shoulders and along the upper back are RGB LED strips with each light individually controllable. The upper back strip is primarily for brake signals and will usually be red while the shoulders when signaling a turn will blink an orangish yellow. On each side of the chest is a bank of white LEDs for forward lighting. All of these are wired to a control box on the lower back. This water resistant lightweight hard plastic box contains the sensors, microcontrollers, and battery system. For user input, a 4 button keyfob is is held in the hand or affixed to handlebars to initiate turns, toggle hazard mode, and toggle the chest lighting.

2.3 Control Block Descriptions

The control subsystem processes button and sensor input to produce control messages for the lighting subsystem. A microcontroller and inertial measurement unit are used to determine a user's positional, velocity, and acceleration vectors. In conjunction with button inputs from the user, control messages are sent to the lighting controller over I^2C when the controller determines a change in signal state.

2.3.1 Sensor Microcontroller

The main microcontroller, decided to be an Atmel SAM D21, is connected to the IMU and Light Controller over a common I^2C bus and to four (4) momentary button control signals. This controller will process IMU data to determine a user's movement path after button initiated signals. Simultaneously, acceleration and velocity are tracked to notice and signal any significant de-accelerations. In the event of any signal state changes, I^2C commands are sent to the lighting controller to request a change in lighting. It lastly uses a mode controlling switch to change between different modes for different forms of transportation (i.e. Sideways oriented modes such as Skateboards vs. forward oriented modes such as bikes and scooters).

2.3.2 Inertial Measurement Unit (IMU)

For this iteration of the project, we use the MPU-9250, a 9 Degrees of Freedom Inertial Measurement Unit (IMU) consisting of an accelerometer, gyroscope, and magnetometer in a single package. This specific IMU comes with the microcontroller we want to use.

Read and controlled by the sensor microcontroller over I^2C, the IMU package will provide rotational data from the gyroscope, linear data from the accelerometer, and cardinal directional data from the magnetometer to be further processed by the microcontroller. As micromobility movements are relatively slow, low full-scale range sensors will allow for precise readings over small movement ranges.

2.3.3 Button Controls

Four (4) momentary button controls allow the user to initiate signals for **hazard**, **left**, and **right** turn signals as well as **toggling the chest lights**. These button controls are needed as, legally, turn signals need to be initiated ~100 feet before the location of the turn itself. A receiver wired to the sensor controller over general purpose digital I/O. To send signal, a 4 button transmitter akin to an automotive key fob will be used.

2.4 Lighting Block Descriptions

The lighting subsystem takes commands from the sensor controller over I^2C to produce appropriate light patterns on the shoulders, back, and chest. Chest lighting gives street legal forward white light while the back and shoulders produce different lighting patterns based on the current signal state.[1]

2.4.1 Light Microcontroller

A low power microcontroller, decided to be an 8-bit AVR controller from the ATtiny line, is interfaced to the main sensor controller over I^2C to take commands for producing various lighting patterns. It is interfaced to the signal lighting over LPD8806 drivers with a clock and data line for each strip. It is interfaced to chest lighting via a pair of MOSFET relays.

2.4.2 Shoulder Light Strips

LPD8806 based Weatherproof RGB LED strips allow for individual LEDs to be separately addressed and controlled by the light microcontroller. The shoulder light strips specifically are for signaling turns with a blinking yellow light common in motor vehicles.

2.4.3 Back Light Strip

Using the same LPD8806 based RGB LED technology as the shoulders, the back light strip provides both rear ambient red lighting and bright red lighting in the case a brake level de-acceleration is detected by the sensor controller.

2.4.4 Chest Lights

To be a comprehensive lighting solution, forward illumination is provided by bright white LEDs. The operation of these lights are controlled by MOSFET relays controlled by the lighting controller.We will make our chest lights to be at max 1200 lumens by combining multiple white LEDs and setting reflecting mirrors at the back of each LEDs to fulfill the bicycle regulations about front light visible in 500 ft.

Considering 60 Watt light bulb and 15W CFL bulb produces 800 lumens, brightness of light bulb and CFL bulb at 500 feet is 0.002724 Lux. To be more visible than with brightness of the light bulb, we will have lux of 0.004.

Inverse Square Law :

$$\mathbf{B} = L\,(4\pi\, d^2)^{-1}$$

B = Brightness

L = Luminosity

d = Distance

B(L = x, d = 152.4 m) = 0.004 lux, L = 1166.86 lumens

2.5 Power Block Descriptions

The power subsystem allows for battery operation of the vest, charging of the on board battery, and distribution of power to the appropriate subsystems in the form of direct battery potential and regulated 3.3V for the microcontrollers and sensors.

2.5.1 Battery

While the exact battery storage value is unknown until we determine system draw in practice, the system will require at minimum 3.3 V at running current for at least 2 hours to maintain the microcontrollers. A single-cell Lithium-Polymer (LiPo or LiPoly) has a nominal voltage of 3.7 V, and safe and easy to charge while connected to the device, and is common for wearables of this scope.

2.5.2 Power Distribution

Consisting of a power switch a 5V booster, and 3.3 V regulator, the power distribution system provides a regulated 3.3 V for the microcontrollers and sensors while providing a safe 5V potential for the rest of the lighting.

2.5.3 Battery Charger

LiPo cells are fairly particular about their charge circuits. Dedicated LiPo charge controllers (such as the MCP73831) exist that can take an input voltage and provide system power while charging a wired up battery. This charging port will take the form of a small form factor USB connector.

