Team Pishro-Nik and Ni
Project Crossroads: Vehicular Ad-Hoc Network for Collision Warning

Sean Morrell, Daniel Marcq, Jarrod LaBarge, and Anthony Swochak

i. List of Figures

ii. List Abbreviations

DOT – Department of Transportation
GUI – Graphical User Interface
MCU – Microcontroller Unit
OBE – Onboard Equipment
PCB – Printed Circuit Board
RSE – Roadside Equipment
SPI – Serial Peripheral Interface
USART – Universal Synchronous-Asynchronous Receiver/Transmitter
VANET – Vehicular Ad-Hoc Network
WRM – Wave Radio Module

iii. Abstract

When vehicles approach an intersection there is the potential risk of colliding with another vehicle. The cause for these collisions are the result of careless action by the driver and obstructions that prevent the drivers from making any preliminary assessments of a potential accident. Vehicular Ad-Hoc Networks provides a means of providing drivers with more information to allow them to assess and react to dangerous situations on the road.

1. Introduction

Until recently, safety features in vehicles have taken a passive form (e.g. seat belts, air bags, and shatter proof glass). Passive safety devices aren't meant to prevent accidents, instead, they are meant to protect the vehicle occupants in the case an accident occurs. We are now beginning to see a shift towards safety devices that take more of an active role in preventing accidents from occurring all together. This technology push is being fueled by the Department of Transportation (DOT), the automotive industry, and a multitude of academic institutions. Vehicle-to-Vehicle (V2V) communication is the leading technology that will enable the development of active safety features. What V2V provides is an ad hoc wireless network by which vehicles are able to send and receive information to and from each other The goal of our project is to implement a Vehicular Ad Hoc Network (VANet) for a proof of concept for an intersection vehicle collision warning system.

Motivation:

To illustrate the projects intentions, consider the following problem situation. There is a roadway intersection where there are two vehicles approaching. Vehicle A has just gotten a green light, and vehicle B (who has just gotten a red light) is approaching the intersection. Vehicle B runs the red light, and collides with vehicle A.

Situations like these are a common occurrence in the United States. Forty five percent of all vehicle collisions occur at intersections. Twenty one percent of these collisions are fatal - amounting to more then nine thousand deaths per year[1]. The result is lost time, money, and, more importantly, lives. Intersection collision warning is considered a high priority by the DOT, but, currently, no adequate solution exists. The project aims to warn of these types of collisions. In addition, the design will be adaptable for other situations (e.g. rear end collision avoidance, blind spot warning, adaptive cruise control).

A major factor contributing to these accidents are the obstructions surrounding the intersection; thus, resulting in a lack of visual warning. Critical decision making by drivers is dependent on extra information about the situation. As opposed to passive solutions, (e.g. air bags and crumple zones) which lower the chance of deaths at intersections, the approach to take an active role in collision warning is noble. V2V communication is the path to providing this information. Using a precise sensory network, coupled with an effective wireless communication system, we aim to provide a proof of concept of vehicle-to-vehicle communication, and it ability to be effective on the road.

In summary, consider an example where all vehicles could intermittently broadcast their current position and speed to surrounding vehicles. If the data suggested that there could be a collision, the drivers would be warned.

2. Technology Theory

GPS

GPS Statistical Analysis and Verification of Accuracy

Please refer to Appendix B for detailed information on GPS analysis.

Background

GPS or Global Positioning System was developed solely as a navigational tool for the military. Over the years its has spread into the commercial realm as a cheap and convenient method for finding where you are on the planet. Utilizing 24 orbiting satellites at a distance of approximately 18000 km above the earths surface, a user can determine latitude, longitude, and altitude positioning anywhere on the planet just as long as at least 3 of the satellites are in view of the targets position. GPS has become such a powerful tool for the general public in terms of its ability to determine location, speed, and heading.

How Does GPS Work?

[2]The way GPS determines a target location on the planets surface involves a means of trilateration (Figure 1). In order to trilaterate a targets location, the GPS receiver determines the distance to each satellite in the area. Using the overall total time for the signal to reach the GPS receiver, and the phase velocity of the signal through the air (e.g. 3E8 m/s), the distances between the satellites and receiver can be calculated by multiplying the total time and phase velocity. Once those distances are established and the location of each of the satellites is known, the targets location can be determined. As the number of satellites in communication with the GPS increases, the accuracy of the measurement increases. Fortunately, the satellite orbits are situated such that there are always at least 6 or 7 satellites in view by the GPS receiver at any moment in time.

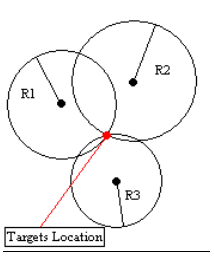


Fig: 1

Accuracy/Errors

The synchronization of the clocks on the GPS receiver and the satellites is essential in obtaining accurate distance measurements between the receiver and satellites. The clocks have to be accurate to within nanoseconds of each other; otherwise, there could be significant delay from the atomic clock on the satellite. Because atomic clocks are very expensive, each GPS receiver is fitted with a quartz clock. By periodically updating the quartz clock with the atomic clock, the GPS receiver is ensured to be synchronized with the orbiting satellites.

Some factors that can affect the accuracy of the GPS unit include multipath effects where the signal transmitted down from the satellite reflect off of terrain. As a result, the transmitted signal could potentially be delayed from the synchronized clock on the receiver; thus, producing inaccurate position readings.

Atmospheric affects can also affect the accuracy in GPS readings. Because this radio signal has to pass through the atmosphere, varying conditions in the atmosphere, such as humidity, can ultimately affect the phase velocity of the transmitted signal. This is key in the synchronization of both clocks in the system; otherwise, delays in timing can result in inaccurate distances between the GPS receiver and satellite. Overall these types of inaccuracies prevent GPS from being as accurate as everyone desires. Other contributing factors to the GPS inaccuracies such as Selective Availability along with multipath and atmospheric effects are covered in more detail in.

Differential GPS

In spite of the apparent misgivings of GPS, in terms of accuracy, there are means by which a fine tuning process can be implemented to give more accurate locations. Differential GPS or DGPS is a system of ground-based stations that are primarily used to correct for the inaccuracies present in GPS. Basically, this system works by taking a already known timing scheme between a satellite in the area and a receiver on the ground. Because the ground station has a fixed location, it can determine and fine tune the resulting time of signal acquisition from the satellite. By comparing this timing scheme with the users GPS device, timing errors can be nullified to a certain extent; thus, resulting in location accuracy to within a meter or half a meter. A more detailed explanation of this type of error correction is found in.

Dedicated Short Range Communication

DENSO is a leading supplier of advanced automotive technology, systems, and components for all the world's major automakers [3]. With the push from the DOT for integrated vehicle-to-vehicle communication, and as the IEEE works to standardize 802.11p (WAVE), DENSO Wireless has developed wireless modules for research and testing. These wireless modules utilize the casing for a wireless router, but the technology inside has been modified to fit the current parameters of 802.11p[4]. The technical specifications for DENSO can be found in Appendix II (WRM).

Through contacts within the Transportation Department at UMass, DENSO has agreed to loan a number of these DSRC transceivers to the department and this project. The reason that these devices look similar to wireless routers is that DSRC is not a complete standard yet, and therefore smaller IC transceivers do not currently exist.

Interface:

The DSRC module utilizes Ethernet for communication to the host device, in this case the OBE. This is very different from most of the other components used, since they are specific to embedded design, and utilize standards such as USART and SPI for communication protocols. This created the need for an Ethernet controller for communication with the DSRC. Details can be seen in section Ethernet.

Theory of Operation:

As stated above, the DSRC accepts packets in IP format. All DSRC configuration information is encapsulated within the option header of the packet, and the payload transmitted via wireless is encapsulated in the payload of the IP packet. Because the DSRC is structured in this manor, packet-by-packet configuration can occur. For a future upgrade, for instance, the OBE could be allowed to transmit on a different frequency from the RSE, seamlessly, without requiring reconfiguration of the entire unit. Communication from the RSE to the DSRC and communication from the OBE to the DSRC occur in the same fashion.

3. Design

In order to address this problem, the system designed for each vehicle would be able to determine its current location, and transmit out warning messages when needed. Our design utilizes a GPS unit, microcontroller, and transceiver; collectively, theses components are called the Onboard Equipment (OBE). In addition, each intersection must be able to transmit current traffic light conditions for each intersecting road; this unit is the Roadside Equipment (RSE), consisting of a laptop driving a transceiver.

