PROJECT SENSE

Ву

Aakash Rangan

Abhay Narayanan

Jerome Dinakar

Final Report for ECE 445, Senior Design, Spring 2023

TA: Vishal Dayalan

03 May 2023

Project No. 67

Abstract

The typical human cannot constantly perceive and process multivariate information, particularly if instantaneous decisions are required on matters of great importance. An excellent example of this quandary can be seen in a very typical situation: driving. Vehicle drivers are frequently unaware of specific positions of smaller, more prone parties, such as bicyclists in a multitude of scenarios. Project Sense was undertaken in order to solve this problem by creating a system that can inform both bicyclists and vehicle drivers about their relative distances from each other. The results of the project determined that the problem could not be sufficiently solved given the equipment and time constraints due to faulty GPS results. The following paper details the implementation of Project Sense.

Contents

| 1. Introductioni |
|--|
| 2 Designiii |
| 2.1 Design Procedureiii |
| 2.2 Design Detailsiv |
| 3. Design Verification |
| 3.1 Project Testingvi |
| 3.2 Requirement & Verificationviii |
| 4. Costs & Scheduleix |
| 4.1 Partsix |
| 4.2 Laborix |
| 4.3 Schedulex |
| 5. Conclusionxi |
| 5.1 Accomplishmentsxi |
| 5.2 Uncertaintiesxi |
| 5.3 Ethical considerationsxi |
| 5.4 Future work & Broader Impactsxii |
| Referencesxiii |
| Appendix A Requirement and Verification Tablexiv |
| Appendix B High-Level Requirementsxvi |
| Appendix C Schedulexvii |

1. Introduction

Throughout the world, roads are diverse in layout, rules, obstructions, structural integrity, levelness and many other factors. As such, there are many distractions or situations that could cause unfortunate collisions between vehicles. While larger vehicles are more noticeable to others, smaller ones such as bicycles are less conspicuous, which could lead to more fatal accidents for the parties using the latter. Therefore, a system that can sufficiently notify both cyclists and drivers about each others presence would go a long way towards solving the problem in question.

The method employed to solve the aforementioned problem involved a pair of Radio Frequency (RF) transceivers (one each for a car and bicycle) that communicate exact location data to one another using GPS modules. The microcontroller of each PCB (Printed Circuit Board) would take its own GPS position and the other's position (received by the transceiver) as input and depict its relative distance from the counterpart through a colored LED system. A red LED would indicate extremely close (less than 10 m), a yellow one would indicate a relative distance of between 10-30 meters, and no glow at all would indicate it being greater than 30 meters.

The performance requirements remain unchanged from the beginning of the project. Please view Appendix B to see them.

Overall, there were two main block-level changes that occurred over the course of the project design. First, as we changed the method to find the relative distance from time of flight to one that uses GPS coordinates, there is a new GPS module in the Control Subsytem. Second, the Display and Speaker were removed from the Car's PCB and replaced with an LED system like the bike's PCB. This was because we revised the scope of the project and made it only compute relative distances in a single direction. In this case, an LED system was sufficient to understand distances, as it is obvious to the user what direction the driver/cyclist is coming from. A minor change was omitting the voltage regulator as all components now ran on 4.7 V.

Considering the components described are cheaply available and not overly complex to incorporate **[1]**, the design is undoubtedly economical to produce on a larger scale, along with being sufficiently reliable owing to GPS coordinates being easily obtained and frequently updatable. The finalized product was further tested extensively, with each component checked for appropriate results. The design requires testing of the transceivers' prompt communication, the GPS' position estimation, and the microcontrollers' final decision, which was depicted on the LED system. This results in a solution that appropriately solves the problem of drivers and bikers potentially being unaware of each others' existence, along with specific distances.

For a visual representation of the aforementioned subsystems, please view the Block Diagram in Figure 1 that shows the two PCBs (Bike and Car), and the three subsystems that each has (Power, User Interface, Control). Naturally the power subsystem is responsible for powering the PCBs (an average of 4.7 V supply). The Control and User Interface Subsystems work in tandem to determine and display relative distances between any given vehicle and bicycle. The signals and data are emitted/received in the Control subsystem, while the results are computed in the microcontroller, which is in both subsytems, so

to speak. This is finally communicated to the cyclist/driver through the LEDs, which is in the User Interface system.





The remainder of the paper further elucidates in the following manner. Section 2 describes and justifies design choices. Section 3 details the design verification. Section 4 addresses specific costs related to project creation, while Section 5 wraps up with a conclusion speaking of the project's scope and any concerns that may arise with its usage. The appendices include diagrams and specific terminology that may aid with understanding of the contents of this paper.