2.6 Functional Overview

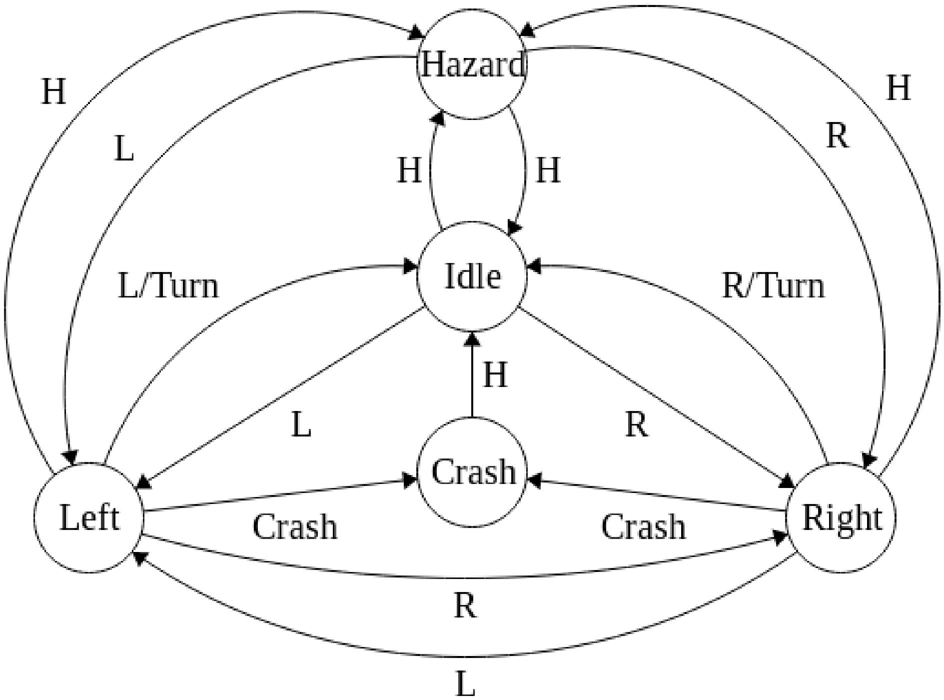
2.6.1 System States

After startup, the vest operates within one of the following states governing sensor and lighting behavior.

1. **Idle** - This is the main state the vest is in when not doing any signaling, it has an ambient dim rear red light that goes bright on brake detection and the chest lighting can be toggled with a button. The sensor controller constantly uses accelerometer data to dead reckon an approximate rider speed using poweron state as zero velocity. Otherwise the system waits for user input or a crash for any state changes.
2. **Left Signal** - This state corresponds to a left turn signal with a blinking yellow left shoulder. Upon triggering, the sensor controller will mark the current location as the origin of the turn. Using accelerometer and gyro data, rider displacement is tracked until the rider is laterally displaced by at least 10 feet (the lower end of standard lane width) to the left from the origin. From here the controller waits for a stabilization of riding direction for at least ten feet before automatically transitioning back to idle. The user may of course move back to idle manually by pressing the left signal button again.
3. **Right Signal** - Equivalent to the Left Signal but using the right shoulder for signal and mirroring detection along the sagittal plane.
4. **Hazards** - Equivalent to a motor vehicle's hazard lights. Shoulders and back both do a slow bright blink. Useful if a rider is off on the shoulder of the road or otherwise in traffic while not actively riding.
5. **Crash** - In the case the sensors detect a vertical drop greater than 2 feet followed by very sudden stop, it is likely the rider had an incident and is now on the ground. The vest will go into a flashing mode similar to the Hazard state but with more rapid flashing including flashing chest lighting. The only way to return from this state to idle is by holding the hazard button down for more than 1 second.

2.6.2 State Transition Diagram

The states are controlled by a combination of user entry via key fob buttons and key events detected by sensor processing. The relation of the system states via these inputs are given in **Fig 3**:



Key	H	R	L	L/Turn	R/Turn	Crash
	Hazard Button	Right Button	Left Button	Left button or Turn	Right Button or Turn	Crash Detected

Fig 3. State Transition Diagram

2.6.3 State to Lighting Encoding

Each State encodes to a specific set of lighting behaviors given in **Table 1**. Of note, not all lighting behaviors are completely controlled by the system state. Brake Signalling operates in its own two state operation based on user movement regardless of signal state. Similarly, chest lighting is toggleable in all non crash states and maintains its toggle state regardless of signal state.

States to Lights	Back Light	Left Shoulder	Right Shoulder	Chest
Idle	Dim Red[1]	Off	Off	Toggleable White[2]
Left Signal	Dim Red[1]	Blinking Yellow	Off	Toggleable White[2]
Right Signal	Dim Red[1]	Off	Blinking Yellow	Toggleable White[2]
Hazards	Blinking Bright Red	Blinking Yellow	Blinking Yellow	Toggleable White[2]
Crash	Rapid Red and Yellow	Rapid Red and Yellow	Rapid Red and Yellow	Rapid White

[1] Becomes steady bright red upon detection of a brake.

[2] Holds toggle state between non crash state transitions.

Table 1. State to Light Encoding

2.7 Software Design

2.7.1 System Control: The SAMD21

The purpose of the SAMD21 sensor microcontroller is to take user and sensor input, process it, and output lighting commands to the lighting microcontroller in a way that implements the states in the Functional Description (Section 2.6) through the following general architecture:

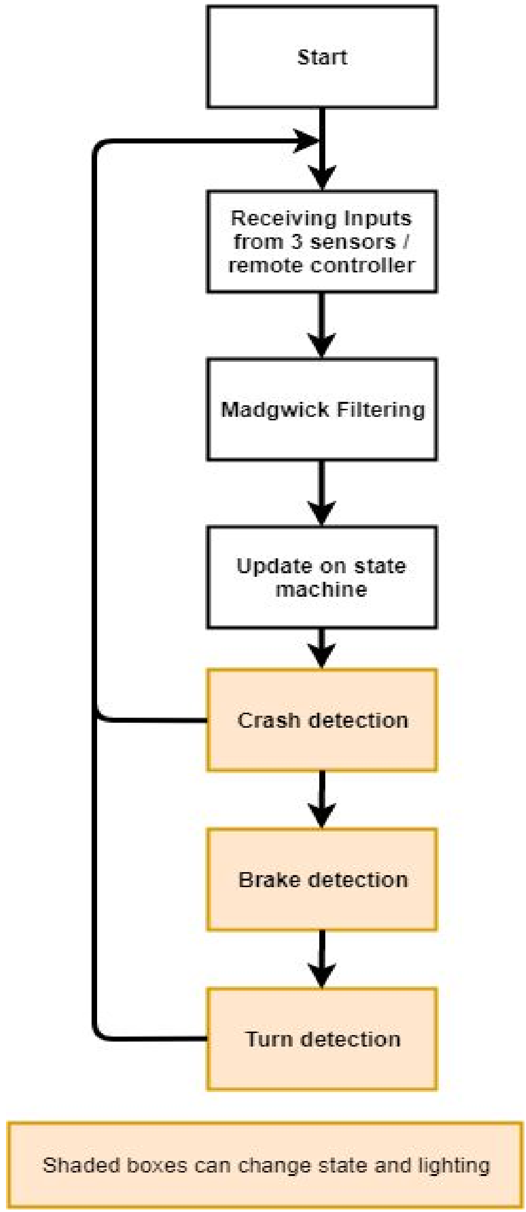


Fig 4. Software Flowchart

2.7.2 Lighting Control: The ATtiny85

The ATtiny85 acts as an I^2C device allowing the the sensor microcontroller to control segments of lighting through a set of registers. The purpose of this is to let the lighting have features such as flashing patterns without having the sensor microcontroller worry about any timing, keeping software design relatively simple. At the same time, the lighting micro controller need not know anything about the sensor situation. The following is the register map for lighting control.