RSE: Road Side Equipment

Objective:

This scenario requires that the environmental variables be portrayed to on coming vehicles. In order to accomplish this task, one module, acting as the RSE, periodically transmits data on the same frequency that V2V communication takes place. The environmental variables consist of coordinates at the center point of each road that intersects at the aforementioned intersection, the current light condition (e.g. red, yellow, green) that corresponds to each given intersecting road, and finally the time interval remaining for that condition. The following diagram illustrates the design of the RSE:

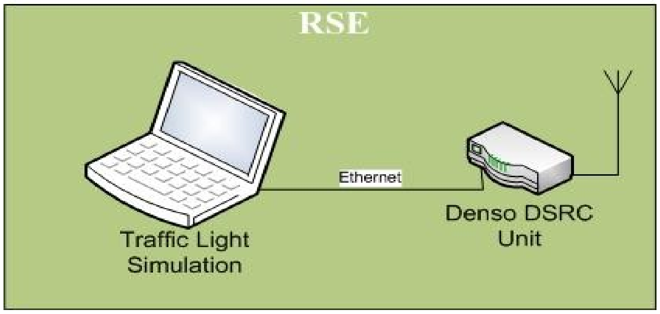


Fig 2:

Implementation:

Integration into the pre-existing DOT infrastructure would be quite simple. Intersections are currently controlled via a microcontroller with a fairly complex algorithm for controlling a dynamic flow of traffic. Many of these intersections are also linked back to a central office for monitoring; this allows for modifications throughout the life cycle of the equipment. The addition and maintenance of communication equipment to this infrastructure would require very little investment or knowledge increase. This scenario, however, is a proof of concept which utilizes transceivers that are not currently set as the industry standard. Therefore, a full simulation for the intersection environment needed to be integrated into the system along with the communication equipment.

Interface:

Since this RSE is for a proof of concept system, it needs to be dynamic in order to allow for multiple test scenarios to be organized. The general section on the interface, allows for general parameters to be set, such as the coordinates for each stop line and the time interval for each light. There are two different operating modes for the RSE, Auto or Manual:

Auto – In automatic mode, the RSE will cycle each stop light at a regular interval, from Red, Yellow, Green and back to Red. While in automatic mode the stop line interval is setup so that perpendicular roads will have opposite light conditions at all times, and a 2 second full red condition will take place prior to an occurrence of a green condition. The RSE message is transmitted at the user defined interval.

Manual – In manual mode each of the stop lights can be set to a specific color. Once the scenario is started, the RSE will transmit at the user defined interval using the preset light conditions. The value transmitted indicating the interval until light condition change in manual mode is set to 99, thereby avoiding any invalid conditions on the OBU.

Stop Line Coordinates:

In order to determine stop line coordinates it is recommended that the area be surveyed by a professional to ensure that the correct coordinates are obtained. For this proof of concept, however, Google Earth was utilized to obtain the coordinates.

Packet Structure:

The packet structure, as discussed in the section covering WRM, resembles an Ethernet packet. Reading a payload on a computer is fairly easy, because of the computing power inherent in the PC. However, since the OBE are microcontrollers with a fairly limited amount of computing resources, in addition to the system being safety critical with severe timing constraints, the payload length and structure becomes an important issue. As much as possible, it is important to keep the majority of computing at the RSE, and only transmit information in a form that is easily usable to the OBE. One important consideration is the size of the information contained in the stop line coordinates. For most intersections there are four intersecting roads. Each one of those roads will have a stop line associated with it as well as the coordinates for that stop line. The coordinates have two components, latitude and longitude; thus, each component ranges from 0 to 90/180, respectively, with a precision of six decimal places when in the form DD.DDDDDD (units of degrees). In addition, each stop line has a two sectors associated with it (e.g. N or S, and E or W). The naive approach to structuring the payload would be to simply line up the data in the following form for each intersecting road:

Latitude Component	Latitude Sector	Longitude Component	Latitude Sector

This would make the total amount of data for each intersection add up very quickly. Since each OBE must, first, determine which light condition corresponds to its respective road, there would also be a significant amount of processing that would be required at the OBE. The following are two alternative approaches to structuring the payload, decreasing data size, and lessening process time requirements:

Common Component:

Even in the form DD.DDDDDD, it is easy to see that there are a number of common digits between the coordinates for each stop line. The average intersection for a two lane road without turning lanes will only vary by approximately .000009 degrees from center North/South and .000017 degrees from center East/West. This implies that the payload could be composed of a common coordinate with a variation for each of the stop lines. The following shows this technique:

Common Latitude	Latitude Sector	Common Longitude	Longitude Sector	Lt 1 Latitude Variation	Lt1 Longitude Variation

The common coordinates would be the first portions of the total, and the variations would be a predetermined number of digits at the end of the coordinate. This case doesn't work when the intersection straddles a change in one of the common portions of the intersections coordinates.

Center Point with Variation:

This method is very similar to the previous example; thus, does not require much explanation. When the coordinate for the center of the intersection is determined, the variation latitude and longitude for each stop line is determined and attached in the same method as above. This method ensures that any location for the intersection would work; however, more processing is required on the OBE for this method.

For RSE GUI image see Appendix II.

OBE: Onboard Equipment

The OBE serves as a CPU for each vehicle. Its purpose is to collect information, and process information from the vehicles sensory network, from other vehicles, and from any RSE in the area. The sensory network provides information about the vehicle's current operating behavior (i.e. location, speed, and direction of travel) through components such as GPS. The OBE uses an algorithm to calculate the probability of an accident occurring (e.g. running a red light), then reacts in two ways. First, it warns the driver, and second, it broadcasts a warning to all surrounding vehicles, alerting everyone of the situation. Figure 3 is a block diagram of the components included in the OBE:

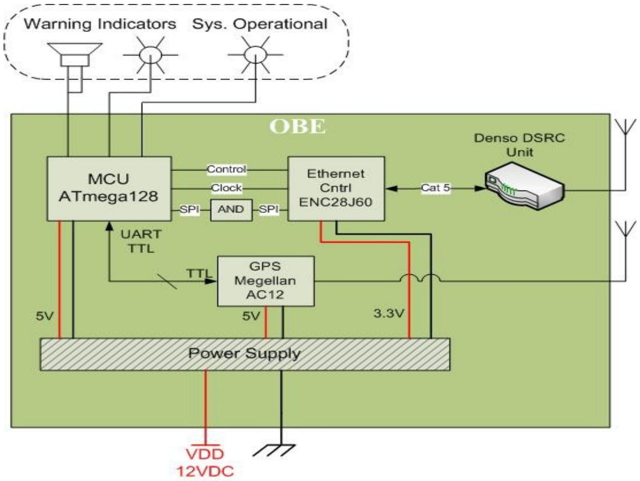


Fig. 3

The operation of the OBE is much more complex than the RSE because it needs to interface with the GPS and any external sensors. In addition, it will have a more complex algorithm to determine if a collision is imminent.

The main loop of this program has a series of polling mechanisms which glean data from subsystems such as the GPS unit. After this is completed, the information obtained is analysed, invoking a sub-process, to determine if any action is required. In addition to the main loop, when a message is received from an outside source, either an RSE or another vehicle, an interrupt is thrown, and the receiver interrupt process is invoked. After the message is received, its contents are analysed, and an algorithm determines if action is required. Upon completion of the interrupt process the main loop continues. This is contrary to normal interrupt operations; however, in our case, even within the relatively small amount of time it will take to process an incoming message, the vehicle location will have changed.

Our end deliverable goal for this project is to focus on the intersection situation. We will set up a hypothetical intersection, and have two cars approach this intersection in such a way that one car will run the red light. In this case, the car's OBE will realize that it is about to run a red light, it will send a warning to its own driver, and then send a similar warning message to the other driver. This message should be received so that both drivers have enough time to react to the warning, and safely slow down their vehicles.

Implementation:

The transceiver for the OBE will also be the DSRC link used in the RSE. Since the units are in router form they will need an Ethernet controller to interface with the microcontroller. The ENC28J60 is a SPI operated Ethernet controller that was selected to complete this task. Since it is a traditional SPI interface, it will meet all standard SPI protocols on standard microcontrollers.

Once using the Ethernet controller, the packets for transmitting must be built and sent out by the microcontroller. The packet size and structure of the OBE is different from the packet size and structure of the RSE. In the case of the OBE all packets transmitted are structured to be exactly 64 bytes long. All packets transmitted will be in the form of IPv4. The message structure will consist of two distinctly different possible packets. These packets will be an alert-all message and a clear-previous-alert message. Two messages are all that is needed, since the OBE will primarily operate as a constant receiver, and only transmit when an accident is probable.

While driving the DSRC unit, the OBE will also be utilizing a GPS unit to orient itself in real time. The OBE will constantly be requesting information from its GPS. Once the GPS has replied back to the microcontroller, the program will then stream in all current location points for the vehicle.