2 Design

The goal of this section is to provide an elucidation regarding a variety of topics relating to specific design choices, equations, and the schematic.

2.1 Design Procedure

To start with the simplest subsystem, the power subsystem has a Battery that runs on 4.7 V on average. This was a very simple design choice to make because it would make implementation much easier if all components could use the same source without need for voltage regulation.

The User Interface subsystem had several methods to implement, for its purpose is to notify the user about the position of its counterpart (a bike or car). For instance, the notification could be voice activated. However, the time for a sentence conveying the information is undoubtedly longer than a simple LED flash. Furthermore, the user would have to pay closer attention to understand the proximity measure to be conveyed. Another alternative was the originally planned display for the vehicle's PCB, that would plot where the bicycles were in relation to it, with the speaker serving as a notification that a new bicycle was detected in range. Not only is this more complex, it started to lose meaning as the project switched from detecting bicycles or vehicles in all directions to just a single direction. There was no longer a need for a scaled visual representation, as an LED indicating proximity is easily understandable when the premise of the device is to detect objects in a single direction. The speaker was not necessary as the LED served to indicate as soon as a bicycle was in range for the car. There was no change in the User Interface for the bicycle's PCB.

The Control subsystem also could've been implemented in many different ways; a typical example is flight time of the signal, which could compute exact distances between a bicycle and vehicle. The reasoning behind not using this method can be easily seen from the following simple computations:

Assuming a car and a bicycle 30 meters apart, a signal (at the speed of light) would be transmitted between them in a mere

 $30/(3 \times 10^8) = 0.1 \,\mu s.$

Even assuming the clocks between the two devices were synchronized (which is not the case), accurately determining the distance using the very short time time computed above is far too inaccurate for transceivers of this quality. The precision of these transceivers would be subpar, as there will be minor differences through different readings. In practice, this would give a large range of distances, which directly fails to accomplish the purpose of the project. For instance, a mere additional 0.01 µs (a potential precision margin for the transceiver), which results in a time of 0.11 µs would result in a perceived distance of 33 meters. At this distance, our system wouldn't even light up an LED. Undoubtedly, this is a significant issue. As such, the time of flight method was infeasible for the Control subsystem. Our chosen method was for each PCB to have a GPS system, as even inexpensive ones would display accurate positions. There is also no need to synchronize clocks of the two systems. The Microcontroller would then take the GPS coordinates of its own PCB and the received coordinates of its

counterpart as input, and use the *Haversine Equation* to calculate the great-circle distance between the two points. This would clearly be more accurate, as clarified in the following paragraph and section 2.2 respectively.

To clarify, the only major design equation that Project Sense requires is the usage of the Haversine Formula, which is a very useful tool to determine distances between two points on a sphere given their respective latitude and longitude coordinates **[2]**. This is otherwise known as a great-circle distance. Note that this premise assumes the Earth is a perfect sphere, which is quite reasonable considering the difference in scale between these distances, and the circumference of the Earth.

2.2 Design Details

Having made all the design decisions previously discussed, the team decided to implement the project as seen in Figure 2:



Figure 2: Visual Detailed Design, representing the 5 major components: The 4.7 V Battery, the Microcontroller, the GPS Module, and the Transceiver.

For the sake of thoroughness, let us walk through a clear application of the Haversine Equation, which is the main method of computing exactly what the users of Project Sense would need.

Consider two points A and B, with latitudes and longitudes represented by φ and λ , we can further label that φ_1 , φ_2 are the latitude of point A and point B, while λ_1 , λ_2 are the longitude of point A and point B respectively.

Let the central angle θ between two points on the surface of a sphere be defined:

$$\theta = d/r$$

where:

• *d* is the distance between the two points, and

• $r = 6.371 * 10^6$ m is the radius of the Earth .