Substrip	Front Shoulder	Back Shoulder	Upper Back	Chest Lighting
Left Side Base Address	0x00	0x10	0x20	0x30
Right Side Base Address	0x08	0x18	0x28	0x38

Table 2. Base Address Listing

Offset	Purpose	Parameter Format
0x00	Color 1 Red	8 Bit Unsigned Integer
0x01	Color 1 Green	8 Bit Unsigned Integer
0x02	Color 1 Blue	8 Bit Unsigned Integer
0x03	Color 2 Red	8 Bit Unsigned Integer
0x04	Color 2 Green	8 Bit Unsigned Integer
0x05	Color 2 Blue	8 Bit Unsigned Integer
0x06	Pattern	8 Bit Enum (See Table 4)
0x07	Pattern Parameter	8 Bit Unsigned Integer

Table 3. Register Offset Listing

Pattern Enum Value	Name	Parameter Use	Purpose
0x0	Off	No Use	Idle
0x1	Steady Color 1	No Use	Brake Light, Chest Lighting
0x2	Alternate Full Between 1 and 2	Milliseconds * 10 Between Colors	Turn Signals, Crash Mode
0x3	Alternate Along Strip between 1 and 2 and crawl along strip.	Speed of Crawl	Hazards
0x4	Demo Mode (moving rainbow pattern)	No Use	Boot Up Fanfare

Table 4. Pattern Value Encoding and Parameter Settings

Example: Assuming we are in an idle state as described in section 2.6, we may set the lighting controller to signal a left turn (a blinking yellow shoulder) by:

1)	Setting Left Shoulder Front Color 1 Red to 255.	Set **0x00** to **255**
2)	Setting Left Shoulder Front Color 1 Green to 255.	Set **0x01** to **255**
3)	Setting Left Shoulder Front Color 1 Blue to 0.	Set **0x02** to **0**
4)	Setting Left Shoulder Front Color 2 Red to 0.	Set **0x03** to **0**
5)	Setting Left Shoulder Front Color 2 Green to 0.	Set **0x04** to **0**
6)	Setting Left Shoulder Front Color 2 Blue to 0.	Set **0x05** to **0**
7)	Set the pattern to Alternate Between 1 and 2	Set **0x06** to **2**
8)	Set the parameter to the blink timing (assume 750 ms)	Set **0x07** to **75**
9)	Repeat for the Left Back shoulder	Repeat for Base **0x10**

2.7.3 Sensor Calibration and Filtering

What we are mainly concerned about here is that the sensor's information needs to be accurate enough in order to ensure the correct functioning of our device. Given that we are going to use a 9 Degrees of Freedom Inertial Measurement Unit (IMU), consisting of an accelerometer, gyroscope, and magnetometer, we have the tools to make a fairly accurate position estimation.

To improve the functioning of our system, we plan on including an algorithm that will help us reduce the noise and get a better estimation on the position data. There are many solutions that could be used, such as the Mahony algorithm, the Kalman filtering or the Madgwick sensor fusion algorithm. We decided to choose this last method due to the fact that it gets fewer error than the Kalman filter, and is also better than the Mahony algorithm for a more complex Inertial Measurement Unit (9doF) like ours.

As for calibration, we can ensure that the accumulated error at its minimum when we pass our data to our control system. Apart from reducing the error using algorithms, calibration must be done at an ideal instant of time so that we do not include unnecessary error in our system. As the turn signal is triggered manually by the cyclist before they make a turn, this as a good time to zero the systems for calibration. For the calibration, we will use the software from the manufacturer along with a large supply of open and public software available for our cosen model of IMU.

2.7.4 Turn and Lane Change Detection

When a turn signal is used, two possible events should trigger a complete turn:

1) A 90 degree cardinal turn
2) A lane change

While just a gyro or magnetometer can handle the 90 degree turn, a lane change is a little more involved but solving it also solved the 90 degree turn at the same time. In this a turn is divided into 3 phases:

1) Signal Initiated
2) Turn Made
3) Turn Completed

When a turn signal is initiated, the position is marked. From there, the horizontal displacement from the mark point is tracked until a threshold is reached at which the turn is considered "made". After this point the user has likely moved over a lane or make most of their 90 degree turn but the turn is not completed. To have the turn complete and return the vest back to the idle state, the user must maintain a fairly straight path for a set amount of distance to be experimentally determined.

Fig 5. General Turn Detection Framework

2.7.5 Brake Detection

The braking subroutine runs during every main loop regardless of state unless the user is in the crash state. It constantly uses processed IMU data to determine the rate of acceleration (jerk) along the axis of motion, regardless of direction over time.

Each sampling period we will calculate jerk by:

1) Getting the magnitude of the projection of the acceleration vector along the axis of velocity.
2) Subtract the result of 1 by the magnitude in the last iteration
3) Take the result of 2 and divide by approximate time between iterations.

When the result of this process exceeds a critical value, the brake light will engage. When acceleration then goes past a certain value the light will be turned off. While testing will best inform the critical and release value, an estimate can be derived from the common situation of stopping at a stoplight:

Distance Covered:
Assuming a signal is initiated before stopping at an intersection, a full stop should happen within the minimum legal distance to initiate a signal of 100 m, let us say about half. Estimate 50 m.

