

December 2015

# IntelliWay: Connecting Bicycles to the V2V Network

Christopher Michael Ellen  
*Worcester Polytechnic Institute*

Dante Alighieri Pace  
*Worcester Polytechnic Institute*

Sunghoon Yoon  
*Worcester Polytechnic Institute*

Follow this and additional works at: <https://digitalcommons.wpi.edu/mqp-all>

---

## Repository Citation

Ellen, C. M., Pace, D. A., & Yoon, S. (2015). *IntelliWay: Connecting Bicycles to the V2V Network*. Retrieved from <https://digitalcommons.wpi.edu/mqp-all/2466>

This Unrestricted is brought to you for free and open access by the Major Qualifying Projects at Digital WPI. It has been accepted for inclusion in Major Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact [digitalwpi@wpi.edu](mailto:digitalwpi@wpi.edu).

# IntelliWay: Connecting Bicycles to the V2V Network

A Major Qualifying Project Report:  
Submitted to the Faculty of the WORCESTER POLYTECHNIC INSTITUTE  
In partial fulfillment of the requirements for the  
Degree of Bachelor of Science  
by

Shawn Yoon

Dante A. Pace

Christopher M. Ellen

Date: December 15, 2015

---

Professor Mohamed Eltabakh, Major Advisor

---

Professor William R. Michalson, Co-Advisor

# Table of Contents

## Contents

Acknowledgement.....	4
Abstract.....	5
Table of Figures .....	6
Table of Tables .....	7
1. Introduction .....	8
1.1 Motivation .....	8
1.2 Cycling Safety Problems.....	8
1.3 Extending the Future Technology.....	9
2. Literature Review .....	10
2.1 Existing Systems .....	10
2.2 Comparing Systems .....	13
Detection.....	13
Collision Detection .....	13
Speed.....	14
Communication .....	14
2.3 Boston Bikes.....	15
2.4 Identifying Critical Elements .....	16
2.5 Other Alternatives: Highway Infrastructure .....	17
2.6 Inter-Car-Communications System.....	18
2.7 Art of Merging .....	18
3. Design Approach.....	19
3.1 Introduction of IntelliWay Bike System.....	19
3.2 System Specifications.....	20
3.3 Testing IntelliWay Components .....	30
3.4 IntelliWay Uniqueness .....	31
4. Implementation and Device Prototyping.....	32
4.1 Obstacles and Changes for Implementation.....	32

4.2 IntelliWay Hardware Device.....	33
4.3 IntelliWay V2V Simulated Broadcaster .....	34
4.4 IntelliWay Android Broadcast Receiver.....	37
4.5 IntelliWay Internal Hardware.....	41
5. Testing & Evaluation .....	44
5.1 Testing V2V Simulated Broadcaster .....	44
5.2 Testing Android Broadcast Receiver .....	47
5.3 Testing PCB.....	48
5.4 Overall Evaluation.....	48
6. Conclusion .....	49
Appendix A.....	56
Appendix B.....	57

# Authorship Page

Abstract - Chris

## 1. Introduction

- 1.1 Motivation- Shawn
- 1.2 Cyclist Safety Problems- Shawn
- 1.3 Extending the Future Technology - Shawn

## 2. Literature Review

- 2.1 Existing Systems - Chris
- 2.2 Comparing Solutions - Chris
- 2.3 Boston Bikes
- 2.4 Identifying Critical Elements - Shawn
- 2.5 Highway Infrastructure - Shawn
- 2.6 Inter-Car-Communications System -Shawn
- 2.7 Art of Merging - Shawn

## 3. Design Approach

- 3.1 Introduction of IntelliWay Bike System - Chris
- 3.2 System Specifications - Shawn - Chris
- 3.3 Testing IntelliWay Devices -Shawn
- 3.4 IntelliWay Uniqueness - Shawn - Chris

## 4. Implementation

- 4.1 Obstacles and Changes for Implementation – Shawn – Dante - Chris
- 4.2 IntelliWay Hardware Device - Chris
- 4.3 IntelliWay V2V Simulated Broadcaster – Shawn - Dante
- 4.4 IntelliWay Android Broadcast Receiver – Shawn - Dante
- 4.5 IntelliWay Internal Hardware - Dante

## 5. Testing & Evaluation

- 5.1 Testing V2V Simulated Broadcaster – Shawn - Dante
- 5.2 Testing Android Broadcast Receiver – Shawn - Dante
- 5.3 Testing PCB - Dante
- 5.4 Overall Evaluation –Shawn - Dante

## 6. Conclusion - Dante

Appendix A

Appendix B

Bibliography - Shawn – Chris - Dante

# Acknowledgement

We would like to acknowledge Professor. Eltabakh for advising the database and programming integration portion of the MQP.

We would like to acknowledge Professor. Michalson for serving as a co-advisor for Electronic and Robotics integration portion of the MQP. His experience as a professional engineering consultant was critical in all of the project phases.

# Abstract

The IntelliWay device aims to assist cyclists to avoid accidents with motorized vehicles. The project summarizes an innovative device that could provide safer travel for cyclists. Related findings on similar solutions such as bike traffic in heavily populated areas, highway infrastructure, Inter-Car-Communication Systems and the art of merging, have been documented. The project will cover approaches to the initial design and the system specifications for the device with future goals and detailed tasks and schedules.

# Table of Figures

Figure 1. BikeShield Application - Map UI .....	11
Figure 2. Volvo City Safety .....	12
Figure 3. Boston Bikes Accident Data .....	15
Figure 4. Boston Bikes EMS Report Data .....	16
Figure 5. Artist Rendition of Final Product .....	19
Figure 6. System Block Diagram.....	20
Figure 7. Artist Rendition of Middle Layer.....	21
Figure 8. Artist Rendition of Outside of Case .....	22
Figure 9. Bicycle Handlebar Grip.....	23
Figure 10. IntelliWay Application.....	27
Figure 11. Solidworks Design of IntelliWay Case.....	33
Figure 12. Broadcaster Architecture.....	35
Figure 13. Android Broacast Receiver Architecture .....	37
Figure 14. IntelliWay LED Driving Circuit with $\mu$ USB-PA5.....	41
Figure 15. LED Driving Circuit Schematic .....	42
Figure 16. LED Driving Circuit 2-Layer Gerber Drawing with Keep-out Layer .....	43
Figure 17. Captured GPS Data around WPI .....	45
Figure 18. Cleaned-up GPS Data Points .....	46
Figure 19. Back of IntelliWay .....	49
Figure 20. Angled View of IntelliWay.....	50
Figure 21. Front View of IntelliWay .....	51



# Table of Tables

Table 1. Competitive Value Analysis Table .....	14
Table 2. Defining Metrics for Battery Value Analysis .....	24
Table 3. Value Analysis of Battery Types .....	25
Table 4. Value Analysis of USB Power Sticks .....	25
Table 5. Defining Metrics for Wireless Adapter .....	28
Table 6. Value Analysis of Wireless Adapters .....	28

# 1. Introduction

## 1.1 Motivation

Every year, new innovative technologies get developed to enhance our daily lives. From backup cameras that can save children's to automatic self-driving cars, various technologies constantly get developed around us to promote more convenient and safer travel and commuting. However, not all industry is developing rapidly, so there are handful of opportunities out there to develop something that could enhance our lives significantly. In this project, we focus on the safety of one activity performed by millions of people on a daily basis, which is the safety of cycling.

## 1.2 Cycling Safety Problems

There are thousands of deadly automobile accidents every year. Various effective technologies were developed to help drivers to avoid deadly accidents, but there have not been significant safety-related technological advancements in the bicycle industry. Although automobiles have several safety features for minimizing injury during accidents, bicycles do not, so when a bicycle and an automobile crashes, the cyclists usually get hurt significantly compared to automobile drivers

In 2013, over seven hundred people lost their lives in bicycle-motor vehicle collisions, which was just over two accidents per day. Of all accidents recorded, 29 percent are due to collisions with an automobile. The rest of all recorded accidents consist of collisions with other bikes, people, animals, as well as poor road conditions and collisions with surrounding objects. A significant portion of the accidents were bike to automobile accidents. The main cause of the accident could be inferred from the time of incident. Twenty-two percent of all accidents also happen between the hours of 6p.m. and 9p.m. The main inferred cause from the accident is limited sight around the specific time, and automobile drivers probably could not spot the bicycle well enough. The amount of total accidents is much higher due to records of these accidents not being recorded by officials as accurately as automobile accidents. Unreported accidents are not fatal and some bikers leave the scene without injury or either party member of the accident pursuing charges of some sort. <sup>[8]</sup>

## 1.3 Extending the Future Technology

Although there are various sets of possible solutions out in the World to improve safety of cyclists, it is important to understand and acknowledge various options and implement effective solution. Building more lights on the road or making a dedicated bicycle road are definitely considerable options, but those options are very expensive. Utilizing vehicle to vehicle communication was a worthy option to utilize (Volvo).

Vehicle-to-vehicle communication, or V2V for short, is not a new concept, but only recently has there been enough advancement in communications technology to make it a reality. V2V involves multiple automobiles sending out signals to communicate with other close by cars, possibly relaying information such as positional and velocity data, as well as driver intentions. As of now, there is a federal mandate in motion that will require all new vehicles to have V2V capabilities by 2020.<sup>[19]</sup>

Given the federal mandate, this project aims to conceptualize and create a system that allows bicycles to also connect to the V2V network and communicate with automobiles. This system will include a device to alert the biker of potentially dangerous situations when automobiles with V2V communication are nearby. It will also alert compatible vehicles of nearby bicyclists. The goal of this project is to prove that such a communication system between bicycles and vehicles is possible and to demonstrate a possible implementation.

## 2. Literature Review

### 2.1 Existing Systems

The BikeCOM project is a cooperative intelligent system using bicycle to car communication. The system uses wireless communication to assist the operator of both bicycle and motorized vehicle in avoiding a collision. This cooperative application was developed by Per Gustafsson, Linus Lindgren, Juan-Camilo Munoz Cantillo, and Christian-Nils Boda as part of their education in the Master Programme for Automotive Engineering at Chalmers University of Technology. Realistic cycling data from the BikeSAFER project was used to derive use cases for this application. By sharing position and dynamics using wireless communication, both operators can be warned by the BikeCOM application to avoid a collision. Threat assessment and warning timing is taken into account for both of the different vehicle dynamics and possible user behavior.

[6]

The BikeShieldApp is a First Layer Crash Avoidance Technology mobile app which establishes communication between cars, motorcycles and bicycles that are sharing the road. When using the app, car drivers get an acoustic signal that warns them about approaching motorcycles and bicycles. These signals raise driver's awareness and reduce distracted maneuvers that may result in a fatal crash with motorcycles and bicycles. The BikeShield API was developed to open their Bike-to-Vehicle (B2V) communication channels to other road apps. This includes apps for cyclists, bikers and car drivers. They are willing to open their B2V channels to other apps, and can be contacted for inquiries about their API. This system requires that each operator has the app installed on their mobile device and activated. For motorcycles and bicycles, all cars that surround the bike with the activated app will receive an acoustic notification 5 to 10 seconds before they can even see or hear the cyclist. For car operators, the app will automatically activate using the mobile device's accelerometer. The app alerts the driver using visual and acoustic signals. Compatible with iPhone, Android and Smart Gear Watches. [4]

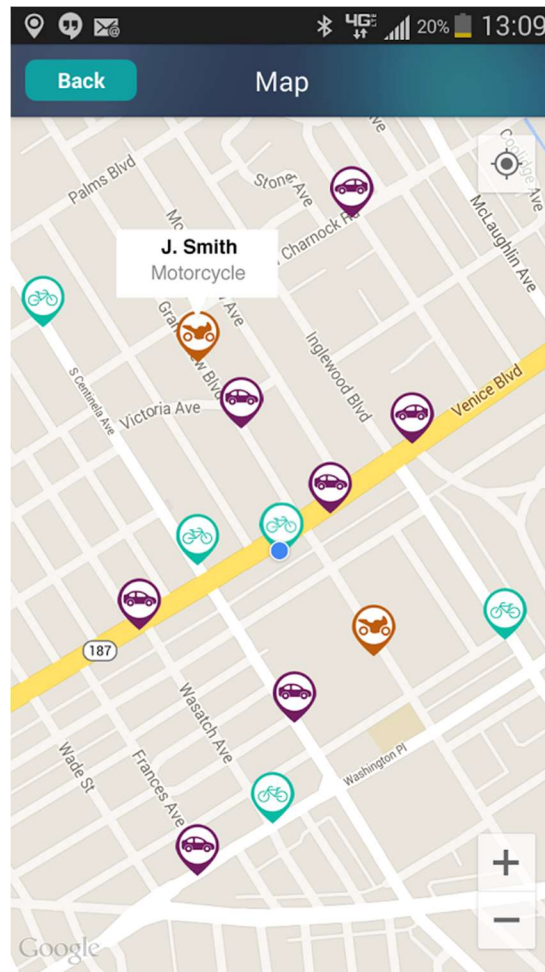


Figure 1. BikeShield Application - Map UI

Volvo currently offers pedestrian and cyclist detection on current production vehicles. They are working on taking the cyclist's safety a step further with a prototype that allows a connected car and connected bike helmet to communicate to avoid collisions. This technology allows a Volvo connected car and the prototype helmet to establish a two-way communication network for offering proximity alert to both drivers and cyclists. Volvo's City Safety system can already detect cyclists and warn the driver about their presence and will take action to avoid a collision. The prototype uses a popular GPS smartphone app for bicyclists to share the cyclist's position relative to the car, and vice versa. If the system detects that a collision is imminent it will alert both the driver and cyclist to possibly avoid an accident. The Volvo driver is alerted by a heads-up display and the cyclist is warned through a helmet mounted light. <sup>[25]</sup>



*Figure 2. Volvo City Safety*

A research team at the Queensland University of Technology is working on a system that allows communication between cyclists and drivers that are about to have a collision. They are working on a smart device app that can locate, track, connect and communicate between devices and warn both rider and driver if there is the chance of a collision. The design is to implement existing GPS, WiFi and Bluetooth systems for sharing information between road users. From their research they have concluded that most cyclist fatalities involve a collision with a motor vehicle and that they typically occur due to human error. [7]

## 2.2 Comparing Systems

Of those systems that are pre-existing, three of the ones we researched utilize GPS for determining the positions of both the cyclist and the driver of a motorized vehicle. The most ideal system for determining position data would be through the use of the GPS. The BikeCOM project focuses more the use of Lidar and radio communications for determining the positions and speeds of both cyclist and vehicle.