And the Haversine of θ (hav(θ)), to be found from the coordinates by:

 $hav(\theta) = hav(\varphi_B - \varphi_A) + cos(\varphi_A) \times cos(\varphi_B) \times hav(\lambda_B - \lambda_A)$

Therefore, as our quantity needed is the value *d*, we can simply rearrange and solve for it as follows:

 $d = r \times \operatorname{archav}(h) = 2r \times \operatorname{arcsin}(\sqrt{h})$. [done by applying the inverse haversine to $h = \operatorname{hav}(\theta)$]

This is explicitly shown in Figure 3:

$$egin{aligned} &d=2rrcsinigg(\sqrt{ ext{hav}(arphi_2-arphi_1)+(1- ext{hav}(arphi_1-arphi_2)- ext{hav}(arphi_1+arphi_2))\cdot ext{hav}(\lambda_2-\lambda_1)}igg) \ &=2rrcsinigg(\sqrt{ ext{sin}^2igg(rac{arphi_2-arphi_1}{2}igg)+igg(1- ext{sin}^2igg(rac{arphi_2-arphi_1}{2}igg)- ext{sin}^2igg(rac{arphi_2+arphi_1}{2}igg)igg)\cdot ext{sin}^2igg(rac{\lambda_2-\lambda_1}{2}igg)igg) \ &=2rrcsinigg(\sqrt{ ext{sin}^2igg(rac{arphi_2-arphi_1}{2}igg)+ ext{cos}\,arphi_1\cdot ext{cos}\,arphi_2\cdot ext{sin}^2igg(rac{\lambda_2-\lambda_1}{2}igg)igg). \end{aligned}$$

Figure 3: Simplification of the Haversine Formula

Plugging in the coordinates in the format (latitude, longitude) for points 1 (12.0000000000, 34.0000114440), and 2 (12.0000000000, 34.000000000), along with the radius, r, of the earth

We obtain d = 1.24471... m, which means that for those given coordinates that represent a car and a bike pair, their relative distance to one another is approximately 1.24 m. This is what Project Sense accomplishes.

3. Design Verification

This section discusses the testing and verification of functionality of the project, elaborating on testing of specific blocks, and other specific predetermined requirements coined in the early stages of the project.

However, please note that this project does not particularly aim to collect data, but rather to simply notify the user about the relative distances. As such, the success of the project can simply be modeled in a binary manner; if the correct LED output is given, the system has succeeded, if the wrong output is displayed, then the system has failed.

3.1 Project Testing

The completed project was a partial success. A majority of the components worked perfectly all the time, while one component (the GPS module) partially failed.

It was rather simple to test the power subsystem. While the device was turned on, the battery was always outputting an average of around 4.7 V. Please note that this doesn't particularly need quantitative data, as whether or not the voltage was correct is a binary check. This can be clarified (as all components that operate at that voltage are working correctly) by Figure 4, or a simple oscilloscope reading.

The User subsystem was also relatively simple to check. As stated before, this was just a matter of whether or not the correct LED was displayed. This occurred 100% of the time whenever input was provided to it. Please note that this doesn't particularly need quantitative data, as whether or not the right LED glowed is a binary check. This is also easily depicted in Figure 4.



Figure 4:An instance where both systems are within 10 meters from one another; therefore, the red LED glows.

The control subsystem had several components to check.

The microcontroller was easily checked (as in Figure 4) by inputting location data and checking whether it performed the distance computation correctly.



Figure 5: The correct distance in meters was obtained (1.24 m) given sample coordinates, verified against online GPS software.

The transceiver was easily checked by verifying that any signal sent by one unit was received by the other and vice versa. If both systems had the correct LED glowing, then it would clearly mean that both transceivers worked correctly as they transmitted the location data. Considering this is the same evidence as what was needed for the user subsystem, figure 4 is sufficient to verify the working of the transceiver.

Due to the fact that the GPS module was quite inconsistent in update speed, the team set an arbitrary threshold at which the update speed is sufficient for usage on a highway. Although it can usually be much faster, the team chooses to set a cap at a frequency of 50 ms⁻¹, or 20 times a second. This was verified by only deeming a successful trial if there were at least 20 readings a second from both GPS systems displayed on a connected computer.

Finally, Table 1 shows the number of readings per second of GPS 1 and 2 respectively. A \checkmark indicates the condition passing (both exceed 20), and a \times indicates failure (one or both do not exceed 20).

| Tria | 1 | Tria | 2 | Trial | 3 | Tria | 4 | Tria | 5 | Tria | 6 | Trial | 7 | Tria | 8 | Tria | 9 | Tria | 10 |
|------|----|------|----|-------|----|------|----|------|----|------|----|-------|---|------|----|------|----|------|----|
| 29 | 15 | 8 | 13 | 16 | 41 | 27 | 32 | 43 | 15 | 23 | 35 | 14 | 4 | 3 | 19 | 23 | 34 | 19 | 18 |
| | × | > | < | > | < | | (| | < | | (| > | < | | < | N | (| | < |

Table 1: A model of the success rate of the GPS system's updates

Table 1 shows that both GPS systems work around 30% of the time. Please note that there are only 10 trials shown to maintain concision. The result obtained was similar when extended for 100 trials.