Starting Velocity:
4.3 m/s [10]

Average Acceleration:
$-(4.3 \text{ m/s})^2 = 2 * (a \text{ m/s}^2) * 50 \text{ m} \Rightarrow a = -0.0462 \text{ m/s}^2$

Time to Brake:
Very rough estimate of 5 seconds from the given velocity.

Average Jerk:
$0.0462 \text{ m/s}^2 / 5 \text{ s} = 0.00924 \text{ m/s}^3$

Based on this analysis, we should start with an engagement jerk somewhere a little below 0.00824 m/s^3 and release acceleration around or greater than -0.0462 m/s^2. The exact value will change with data collection, but our minimum acceleration resolution should be on this general scale to obtain coherent data at this magnitude.

2.7.6 Crash Detection

Like the brake detection, this subroutine is constantly using sampled IMU data to determine an instance of a user crash. As a general model, a crash is the situation of a reasonably large vertical drop followed by a very sudden vertical stop with a high amount of jerk.

To do this, broadly the subroutine will track a “falling” state triggered by a strong enough instantaneous vertical velocity and a “shock” state caused by a very sudden acceleration change in the opposite direction of the falling motion. In the case a falling state is followed by or simultaneous with a shock state, the vest will consider a crash detected and lock into the high visibility crash state with overrides any other signaling.

The exact parameters for this subroutine in particular will likely have to be tuned by data from some form of (safe) testing to avoid false alarms and statistical misses but the following is a theoretical ballpark of the values relevant:

Falling Acceleration:
We assume crashes are not freefall, but fairly close to it. Estimate 9.0 m/s^2
User Hip Height Minimum:
Find a round value less than the male and female adult 50th percentile for hip height but within a standard deviation [8] Estimate 900 mm .
Time to Fall to the Ground:
Hip Height = ½ * **Fall Speed** * t^2, t = 0.447 s
Velocity at Ground Impact:
v = t * **Fall Speed** = 4.023 m/s
Amount of Time for Collision:
We assume a reasonable time to decelerate over the span of the collision. Estimate 0.250 s
Crash Deceleration:
Impact Velocity / **Collision Time** = 16.09 m/s^2

Based on this analysis, we should assign a “falling” criteria after downward velocity exceeds 4 m/s. A shock should be assigned with an upward acceleration greater than 16 m/s^2. If the shock occurs within 250 ms of a falling state, we should initialize the crash state. This also means our accelerometer should be rated with a range magnitude of at least 1.5g.

2.8 Board Schematics

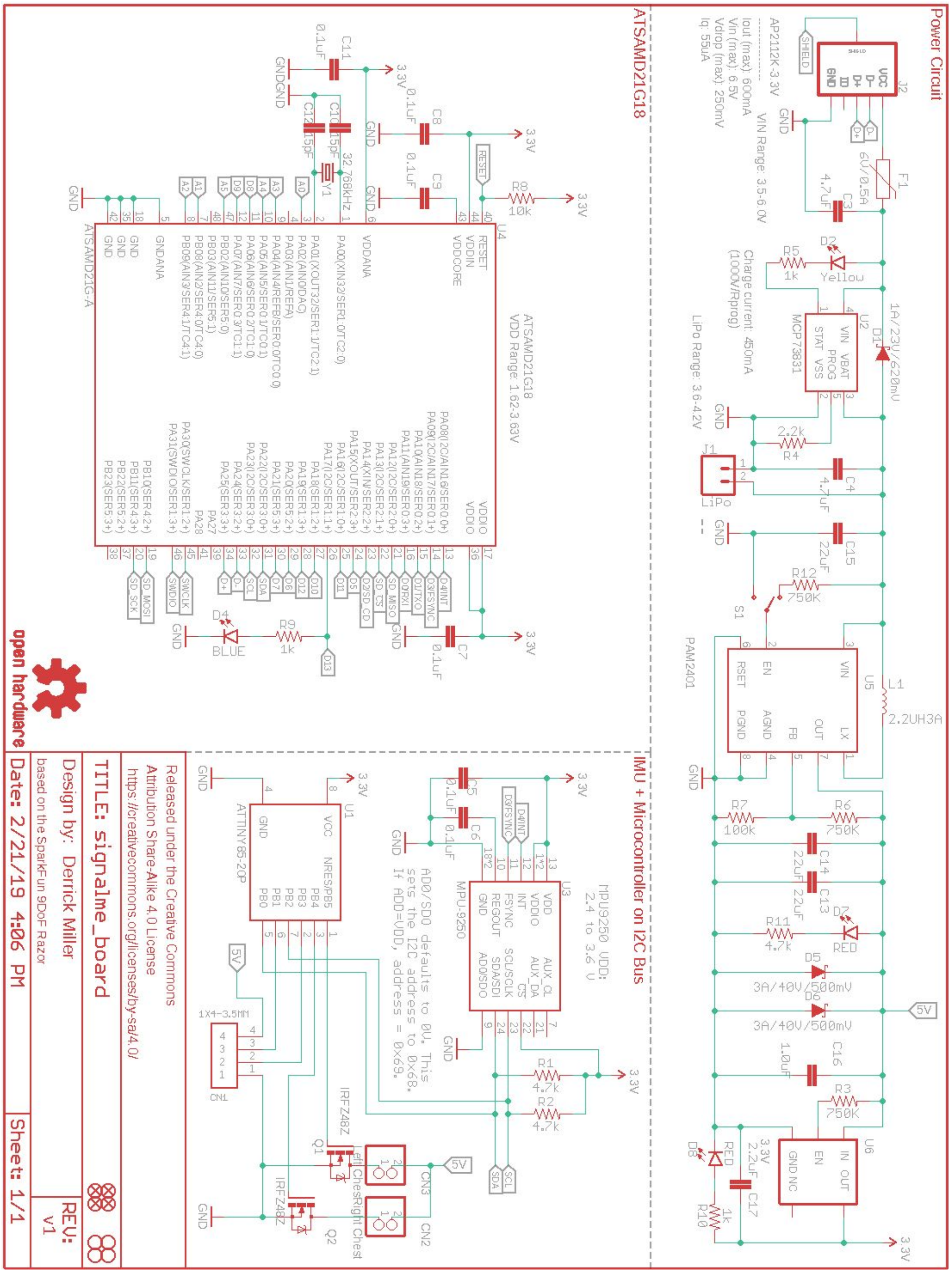


Fig 6. Board Schematics

2.9 Requirements and Verification

Each subsystem largely solves a high level requirement. As such each of a block's requirements go towards the high level goal of the subsystem. **Control** is responsible for determining proper state changes in the system required by **high level requirement 1**. **Lighting** provides street legal brightness and light colors in line with **high level requirement 2**. **Power** then not only allows the other two to happen, but also is burdened with battery life in **high level requirement 3.**