### Detection

The detection abilities of the BikeCOM app and Queensland project are decent, as they can detect vehicles in cases of possible threat. With BikeCOM, both parties require the application, and drivers may forget to turn it on when getting behind the wheel. The Queensland University project is more dedicated to detecting vehicles that are on a collision course with a cyclist. The Volvo City Safety system is very good at communicating with cyclists the location of other cars, but only with Volvos. The BikeShield App is the closest to what is desired in this project. There is a map, as shown in Figure 1, of all nearby vehicles, including motorcycles, as well as other bicycles.

### Collision Detection

BikeCOM was designed to avoid collisions between bicycles and vehicles. Using naturalistic cyclist data, they derived use cases to ensure that this application can effectively detect possible collisions and give both parties ample time to react. Volvo's City Safety System works like most commercial collision detection systems work, but they keep much of it a secret from the public. The Volvo project relies on a specialty helmet that communicates with Volvo cars, which in turn lowers the detection ability of this system. The BikeShield application detects every motor vehicle and bicycle connected with the application, and therefore can alert both parties of any imminent dangers. The Queensland University project is designed to track and locate both parties in real-time and warn them, with ample time, of collision threats.

## Speed

The BikeCOM application uses a wireless network between other BikeCOM users, similar to how BikeShield and Volvo City Safety System work. Volvo has an extra step in their process, which involves communicating with a cloud server before contacting the Volvo vehicles. The Queensland project uses a combination of WiFi, Bluetooth, and GPS. Bluetooth isn't as fast a standard as the other systems, but not slow enough to make the system obsolete.

## Communication

BikeShield, the Queensland project, and BikeCOM uses a similar method to the proposed method for IntelliWay, which is an application-based bicycle-to-vehicle network, but requires the bicyclist and driver to have the application installed. Volvo doesn't require both parties in their system to have the application installed, as Volvo will have the technology preinstalled in their vehicles. BikeCOM and the Queensland project were given lower scores due to these still being in development.

*Table 1. Competitive Value Analysis Table*

		BikeCOM		BikeShield		Volvo		QUT	
Factor	Weight	Rating	Score	Rating	Score	Rating	Score	Rating	Score
Detection	4	2	8	3	12	2	8	2	8
Collision Detection	3	3	9	3	9	3	9	2	6
Speed	3	2	6	3	9	3	9	2	6
Communication	3	1	3	2	6	3	9	2	6
Total:			26		36		35		26



## 2.3 Boston Bikes

Boston Bikes, a group launched by Mayor Menino in 2007, conducts an annual survey tracking the demographics of bicyclists, crashes, events, and other issues. According to their results, 44 percent of cyclist reported that they chose to commute to work on their bikes because it was the best option. A helmet survey at Boston University showed that in 2010, 72% of bicyclist self-reported using a helmet while riding a bike. The percentage of riders wearing helmets requiring EMS is 45%. In 2009, Massachusetts Avenue and Commonwealth Avenue were reported as the roads with the most bicycle accidents. Massachusetts Institute of Technology and Boston University are located on these streets, respectively. These concentrations of population can help explain why these streets are hotspots for bicycle activity. Figures 3 and 4 show bicycle crash data and EMS response data from Boston, respectively. Responders to the Boston Bikes surveys also report that they believe riding on off-road paths are “significantly more safe than biking on Boston streets.” Thirty-two percent of those who replied that they sometimes or never wear a helmet say that it’s due to the affordability of bicycle helmets. [2]

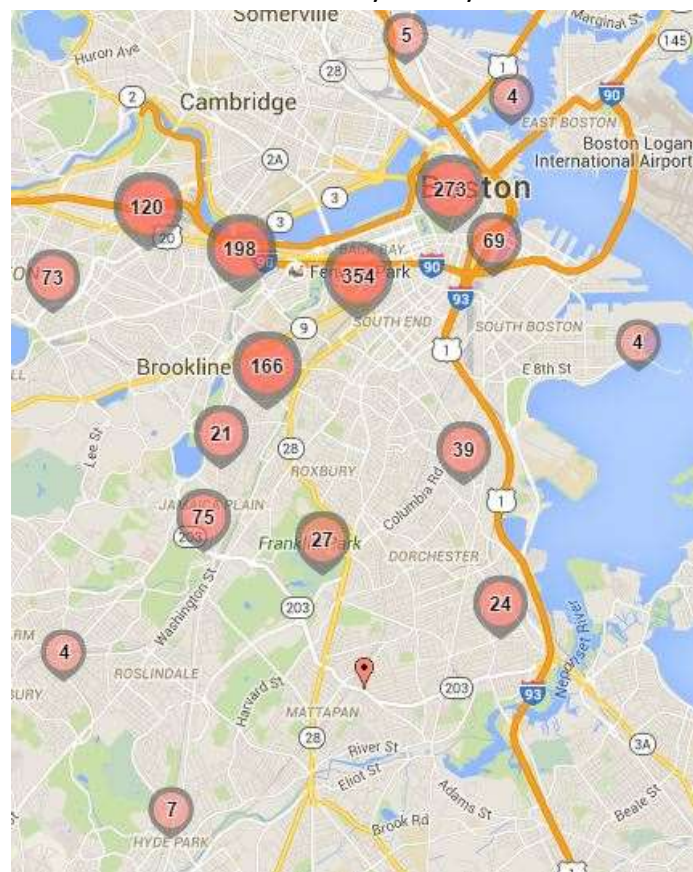


Figure 3. Boston Bikes Accident Data

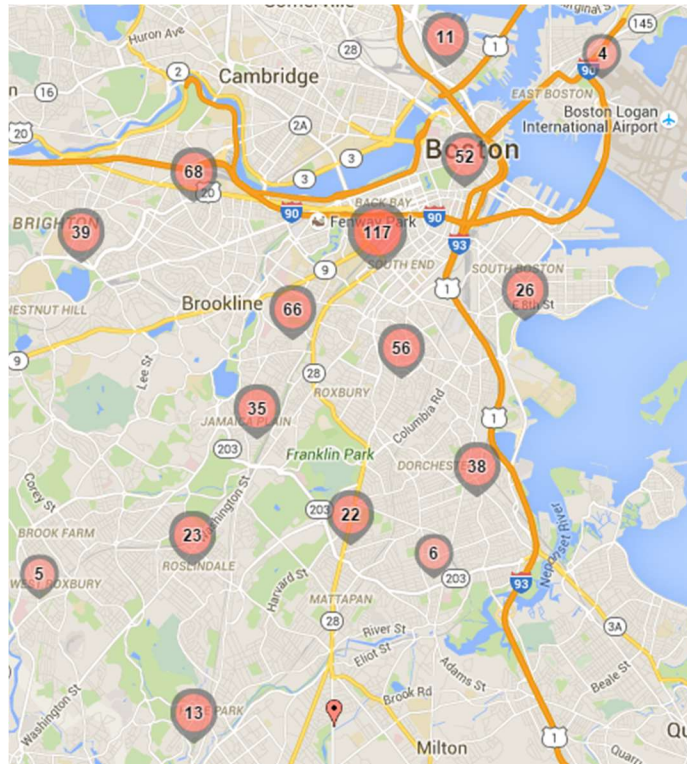


Figure 4. Boston Bikes EMS Report Data

## 2.4 Identifying Critical Elements

It is important to understand and identify critical elements in various technologies in bicycle industry. After comparing and contrasting various options, critical elements in those options need to be analyzed and combined to make a useful product. Useful products need to be general, easy to access, relatively low cost, and easily scalable.

Critical element from BikeCOM, and BikeShieldApp is communication system. The idea of wireless communicating from vehicle to vehicle is a critical element because IntelliWay system can easily intercept this system and start using the same communication channel.

Critical element from Volvo's B2V alert system is communication layer with smartphone mobile application. Smartphones are everywhere in modern society, but smartphones are really powerful, so effective smartphone utilization can result in low cost and simple accessibility. Volvo utilized smartphone to alert the bikers, but the company used bike helmets as part of the required device. However, major portion of bikers do not utilize helmet or desire to purchase one.

The identified critical elements are the following: V2V signal interception for easy scalability and no production cost (Tapping the system). Smartphone signal utilization for easy scalability, easy to access, and no production cost. Final critical element is recharging system for cyclists that comes with low production cost.

## 2.5 Other Alternatives: Highway Infrastructure

Although highway infrastructure was the first option we considered, but we wanted to make sure that what we are building is financially and legally viable. This project is definitely possible with significant amount of grant or funding. However, the project would cost a lot since the machine has to be implemented all of the merging part of the highway segments. Furthermore, the maintenance cost is anticipated to be significant because the machine could wear off by weather and time. We've explored couple journals and documentations, and discovered that current way of integrating new infrastructure and accessing data is inefficient. [31]

One of the paper described a new hypothetical framework for integrating infrastructure. We learned that it's probably not a good idea to try and go for a stand-alone device to add to the current highway system.

Another documentation implied that creating highway infrastructure is extremely expensive. The cost for just creating highway infrastructure is significant enough, so that the sustainability wasn't even considerable: "To date, however, there have not been many financial assessments on the sustainability aspects of highway projects. This is because the existing life cycle costing analysis models tend to focus on economic issues alone and are not able to deal with sustainability factors." [10]

One of other document suggests that there has always been debate on whether or not highway infrastructure investments will cause significant economic growth. The document extends on possible benefits of investing in the infrastructure, but more so of changing federal transportation policy. Not all infrastructure investments have as much an effect as others. This project could potentially benefit a lot of drivers, or it could be completely irrelevant. [30]

## 2.6 Inter-Car-Communications System

After investigating possibility of static highway structures, we also investigated couple of inter-car-communication system and its impact. One of interesting article summarizes that Google's automated cars had accidents. However, the paper indicates claims from Google that the automated cars were not the primary causes of accidents. Google claims that it is the drivers who caused the accident, and the self-driving cars were simply victims of those accidents.

Another interesting article was about automating the turn signal. One of the interesting quote from the document says: "In fact, that's a good litmus test. Any time you wonder why a particular thing hasn't been made automatic yet, think about whether you would arbitrarily hand it off to someone else sitting in a different seat..."<sup>[23]</sup> It makes a concise point about surveying and testing before actual implementation of the solution.

Another article summarizes the difficulty and complexity of car-to-car communication system. It says: "Creating a car-to-car network is still a complex challenge. The computers aboard each car process the various readings being broadcast by other vehicles 10 times every second, each time calculating the chance of an impending collision. Transmitters use a dedicated portion of wireless spectrum as well as a new wireless standard, 802.11p, to authenticate each message." This poses a lot of security concerns regarding broadcasting and receiving data between machines.<sup>[20]</sup>

Extending from the previous article, another article talks about GPS tracking and data collection regarding privacy concerns. With all modern technologies, a lot of creative implementations are viable, but we have to worry about privacy concerns with a lot of our data implementation. The article also includes an automated break-signal system.<sup>[26][27][28]</sup>

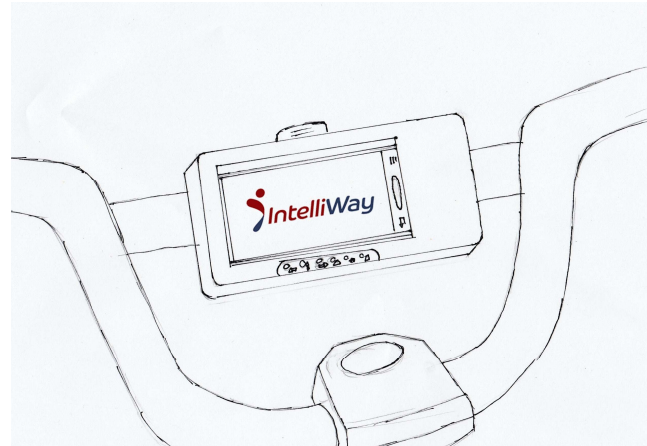
## 2.7 Art of Merging

A couple of merging protocols that were built by other individuals including a researcher in Columbia University in New York were investigated as part of our research. His project used RFID to broadcast and receive signals from nearby vehicles.<sup>[1]</sup> This poses significant security problems. However, theoretically, anything is hackable. We just needed to make sure out it isn't easily hackable. The main reason that this project did not get implemented is because of security problem. There were couple other documents that contain merging algorithms and protocols we plan to investigate some part of algorithm if we decide any part of them are relevant to what we are doing.

## 3. Design Approach

### 3.1 Introduction of IntelliWay Bike System

The approach taken towards designing the IntelliWay Bike System was to maximize both the effectiveness of the alert system and the convenience to the user. If the system is ineffective at relaying important information to bikers and drivers, then the entire system would be rendered useless. More importantly, this issue could lead to accidents due to erroneous or misinterpreted signals. Convenience is also something to consider when trying to market a product. If the IntelliWay system is deemed too tedious to work with by users, they will not want to invest in and utilize the product. The IntelliWay Bike system will try to utilize the Vehicle to Vehicle communication system and alert both drivers and cyclists in dangerous situations.



*Figure 5. Artist Rendition of Final Product*

Instead of directly evaluating effectiveness and convenience as metrics, each block in the System Block Diagram will have its own set of unique metrics. These metrics will provide the basis for all design decisions made and can be used in the future if changes need to occur. Figure 5, the System Block Diagram, denotes all the major functional blocks in the IntelliWay Bike System. Each block will be explained in a later section.