GPS and MCU Interface Procedures:

The GPS is a key component in this project, because it calculates position and horizontal velocities. Because we desired to integrate the GPS with a MCU, we needed to determine how we could translate the messages familiar to the GPS into messages that the MCU could produce. This became a matter of understanding how the messages from the MCU to the GPS had to be structured. We conducted tests using Windows Hyper terminal, and were able to established a similar communication link as our Ashtech GPS Software counterpart. In order to facilitate this type of communication, we had to setup circuitry that could support the conversions of RS232-to-TTL and TTL-to-RS232. The following figure shows the circuit model we used to conduct this communication experiment:

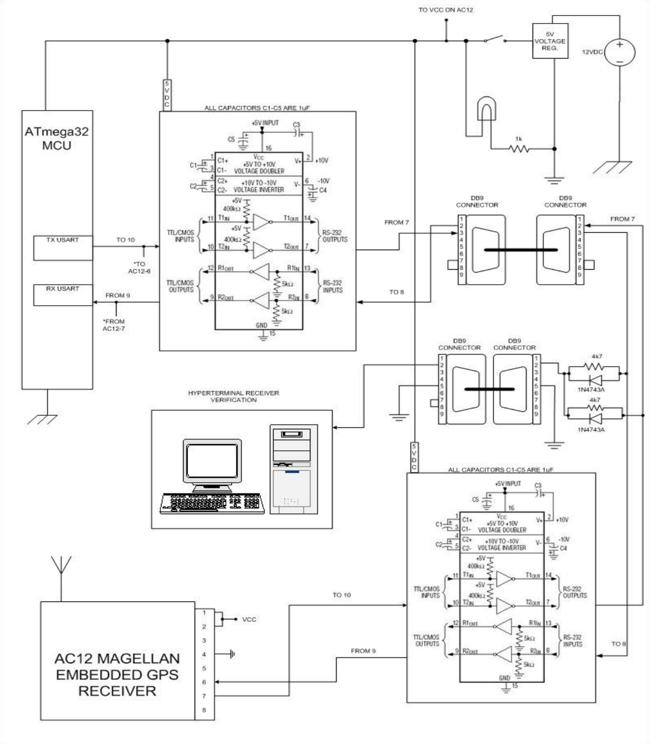


Fig: 4

Once we created this circuitry, we quickly realized that each of the commands to the GPS could be sent as a series of 8-bit ASCII characters. The following command shows the typical structure of the commands sent to the AC12 GPS Unit:

$PASHS, PRT, A <CR> <LF>

The details of this command are not necessary for understanding our methods behind establishing communication between the GPS and MCU. The <CR> (Carriage Return, 0x0D) and <LF> (Line Feed, 0x0A) are used to complete the process of serial communication between the GPS and MCU. When constructing the message structure in code, we had to take into account the 8-bit resolution of our MCU. As stated before, we can divide the commands sent to and received from the GPS as a sequence of ASCII characters. The 8-bit USART on the MCU allowed us to take advantage of this characteristic in the message. You are probably wondering why there are two redundant instances of a MAX232 IC; this is because we initially used an ATmega32 Development board, which featured a built-in RS232 conversion stage.

In order to validate the commands sent to and received from the GPS, we setup a spy circuit, which consisted of two pairs of 1N4743A diodes and 4.7kΩ resistors connected in shunt. This circuitry provided the capability of reading both the receive and transmit lines of the GPS to Windows Hyperterminal for verification. Because the MCU and GPS both support TTL interfaces, they are connected directly. For our implementation, we have the MCU and GPS connected directly, and a separate UART dedicate for debugging using Window Hyperterminal.

Main Algorithm:

The main algorithm will start by first initializing the OBE system. Following initialization, the OBE simply runs a continuous loop shown in Figure 5. What this loop does is the following:

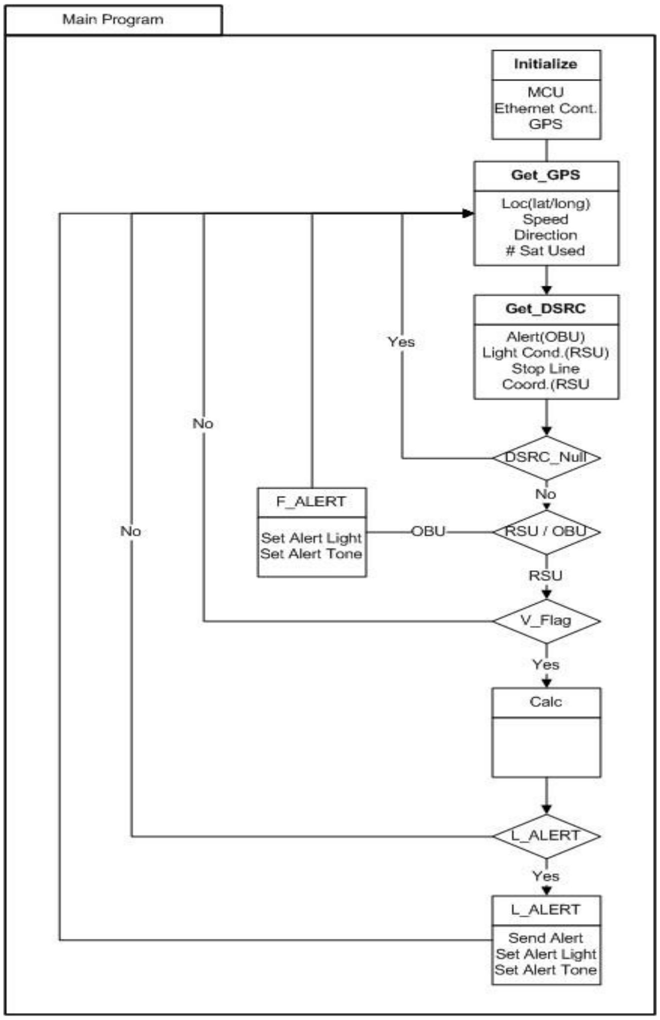


Fig: 5

First, the system continuously gets new data from the GPS and DSRC. Getting information about the vehicle's position from the GPS will never result in a warning, so the only thing that should cause the program flow to leave this area is getting information from the DSRC. Once data is on the DSRC, the data flow will leave this portion of the code.

There are only two different types of data that can be received by the DSRC. One is a warning message from another OBE, and another is information about a nearby intersection sent from RSE. The following diagram illustrates this process:

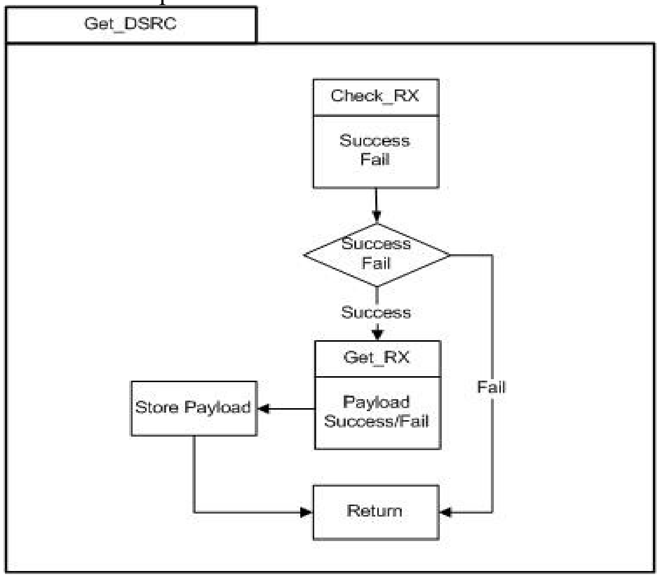


Fig: 6

If the data is a warning message, then the driver of the vehicle should be warned refrain from entering the intersection. Once this happens, the code will return to the top of the main loop, and continue to update the data from the GPS and check for information from the DSRC. The following diagram provides a visual illustration of the Get_GPS function:

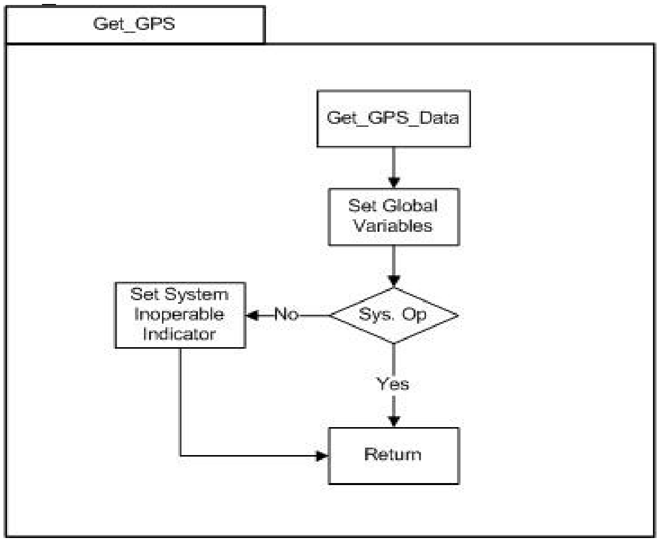


Fig: 7

If the data received contained information regarding a nearby intersection, and as long as the GPS data is currently valid, the data flow will enter a calculate function that will determine if the driver of the vehicle will run a red light.

The first thing that each vehicle needs to determine is relative light status and stop line data sent out from the RSE. The current implementation for this determination is simply finding out which stop line is closest to the vehicles current location. Having determined this, the light status associated with the intersection is now known. From here, the code is divided up into three sections: green light, yellow light, and red light.