	Block	Requirement and Verification
Control	Sensor Microcontroller	Track at least 1 minute of location/IMU data. ***Verification:*** We will make a software program to store IMU data and test out the if the data is correctly stored for in a range of 1 minute to 1 minute and 15 seconds.
	Inertial Measurement Unit	Gyroscope Range of ±250-500°/sec and accelerometer Range of ±1.5-2g ***Verification:*** We will do a physical test on our unit to prove that our system is able to detect small measurements in real life with an accuracy of ±250-500°/sec for the gyroscope and ±1.5-2g for the accelerometer. Operating Current below 5 mA ***Verification:*** We will use multimeters in the lab to measure current across IMU to ensure current is below 5mA.
	Button Controls	Sensor MCU acknowledges a button press within half a second. ***Verification:*** We will make a program that connects LED with the buttons and see the process took less than half a second.

Table 5. Requirements and Verification Table. Control Subsystem

	Block	Requirement and Verification
Lighting	Light Microcontroller	Under 1 mA during signal operation. ***Verification:*** We will use multimeters in lab to measure current during signal operation and confirm it is under 1mA.
	Shoulder Light Strips	Yellow blinking light is visible at least 100 feet away in a 270 degree cone centered in the direction of that signal's side under clear night conditions. ***Verification:*** We will provide proof that the light is visible from the cyclist's position up to 100 feet away, excluding the 270 degree cone centered on the opposite side of the signal.
	Back Light Strip	Red light is visible from at least 100 feet away from the rear when the brake signal is engaged under clear night conditions. ***Verification:*** While the system is in brake state, we will observe the light is visible from 100 feet away at night.
	Chest Lights	White light is visible from the front at at least 500 feet away in clear night conditions. ***Verification:*** While the front white light is turned on, we will observe the light can be visible from 500 feet away at night.
Power	Battery	A Full charge provides at least 2 hours of functionality. ***Verification:*** We will observe how long battery will last under following conditions: 1) Battery is fully charged with no activation before testing. 2) Full LED is set during experiment, including front LED turned on full time. 3) LED is clearly visible without any dilution at dark.
	5V Booster	Provide at least 1 A over the regulated 5V rail. ***Verification:*** We will connect multimeters to the output of 5V Booster to confirm that 5V regulator provide at least 1A.
	3.3V Regulator	Provide at least 500 mA over the regulated 3.3V rail. ***Verification:*** We will connect multimeters to output of regulator to check 3.3V regulator provide 500mA or higher.
	Battery Charger	Provide a charge current of at least 450 mA. ***Verification:*** We will use multimeters in lab to test current from battery provide 450 mA or higher.

Table 6. Requirements and Verification Table. Lighting and Power Subsystems

2.10 Tolerance Analysis

The critical part of our project will be the sensor subsystem. The functioning of our project depends heavily on information from the gyroscope, accelerometer and magnetometer, and thus the data recorded needs to be as precise as possible.

Additionally, as a wearable, there will be a level of introduced noise in the form of rider motion which differs both between modes of transport and between riders. In all of these cases, this unwanted noise provides either false or scant IMU information and may result in incorrect turning on/off of LED light negating the core purpose of this project. If this happens in the middle of a turn, this could turn out to be dangerous.

For these reasons stated above, we found that it is essential that we filter our sensor data. We chose Madgwick's sensor fusion algorithm because it is the most accurate for our exact implementation. Another common solution in small embedded systems is the Mahony algorithm; however, Madgwick is more precise for systems such as 9 degrees of freedom Inertial Measurement Unit, like ours. Another of the options we looked into was the Kalman filtering, but Madgwick's algorithm is still a little more accurate, as can be seen in the typical example from Madgwick's report [11] in ***Fig 7.***

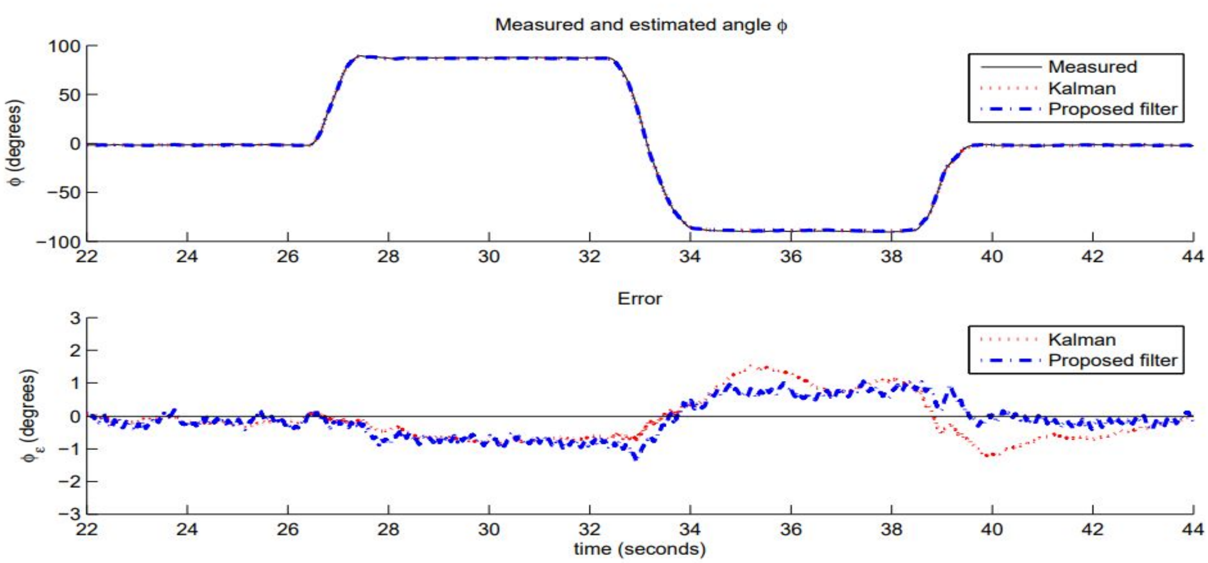


Fig 7. Typical results for measured and estimated angle (top) and error (bottom)

As for the error estimation, Madgwick's report [11] estimates a dynamic error smaller than 7º for a 10 Hz sampling rate. For our project we will be using a sample rate of 256 Hz for gyroscope and accelerometer, and 100 Hz for the magnetometer, because that is what it is limited to. These measurements provide a high level of accuracy and stays inside the sample rate rank of the three devices. We will then get an error of less than 1º RMS, as pictured in ***Fig 8***.