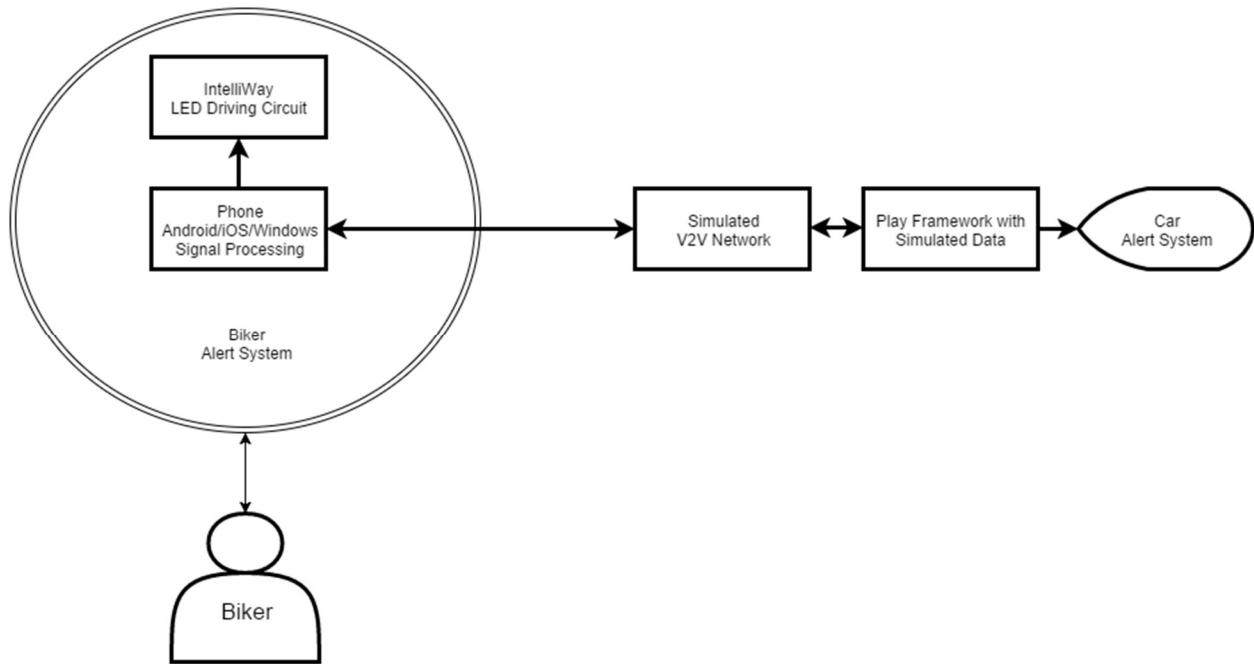


Figure 6. System Block Diagram

### 3.2 System Specifications

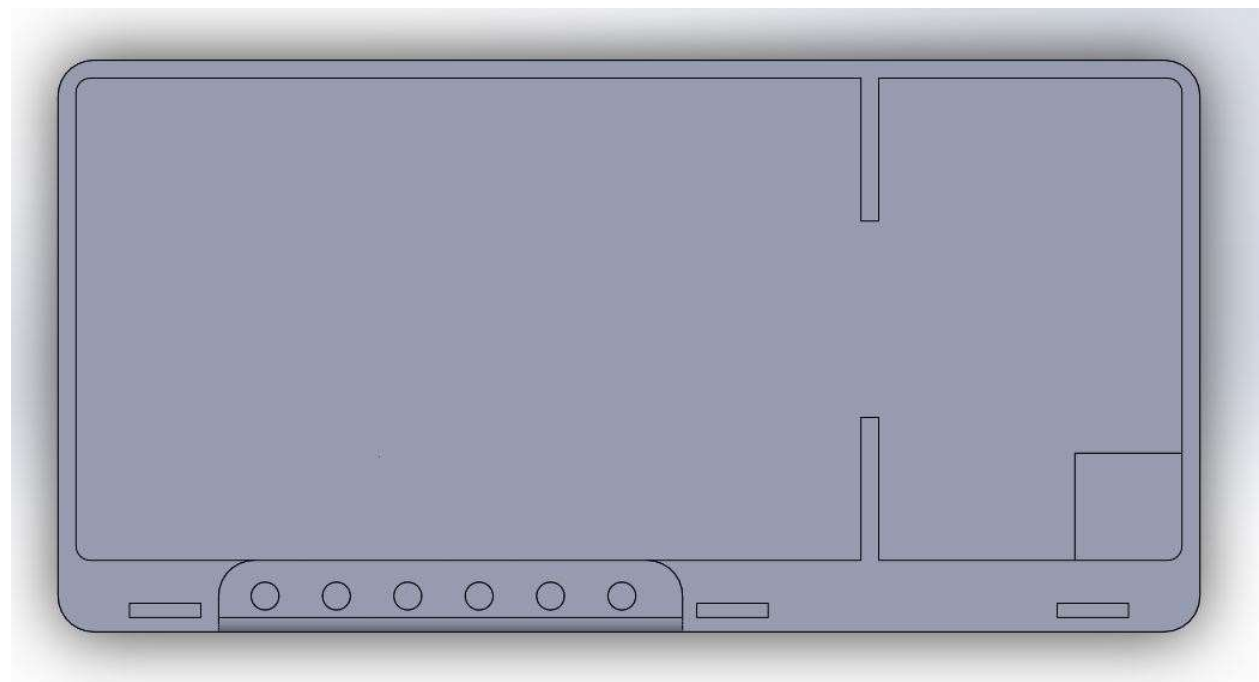
This section will give an overview of each section in the System Block Diagram. Each section will have a brief description of what they are meant to achieve and what sort of interfaces exist between blocks. Afterwards, there will be a description of the value analyses done in order to justify all decisions made. In the analyses, each section was given a set of metrics that were deemed important qualities for each block to possess. Each metric was given a weight to emphasize the importance of that attribute to the functionality of the system block. Each metric is then rated on a relative scale which is defined separately for each metric. Some design decisions will be better for a certain specific attribute, but that attribute may not be important to the overall design. Doing value analyses allows for optimizing the design decisions and is important in making solid engineering decisions.

## **IntelliWay Alert Device**

The IntelliWay Alert Device will be the collection of components attached to the bicycle. There will be LEDs to alert the bicyclist of potential dangers and various warnings. This will be supplemented by the smartphone screen. The circuitry and phone will be kept inside a waterproof container with a plastic membrane over the phone to keep it useable.

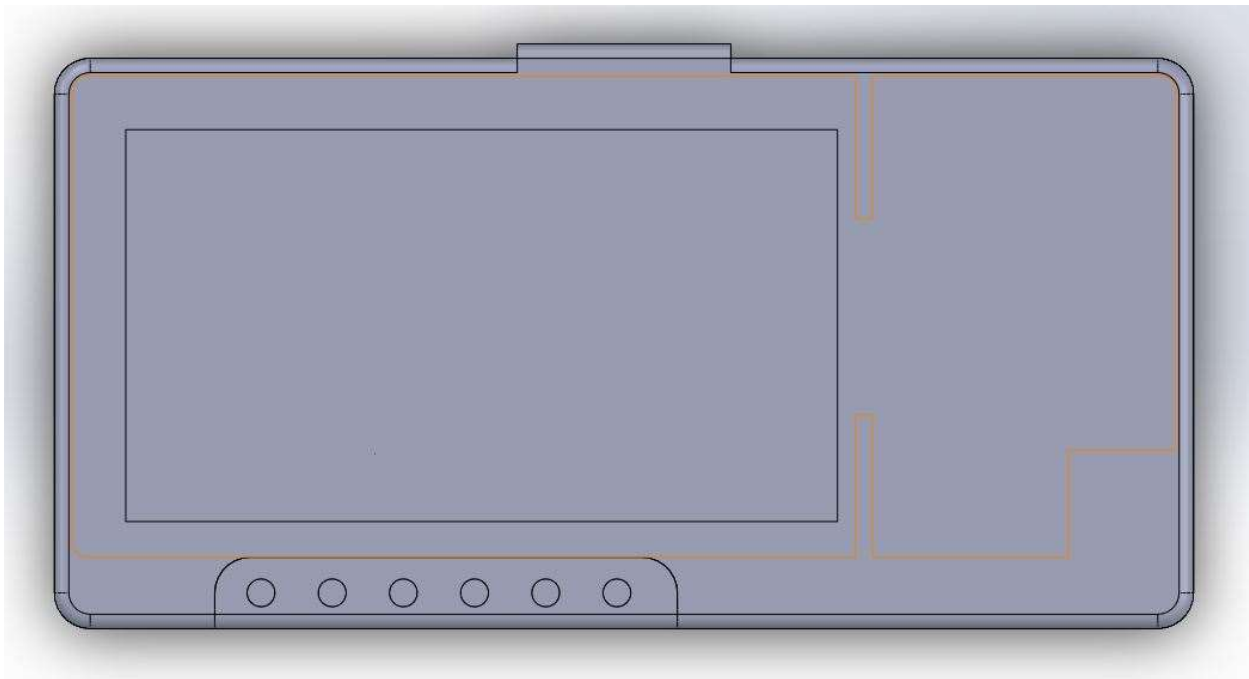
The LEDs will be driven by an integrated circuit (IC) and resistors to limit the current draw. This IC will take a serial input from the smartphone using a USB connection

The device will draw power from a USB power stick that can be easily removed for recharging. There will be a port where the USB power stick connects to the board that drives the system. The battery, wiring, and any electrical circuits will be contained within the casing, this is necessary to protect the sensitive equipment from the elements. The board will then be wired to a female USB port on the upper level of the device where the phone is stored, as shown in the figure to the right. With a USB connection to the phone, this will allow for compatibility for different devices. The upper level, will have two cavities, one to house the phone and the other to house the cable.



*Figure 7. Artist Rendition of Middle Layer*

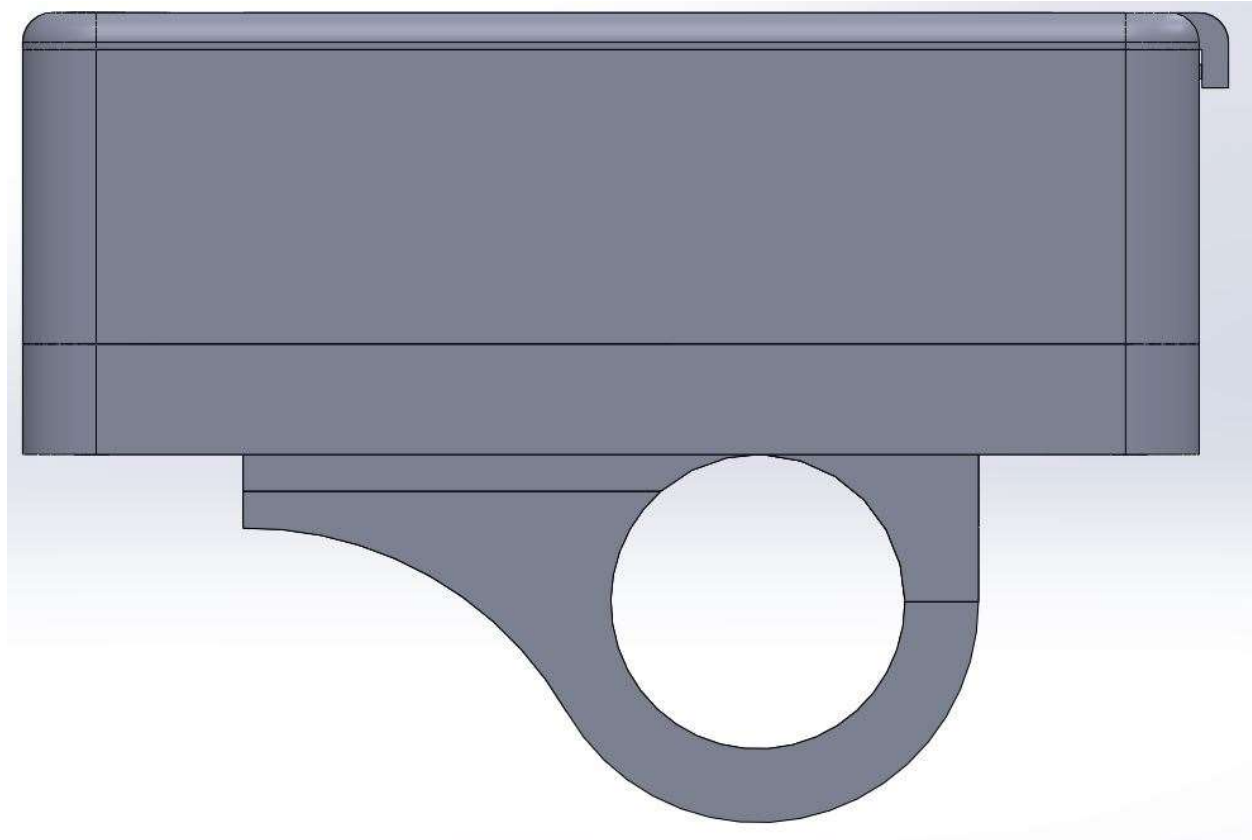
The upper level of the device will have two separate cavities, one will store the phone and the other will store the cable and is where the phone connects to the rest of the device through a USB port as shown in the figure above. This level will house multiple LEDs for simple alerts such as low battery, close proximity of a car, and proceed with caution. The dimensions of the prototype will be approximately 6.5" by 3.5". The final dimensions of the device will need to be adjusted to fit a majority of smartphones.



*Figure 8. Artist Rendition of Outside of Case*

There will be a separate cover plate for protecting and securing the phone as shown in the figure above. The cover plate will have a soft membrane so that the phone can still be operated while within the device. The cover plate will securely snap into place with an easy remove tab for quick access to the phone.





*Figure 9. Bicycle Handlebar Grip*

The device will have a cavity on one side where the USB stick can be inserted as shown in the figure above. There will be two eye loops with rubber ribbing for securely gripping the handlebars of the bike. This will allow for the operator to adjust the device to the desired angle without it slipping.

## **Power Supply**

The power supply for the IntelliWay Alert Device will be a lithium-ion USB rechargeable power supply. With this form factor, it will require 5V to recharge and will also output 5V. There will be protection circuitry built into the power stick to avoid overcharging and discharging accidents. It also provides the user with multiple opportunities to charge the battery. If they have a powered USB port anywhere nearby, such as on a laptop, desktop, or even a mounted wall unit, they can simply plug the power stick in and let it charge.

The power supply will supply both the IntelliWay Alert Device and also act as an extension of the smartphone battery. Different models of smartphones have different power consumption rates, but for the purposes of calculating worst case power consumption, the Samsung Galaxy S6 and iPhone 6 will be the two smartphones looked at.

The iPhone 6 uses 130 mA during “heavy usage”, while the Samsung Galaxy S6 drains 266 mA while net browsing on a 4G network.<sup>[14][21]</sup> These numbers will be averaged out to 250 mA, as to have a safer estimation of power consumption.

The IntelliWay Alert Device has varying degrees of power usage depending on what LEDs are active. The low-power IC has negligible power consumption, so for worst case power consumption, the IntelliWay will draw 100mA. This estimation brings the total power consumption of the system to 350mA.

*Table 2. Defining Metrics for Battery Value Analysis*

Battery	Weight	Rating Range
Size	2	1-3
Capacity	4	1-3
Safety	3	1-5
Protection Circuitry	3	1-3
Price	2	1-3
Durability	3	1-3

Using the Digikey Electronics online catalog, Adafruit, and other electronics retailers, the datasheets of three kinds of batteries, one of each chemical composition, were examined and used for the value analysis. The MaxSale 18650 was used for the value analysis for lithium ion, while the HHR-210AAC4B was the model nickel metal hydride and the KR1800SCE is a nickel cadmium battery. All datasheets can be found in Appendix B.

Table 3. Value Analysis of Battery Types

		Lithium Ion		Nickel Metal Hydride		Nickel Cadmium	
Factor	Weight	Rating	Score	Rating	Score	Rating	Score
Size	2	3	6	3	6	2	4
Capacity	4	3	12	2	8	2	8
Safety	3	3	9	4	12	3	9
Protection Circuitry	3	2	6	1	3	3	9
Price	2	2	4	2	4	3	6
Durability	3	3	9	1	3	3	9
		<b>Total:</b>	<b>46</b>	<b>Total:</b>	<b>36</b>	<b>Total:</b>	<b>45</b>

After choosing three commonly used types of rechargeable batteries and performing a value analysis, it was decided that a lithium ion battery is the best choice for this device. Lithium ion batteries are much more energy dense than nickel metal hydride or nickel cadmium batteries, but they are more expensive and require more protection circuitry. If maintained properly, lithium ion batteries also outlast the other types of batteries.