This can be seen in the calculate diagram shown in the following figure:

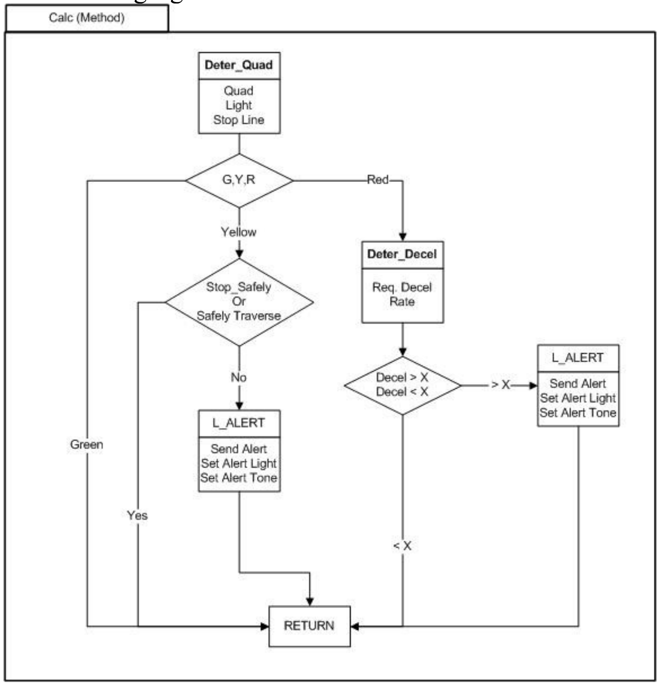


Fig: 8

If the light is green, then no calculations need to be made. At each intersection the duration of a yellow light is sufficiently long. If the green stage suddenly changes to yellow, the driver will either make it through the intersection easily, or be able to stop on time. It is, therefore, impossible to generate an alert if the light is green.

If the light is red, then only one calculation needs to be made - “Can the driver stop in time before reaching the stop line?” This, of course, is going to be determined by three pieces of information: The distance between the stop line and the vehicle, the speed of the vehicle, and the maximum deceleration force of the vehicle. There are various error constraints associated with these three piece of information; they will be discussed in a later section. The distance between the light and vehicle is known due to information from the RSE and the GPS. The speed of the vehicle is also known from the GPS. Unfortunately the maximum deceleration force of the vehicle is variable. It has to do with the tires on the vehicle, the weight of the vehicle, the type of pavement, the weather conditions (e.g. rainy and slippery, snowy or icy, etc.) This information is not known. In this case a reasonable assumption must be made about the vehicle’s ability to slow down. The default value for this is to assume that no car can decelerate any greater then 3 m/s^2, or roughly 1/3 of gravity. According to the DOT, the average vehicle can decelerate at approximately 4 m/s^2. In practical use, this number would be predetermined for each vehicle separately, and in future use, this number could even be updated based on the weather report for road conditions.

Using this information, the microcontroller can determine if the driver will be able to stop before the stop line. If the driver can stop, then no alert is generated on this run-through of the code. If not, then an alert will be sent out to all nearby vehicles, and the loop will reset.

If the light is yellow, then two different calculations need to be made. Can the driver stop in time before reaching the line? Can the driver safely traverse the intersection? If the answer to both of these questions is ‘no,’ then an alert will be sent out to all nearby vehicles, and the loop will reset.

To determine if the vehicle can stop, the microcontroller utilizes the same portion of code discussed above in the red light scenario.

To determine if the vehicle can safely traverse the intersection, the following three pieces of information are needed: The vehicles speed, the distance between the vehicle and the stop line, and the duration of the yellow light. Using this information, the microcontroller can determine if the driver can safely traverse the intersection.

There are various timing constraints within this system, which we have minimized in our design. In order to keep GPS data up-to-date, upon arrival of information on the DSRC, the GPS is queried on a regular basis; this prevents wasted time communicating with the GPS during the calculation portion of the code. In order to alert people in the shortest amount of time, in any situation, upon notification of an alert, the system reaches a portion of code to alert the drivers instantaneously.

There are also sources of error which could cause the system to not function properly. The only problem that could happen due to error is lag time during alert determination. As a result, the system would not be able to warn of a collision in time to avoid it. This issue has been addressed in the following ways.

First, a functional indicator light has been included which simply states that if the error in GPS data is too large, then the system is currently not functioning. As a result, the OBE will only be able to warn of other drivers’ warning transmissions, and not produce any valid warnings of its own.

Second, the maximum deceleration speed for the vehicle used as a default is much slower then the actual maximum deceleration speed for the average vehicle. This gives the driver time to realize what has happened, and hit the breaks. This technique also accounts for minimal error from the GPS.

Design Alternatives

During the initial stages of the research and design, we determined that GPS was probably the best sensor for the scenario. Attempts at using other forms of technology to track vehicle locations relative to other objects has been researched by other groups; however, these methods aren't adaptable into other situations.

The discussion of design alternatives very closely mimics the discussion of the modular design of the project. There are many applications of VANETs, and after looking at all of the options, the decision to focus on the intersection was validated.

One such alternative would be to use forward looking proximity sensors or radars to detect front end collisions. This would, of course, work to detect collisions with vehicles not connected to the ad-hoc network. For instance, this could, potentially, be used to alert a driver of tail-gating car to back off, and with a system like that in-place the tail-gating driver could be alerted of a possible rear end collision. The decision to move away from this was made with respect to the overall success of the project, and budget constraints.

Another design alternative that we considered was to focus on a blind spot warning, instead of an intersection collision warning system. If a vehicle is in your blind spot, a light would simply light up on the dashboard indicating that a vehicle was in your blind spot. In the end, we voted against this situation because, although it would use VANETs such that the leading car would receive the location of the car in the blind spot, there was really no packet sent over the air to warn drivers of a collision; thus, the intersection situation seemed like a much more complete example of the capabilities of VANETs.

For a full circuit layout, PCB design layout, and final design of the OBE refer to Appendix II.

4. Considerations

RSE:

Originally the design plan for the RSE was that of an embedded system, just as the OBE. However an embedded RSE was not useful in terms of evaluating, debugging, and testing system performance. Using a laptop to simulate the RSE did not change the overall functionality system nor did it effect the design of the OBE. Using a laptop for the RSE also allowed us to effectively debug the OBE more easily. Large scale implementation of this system would require the cooperation of the DOT and public works departments across the country to integrate additional equipment into intersection equipment. It is important, however, to realize that most modern intersections outfitted with stop lights already posses the equipment necessary to serve as the RSE. The only addition to this equipment would be a suitable transceiver and additional programming to control the transceiver.

OBE:

Each OBE prototype was contained within a “black box” enclosure where the key device components talk to each other; these components include, the embedded GPS unit, the microcontroller unit, and the Ethernet controller.
The microcontroller for the OBE helped bring all of these subsystems together into one cohesive system. The device drivers we programmed for both the GPS units and Ethernet controllers (DSRC communication) make it possible to gather the information from these devices and thus enable the full capabilities of our collision warning algorithms.

Similar to the RSE, if this system were to be implemented on a large scale, it would require the cooperation of many organizations and vehicle manufacturers. The OBE would be fully integrated into the pre-existing computer readily available in most modern-day vehicles. With the addition of a transceiver the system would be complete. Currently the transceiver used for this project is an up and coming standard for Vehicle to Vehicle communication.

Philosophical Issues:

The purpose that our project serves is to prevent accidents from occurring at a traffic light intersection. Because of the aforementioned communication link present within the system, many potential users of the system feel uncomfortable about information being transmitted and the possibility of the information being used against them. For instance, logging and reporting traffic violation to the local authorities. An additional concern is that the system would provide a false sense of safety and result in people using this system to run red lights more frequently. If the system works then surrounding cars will be warned before they enter the intersection.

Another issue that is worthy of discussion is the potential for hackers to jam transmissions preventing warnings and defeating the system. This could be used as a tool to cause more accidents. More research and development needs to take place that addresses these issues. Obviously, for reasons involving time and money, we were unable to address these problems in this implementation, but we do realize the problems that these security issues pose and that they should be addressed in the real-life application of this system in order to ensure the safety is the top priority.

5. Testing and Results

The test location has been selected to be the small intersection in front of the UMass football stadium. Stop line coordinates for each intersecting road were determined using Google Earth; however, at this particular location we were only concerned with three of the intersecting roads. Since the RSE interface was designed to be dynamic, the coordinates for each intersection could be easily input to the system. Figure 9 shows the survey of the test site.

Fig: 9

Testing scenarios consisted of a car running a read light at various speeds, a car traversing the intersection through a yellow light while varying the amount of time remaining until next light change, and a car traversing the intersection through a green light. We also made sure that if the vehicle stopped in time at a red light that the alarm did not sound. Our result showed that the alarm would sound in adequate time to warn drivers in the surrounding area of a possible collision.

For additional information about testing please refer to videos included with project documentation.

6. Further Work

There are many improvements that have been considered for this system but have yet to be implemented. While we realize that GPS has its advantage in terms of navigation, it also has its disadvantages in a safety critical system. The major problem with GPS in this system, other than accuracy, is latency. GPS only updates at a rate of 1Hz; a rate that is quite slow when complex calculations of deceleration and current location are taking place in order to save lives. Additional prediction can, however, be helpful to alleviate this problem. If the last known speed is assumed to be constant over the time between GPS updates, the OBE can effectively predict current location using the last known location. In addition if the OBE were tied directly into the vehicles Onboard Diagnostics (OBD-II)[5], speed would no longer need to be assumed thereby increasing accuracy again. OBD-II capability would also allow for increased accuracy in the calculation of deceleration time required by providing information such as brake pad condition, vehicle mass, tire pressure, tire wear, and possibly even driver habits.