3.2 Requirement & Verification

Due to design changes that are fully elaborated on in section 2.1, all the requirements discussed below were not met.

The requirement for the Power subsystem (maintaining appropriate voltage for each component using the voltage regulator) was not met, as the components changed. They all were usable with the same voltage, and thus did not need a voltage regulator. Needless to say, the appropriate voltage was managed (4.7 V); it merely didn't correspond to the initially expected values.

In regard to the User subsystem, the original requirements mandated the buzzer to periodically notify drivers of a new perceived cyclist, and for the buzz to be sufficiently audible. Additionally, the display was also meant to update every 10 ms to guarantee the driver being knowledgeable of nearing cyclists. Neither of these were met because both the buzzer and the display were omitted from the final design for reasons discussed in depth in section 2.1. However, the LED system that replaced them in the Car's PCB managed to accomplish the purpose meant for the display and buzzer with a 100% success rate every time the GPS "succeeded" according to Table 1.

4. Costs & Schedule

The costs related to the project design include component and labor costs respectively. These are further detailed below in their respective subsections.

4.1 Parts

Please note that the Bulk Purchase Cost in Table 2 assumes purchase of required pieces of each part to create 50 car/bike sets, and that the Actual Cost is the Retail Cost multiplied by the Quantity (department paid retail cost for our shipping).

| Part | Manufacturer | Quantity | Retail Cost (\$) | Bulk Purchase Cost (\$) | Actual Cost (\$) |
|---|----------------------------|----------|------------------|-------------------------------|---------------------|
| RFM95W LORA RADIO TRANSCEIVER BR [1] | Adafruit Industries LLC | 2 | 19.95 | 1965 | 39.90 |
| GPS Module GPS NEO-6M [3] | MakerFocus | 2 | 9.99 | 949 | 19.98 |
| ATMEGA328P- PU [4] | Microchip Technology | 2 | 3.03 | 257.6 | 6.06 |
| LED CLEAR T-1 3/4 T/H [5] | Lite-On Inc. | 4 | 0.36 | 17.94 | 1.44 |
| Energizer Max AA Batteries - 6 Pack [6] | Energizer | 1 | 6.99 | 255 | 6.99 |
| Total | N/A | N/A | 40.32 | 3,444.54 | 74.37 |

Table 2: Parts Costs

4.2 Labor

Given that the ideal hourly salary is \$41/hour, and that the project took on average 55 hours (per partner), the labor cost for each partner for Project Sense is:

41 × 55 × 2.5 = \$5,637.5.

Accounting for the three partners required for this project, the total labor cost required is **\$16,912.5**.

4.3 Schedule

Please consult Appendix C to view the schedule.

5. Conclusion

5.1 Accomplishments

The overall design of the project was seen to successfully incorporate the separate components to give accurate results during trials. To start with, the microcontroller was correctly programmed to compute distances given sample input data to compute distance to high accuracy. Multiple trials yielded results of high precision, with values remaining extremely close to each other, and to the ideal value measured by existing software for distances. The transceiver pair also worked well in tandem with one another, where one system transmitted its signal to the other first. The other system decodes it, and then relays its own signal to the initial system, completing the communication phase. The decoded signal relays the position information to the user (through varied voltage level to the LEDs) after being converted into a data signal. The whole cycle periodically repeats, resulting in the bicycle and vehicle easily being able to update one another regarding their positions.

5.2 Uncertainties

Over the course of the project, there were sparse cases of the GPS systems falling to provide updated frequent results. Although the position's update speed for similar systems is on the order of milliseconds, sometimes even waiting minutes at a time would still be insufficient for the GPS to provide position readings. However, when readings were obtained, they were extremely accurate. As a result, the team tested whether several conditions such as current, voltage, and satellite exposure were in line with specifications required by the GPS's datasheet. Considering these variables were sufficiently monitored, the most likely conclusion that we can draw is that the component was simply faulty due to the fact that its results were not as regular as it would need to be for the purpose in question. As such, in certain conditions, the implementation of Project Sense may not be in perfect working order.

5.3 Ethical considerations

The project itself, being economically and harmlessly made, does not have particularly notable ethical concerns. Nevertheless, the remainder of this section is included to cover any minor cases for the sake of thoroughness.