In keeping with the design decision to have the battery supply power via USB connections, three models were taken from the market and went through another value analysis. This analysis is for deciding which model we will be using for the prototype device and will serve as the basis for designing one for the IntelliWay Bike System in the future.

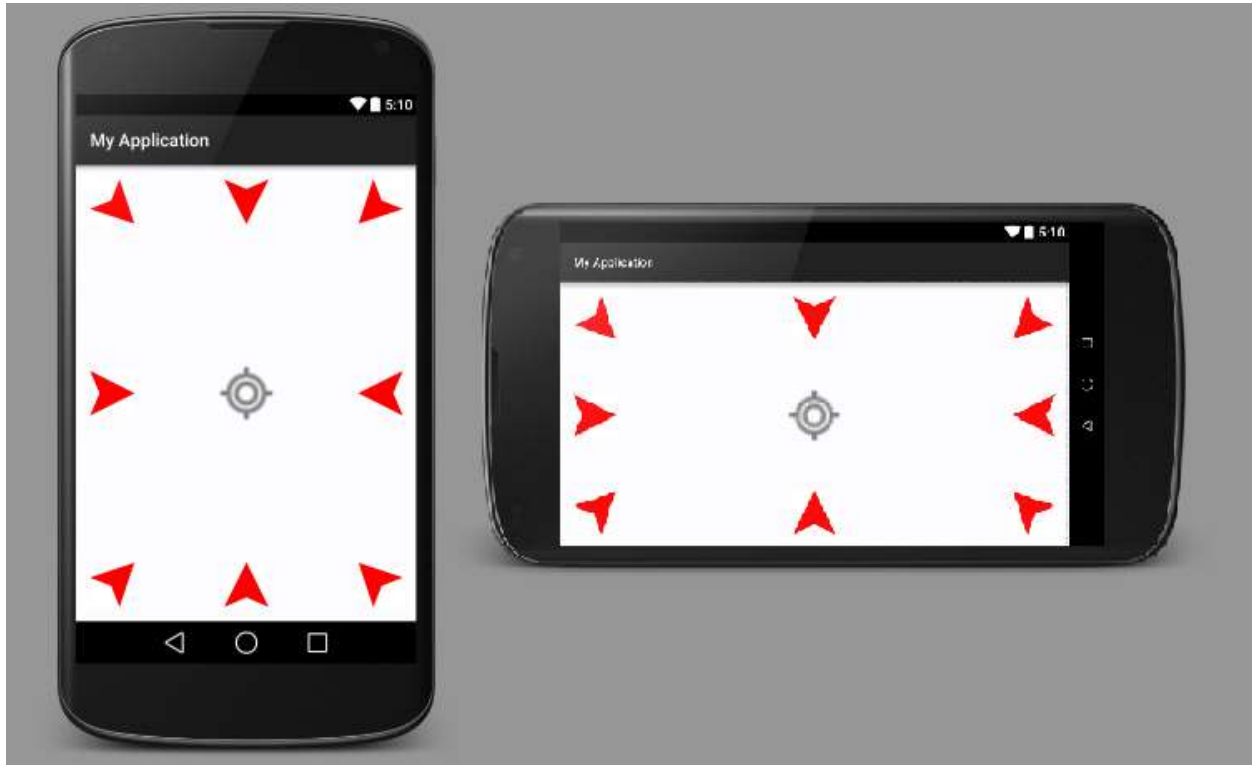
Table 4. Value Analysis of USB Power Sticks

		Mota		Killer Concept		Hype	
Factor	Weight	Rating	Score	Rating	Score	Rating	Score
Size	2	2	4	3	6	3	6
Capacity	4	3	12	3	12	2	8
Safety	3	3	9	4	12	4	12
Protection Circuitry	3	2	6	2	6	3	9
Price	2	1	2	2	4	3	6
Durability	3	3	9	3	9	3	9
		<b>Total:</b>	<b>42</b>	<b>Total:</b>	<b>49</b>	<b>Total:</b>	<b>50</b>

For the purpose of prototyping this device, the Hype 2200mAh Power Stick will be the best choice. It has slightly less charge capacity, but it makes up for that in its smaller size and price. The Hype Power Stick is actually a lithium ion polymer battery, which allows it to be smaller and is more resistant to overcharging. In the end, the MaxSale 18650, which is most similar to the Hype Power Stick, will be configured to run using a USB input and output, allowing for more control of the entire system and not having to rely on a third party manufacturer.

When all LEDs on the IntelliWay Alert Device are active and the smartphone is following the worst case power consumption model, the Hype Power Stick can keep the system working for 6.28 extra hours. The Galaxy S6's 2550mAh battery can last for 9.6 hours on its worst case power consumption estimation, while the iPhone 6's 1810mAh can last up to 14 hours.<sup>[14][21]</sup> Due to the smaller current drain of the iPhone 6, the Hype Power Stick is even more effective at keeping the system on longer than an Galaxy S6.

## Smartphone



*Figure 10. IntelliWay Application*

Smartphone will be a key component of IntelliWay system. Smartphone receives the V2V signal from the simulated V2V device (Raspberry Pi) and processes the data received. The data will have two different formats with same information. Both JSON for Object formatted data will contain positional data of latitude and longitude with degree direction of north as 0, and current speed. Smartphone will also calculate those data based off of GPS and broadcast same data to nearby devices. If the car is approaching in a dangerous speed and within a danger zone, the phone will display the direction, speed, and type of vehicle (If type is applicable) on a screen for 2 seconds. Smartphone will also broadcast its positional data to nearby devices. The mobile application will be a background application, so that at any moment, users will get interrupted with the message. There will be an option to turn this option off as well. The mockup is not final, but it is a basic idea. The way information is represented could be changed with different pictures instead of what is currently on the mockup.

In terms of the code development, the Android will get developed with Android SDK and Java libraries, and iOS will get developed with Xcode with Swift libraries. The project will develop the prototype for Android firsthand and move to iOS applications in the future due to limited Apple resources. Android devices will take input as 5V Power from power supply in order to extend battery life of phone, and GPS data from V2V simulated network. The output will be broadcasted GPS data to Pi and serial signal to TI chip for LED lights.

### **Simulated V2V Network**

To simulate the V2V Network, a Raspberry Pi Model B+ will act as a car with active V2V communication. To allow the Raspberry Pi to communicate with the smartphone, a wireless network will need to be established. Depending on the kind of wireless adapter connected to the Raspberry Pi, the process of creating this communication link will vary in difficulty. Below is a value analysis chart for three different possible choices for wireless adapters that have been proven to work with Raspberry Pi, but have very different attributes.

*Table 5. Defining Metrics for Wireless Adapter*

Wireless Adapter	Weight	Rating Range
Size	1	<b>1-3</b>
Price	2	
Speed	1	
Range	3	
Complexity	2	

*Table 6. Value Analysis of Wireless Adapters*

		Edimax		Wi-Pi		nRF24L01	
Factor	Weight	Rating	Score	Rating	Score	Rating	Score
Size	1	3	3	2	2	1	1
Price	2	3	6	2	4	1	2
Speed	1	3	3	3	3	3	3
Range	3	3	9	2	6	2	6
Complexity	2	3	6	3	6	1	2
		<b>Total:</b>	<b>27</b>	<b>Total:</b>	<b>21</b>	<b>Total:</b>	<b>14</b>

From the value analysis chart, the Edimax EW-7811Un is shown to be the best choice for this application. It is more affordable than the Wi-Fi and nRF24L01, with comparable speed and ranges, and is also very easy to set up on a Raspberry Pi. Size and speed were deemed as low priority attributes due to this only being a simulation of what will be programmed into future vehicles with V2V capabilities.

Pi will simulate the positional data of a vehicle. The data will have two different formats with same information. Both JSON for Object formatted data will contain positional data of latitude and longitude with degree direction of north as 0, and current speed in miles.

### 3.3 Testing IntelliWay Components

Since the entire system is a combination of multiple components, each component needs to be tested separately, and then the entire system as a whole. The detailed outline of schedule is proposed in next section, but each component will be tested as soon as it is created.

#### **Power Source Testing**

To test the effectiveness of our power supply, it will be connected to a circuit with a multimeter and will drain until a predetermined drop in voltage and current occurs. When it reaches this point, then the device will be recharged. This process will be repeated ten times to ensure that this power supply is reliable.

#### **Smartphone Testing**

For smartphone app, simulated data will be used to test whether the smartphone actually displays proper message. The phone will also broadcast simulated positional data of the bike and print those data for confirmation.

#### **V2V Communication System Testing**

V2V simulated device will be completely simulated. Once both V2V simulated device and smartphone are ready to be tested, both of the devices will be broadcasting and receiving data. The Raspberry Pi will need to be tested to ensure it sends out GPS data properly at a reasonable range. The smartphone will need to be tested to ensure proper serial data transfer to the IC in the IntelliWay Alert Device.

#### **IntelliWay Device Testing**

IntelliWay Device will be tested by sending simulated data to the device. Once the IntelliWay Device receives the signals from the source, the device will be tested whether proper LED lights light up or not. Once the Smartphone portion is completed, IntelliWay Device will be connected with the phone to test whether it properly lights up the LED or not.



### 3.4 IntelliWay Uniqueness

The IntelliWay device will provide the operator a simulated dashboard using integrated features with a smartphone. The smartphone will be used to process all calculations and relay information back to the operator. This information can be anything from maps, to the position of other cyclists and motor vehicles. It will include a mounting system that securely grips the handlebars and a waterproof casing for which the phone and all other components are stored. Unlike other devices, this one has the potential to warn operators of high-traffic areas, dangerous areas, and recent accidents in the surrounding area. Furthermore, there are various technologies developed for motor vehicles. However, the ones that are in most danger on the road are pedestrians and cyclists. Since bike is a fast moving object, car drivers experience difficulty in low visibility environment. Although many previous design attempts tries to resolve this issue, many of those required an unnecessary purchase such as helmet. This project attempts to interfere with current system such as smartphones and V2V communication system, so that the new device can easily adapt technological changes without requiring unnecessary purchases from the cyclists.

## 4. Implementation and Device Prototyping

### 4.1 Obstacles and Changes for Implementation

There were a couple of software obstacles during the implementation phase. The original plan was to implement the IntelliWay broadcaster using C or C++ along with real GPS data read from the GPS logger and broadcast over the WiFi. However, there were multiple problems understanding raspberry Pi and the underlying architecture.

The serious problems started when the Raspberry Pi could not connect to WPI-Wireless through one of the setup programs on WPI helpdesk. Eventually, the WiFi connection issues were resolved through a custom solution posted by a former WPI student. However, once this solution was applied, we learned that the WPI-Wireless does not allow anyone to broadcast messages over a subnet, and WPI NetOps has a very strict policy regarding the network administration on campus.

Instead of using WPI WiFi, it was decided on to use a locally hosted network, also known as hotspots. With the custom WiFi settings applied to the raspberry Pi, switching between networks was problematic. The Raspberry Pi also did not have enough memory to support the Java applications that was needed to process and send out GPS data. At this point, the broadcaster was not going anywhere. The wireless setup was problematic and trying to learn enough C and C++ to create a broadcaster were all bottlenecks in the design process.

With time running out and the lack of completion on the broadcasting portion of the simulated V2V network, the project switched gears towards using Java layers and mocked GPS data. There was no longer a need to spend excessive amount of time to get the GPS broadcasting working with the Raspberry Pi. The concept of using the Raspberry Pi as a simulated car on a V2V network had been officially scrapped. This also included any wireless adapters that would be required for the Raspberry Pi.

The last major change involved the external power supply for IntelliWay. Previously, it was assumed that it would be possible to both charge the smartphone and power the LED driving circuit. This assumption turned out to be true, as using USB On-The-Go (OTG) would allow the smartphone to act as the USB host to the driving circuit, while also acting as an accessory looking out of the external power source. Setting up this three-way system required further research into USB OTG specifications. This part of the project was deemed uncritical in the successful completion of the MQP, and was left out of the rest of the design.

## 4.2 IntelliWay Hardware Device

The casing for the prototype was designed in Solidworks. The casing consists of four unique parts. The cover plate, which encloses the phone with an opening for a potential membrane so that a smart phone can still be accessible while in the device. The phone midsection, has a cavity to store the phone along with the phone's cable. There are six open holes on the top of the piece for any necessary LEDs. On the midsection piece, there is an opening for a female-USB port, so that the phone can then be connected to the rest of the device. The bottom plate, is where all the internal hardware is stores this includes any potential batteries and circuitry. To securely attach the device to the bike, there are two clamps that are used to grip the handlebars of any bike.

The original design had some minor complications, which were easily correctable. The phone cavity was originally too small to fit the newer, much larger, smartphones. The base plate was also too shallow and needed to be deepened to fit the hardware. All of the corrections were made within the source files. However, to save time from printing instead of reprinting the midsection and baseplate a separate  $\frac{1}{4}$  inch spacer was printed to deepen the baseplate and the inner walls of the midsection were cut out and glued back in at the correct spots.

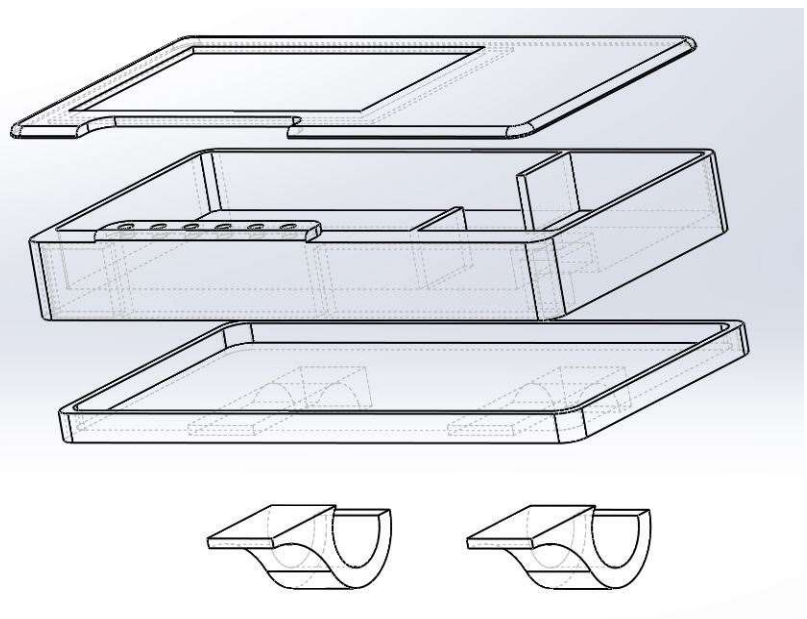


Figure 11. Solidworks Design of IntelliWay Case

### 4.3 IntelliWay V2V Simulated Broadcaster

The Wi-Fi subnet broadcast was implemented in the following way. The application uses Play Framework's Datagram and Socket Java Libraries. Furthermore, the application needed to be asynchronous, so Java Runnable, which is like a thread, can be utilized to run the program asynchronously without any hanging problem.