While this project was considered a success, wide scale implementation seems to be a notion that will take years of work. However it is important to realize the benefits of this project as a bridge for further work in V2V and V2R communication. The importance of this project isn't that it is a means to an end, but that it provides proof that V2V can be used as a viable solution[6].

7. Conclusion

To sum it up, our implementation of a VANet is being used to actively prevent collisions, rather then passively protect the vehicle occupants in the event of a collision. This was the selected situation for many reasons. One major reason is that collisions at intersections are a major cause of death in the United States, amounting to over 9,000 deaths each year.

The deign includes two major pieces of equipment: the OBE and the RSE. The RSE is stationed at each intersection, intermittently broadcasting the position of the stop lines, and the current status of the lights to each of the vehicles in the surrounding area. The OBE is in each vehicle, which receives the information from both the RSE and other OBE, as well as an internal GPS. If the OBE determines that it is going to run a red light, it sends a warning to all nearby vehicles to not enter the intersection.

Currently the system is fully integrated and has worked correctly during in-lab testing. Once the OBE is integrated onto its PCB the system will then be field tested. During these field test cameras will be used to video record our results. Also reliability and success charts will be developed as a way of reflecting the overall efficiency of the system as a whole.

8. References

[1] http://www.personal-injury.com/practice_areas/Auto_Accident_Statistics.asp
[2] http://en.wikipedia.org/wiki/Global_Positioning_System
[3] http://www.densocorp-na.com/corporate/philosophy.html
[4]http://www.leearmstrong.com/DSRC/DSRCHomeset.htm
[5] http://en.wikipedia.org/wiki/On_Board_Diagnostics
[6] http://www.ece.osu.edu/citr/Demo97/osu-av.html
[7] http://en.wikipedia.org/wiki/Chi-square
[8] http://en.wikipedia.org/wiki/Rayleigh_distribution
[9] http://ieeexplore.ieee.org/iel5/7690/21011/00974445.pdf
[10]http://www.cs.ucdavis.edu/~ghosal/Research%20Talks/AHMCT-Talk.ppt
[11]http://homepage.fudan.edu.cn/~yubohome/References/VANET/VANETResource.htm
[12]http://www.tech.plym.ac.uk/see/research/cdma/Papers/Paper%20for%20UBIROADS%202007.pdf
[13] http://www.trimble.com/gps/index.shtml
[14]http://www.winlab.rutgers.edu/pub/IAB/fall_2006/sangho-v2vcom3.pdf

9. Appendix 1

A. Application of mathematics, science and engineering

This section provides material that demonstrates how we applied our knowledge of mathematics, science and engineering. Listed below are three courses that have directly contributed to this project and a brief description of how we used material from each. One detailed example is also included.

ECE 314 – Intro, Probability & Random Processes
We used the knowledge gained from ECE314 in order to gain a better understand of the accuracy of GPS. This involved developing probabilistic models using Matlab. This process is described in more detail in Appendix 1B.

ECE 353 – Computer Systems Laboratory I
We are employing interfacing, programming techniques, and debugging skills learned from ECE 353 on the embedded microcontroller, OBE. We also used GoLogic to analyze the system on a high speed level. This way we could see what was being received and transmitted by our OBEs. Using this test equipment gave us the ability to determine that the OBE's receive and transmit functions worked properly.
ECE 584 – Microwave Engineering I
Prior to the availability of the Denso Wireless DSRC modules, we used 2.4GHz radios for which we had constant debugging issues. A tool we used in this debugging process was the Vector Network Analyzer. Using a spectrum analyzer assisted us in getting concrete evidence of our RSE transmitting within the desired band. We were able to obtain plots that showed the resonant peaks during transmission; this gave us evidence that the device had effectively been configured to act as a transmitter and was, in fact, transmitting data. The following figure is a plot of the spectral energy emitted from the 2.4GHz radio:

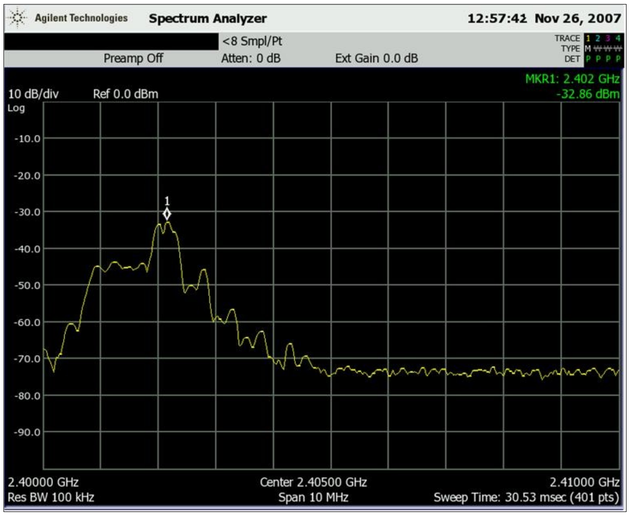


Fig: 10

To obtain these results, we used the max hold on the Spectrum Analyzer, which ensured that all of the transmission activity was falling within the channels bandwidth. The bandwidth of each channel is 8MHz, which is validated since the transmission power falls completely within 2.4GHz and 2.408GHz.

What we are seeing in this output is effectively channel one being active on the radio during transmission. When we ran this test the transmitter was set to have its output across channel one.

B. Design and performance of experiments, data analysis and interpretation

In order to build our algorithms correctly for our final design we needed to test the accuracy of our GPS unit. Utilizing Ashtech Evaluation Software we were able to glean the required data for analysis. This software makes a connection to the GPS receiver and polls for position updates at a rate of 1Hz, which was then logged to a text file for further analysis. There are two main types of tests that can be used to test the accuracy of GPS: Static and Dynamic.

Experiment

Static:

The static tests are used to show how the received location from the satellites can change over time when the GPS receiver is held at a set location. In addition, if the placement of the GPS receiver is at a known location, the static test can be used to show how the received location varies relative to the origin. We used Google Earth to find the coordinates of a parking space on campus and used this for the known location data (Figure 11). Using a function within Ashtech we were able to plot the positions received around the known origin (Figure 12).

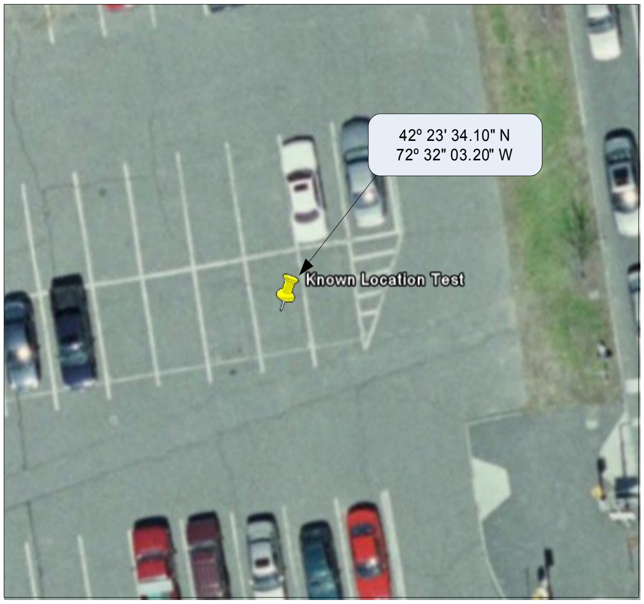


Fig: 11

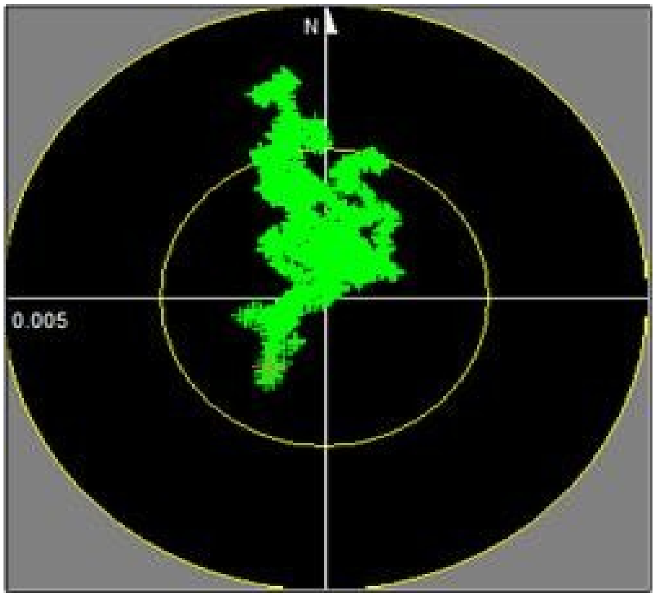


Fig: 12

Dynamic:

The dynamic test has become more important as GPS gains more in mobile applications. This test is used to show how the received signal can vary while the receiver is mobile(e.g. taking turns, varying speeds, changing reception quality). In this test we also used Ashtech in normal mode logging the current position and speed information. This test was meant to show that we the GPS could accurately track our location over a long distance, but the majority of the trip was at a constant speed of 80 km/hr (50MPH) so that we could see if the GPS data would match the test (Figures 13 & 14).