A tenet of the IEEE code of ethics is "to protect the privacy of others". As such, a potential concern that arises from the project is the device being able to detect the location and a unique id for another driver/ biker. Naturally, we understand that users may express reservations regarding privacy, as the product works by allocating unique ids for all our users. Naturally, this could be used to uniquely identify consumers. We hope to assuage these concerns by reallocating unique ids for each consumer every 24 hours to ensure user anonymity over longer periods of time.

Another tenet of the IEEE code states "to hold paramount, the safety, health, and welfare of the public". One potential concern related to this is the detrimental effects of EM waves on humans. According to the Federal Communications Commission, the maximum RF limit for an individual from a cell phone or similar device is 1.6 W/kg. Considering such an amount is widely used safely, maintaining these values should be sufficient in assuring the user's safety, which is simply accomplished by using commonplace emitters.

In addition, considering the fact that the LED system for the bike and car are both serving to alert the driver/cyclist, we must disclose that users particularly susceptible to lighting be cautious in the use of the product, as it may be potentially jarring to them. Nevertheless, the LED brightness is quite standard for everyday use, and shouldn't cause problems for the vast majority of users. Naturally, the alerts were still easily noticeable in testing in different situations, meaning that users are unlikely to completely miss the LED lighting up.

5.4 Future work & Broader Impacts

A potential path an extension of Project Sense could take is the implementation of omnidirectional sensing of cars and bikes relative to one another. This could extend the application of the project to a greater scope. For example, it could now be used to detect parties even at blind corners or in high-congestion areas, where road traffic isn't always 2-dimensional. This could potentially be implemented with the usage of UWB (Ultra-wideband) technology which computes positions with even greater accuracy through angles of arrival.

Considering the project created for demonstration purposes was only feasible for a one car-one bike scenario, the next step is naturally to consider multiple cars and bikes, for which we must build functionality for tracking each unique device by identifying them using tags. In order to address concerns brought up in section 5.3, we must also code it such that these tags are erased periodically to protect user privacy.

A final salient extension that can be made to the project is to use superior GPS units. This is in order to rectify the slow-update issue, resulting in more accurate results over time.

Naturally, considering this is a concept that is capable of being mass produced and actually used on a much larger scale, it is of the utmost importance that we responsibly consider potential broader impacts that it could cause in multiple contexts. For instance, naturally, like any mass production, there will be an initial negative impact in an environmental context. This could be attributed to the raw materials needed to produce and distribute the good, along with other potential wastage. However, in the longer run, this could even have a positive impact on the environment; the increased safety that cyclists would enjoy as a result would undoubtedly encourage people to switch to cycling to a greater extent than their vehicle usage. This would reduce vehicle pollution that would otherwise have been emitted, as biking is undoubtedly a more sustainable form of transport. This would also have a positive impact on people's physical health due to the increased exercise, potentially augmenting happiness on a global or social context. Needless to say, it would even be more economically friendly for the commuters, likely making it a very helpful product!

References

 Adafruit (PID 3072 RFM95W LoRa Radio Transceiver Breakout - 868 or 915 MHz, web page. Available at:

https://a.co/d/6uo67tD. Accessed May 2023

[2] Haversine Formula – Calculate geographic distance on earth. Available at:

https://www.igismap.com/haversine-formula-calculate-geographic-distance-earth/. Accessed May 2023

[3] GPS Module GPS NEO-6M, web page. Available at:

https://www.amazon.com/dp/B07P8YMVNT?ref_=cm_sw_r_cp_ud_dp_0AA2ASF82XE2ERJEM1BX. Accessed May 2023.

[4] ATMEGA328P-PU, web page. Available at:

https://www.digikey.com/short/m0q28zbd. Accessed May 2023

[5] LTL-4224, web page. Available at:

https://www.digikey.com/en/products/detail/liteon/LTL-4224/217584. Accessed May 2023

[6] Energizer Max AA Batteries - 6 Pack. Available at:

https://www.amazon.com/dp/B000JJVB5K?ref_=cm_sw_r_cp_ud_dp_MAFF4ZS92S3CA60FA1NM. Accessed May 2023