#### **Technical Problems with Raspberry Pi & Wi-Fi**

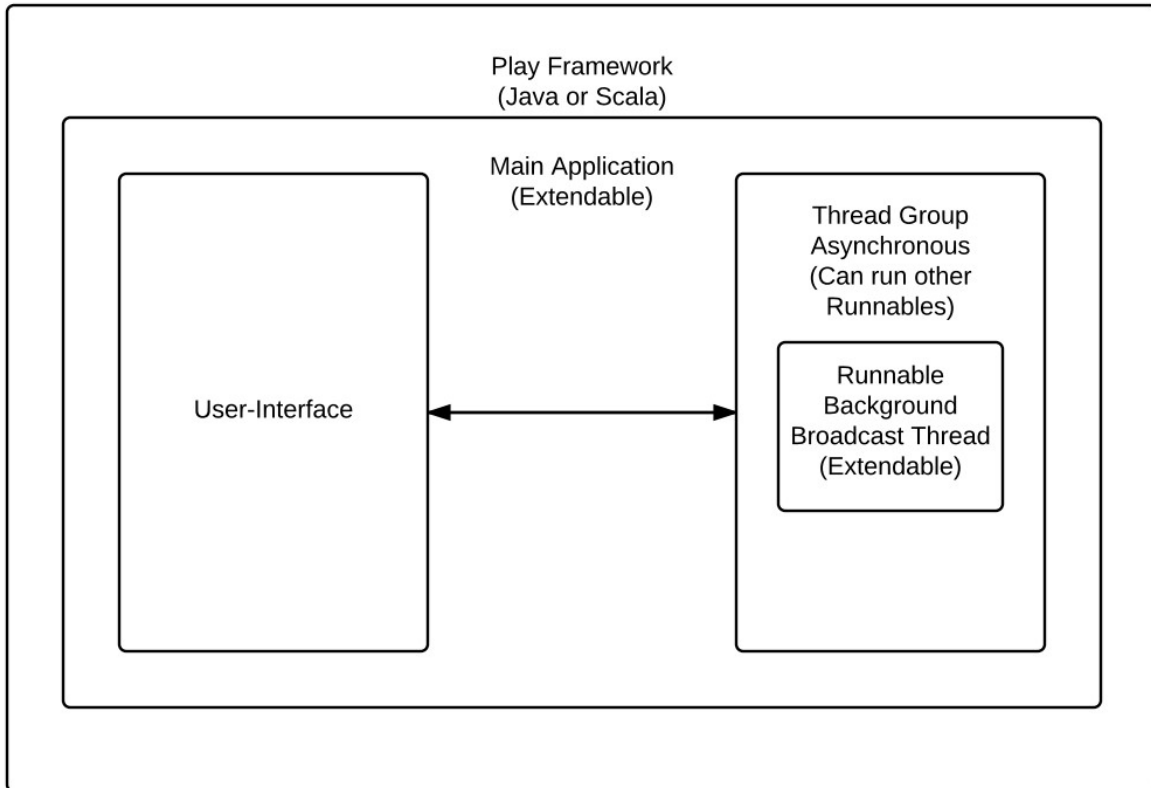
The original idea was to use C and C++ to implement the broadcaster. However, there were couple technical problems with this implementation.

The Raspberry Pi had to be wirelessly connected, but WPI does not have an official program that supports Wi-Fi registration for Raspberry Pi. Thankfully, we were able to connect to WPI Wireless using a previous student's walkthrough. WPI does not grant permission for students to freely broadcast messages over Wi-Fi and the Raspberry Pi was customized so that it could only connect to WPI Wireless instead of other custom networks.

Furthermore, the GPS data collection was not possible from Raspberry Pi. The original Wi-Fi Module that we purchased was not compatible with Raspberry Pi, so Raspberry Pi was not able to gather any real GPS data from a Wi-Fi module.

The Raspberry Pi was very slow in terms of development. Unfortunately, there was a lack of deep understanding of C or C++ programming. By this stage, three weeks have been wasted on the Raspberry Pi, so we had to make an important decision regarding the broadcaster and switched to a JVM based Play Framework to write Java-based program that could run on Raspberry Pi or other Java-based machines. After switching to Play Framework, the initial broadcaster was successfully tested with a simple broadcast message.

## Design & Architecture



*Figure 12. Broadcaster Architecture*

In order to effectively and efficiently modify and extend the broadcaster, a modularization principle was applied to the software design process. The IntelliWay V2V Broadcaster is only in the prototype stage, so a lot of work will be required to be fully implemented. Therefore, the initial software design had to be easily extendable. The V2V simulated Broadcaster is designed in two portions: the main running application and extendable thread running group as shown in Figure 12. The main application would start and initiate a broadcasting thread group. Each thread group would run and simulate a V2V broadcast.

Originally three different broadcasting methods were employed to test various broadcasting mechanisms. It was decided on to use the instant broadcast to all device mechanisms because other mechanisms took longer time to broadcast individual subnet addresses. The asynchronous thread group was separated from the main application. Therefore, the user interface thread would never hang, as well as the network thread, allowing to run various jobs without hanging. The main application can run other applications or extend itself and multiple threads. Extended runnable objects can be run on asynchronous thread group to allow flexible design. Furthermore, the user interface can be modified and changed, as well as communicate with the asynchronous runnable threads to update the user interfaces.

### **GPS Data Object**

The intervals between sent data points were set to 1 second to simulate a car broadcasting GPS data. In order to simulate real GPS data sent from another vehicle, GPS Data objects were created to simulate the GPS information. GPS Data objects contains the latitude, and longitude in doubles format with unique vehicle information, along with the DateTime object representing the time the GPS data was recorded. Every second, the Discovery Thread gets GPS Data from a queue, which was implemented through linked lists, using an iterator, and converting the GPS Data objects to byte arrays using ObjectMapper.

### **Broadcasting Data over Wi-Fi**

In order to start broadcasting messages, a port need to be reserved on the sender and receiver. Since some of port numbers are used for operating systems and other applications, IntelliWay uses a port number greater than 5000. Port 8888 was reserved for broadcasting messages. Initially the port was not opening correctly for an unknown reason, but the issue was fixed after a recompilation.

After the port was reserved, the datagram socket needed to be initialized and set for broadcasting. Once the socket was properly opened, DatagramPacket needed to be initialized with the byte array created from ObjectMapper, along with the reserved port. The packet also required a destination address which can be specified to one specific destination IP address or an entire subnet. In order to broadcast to all devices in the subnet, the destination address had to be 255.255.255.255.

For an unknown reason, the broadcasted messages would not arrive to the Android receiver, and the source of the issue could not be identified at first. The Android device was not receiving any information, and neither application was returning an error messages. The Android Studio Debugger was utilized to find the source of problem, and later the issue was found on the Android side port reservation. After the port problem on the receiver side was fixed, the broadcaster was confirmed to be broadcasting successfully.

## 4.4 IntelliWay Android Broadcast Receiver

The current version of the smartphone application is only operational on Android smartphones. Future implementations would include the iOS version of the mobile application, but Apple holds a lot of security changes to themselves. However, if the iOS devices are really required for future releases, having a central web-server to broadcast all registered iOS application could potentially resolve the Apple's security issue. Another alternative is editing the info.plist file on the iOS development side, which can be very restricted.

The Android Broadcast Receiver was designed to receive a broadcasted message over Wi-Fi and compute all necessary computations and display potential hazards to the users. The Broadcast Receiver can also send or broadcast messages to various devices on subnet as well, so two-way communication is possible, but not implemented for the IntelliWay prototype.

The Android Broadcast Receiver is a bit more complex than the broadcaster device because of different Android standards that had to be considered while also managing the user interface layer.

### Design & Architecture

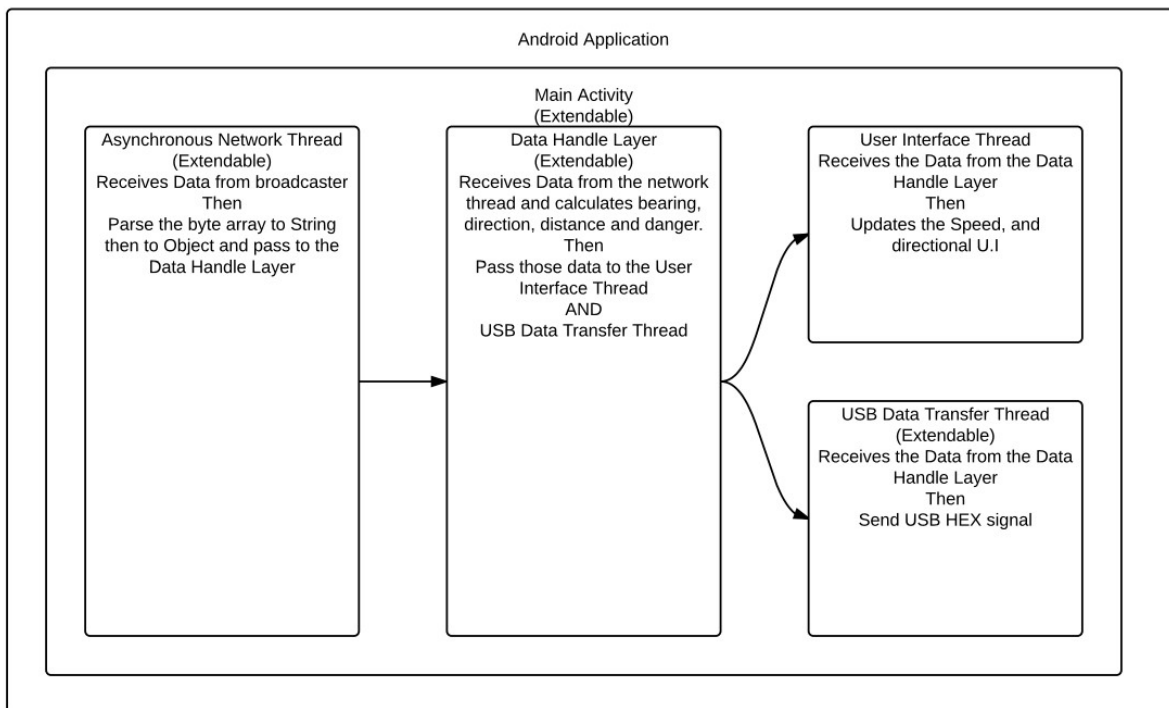


Figure 13. Android Broadcast Receiver Architecture

Designing the architecture of the Android Broadcast Receiver was more complex due to the extra data-handle layer for processing GPS data from the broadcaster and another extra layer for connecting to the USB device in host mode to transfer the byte signal to IntelliWay LED Driver. Within the Android application, Main Activity gets launched. Main Activity initializes the user interface layer from the XML layout, creates a USB Data Transfer thread, starts the asynchronous network threads to receive broadcasted messages, and connects each layers with the data handle layer. Just like the V2V broadcast simulator, every component was modularized in order to make future implementations and extensions smoother. Android activity, the network thread, the data handler, and the USB Data Transfer thread can be easily extendable and modifiable as shown in Figure 13.

### **User Interface Layer**

When the Android application starts up, the user interface layer gets initialized. Each unique element in the user element layers can be specified through a unique String of id, and can be referenced through the Java layer. Each hazard direction arrows were implemented using ImageButton class to easily embed arrows.

### **Asynchronous Network Layer**

Not only does the application initialize the user interface layer, but the application also initializes the network thread using the AsyncTask object and starts the location manager to grab the current location object, which contains latitude, longitude, and the current bearing. The original implementation was copied from the broadcaster with Runnable, but, unfortunately, Runnable implementation was not acceptable in the Android Broadcast Receiver because the network thread had to run in AsyncTask. AsyncTask is an asynchronous network thread that does all the network operations in the background, so the Runnable implementation was migrated to the AsyncTask job. AsyncTask waits for the broadcasted message by listening to network address of 0.0.0.0 on port 8888.



### **Data Handle Layer**

Broadcasted messages get delivered to the smartphone. The smartphone receives the broadcast messages and sees if there is previous GPS data cached to the device. If there was no previous data, it stores the current data with a specific vehicle ID and does nothing. However, if there was previous GPS data from that specific vehicle, it grabs the current broadcasted data in bytes then parses the byte array into a String object. Once the String is successfully parsed, it parses the String representation of the byte array into a custom GPS object using GSON JSON parser. After the GPS data is created, the application calculates the distance between the current device's location and the vehicle's location and determines whether the vehicle is approaching towards the device. If the application determines that such movement is dangerous, the user interface updates and shows the vehicle approaching towards the current location. Once the vehicle passes beyond the current location and is determined to no longer be a threat, the user interface is updated, showing no hazards.

The key for updating the user interface is that the user interface update must be run on the UI thread. The Android reserves a UI thread so that anything else that runs other than the UI thread will not properly update the user interface layer.

The user interface has total of 8 directions: North, Northeast, East, Southeast, South, Southwest, West, and Northwest. Each direction is calculated after the bearing gets calculated from the current given latitude and longitude information to the threat vehicle's latitude and longitude. The calculation formula is major direction  $\pm 22.5$ . For example, North would be between  $360 - 22.5 > \text{bearing}$  or  $\text{bearing} < 22.5$ .

### **USB Host Data Transfer Layer**

Once the USB device is connected through an Android Broadcast Receiver, the receiver detects a USB device and requests for host permissions. This portion of the code was initially separated into a different module for testing only. Afterwards, this module was connected with the data handle layer. The USB Host Data Transfer layer requires a `targetVendorID` and `targetProductID` to search for the desired product. If the device is found, that device information gets cached into the `UsbDevice` object. If the desired IDs are not found, any connected USB device gets connected after requesting host permission. The key thing to note here is that the `UsbDeviceReceiver` manager must be initialized and registered in order to start asking permissions for the USB device.

Once a USB device is properly recognized with correct permission, the USB Endpoint should be properly configured. Since `bulkTransfer` is used to transfer data, `USB_ENDPOINT_XFER_BULK` from `UsbConstants` class should be used and `USB_DIR_OUT` from the `UsbConstants` class should be used for getting direction. The USB Device must be open and initiates access control with `controlTransfer` and the correct `requestType` hex bytes as well. There are encoding settings for different devices. The encoding setting used was `[0x80, 0x25, 0x00, 0x00, 0x00, 0x00, 0x08]`, for a total of 8 data bit and 1 stop bit.