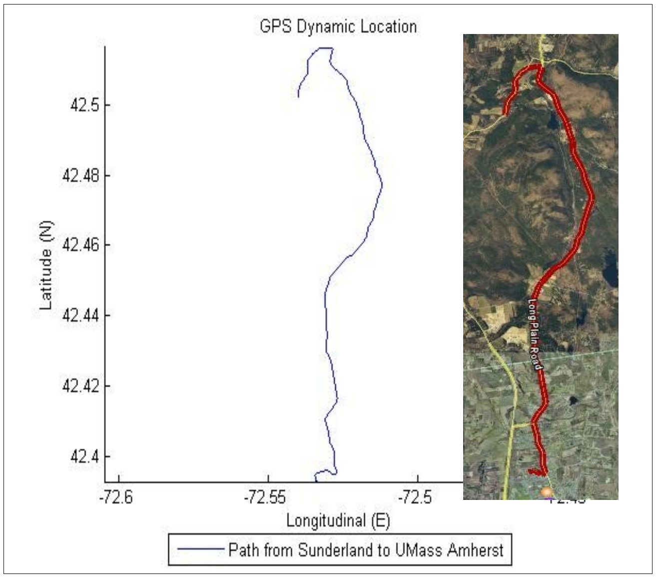


Fig: 13

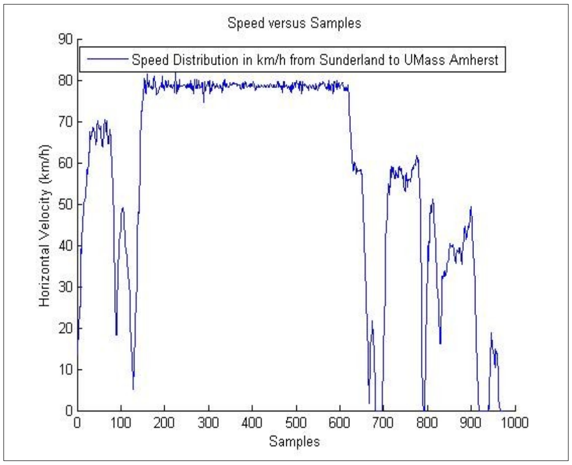


Fig:14

Analysis

Text Parsing: Once we had the log files from our experiments our first task was to parse the data so that we could analyze it.

Results Following Parser

```
42,23.57303,07,232.05458
42,23.57302,07,232.05460
```

```
000.0
000.0
```

Matlab Analysis:

The goal of our analysis was to gain an understanding of the amount of variation we could expect to see in the received position information. In order to do this, we used Matlab to develop a histogram of the euclidean distances from the origin.

Note: > 2000 total data points

I. Comparing Latitude and Longitude Data:

The first step was to create a histogram from the latitudinal and longitudinal in order to compare these results and see if they had a Gaussian distribution (Figures 15 & 16).

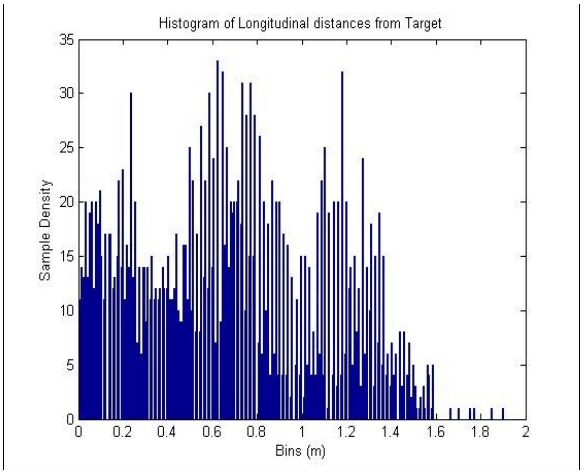


Fig. 15

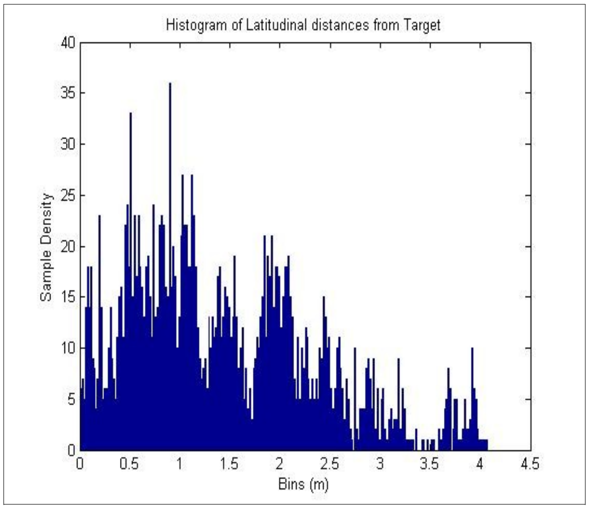


Fig: 16

II. Assuming that the latitude and longitude are both independent, the next step was to calculate the variance of both sets and verify that the results were reasonably close to each other.

$$\sigma^2_{Latitudinal} = 0.8792$$
$$\sigma^2_{Longitudinal} = 0.1696$$

III. Although the variances are not approximately equal, we take the geometric mean between them, and assume that the latitudinal and longitudinal histograms are Gaussian. Based on these three pieces of information , we can form a new random variable called Y. This random variable is defined as follows:

$$Y = X^2_{Latitudinal} + X^2_{Longitudinal}$$

The following plot shows the histogram of this new random variable Y:

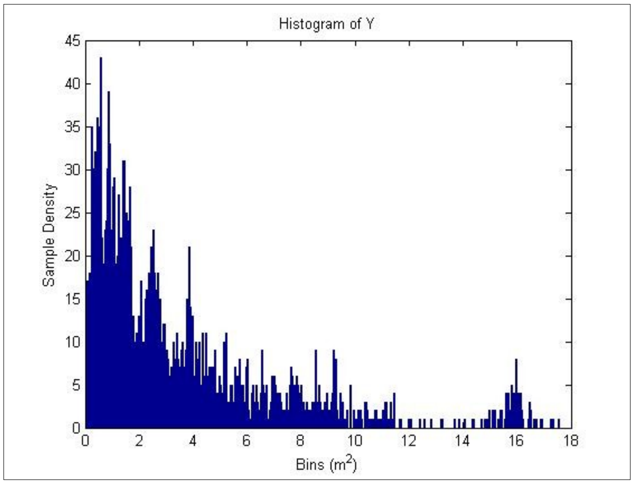


Fig. 17

Notice that this distribution resembles the following Noncentral Chi-Square PDF with 2 Degrees of Freedom:[7]

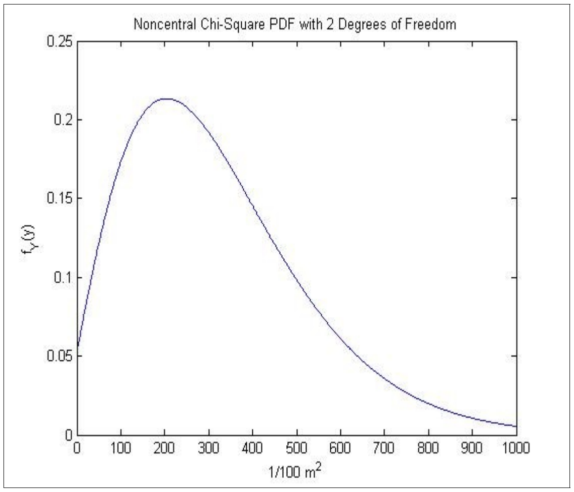


Fig. 18

$$p_Y(y) = \frac{1}{2\sigma^2} e^{\frac{-(s^2+y)}{2\sigma^2}} I_0\left(\frac{\sqrt{y}\, s}{\sigma^2}\right)$$

$$s^2 = \mu^2_{Latitudinal} + \mu^2_{Longitudinal} \qquad \sigma^2 = \sqrt{\sigma^2_{Latitudinal}\, \sigma^2_{Longitudinal}}$$

By integrating the probability density function over the entire distribution, Matlab was able to form the following commulative distribution function:

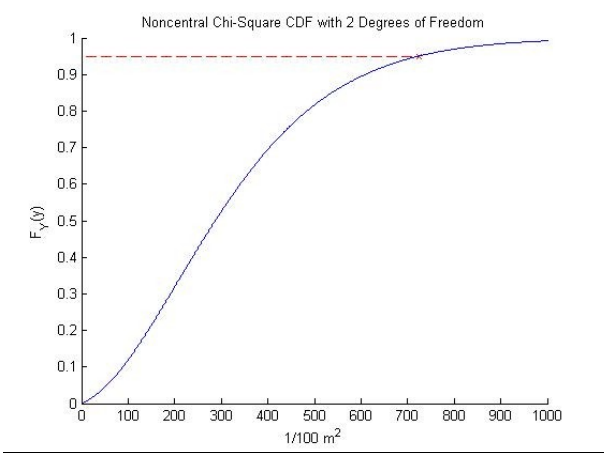


Fig. 19

From this plot, we were able to determine a 2.6m accuracy with 95% certainty.