Appendix A Requirement and Verification Table

| Requirement | Verification | Verification status (Y or N) |
|--|--|------------------------------------|
| While varying the input voltage level, we should maintain the aforementioned 5,4, and 2.5 volts. (Power Subsystem) | Power on the PCB in standard room temperature conditions Using a Multimeter or an Oscilloscope, measure the input pins of the voltage regulators to be 9V | Ν |
| | Measure the output pins of the voltage regulators to confirm their values at 4 and 5 volts | |
| | Now varying the input voltage between 5 and 13 volts, confirm the voltage regulator outputs remain at 4 and 5 volts | |
| Since Bikers may be using our device in subfreezing or extremely hot temperatures, our circuit must operate with consistent voltage levels between -10 to 40 degrees Celsius. (Power Subsystem) | Assuming the first requirement has been satisfied, repeat the first requirement in a variety of temperatures to confirm changes in resistance do not impact the expected voltage levels. | Y |
| Given any arbitrary set of control signals, the correct color LEDs should light up. | Power on the design and check that the necessary 4 V rail has the correct value using a multimeter or oscilloscope | Y |
| | 2. Assuming for the transmission gates a digital 1 value is 4 volts and a digital 0 value is 0 volts, try all 8 possible digital combinations to confirm the correct LEDs light up correctly. | |

Table 2: System Requirements and Verifications

| The buzzer must produce a beeping sound. (User subsystem) | Power the board and check the 4 V rail with a multimeter or oscilloscope to confirm. Using the oscilloscope, confirm if a squarewave is being produced from the oscillator. Using a signal generator, test the transmission gate with both a high and low digital value to confirm the squarewave is correctly being passed to the buzzer. Audibly verify the sound of the buzzer. | Ν |
|---|---|---|
| The buzzer must be audible with moderate background noise from driving in a vehicle (User subsystem) | Power the board and verify the sound of the buzzer is audible in a silent room. Play audio at around 80-90 dB while turning on the buzzer to confirm the buzzer is audible. | Ν |
| The LCD Screen must update every 10 ms to display the bike system location to the car system. (User subsystem) | Load the code into the microcontroller to display information on LCD screen Power board and verify that receiver is picking up signals from the bike system's transmitter Verify data and power signals are being produced using an oscilloscope. Visually verify if correct output is being seen on LCD screen | Ν |

Appendix B High-Level Requirements

• The vehicle and bicycle should be able to detect and inform the driver/cyclist about their approaching counterpart starting from the time where the distance between them is at least 10 meters. This is to give them sufficient warning to respond to the situation.

• The vehicle and bicycle should be able to detect movement of their counterparts that are of speeds of 10 miles/hour at the minimum. This is simply a metric to start with. If the device can detect approaching vehicles at higher speeds, it would be ideal.

• The system should update both the bicycle and vehicle regarding their counterpart's position once every 10 ms (milliseconds). This is to ensure that both the cyclist and driver always have updated information, even in a fluid situation.

| Week | Task | Member(s) |
|-----------------|---|----------------------|
| Feb 27 - Mar 6 | Order listed parts on PCBway | All |
| | Begin work on Power subsystem for bike | Abhay |
| | Research systems that achieve efficient depiction and monitoring of moving bodies, bikes in this case (for the car) | Jerome |
| | Begin work on Power subsystem for Car | Aakash |
| Mar 6 - Mar 13 | Establish and power connection for bike receiver and controller | Abhay |
| | Establish and power connection for car receiver and controller | Jerome |
| | Revise PCB design to improve/facilitate device function. | Aakash |
| Mar 13 - Mar 20 | Work on appropriate communication between both transceivers. | Jerome and Aakash |
| | Begin plotting LED output from logic/conditional input | Abhay |
| Mar 20 - Mar 27 | Place orders on PCBway for second round for additional needs | All |
| | Complete signals and computation units of both bike and car | All |
| | Test with diagnostic LEDs if cars and bikes are appropriately detected | Aakash |
| Mar 27 - Apr 3 | Perform connections between bike's signals & computation unit and LED system to indicate distance | Jerome |

Appendix C Schedule

| | Wire GPS module's communication with transceiver units. | Aakash |
|-----------------|--|---------------------|
| | Perform connections between car's signals & computation unit and speaker to indicate a detected bike in range | Abhay |
| Apr 3 - Apr 10 | Perform test runs of both sections together. Check that results of both car and bike are identical at varying cases. | Abhay and Aakash |
| | Revise PCB design to account for unexpected behavior. | Jerome |
| Apr 10 - Apr 17 | Fine-tune devices in different environments to check GPS consistency. | All |
| | Deliver mock demo to TA. Demonstrate progress | Everyone |
| Apr 17 - 24 | Make final adjustments and incorporate TA feedback in time for the final demo | Everyone |
| Apr 24 - May 1 | Deliver final demo to instructor and TAs for functionality | All |
| May 1 - May 4 | Deliver the final presentation and complete other end-of-project assignments and formalities | All |