The data transfer between the Android Devices and the USB Device has to be `bulkTransfer`. There are two types of data transfer available: `bulkTransfer` and `controlTransfer`. `BulkTransfer` is used for sending byte array data to the USB device whereas `controlTransfer` is used to gain permissions or other controls from the USB Device.

The most important portion of the implementation that should never be ignored is properly releasing the USB Device. Upon any disconnect, the USB device must be properly released in order to minimize any side effects and memory leaks. When a USB device gets detached, the USB connection must be closed, the USB interface released, and the USB Device set to null at each endpoint. All the code is available in the Bitbucket repository.

## 4.5 IntelliWay Internal Hardware

The LEDs on the IntelliWay device are part of this device to add additional visibility during times of intense daylight, when the phone screen is not easily visible. The MSP430G2553, a microcontroller part of a low-power series made by Texas Instruments, was the chosen microcontroller used to drive the LEDs.

The use of the MSP430 LaunchPad and Energia was critical in setting up the MSP430G2553. The MSP430 LaunchPad allows the quick prototyping of the MCU programming. Four of the GPIO pins on the MSP430G2553 have been defined as outputs. These pins are the pins that turn on the LEDs. The reset pin has an RC circuit attached to it. The values of the resistor and capacitor are defined in the MSP430 LaunchPad schematic.

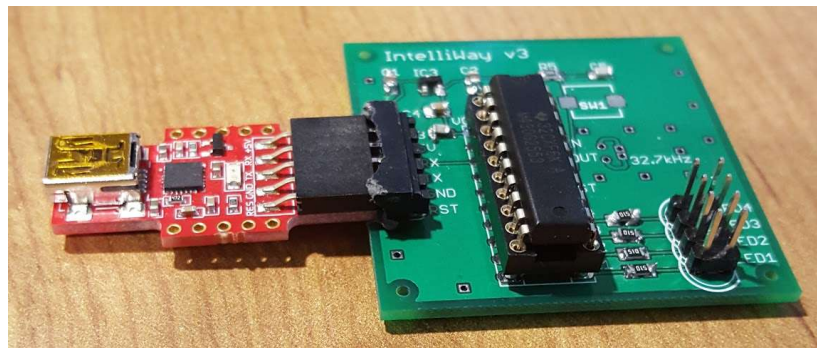


Figure 14. IntelliWay LED Driving Circuit with  $\mu$ USB-PA5

The hardware serial interface is used by a few lines of code due to the libraries that come predefined in Energia, a prototyping platform developed to interact with the MSP430 LaunchPad. The MSP430G2553 receives serial data, and depending on the value of the byte received, one or more of the LEDs will light up.

To receive USB data from the smartphone, there needs to be a conversion process to turn USB data into serial data. Normally, on the MSP430 LaunchPad, this is done using a combination of the TUSB3410 USB-to-Serial Bridge and the MSP430F16x. Without these, there is a need for an external system to do this process. The  $\mu$ USB-PA5, a USB-to-Serial Bridge, will handle the data conversion between the smartphone and the MSP430G2553. The  $\mu$ USB-PA5 interfaces with the microcontroller's UART pins, the input of the voltage regulator, the ground connection, and a reset pin.

A printed circuit board (PCB) was custom designed in Eagle. The design is a simple two-layer board with 0805 package sizes for resistors and capacitors, a SOT23 footprint for the voltage regulator, a pad area for the attachment of a hard reset button, a surface mount pad area for the attachment of the  $\mu$ USB-PA5, a through-hole area for the attachments of LEDs, and a through-hole area for the attachment of a 20-pin DIL socket component. The socket will allow the removal of the MSP430G2553 in case any adjustments to the programming is necessary. Each LED has a current limiting resistor to make sure that they don't burn out. There are several bypass capacitors in the circuit itself to make sure that regulated voltages don't experience any irregular drops. The reset button is connected to an RC circuit with values that were taken from the MSP430 LaunchPad datasheets.

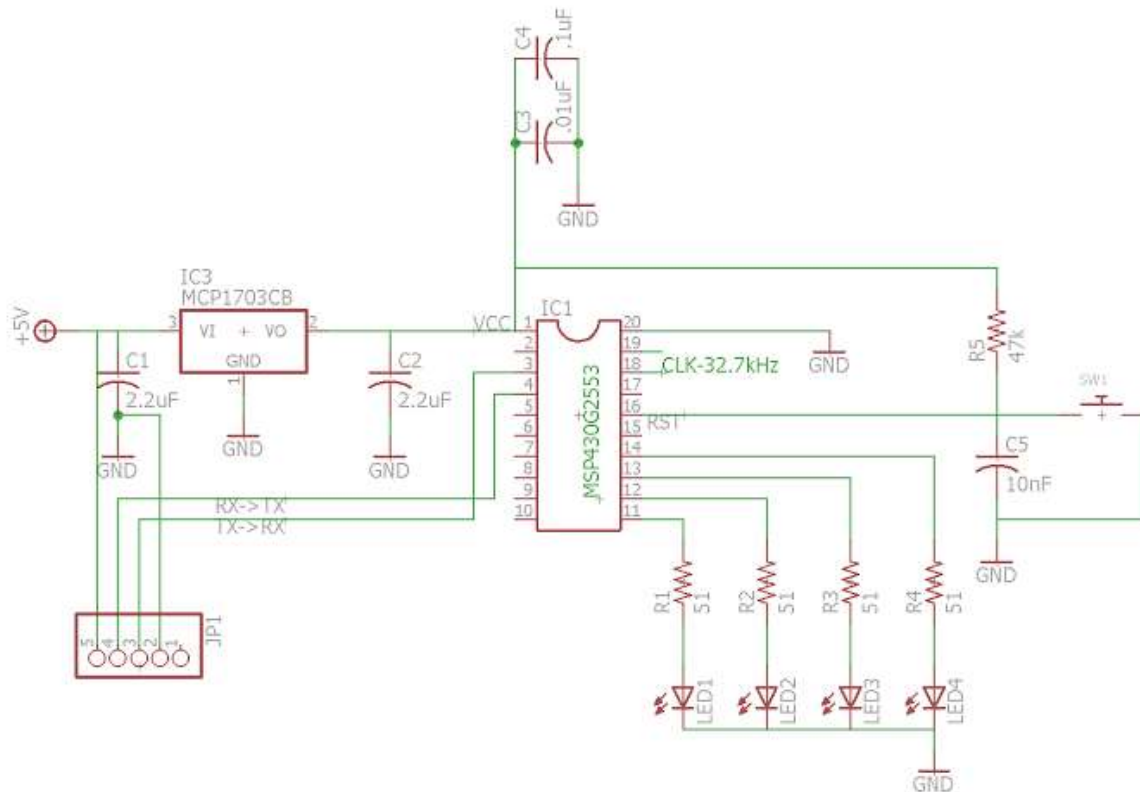


Figure 15. LED Driving Circuit Schematic

In the first iteration of the design, it was assumed that an external clock source would be required for the functionality of the MCU, but the serial command seems to automatically set a clock frequency from the internal voltage oscillator. If this weren't the case, then the external watch crystal would be mounted to pads connected to the XIN/XOUT pins of the MSP430G2553. A ground island would then be placed around this component to reduce electromagnetic interference from the board.

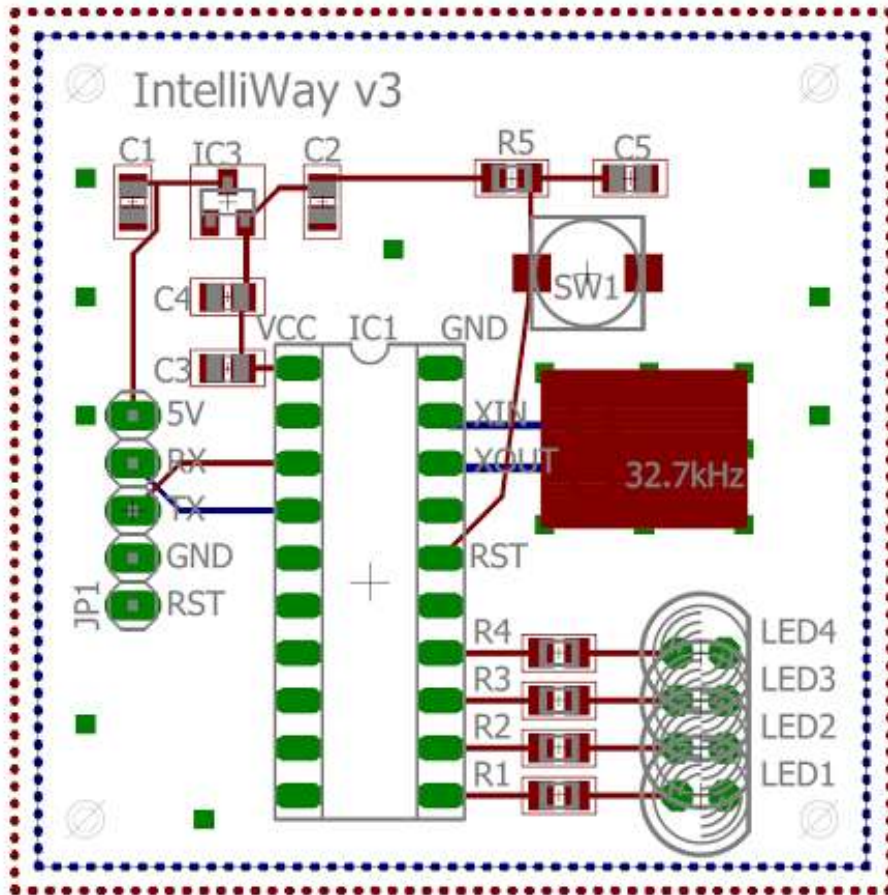


Figure 16. LED Driving Circuit 2-Layer Gerber Drawing with Keep-out Layer

## 5. Testing & Evaluation

### 5.1 Testing V2V Simulated Broadcaster

The V2V simulated broadcaster was a simulator, so the actual data gathering from the GPS device was ignored. However, the key component in the broadcaster was that it was actually broadcasting data over entire subnet.

Three different Android devices were connected to a mobile hotspot hosted by a Samsung note 5 because of network security issues in Worcester Polytechnic Institute's networking policy. In order to see whether the actual broadcasting was happening, various Android devices were connected to a single subnet and tested to see whether broadcasted messages actually arrived and contained exact data that was broadcasted. After testing, Galaxy S6, Note 3, and Note 5 were able to successfully receive a broadcasted messages constantly.

Although mobile hotspot served as a WiFi subnet, we wanted to make sure it would still work in an actual WiFi router environment. Furthermore, we wanted to make sure to disable the security firewalls on a WiFi router if required to be, so we tested the broadcaster with Comcast powered internet router. The broadcaster worked as if it was working in a hotspot and successfully broadcasted to all connected devices without any security limitations.

Initially the GPS data was all mocked, but the broadcaster actually had to simulate a moving vehicle, so we utilized an Android mobile app called GPS Logger that would log GPS data into an Android device for a certain duration. GPS Logger would support multiple GPS data formats for recording GPS data, but all we needed for the data points were latitude and longitude, so we used a GPX format which includes both data points with 1 second interval for GPS logging.

Since the Android Broadcast Receiver supports 8 different direction, we wanted the test data to drive in various direction. However, there are no real intersection with 8 different direction combined, so we decided to use a 4 way intersection and drive 4 direction directions both inbound and outbound.



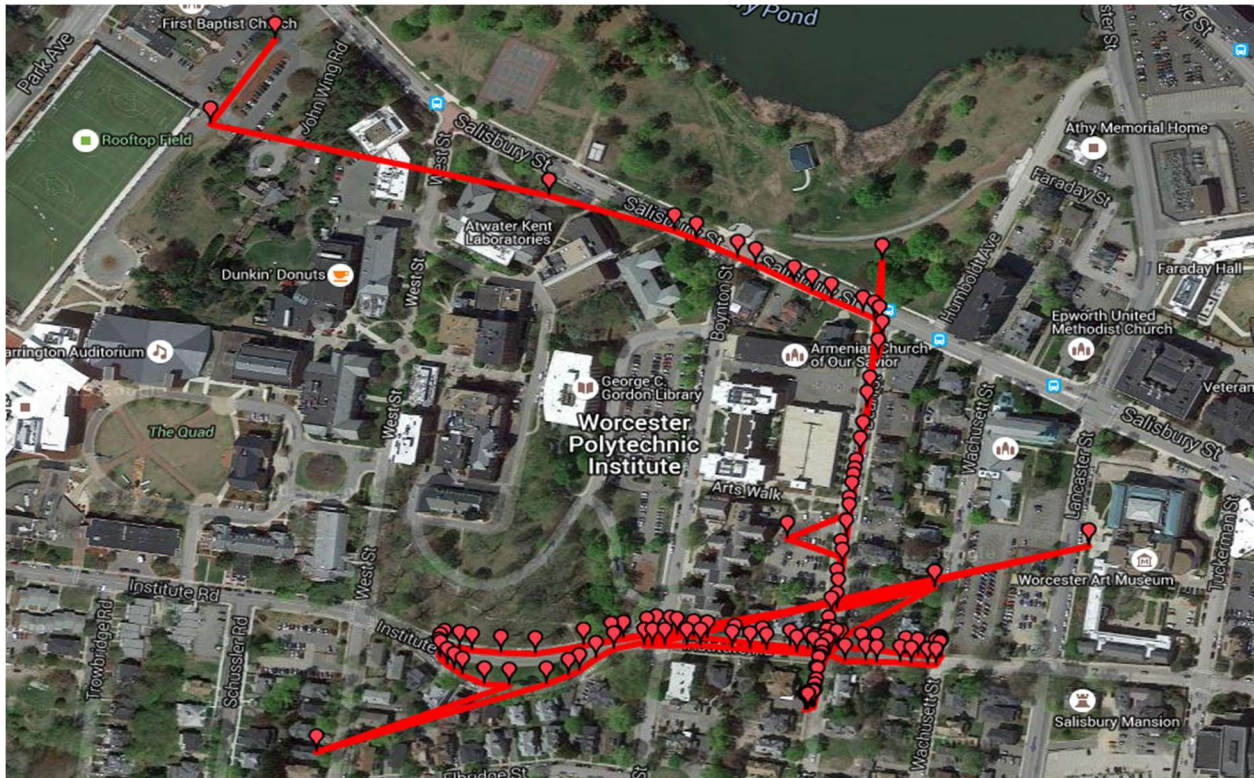


Figure 17. Captured GPS Data around WPI

Although we could use the entire GPS data that was logged, we thought normalizing and cleaning up the collected GPS data would be helpful in the future. GPX is a raw GPS coordinate data in a form of XML, so we thought visualizing the gathered data would be helpful. Online GPX data visualizer and editing tool WTracks was utilized to analyze the recorded path and normalize the location data.