Interpretation:

This data that has been collected and analyzed shows the range of inaccuracy that we can expect to see from the GPS unit. Since the antenna will be located on the center of the roof of our test vehicle, we need to build an imaginary radius surrounding the vehicle into our algorithm. Based on our analysis we can establish what the radius should be with high confidence.

C. Design of system, component or process to meet desired needs within realistic constraints

In this section we have included the following material:

1) A description of the system requirements of our design.
2) A description of at least two realistic constraints imposed on our design (chosen from the following categories: economic, environmental, social, political, ethical, health and safety, manufacturability and sustainability).
3) A description of how our design to date addresses the requirements and constraints.

System Requirements:

The system requirements of our design are in terms of the overall range, data rate, reliability, and timing of our designed units.

The selected transceiver must have a large radius because it is imperative that the cars are in contact with one another in order to predict accidents. If the cars are not in contact, the range is two small, then in some cases we will be predicting accidents in instances where there is not enough time to safely warn the drivers. The DSRC's were field tested and achieved a maximum range of approximately 500 feet.

Time directly ties our microcontrollers programming and our transceiver together. For our program, we must process our information as fast as possible. The more time we take to process our data the less time there is available for the drivers to avoid their accidents. So we must build clear and concise algorithms that will predict the chances of an accident fast enough to safely warn our drivers. Also the transceiver must have a data rate high enough to transmit our packets to the other vehicles in a timely manner. So, to achieve this goal we plan to keep our packet designs as small as possible, and take advantage of a data rate from the transceiver that will transmit the package the fastest. While the selected data rate must transmit our packets as quickly as possible, the data rate will also directly affect the overall range of our units.

Reliability is the most important part of the design requirements. This will tell the user whether or not their OBE is currently working properly. To complete this there will be an extra light indicator on the user interface. This light will turn on if the system is getting enough reliability out of its GPS unit. This will be determined by the amount of satellites currently available to the GPS unit. If the number of satellites is below the critical line then the light will turn off to let the user know the system is currently not reliable. With the addition of this part, we believe the overall success of our project will increase.

Realistic Constraints:

- Project budget of $600 set by course standard.
- GPS Accuracy/Latency
- Transceiver range, mobility, data rate (the faster the better)
- Timing is critical
- Security (hacking issues)

Security is an important social and political issue for our final design that must be considered while implementing our design. If our design is left without protection malignant individuals, then the project is at risk of being hacked. This of course could lead to people using the device to assist them in breaking the law. Situations such as running red lights can be achieved by a hacker if they reprogram our design to send off warnings to all other drivers. Also a hacker could use the device to cause accidents by simply overshadowing pending accidents. This can be as simple as warning drivers of fake accidents that initiate actual accidents. The vulnerability of our design produces concerns within government because of the projects potential success in the world market.

Our budget applies constraints to all aspects of our final design. We can only make our design as effective as the budget can afford. One main constraint came in the form of our transceiver selection since the new IEEE protocol, Dedicated Short Range Communication, has not hit the market yet. To acquire one of these units, at the moment, will cost us a few thousand dollars because they are only found in black boxes. So this forced us to find an affordable solution that we can use in order to achieve our planned goals.

The range of our transceiver and its data rate are equally important, but, unfortunately, both constrain one another. In order to have a large data rate, we must sacrifice our overall range. This constraint will be the hardest for us to work around because range must be the highest possible so the cars are in contact for longer periods of time. To work around this issue we will begin by keeping our transmitted data as low as possible to see if we can achieve our timing goals.

Design Decisions Which Address These Constraints:

To work around our budget constraint we ordered a transceiver that can achieve the necessary goals and only cost us $21.95. Also we were able to use a GPS module that the team from last year never actually implemented into their design. For our microcontroller design, we chose the Atmega32, which was donated to our team from the school. At the time, we were close to being $500.00 dollars under budget which gave us room to expand our project.

Even though we purchased these transceivers they ended up not being able to complete the job for us. They were very unreliable and very difficult to debug. Since then, we have moved on to donations from Denso Wireless. The donation was, of course, the DSRC units mentioned in previous section of the report. These units not only helped our budget issues but also gave us a better range for the transceiver.

The range and data rate constraint has applied a large amount of pressure and concern on our present design. This constraints will be the hardest for us to work around because range must be the highest possible so that the cars are in contact for longer durations. To work around this issue we will begin by keeping our transmitted data as low as possible to see if we can achieve our timing goals. If the timing goals can be achieved on the lowest data rate then we will be able to take advantage of the measured range of 500 feet. Currently, we can operate around 5.9GHz with a overall range and data rate of approximately 500 feet and 10Mbps, respectively.

D. Multi-disciplinary Team Functions

Jarrod LaBarge, CSE

Scheduler
Software Designer

Jarrod has focused on developing the main algorithm. He developed the ordering of methods and the overall structure of the final main of the OBE.

Daniel Marcq, EE

Team coordinator
Transceiver Specialist
Embedded Programmer

Dan has been working on establishing both transmit and receive for the OBE. He has also helped develop the flow and program for the main algorithm.

Sean Morrell, CSE

Embedded Programmer
Website manager
Graphics Specialist

Sean has focused on the development of the RSE and its user interface. Also, he has worked with the GPS programming and with the main algorithm.

Anthony Swochak, EE

MATLAB Programmer
GPS Analyst
PCB Designer

Anthony has been working directly with the PCB. Anthony has also assisted in the circuit design for the GPS portion of the OBE and the programming behind it.

E. Identification, formulation and solution of engineering problems

The engineering problem we are trying to solve with our design is setting up an effective vehicle-to-vehicle communication system. Along with this we are also trying to use our designed network to help decrease the chances of an accident taking place at an traffic light intersection.

F. Understanding of professional and ethical responsibility

As engineers we are responsible for making our decisions with ethics in mind. With our design, we have to understand that there is a chance someone could use our design to commit crimes. Since our design focuses on the intersection, a person could use the device to assure themselves to safely run a red light. Confidence toward the systems capability to warn others could potentially encourage such behavior. Of course the warning comes before they actually run the light, which means that all other drivers would have avoided entering the intersection. Even though our final design will not be accounting for events such as these, we will be designing it with modularity in mind. In the future, a simple upgrade to both the RSE and the OBE could be made. Some upgrades to the OBE could relay a violators VIN number to the RSE unit, which could then store this information for later collection by the proper authorities.

G. Team Communication

The team has a weekly meeting scheduled every Friday at 1:30pm with our advisors. During this meeting we cover the success of the last week and begin making plans for the upcoming week.

We also have team meets almost daily in lab to discuss debugging issues and to move work around the group.

Outside of our team meeting the group is responsible for maintaining their own side projects and lab times. Once accomplishments are achieved or if someone needs some help working on their particular aspect then they will email the team with this information. Also phone numbers were exchanged over the summer, and all team members were encouraged to maintain phone contact with one another through out the week.

H. Understanding the impact of engineering solutions in a global, economic, environmental and societal context.

Our final design will be one of the first active applications on the market that actively attempts to warn of imminent accidents. Until now, the only solutions we have to date are post accident technologies that simply attempt to assure the survival of the occupants. These devices come in the form of air bags, seat belts and crumple zones. The problem with these forms of safety applications is in order for them to work the accident must take place. Once the accident takes place, even with these safety measures, the occupant's chance of survival is not guaranteed. We aim to prevent the accident completely, which in turn will keep our drivers safe.

The choice of the intersection came from the amount of accidents that take place at an intersection every year. The majority of accidents will take place at an intersection, and, unfortunately, a large portion of these accidents may lead to fatalities. By using our design, the driver will be able to make a decision that may save their life.

I. Application of material acquired outside of coursework. While designing our system we streamed information from many sources that were not related to our course work.

Sources of information that we used include:

(i) We utilized manufacturer's specification sheets for their products. Some of these manufacturers were Atmel and Denso. A full list of references, including these, can be found at the reference section of our report.

(ii) While establishing communication, we used some coding designs by a man named Guido Socher. We came into contact with him while posting on the AVR Freaks forums. He was able to assist us with code that helped us develop our final code for transmitting and receiving on the OBE.

(iii) Another source of information was the AVR Freaks. We used this to get input from other engineers on various design concerns we had while coming up with a specific design for our system.

J. Knowledge of Contemporary Issues

Vehicle collisions at intersections account for a large percentage of overall traffic accidents, injuries and fatalities in the United States. In fact, it is estimated that approximately 70% of fatal traffic accidents (amounting to over 25,000 fatalities per year) occur at intersections. This has motivated a current emphasis on improved intersection safety as part of the U.S. Department of Transportation's Vehicle Infrastructure Integration (VII) initiative. One important component of improved intersection safety is a reduction in the number of accidents caused by vehicles that fail to stop at red lights or stop signs.