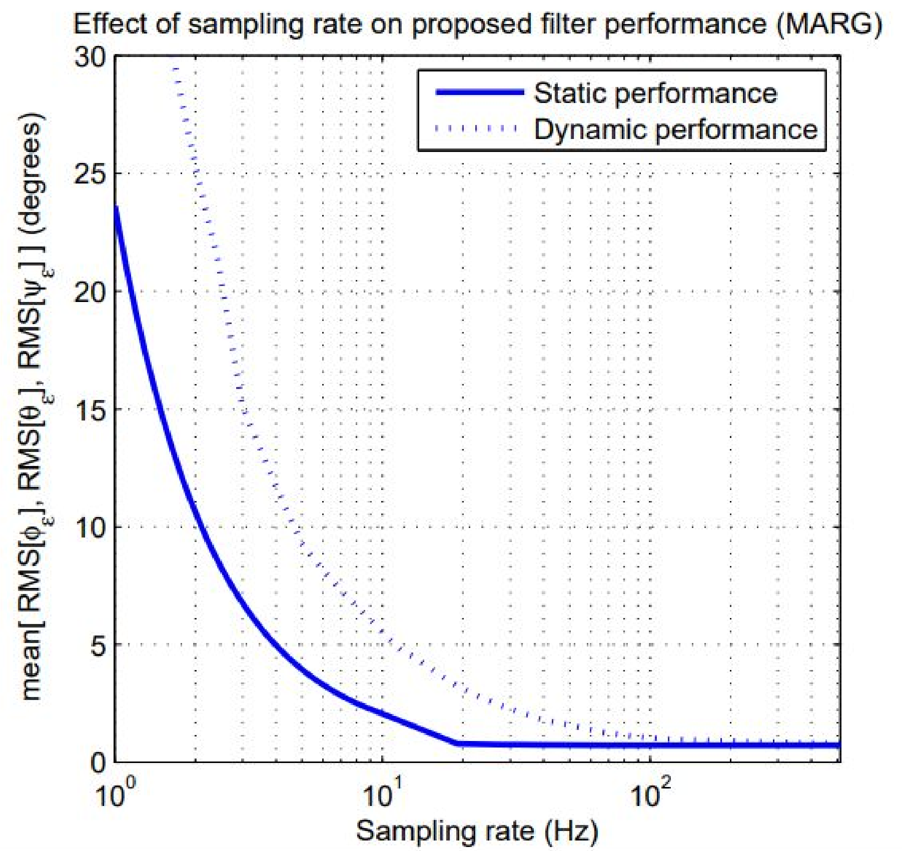


Fig 8. Sampling rate vs Performance

Furthermore, we estimated the behavior of our system for this sample rate chosen, simulating a rotation from 0 to 90 degrees and then to -90 degrees around X,Y and Z axis, using an open source Matlab Code [13]. After applying the filtering we would get the following output:

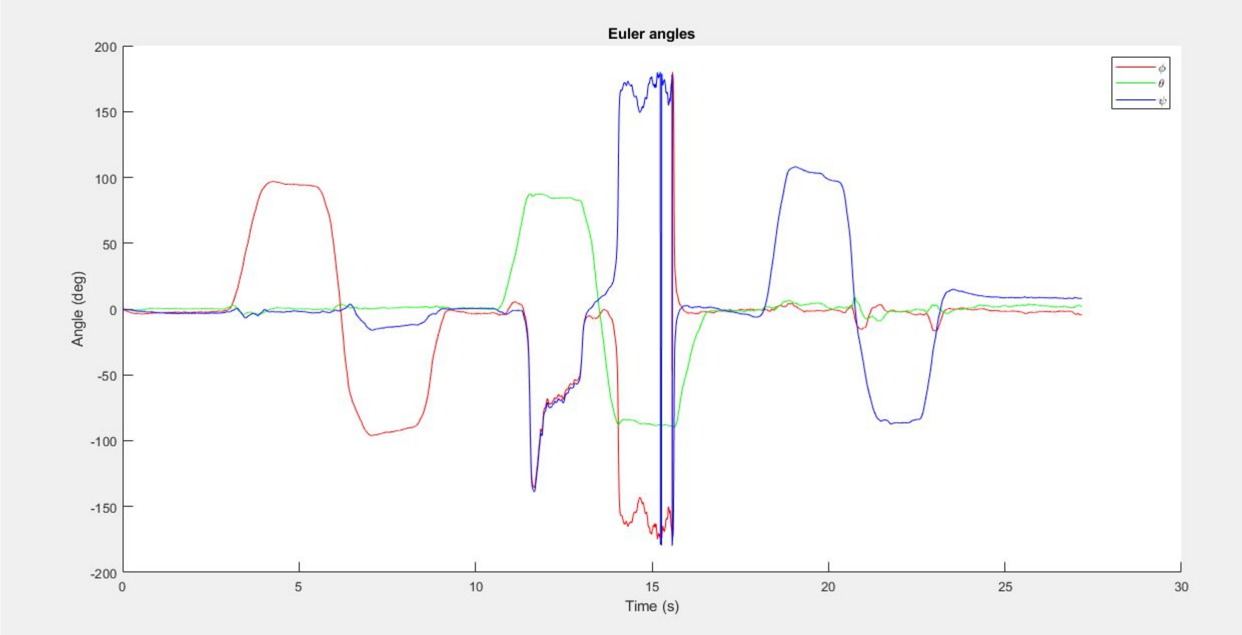


Fig 9. Output of Madgwick algorithm over a typical dataset

As we can see in the graphics above, we get much error when the simulation gets to a vertical position (±90 degrees). Although we should not be dealing with this situation, we should take into account that if this problem arises we could approach it with quarternion matrices. Furthermore, we estimated that a complete turn of a cyclist should not take more than 30 seconds. As we can see from the graphics above, the behavior over this time does not include noticeable errors.