Initially GPS logger started from the WPI Gym then ended at East Hall parking garage. However, the data was really dirty because it recorded unnecessary data points while driving down to the resident hall. Furthermore, the android device connected various WiFi access points while driving around campus, creating a chaos due to recording WiFi location instead of actual precise location for the GPS signal received from satellite. In order to make testing easier, unnecessary data points and irrelevant WiFi location points have been removed. After the normalization, the data points look like the following.

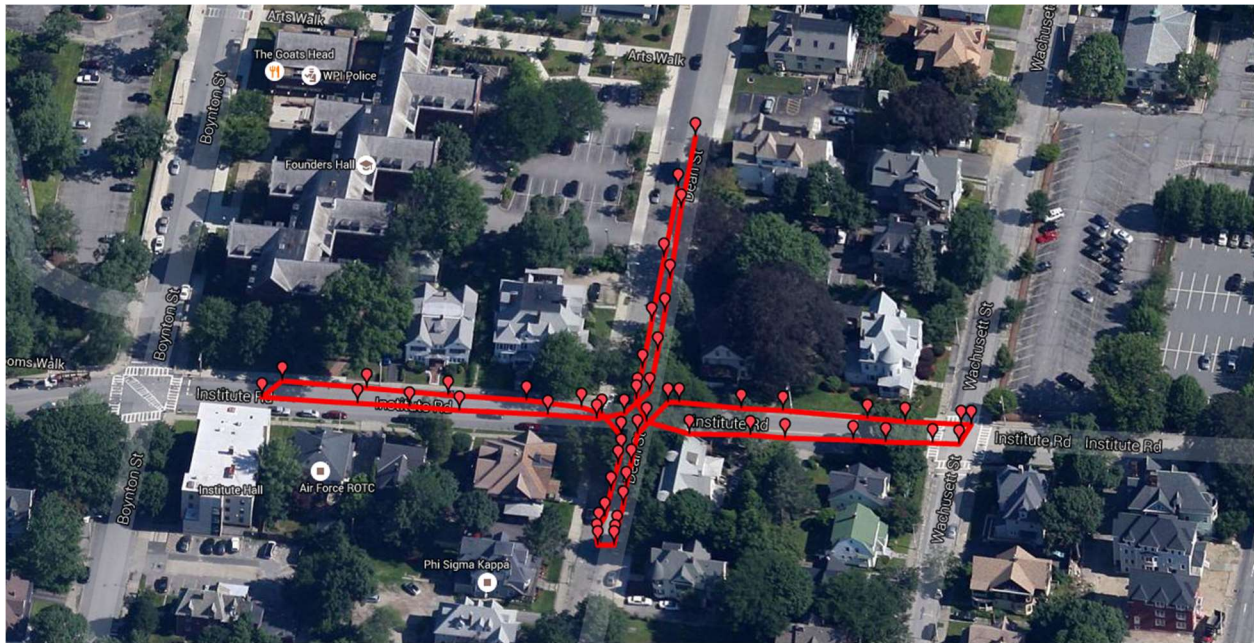


Figure 18. Cleaned-up GPS Data Points

Lastly, the normalized data was exported out to GPX format. Unfortunately, we had to convert the GPX XML format to Java GPSTData object. There are various tools out there for java based libraries to parse the XML nodes, but I felt that was overly complicating for importing test data, so I simply utilized text editing tool, sublime text, to clean and update from XML to Java object.



## 5.2 Testing Android Broadcast Receiver

There are two major component in Android Broadcast Receiver: receiving broadcasted messages and updating the user interface based on the received data. Testing the Android devices were more complex because there was an extra layer in Android Broadcast Receiver that would have to calculate and handle the user interface changes or updates.

Android Debugger was used to confirm whether data received was the data broadcasted. With each given data point after data broadcast, the broadcaster would output a data that was broadcasted. On the receiver end, once the data arrives, debugger would hang the thread to show what data has arrived and logged for the record. The output for both broadcaster and the receiver was compared and determined to be same.

Once the broadcast receiver receives the GPS Data, the receiver calculates relative bearing from current receiver position to the moving vehicle, distance from current receiver to the vehicle, and the speed of the vehicle. Bearing and distance could be calculated by two different data points, so the formula and manual test values were compared using an online tool developed by Andrew Hedges.

Once the calculations were determined to be correct, the user interfaces were tested. User interface simply takes all the calculated data and updates the user interface layer, displaying where the possible threat is and whether it is approaching. Because we gathered a good set of data covering at least 4 different direction, various scenarios could be tested using those data. During the testing phase, we discovered that displaying east and west directions were switched. When the car was approaching from east, west direction was displayed and vice versa. We have detected and analyzed that the bearing calculation was correct, so we thought the user interface layer was calling incorrect layer. In fact, that assumption was correct, and the user interface layer was properly updated. After the user interface layer was correctly fixed and updated, it was determined that all 4 direction properly calculates and updates the user interface layer.

## 5.3 Testing PCB

Two PCBs were purchased for prototyping. If anything happens to the first board, there is a backup. The board will need to be tested to ensure that all signal nets are properly connected. With so many parts coming off of the board itself, it is critical to ensure that all connections are sufficiently secured. The socket for the MSP430G2553 will be mounted on a through-hole connection, and if any of the pins are soldered together, then there could be serious problems. The positive terminals of the LEDs can't touch or the brightness will drop dramatically. If the button for the reset doesn't drive the pin low for enough cycles, then there will be no way of refreshing the chip in a situation that requires it.

A multimeter will serve as the testing equipment for making sure the connections are isolated and well established. It can be used to see if the power coming into the board is at the correct voltage. The output voltage of the regulator also needs to be at 3.3V, and that can also be tested with a multimeter. The LEDs won't need to be tested using the multimeter, as they will either light up or turn off depending on the output of the associated pin.

## 5.4 Overall Evaluation

The  $\mu$ USB-PA5 functions without any hitches when connected to a computer with Energia, but when connected with the smartphone, several problems began to arise. With the power coming from the  $\mu$ USB-PA5 5V power line, the output voltage of the regulator was measured at 2.72V, which is still well within the operating region for the MSP430G2553. This lower operating voltage does have an impact on the drive power. With 3.3V, the output pins could support a high enough voltage to turn on all LEDs, but with this lower voltage only blue LEDs are able to turn on.

The serial monitor in Energia sends out the data to the USB-to-Serial Bridge and the MSP430G2553 processes the data. Problems arose when the smartphone applications handling USB transfer protocols would not successfully interface with the  $\mu$ USB-PA5. In the code, there is a section that calls for control transfers between the smartphone and the USB device. There is no specific documentation on what values allow for this routine to be successful. Much of the documentation found involves using Linaro 13.10, an old release of Linux, and delving into the source code to find the drivers. This proved difficult to finish in the amount of time that was left, and it was much easier to allow the LaunchPad to handle the USB-to-Serial conversion, as that had already been proven successful with the smartphone.

## 6. Conclusion

For IntelliWay to be a viable product idea, it needs to compete with already existing systems that attempt to have bicycles and cars communicate with each other. With this in mind, IntelliWay was designed to keep the product requirements in mind. It has a simple UI that can effectively show the direction of the nearest approaching vehicles, but only as long as the vehicle is on the V2V network. With the recent push to equip all new cars with V2V technology, more and more cars will start using the V2V network, which in turn will make IntelliWay better and better at detecting approaching vehicles. Unlike the Volvo City Safety System, there will be no cloud connection between the smartphones and vehicles. The smartphone will have the capability of directly accessing the V2V network and cars will transmit the GPS data directly to the cyclist. The BikeShield Application requires all cyclists and drivers to have the application installed and running on their smartphones. With IntelliWay, only the cyclist will need to have the application. The IntelliWay software will utilize the existing V2V system to communicate directly with vehicles instead of between smartphones.



*Figure 19. Back of IntelliWay*

IntelliWay has gone through many intense changes from start to finish, including the intent of the product. At first, IntelliWay was conceived as an addition to highway infrastructure to assist motorists in making safe merges and entrances on and to the highway. After doing much research on the efficiency of integrating new infrastructure to the already expansive highway system, this idea was scrapped. The next idea was to take the concept of merging onto highways and add a system directly to the vehicles. This avoided making any changes to the highway system and was much more easily implementable. New vehicles could be outfitted with this new system and older ones could have it installed aftermarket. The issue that came with this was the already expansive literature and implementations of this system already in place. It didn't seem like a logical direction to take in a marketing sense if we had to compete with systems developed by Google or Tesla. With this in mind, the focus shifted towards being able to prototype a system of communication that no other company has perfected yet. This is when it was decided that this project will aim to showcase the possibilities of incorporating bicycles into the upcoming V2V network standards.



*Figure 20. Angled View of IntelliWay*

With IntelliWay, there are two outcomes that are trying to be achieved. The first, and most important one, is to create a system for cyclists to be more aware of the vehicles around them. This device aims to make sure that cyclists are safer when they embark on their commutes or when they are out for fun. The second outcome is to set up a framework for utilizing smartphones and LEDs to create a device that can interface with a network and share GPS data.



*Figure 21. Front View of IntelliWay*

The focus of this project has been to find a way to enhance the cycling experience by finding a way to allow cyclists to tap into the upcoming technologies of Vehicle-to-Vehicle Communication. Statistics show that there are real dangers that cyclists face. Many of these dangers revolve around automobile collisions. In dense urban areas where there are many cars and bicycles sharing the road, the chances of a collision increase greatly. By creating and prototyping IntelliWay, this project aims to show how it is possible to connect cyclists with the upcoming V2V network by using smartphones.

# Bibliography

1. Woldesenbet A, Jeong HD, Park H. Framework for Integrating and Assessing Highway Infrastructure Data. *Journal of Management in Engineering*. 2015:4015028. <http://ascelibrary.org/doi/10.1061/%28ASCE%29ME.1943-5479.0000389>
2. Goh KC, Yang J. Managing cost implications for highway infrastructure sustainability. *International Journal of Environmental Science and Technology*. 2014;11:2271-2280. <http://link.springer.com/article/10.1007%2Fs13762-014-0572-5>
3. Shatz HJ, Ebrary Academic Complete, Rand Transportation, Space, and Technology (Program), Books24x7 I. Highway Infrastructure and the Economy: Implications for Federal Policy. Santa Monica, CA: Rand Corporation; 2011. <http://au4sb9ax7m.search.serialssolutions.com/?V=1.0&L=AU4SB9AX7M&S=JCs&C=T C0000523150&T=marc&tab=BOOKS>
4. Claburn, Thomas. (5/12/2015). *Google Automated Car Accidents: Not Our Fault*. <http://www.informationweek.com/software/enterprise-applications/google-automated-car-accidents-not-our-fault/d/d-id/1320374>
5. Munson, Glenard (2/17/2015). *Why haven't turn signals been automated?*<http://www.quora.com/Why-havent-turn-signals-been-automated>
6. Knight, Will (3/4/2015). *Car-to-Car Communication: A simple wireless technology promises to make driving much safer*. <http://www.technologyreview.com/featuredstory/534981/car-to-car-communication>
7. Marcus, Olivia (3/4/2015) *Car-to-Car communication May Hit Roads Soon*. <http://www.usnews.com/news/articles/2015/03/04/car-to-car-communication-may-hit-roads-soon>
8. Kallstrom, Henry (2/5/2015). *Government regulations push auto research and development*.<http://marketrealist.com/2015/02/government-regulations-push-auto-research-development/>

9. Kalra, Nidhi (1/29/2015). *To Hit the Road, Driverless Cars Must Be Safe, Not Perfect.* <http://www.rand.org/blog/2015/01/to-hit-the-road-driverless-cars-must-be-safe-not-perfect.html>
10. Kim, B.H., Maxemchuk, N. *A Safe Driver Assisted Merge Protocol.* <http://safevehicle.ee.columbia.edu/docs/Kim12.pdf>
11. Choi, B., Lin, S., Peters, E. *Extended Driver-Assisted Merging Protocol* [http://www.ee.columbia.edu/~nick/EE6778/Final\\_Reports/Merge.pdf](http://www.ee.columbia.edu/~nick/EE6778/Final_Reports/Merge.pdf)
12. Greenberg, Andy. (7/21/2015). *Hackers Remotely Kill A Jeep on the Highway-With Me in It.* <http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/>
13. Pagliery, Jose. (6/2/2014). *Your car is a giant computer - and it can be hacked.* <http://money.cnn.com/2014/06/01/technology/security/car-hack/index.html?iid=EL>
14. Pagliery, Jose. (6/10/2014). *Talking cars: the next hacking target.* <http://money.cnn.com/2014/06/10/technology/security/talking-cars-hacking/>
15. Pagliery, Jose. (3/31/2014). *Tesla car doors can be hacked.* <http://money.cnn.com/2014/03/31/technology/security/tesla-hack/?iid=EL>
16. Glon, Ronan. (8/6/2015). *Tesla Issues Software Patch After Hackers Take Control of A Mode S.* <http://www.digitaltrends.com/cars/tesla-model-s-hack-shuts-down-car-news/>
17. Fox-Brewster, Thomas. (2/2/2015). *BMW Update Kills Bug In 2.2 Million Cars That Left Doors Wide Open To Hackers.* <http://www.forbes.com/sites/thomasbrewster/2015/02/02/bmw-door-hacking/>
18. Honda WorldWide. (3/8/2007). *Honda Auto Parts Manufacturing Co., Ltd. Begins Operations in China.* <http://world.honda.com/news/2007/c070308AutoPartsManufacturing/>

19. Notte, Jason. (5/5/2014). *10 Most American-Made Cars of 2014*. <http://www.thestreet.com/story/12690707/1/10-most-american-made-cars-of-2014.html>
20. Kiplinger. (July, 2011). *10 Features Your Car Will Have By 2020*. <http://www.kiplinger.com/slideshow/cars/T009-S001-10-features-your-car-will-have-by-2020-slide-show/index.html>
21. Deloitte. (2009). *A new era Accelerating toward 2020 -- An automotive industry transformed*. [http://www2.deloitte.com/content/dam/Deloitte/pt/Documents/manufacturing/pt\(en\)\\_dc\\_accelaratingtoward2020\\_05112009.pdf](http://www2.deloitte.com/content/dam/Deloitte/pt/Documents/manufacturing/pt(en)_dc_accelaratingtoward2020_05112009.pdf)
22. Goldstein, Phil. (7/18/2014). *Lawmakers introduce bill to open up 5.9 GHz band for Wi-Fi use*. <http://www.fiercewireless.com/tech/story/lawmakers-introduce-bill-open-59-ghz-band-wi-fi-use/2014-07-18>
23. Dozza, Marco. (3/7/2013) *BikeCOM: A cooperative intelligent system using bicycle to car communication*. [https://www.youtube.com/watch?v=kK\\_G9wShj2g](https://www.youtube.com/watch?v=kK_G9wShj2g)
24. Carl. (2014). *The BikeShield App*. <https://thebikeshieldapp.com/>
25. Newcomb, Doug. (12/30/2014). *Volvo Develops Car-to-Bicycle Communication Prototype*. <http://www.cthereport.com/volvo-develops-car-bicycle-communication-prototype/>
26. Fallah, Alborz. (2/26/2014). *Bicycle-to-vehicle communications system in the works*. <http://www.caradvice.com.au/272112/bicycle-to-vehicle-communications-system-in-the-works/>
27. Ho, Joshua. (4/17/2015). *The Samsung Galaxy S6 and S6 edge Review*. <http://www.anandtech.com/show/9146/the-samsung-galaxy-s6-and-s6-edge-review/3>
28. Kypreos, Evan. (9/22/2015). *iPhone 6 Battery Life*. <http://www.trustedreviews.com/iphone-6-review-battery-life-and-verdict-page-6>



29. Federal Highway Administration. *Pedestrian and Bicyclist Crash Statistics*.  
[http://www.pedbikeinfo.org/data/factsheet\\_crash.cfm#No1](http://www.pedbikeinfo.org/data/factsheet_crash.cfm#No1)
  
30. City of Boston. Bike Statistics. <http://www.cityofboston.gov/bikes/statistics.asp>

# Appendix A

## Power Consumption of MSP430G2001

### Typical Characteristics - Active Mode Supply Current (Into $V_{CC}$ )

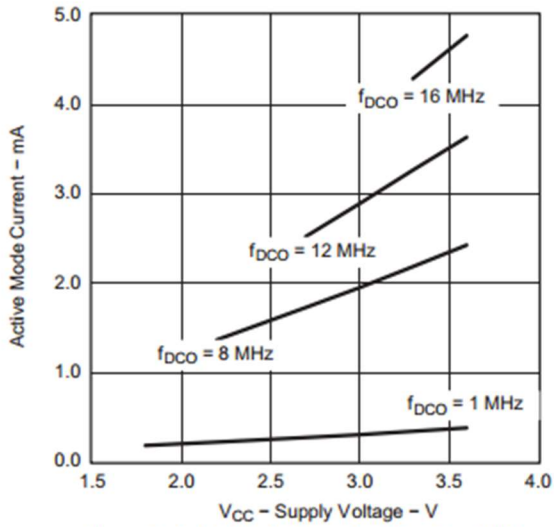


Figure 2. Active Mode Current vs  $V_{CC}$ ,  $T_A = 25^\circ\text{C}$

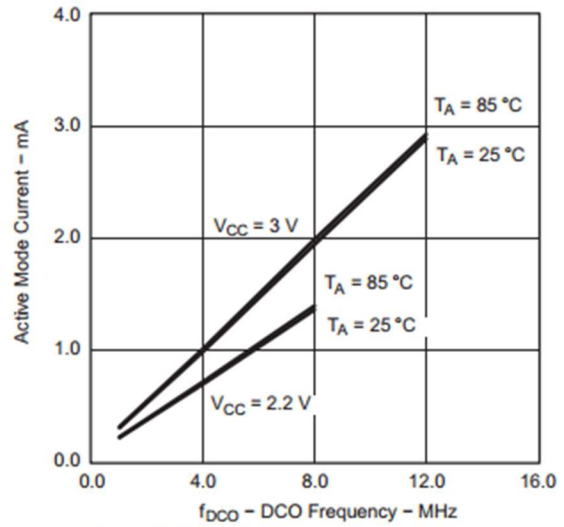
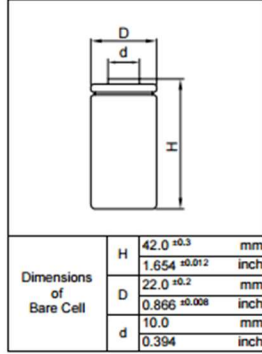


Figure 3. Active Mode Current vs DCO Frequency

# Appendix B

## Battery Datasheets - Discharge Characteristics

**SANYO**  
**Cadnica**

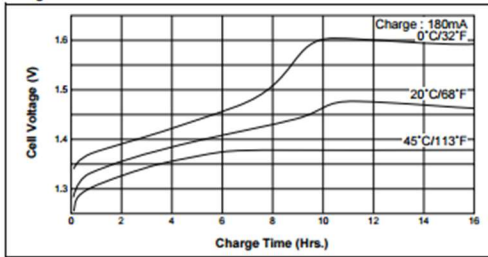


### Cell Type KR-1800SCE Specifications

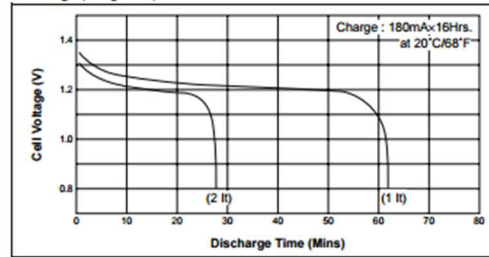
Nominal Capacity		1800mAh
Nominal Voltage		1.2V
Charging Current	Standard	180mA
	Fast	2700mA
Charging Time	Standard	14 to 16Hrs.
	Fast	about 1Hr.
Ambient Temperature	Charge	Standard 0°C to +45°C [+32°F to 113°F]
	Discharge	0°C to +45°C [+32°F to 113°F]
	Storage	-20°C to +60°C [-4°F to 140°F]
Internal Impedance (Av.) (at 50% discharge)		6.5mΩ (at 1000Hz)
Weight		49g/1.73oz
Dimensions(D)×(H) (with tube)		22.9 <sup>0</sup> <sub>-1</sub> × 43.0 <sup>0</sup> <sub>-1.2</sub> mm 0.90 <sup>0</sup> <sub>-0.04</sub> × 1.69 <sup>0</sup> <sub>-0.05</sub> inch

### Typical Characteristics

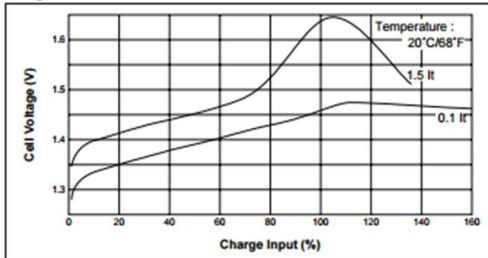
#### Charge



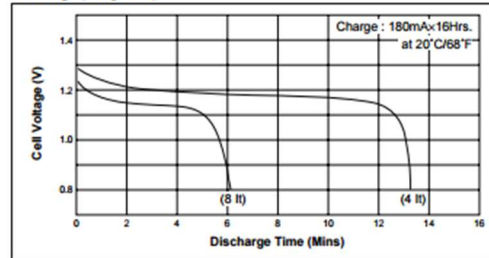
#### Discharge (at high rate)



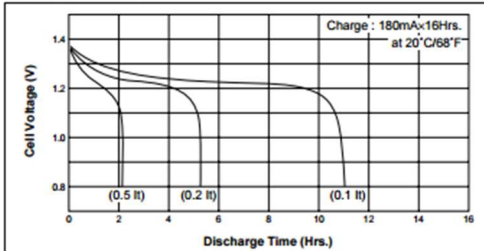
#### Charge



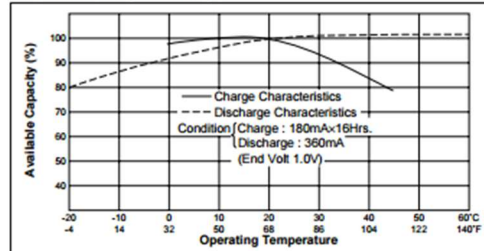
#### Discharge (at high rate)



#### Discharge (at low rate)



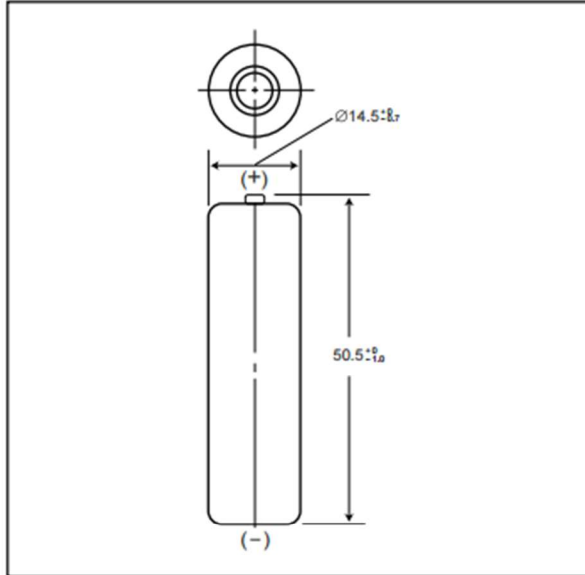
#### Temperature (Charge & Discharge)



CD030825

# HHR210AA/B Cylindrical AA size (HR 15/51)

## Dimensions (with Tube) (mm)

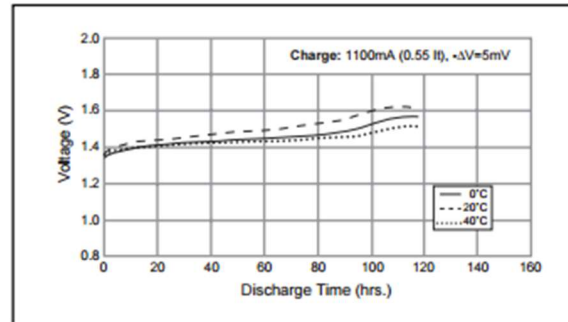


## Specifications

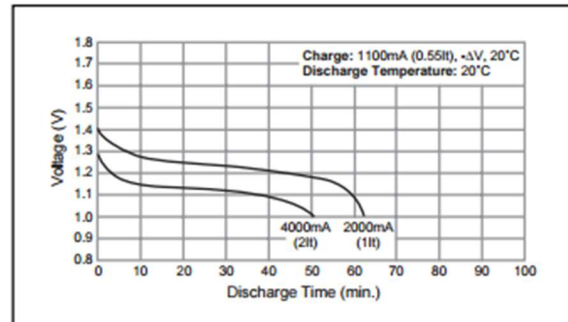
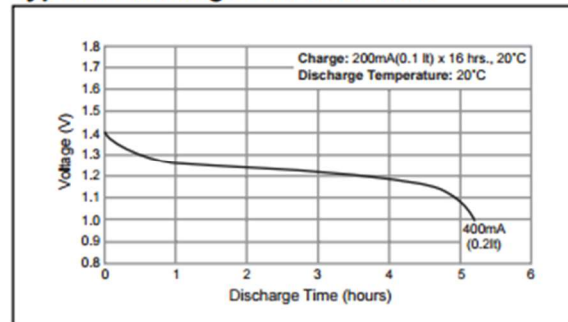
	mm	inch
Diameter	14.5 +0/-0.7	0.57 +0/-0.3
Height	50.5 +0/-1.0	1.99 +0/-0.5
Approximate Weight	Grams	Ounces
	29	1.02

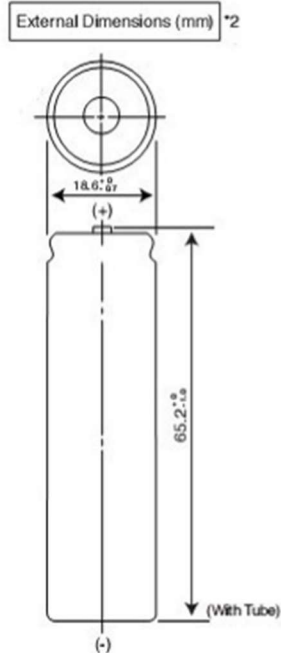
Nominal Voltage		1.2V	
Discharge Capacity*	Average**	2080mAh	
	Rated (Min.)	2000mAh	
Approx. Internal Impedance at 1000Hz at charged state.		25mΩ	
Charge	Standard	200mA (0.1It) x 16 hrs.	
	Rapid	1200mA (1It) x 2 hrs.	
Ambient temperature	Charge	Standard	0°C to 45°C 32°F to 113°F
		Rapid	0°C to 40°C 32°F to 113°F
	Discharge		-10°C to 65°C 14°F to 149°F
		< 1 year	-20°C to 35°C -4°F to 95°F

## Typical Charge Characteristics

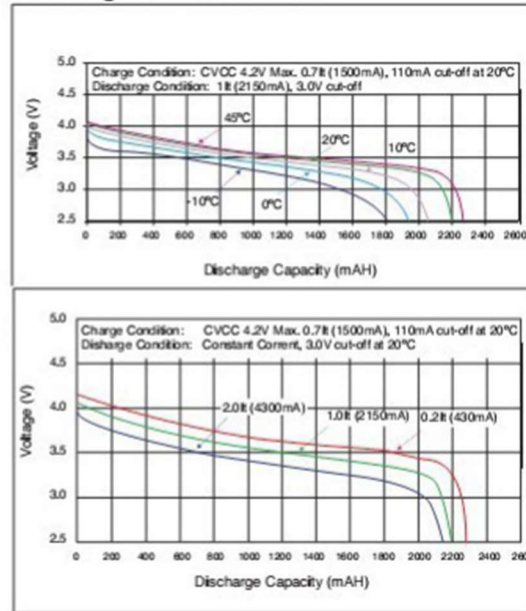


## Typical Discharge Characteristics





## Discharge Characteristics



## 8. PCM

### ELECTRIC CHARACTERISTICS (at 25°C)

- |  |                |
|--|----------------|
| 1) Overcharge Detection Voltage        | 4.2V±150 mV    |
| 2) Overcharge Release Voltage:         | 4.05V±100mV    |
| 3) Overdischarge Detection Voltage     | 4.8V± 160mV    |
| 4) Overdischarge Release Voltage:      | 3V±1200mV      |
| 5) Overcurrent Detection Voltage:      | 0.15V±120mV    |
| 6) Overcurrent Detection:              | 4.00A--7.00A   |
| 7) Overcharge Detection Delay Time:    | 0.55S--2.06S   |
| 8) Overdischarge Detection Delay time: | 67--141ms      |
| 9) OverCurrent Detection Delay Time:   | 6.3--14.7ms    |
| 10) Consumption Current                |                |
| --Operating mode:                      | Max 14.2 u A   |
| --Power — saving mode:                 | Max 0.2 u A    |
| 11) Maximum Operating Temperature:     | -40°C-- +85°C  |
| 12) Maximum Storage Temperature:       | -55°C-- +125°C |
| 13) Maximum Input Voltage:             | 4.2VDC         |
| 14) Maximum Charge Current:            | 1.8A           |