As for the position error tolerance, since road lane width varies from 8.2 to 10.7 ft, the average road lane width would be of approximately 9.5 ft. Assuming the cyclist would ideally be on a 10% tolerance of displacement, that would leave approximately 4.5 meters at each side of the cyclist. Since the error estimation of this model is of the order of 1%, as demonstrated in Madgwick's report [11], and the estimated time for a complete turn is 30 seconds, the amount of error in total should be small enough so it doesn't influence our model.

3 Project Logistics

3.1 Cost Analysis

3.1.1 Bill of Materials

Qty.	Part	Distributor	Cost Per Unit @ 1 Vest	Cost Per Unit @ 100+ Vests
Control Subsystem			**$26.09**	**$23.0206**
1	MPU-9250	Digikey	$10.63	$10.63
1	ATSAMD21G18	Digikey	$3.15	$2.5956
1	32.768kHz Crystal	Digikey	$0.41	$0.2750
1	Simple RF M4 Receiver - 315MHz Momentary Type	Adafruit	$4.95	$3.96
1	Keyfob 4-Button RF Remote Control - 315MHz	Adafruit	$6.95	$5.56
Lighting Subsystem			**$18.73**	**$17.91**
1	ATtiny84	Sparkfun	$2.95	$2.66
1	LPD8806 Weatherproof LED Strip	Amazon (Mokungit)	$11.88	$11.88
2	JST 2-Pin Connector	Adafruit	$0.75	$0.60
2	N-Channel MOSFET 60V 30A	Sparkfun	$0.95	$0.86
1	4-pin JST connector	Sparkfun	$0.50	$0.45

Table 7. Part Quantity and Cost. Control and Lighting Subsystems

Qty.	Part	Distributor	Cost Per Unit @ 1 Vest	Cost Per Unit @ 100+ Vests
	Power Subsystem		**$10.40**	**$7.0614**
1	MCP73831	Digikey	$0.58	$0.4326
1	PAM2401	DigiKey	$1.08	$0.7725
1	BAT20J Schottky diode 1A/23V/620mV	DigiKey	$0.41	$0.2333
1	USB Micro-B SMD Connector	Mouser	$1.28	$0.99
1	SPDT Switch	Distributor	$2.24	$1.8052
2	Red SMD LED	Digikey	$0.22	$0.0853
1	Blue SMD LED	Digikey	$0.48	$0.1749
1	Yellow SMD LED	Digikey	$0.27	$0.1073
1	LiPo Connector	Sparkfun	$0.95	$0.86
1	3.3V 600mA CMOS LDO Regulator w/ Enable	Mouser	$0.47	$0.15
1	Resettable Fuse PPTC 6V/0.5A	Mouser	$0.34	$0.31
1	Inductor 2.2UH3A	Mouser	$1.08	$0.62
2	B320 - B360 Schottky diode 3A/40V/500mV	DigiKey	$0.39	$0.2175

Table 8. Part Quantity and Cost. Power Subsystem

Qty.	Part	Distributor	Cost Per Unit @ 1 Vest	Cost Per Unit @ 100+ Vests
Manufacturing, Vest, Enclosure, and Assorted			**$48.9675**	**$36.5566**
1	PCB Manufacturing	PCBWay	$5.80	$1.42
1	Vest	Distributor	$15.99	$15.99
1	3D Printed Control Box	MakerBot	$20.77	$15.72
6	0.1μF ceramic capacitors	Sparkfun	$0.25	$0.23
1	1μF ceramic capacitors	Mouser	$0.18	$0.066
2	15pF ceramic capacitors	Mouser	$0.54	$0.252
1	2.2μF ceramic capacitors	Mouser	$0.33	$0.12
3	22μF ceramic capacitors	Mouser	$0.32	$0.124
2	4.7μF ceramic capacitors	Mouser	$0.31	$0.1215
1	100kΩ resistor	Mouser	$0.36	$0.256
1	10kΩ resistor	Sparkfun	$0.25	$0.23
3	1kΩ resistor	Sparkfun	$0.1425	$0.043
1	2.2kΩ resistor	Digikey	$0.10	$0.0258
3	4.7kΩ resistor	Digikey	$0.10	$0.0168
3	750Ω resistor	Digikey	$0.10	$0.0168
Part Subtotal			**$104.1875**	**$84.5486**

Table 9. Part Quantity and Cost. Manufacturing, Vest, Enclosure, and Assorted. Part Subtotal

3.1.2 Labor Value Added

Using reported graduating salaries divided by salaried hours in a year we can get an hourly rate. [9] Each part of the project has an estimated man hours of work. Finally everything is multiplied by a 2.5 fudge factor.

Board Design	2 people at	25 hours	50 hours
Board Assembly and Soldering	2 people at	30 hours	60 hours
Board Testing	2 people at	30 hours	60 hours
Sensor Code Development	2 people at	45 hours	90 hours
Lighting Code Development	1 people at	30 hours	30 hours
Control System Testing	2 people at	20 hours	40 hours
Lighting System Testing	1 people at	15 hours	15 hours
Physical Integration	3 people at	10 hours	30 hours
Integration Testing	3 people at	15 hours	45 hours
System Tuning	3 people at	10 hours	30 hours
Presentation Polish	3 people at	5 hours	15 hours
Hour Sum			**450 hrs**

Wage: 80,000 $USD / 2,080 hours = 38.46 $/hr

Labor Cost: 2.5 x Wage * Hour Sum = 2.5 x 38.46 $/hr * 450 hrs = $43,267.50

Table 10. Labor Value Analysis

3.1.3 Cost Totals

For Development purposes we can assume parts cost at the scale of 1 vest. For a fudge factor, as there will be unexpected costs and multiple iterations, let us multiply it by 1.5 and finally sum with labor cost:

Total Cost: $104.1875 x 1.5 + $43,267.50 = $43,423.78

3.2 Development Schedule

Week	Task and Assignments	
2/25 - 3/3	**Board Design** *Paloma and Derrick*	**Sensor Code Dev** *Wonyong*
3/4 - 3/10	**Board Assembly and Soldering** *Paloma and Derrick*	
3/11 - 3/17	**Board Testing** *Paloma and Derrick*	
3/18 - 3/24	**Lighting Code Dev** *Derrick*	*Paloma joins sensor and control from boardwork*
3/25 - 3/31		
4/1 - 4/7	**Lighting Testing** *Derrick*	**Control Testing** *Paloma and Wonyong*
4/8 - 4/14	**Physical Integration** *Everyone*	
4/15 - 4/21	**Integration Testing** *Everyone*	
4/22 - 4/28	**System Tuning** *Everyone* **Presentation Polish** *Everyone*	

Table 11. Planned Project Schedule

4 Safety and Ethics

There are handful of safety concerns within our project. Usage of a Lithium polymer battery is one of them. We chose the Lithium polymer battery as a mobile power source due to its high energy density and discharge rate. With such features, the LiPo cells have a distinct risk of fire. There are two common causes of fire; when the battery is overcharged, and when it is exposed to too much heat such as from its own discharge. LiPo batteries are also quite soft making puncture and associated fires a very real risk. To avoid these incidents, we will ensure our charge circuit has a cutoff point and instruct users to not charge unattended. To protect the user and vest we will create an enclosure for the control and power systems that will protect battery from heat exposure and puncture [6].

As we are billing ourselves as a “comprehensive micromobility signaling system”, it is our responsibility to follow part 3 of the IEEE code of ethics, and 1.3 of the ACM Code by providing a system that fulfills the common sense requirement of such a project. To do so the project will allow users to follow state regulation for bicycle lighting on the road.[2][3] The regulation states that bicycle safety equipment should compose of red rear reflector visible from 100 to 600 ft, horn or bell that can be heard up to 100 ft, and front light visible at least 500 ft[5]. All of these lights should point down to not distract any drivers from driving on the road. Since we are making vest with LED lights, we need to account that extreme brightness and darkness can distract drivers more, and find valid range of brightness that will suit both state regulation and driver’s comfort.

As an outdoor device that uses electricity, we must acknowledge and follow IP67 guidelines to avoid potential safety hazard. According to IP67, the device needs to be protected from total dust ingress, and water immersion between 15 cm to 1m depthq] which the enclosure would also satisfy. Another concern is failure of the device in the middle of the ride, particularly in the middle of turn signals along with erroneous brake lights. This problem could be crucial since it can guide to a false communication between rider and other drivers leading potential accident. To solve this problem, we will have a remote controller that can override any signals between sensor and the vest to ensure rider to manually signal their desired turn.

Experiments outside the lab is another safety concern for our project. To ensure that our project will work on the road and reduce errors, we will need to collect data from outside the senior design lab. Because we will be riding personal transportation on the road, it is important for us to learn various hand signals with different transportations accordingly and provide such information to potential users as reference for additional safety.

Lastly, working on senior design with other groups in the lab also follows risks. While one group does not have much safety concerns, other groups may have several safety hazard they must consider carefully. As students whose sharing working space with other groups, we must aware of other safety hazards and precautions. To prepare for such concern, we will complete lab safety tutorials and visit frequently on DRS safety program before entering the lab [12].

5 References

[1] Stillwell, Keven; "Bike Lights and Night Riding: Laws and Suggestions", *Neutral Cycle*, Feb 27 2015 [Online]

http://www.neutralcycle.com/bike-lights-night-riding-laws-suggestions/

[2] "IEEE Code of Ethics", *IEEE.org* [Online] https://www.ieee.org/about/corporate/governance/p7-8.html

[3] "ACM Code of Ethics and Professional Conduct", *ACM.org*, June 22 2018 [Online] https://www.acm.org/code-of-ethics

[4] "IP enclosure ratings & Standards explained", *RainFord Solutions*, Jan 15 2019 [Online] https://www.rainfordsolutions.com/ip-enclosure-ratings-and-standards

[5] "Bicycle, Rules of the Road", Office of the Secretary of State of Illinois, May 2018 [Online] https://www.cyberdriveillinois.com/publications/pdf_publications/dsd_a143.pdf

[6] Reade, John; "LiPo Batteries and Safety for Beginners", *CNY Drones*, Jan 9 2018 [Online] https://www.cnydrones.org/lipo-batteries-and-safety-for-beginners/

[7] "Bicycle Safety", *National Highway Traffic Safety Administration* [Online] https://www.nhtsa.gov/road-safety/bicycle-safety

[8] "Chapter 2. Definition and applicability of the Anthropometry Data", Dr. Alan Chan Teaching Page [Online]

http://personal.cityu.edu.hk/~meachan/Online%20Anthropometry/Chapter2/Ch2-5.htm

[9] "Salary averages", Electric & Computer Engineering - University of Illinois, 2019 [Online]

https://ece.illinois.edu/admissions/why-ece/salary-averages.asp

[10] "Bicycle performance", Wikiwand [Online]

https://www.wikiwand.com/en/Bicycle_performance

[11] Madgwick, Sebastian O.H.; “An efficient orientation filter for inertial and inertial/magnetic sensor arrays”, University of Bristol (UK), April 30 2010.

[12] ”Laboratory Safety Management”, Division of Research Safety, Sept 10 2018 [Online]

https://www.drs.illinois.edu/Programs/SafetyManagement

[13] “Open source IMU and AHRS algorithms”, x-io Technologies, July 31 2012 [Online]

http://x-io.co.uk/open-source-imu-and-ahrs-algorithms/

[14] “MPU-9250 Product Specification Revision 1.1”, InvenSense, June 20 2016 [Online]

http://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf

[15] “Pedestrian and Bicyclist Crash Statistics”, Pedestrian & Bicycle Information Center [Online]

http://www.pedbikeinfo.org/data/factsheet_crash.cfm

[16] S Fotios; HF Castleton; “Lighting for cycling in the UK—A review”, SAGE journals, October 6 2015 [Online]

https://journals.sagepub.com/doi/full/10.1177/1477153515609391