K. Use of modern engineering techniques and tools

Some of the techniques we used during our design involved the following:

- MATLAB
- Statistical analysis
- Use of tools such as Oscilloscopes and Spectrum Analyzers
- GoLogic

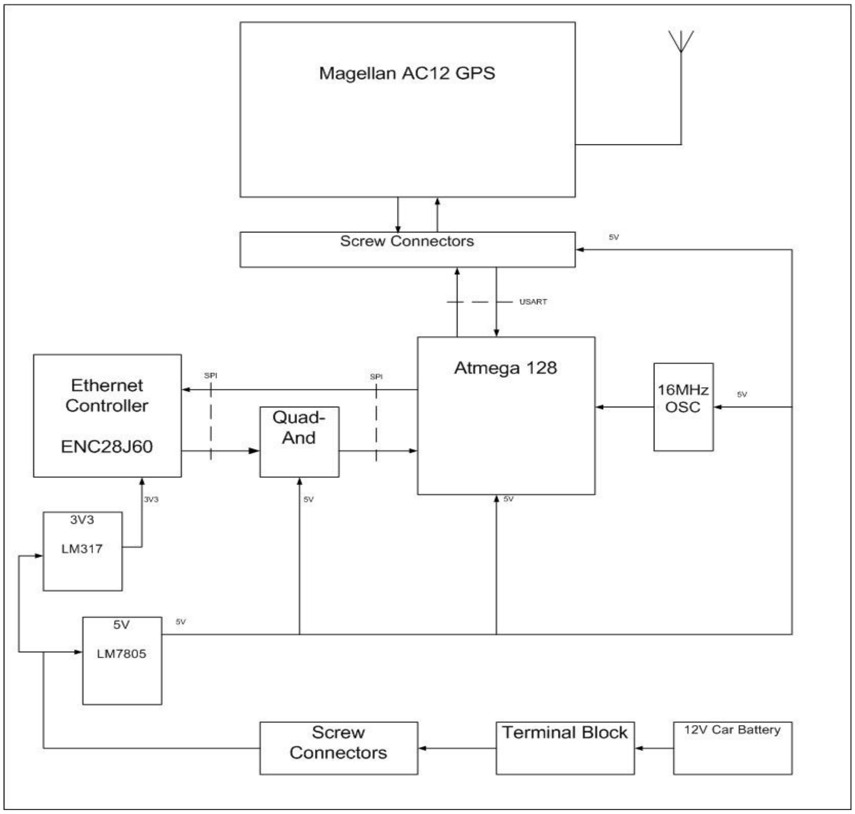
For a detailed example of these please refer back to Appendix B for a MATLAB and statistical analysis example.

Appendix II:

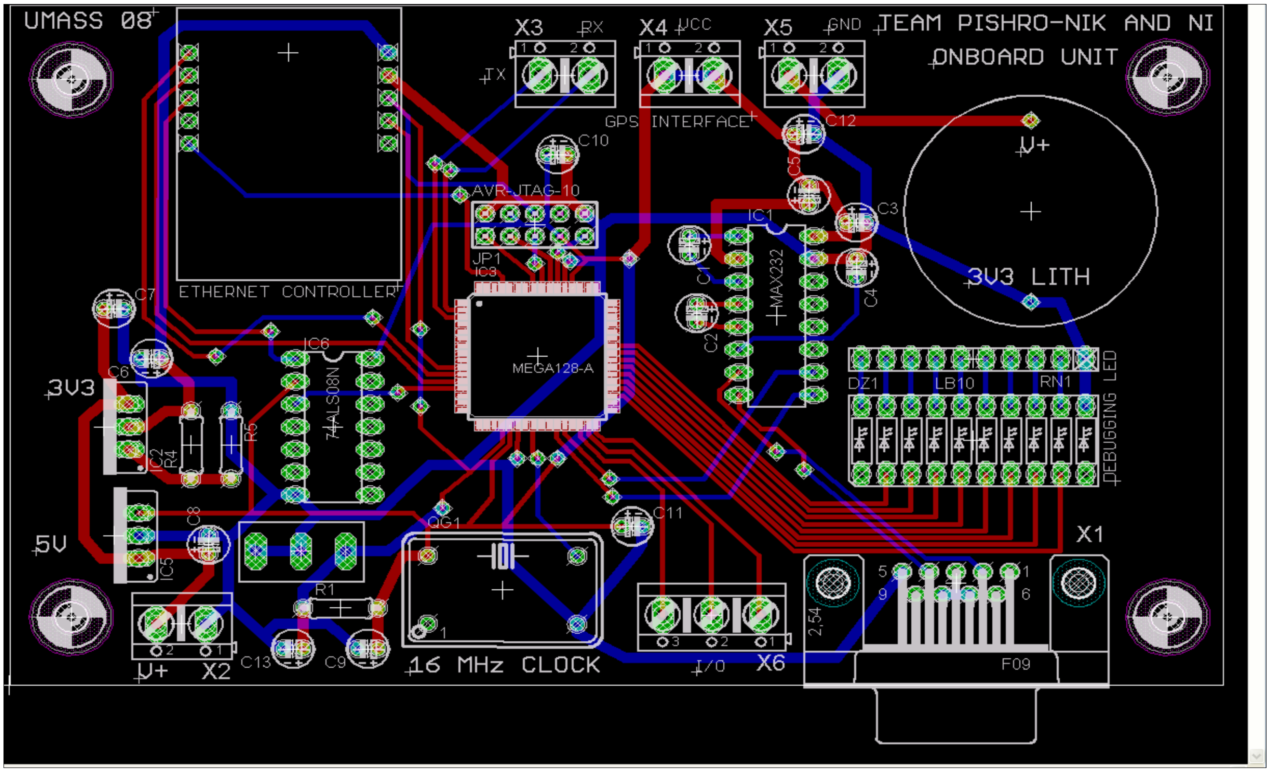
Specifications: (WRM)

FCC Certification	Certified to Part 15 for 802.11a operation (subject to FCC restrictions)
	Tested by DENSO for Part 90 (conducted emissions) in WAVE band
Operating Modes	WAVE, 802.11a
Frequencies	5.725-5.825GHz; 5.85-5.925GHz (WAVE)
Data Rates	3–27 Mbps (10 MHz channels), 6-54 Mbps (20 MHz channels)
Tx output power	0 to 20dBm (rate-dependent), measured at SMA connector
Operating Ambient Temperature	0° to 40° C
Size with housing	7.5"W x 4.7"L x 1.25"H
Total weight (w/o DC converter)	0.6 lb
Command Formats	Internet Protocol (IP) or Telnet
Unit Modes	OBU, RSU

High-Level Circuit Model (OBE):

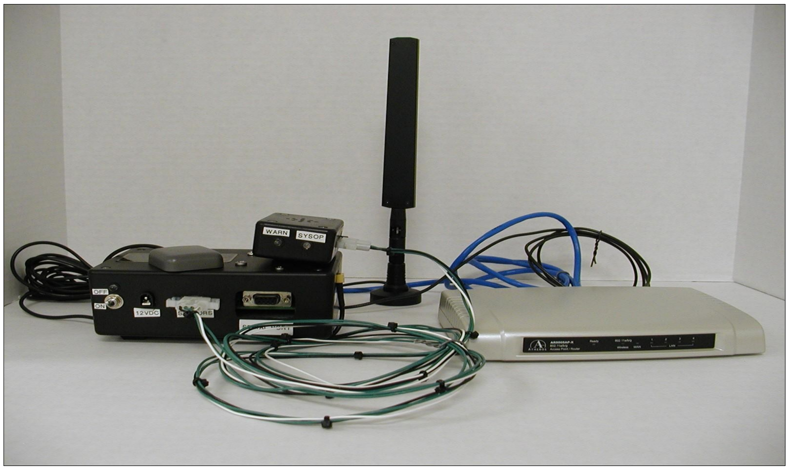


PCB (OBE):

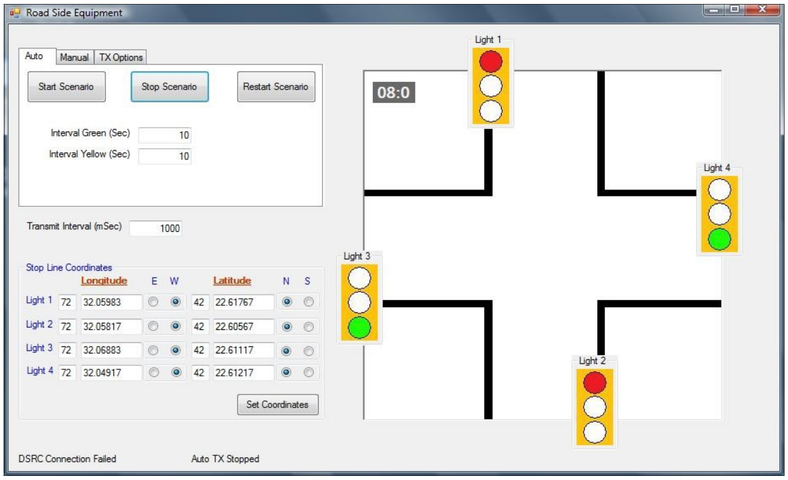


Inside the OBE:

Realized OBE:



RSE GUI:



Interval Green (Sec) 10
Interval Yellow (Sec) 10

Transmit Interval (mSec) 1000

Stop Line Coordinates

	Longitude		E	W	Latitude		N	S
Light 1	72	32.05983	○	●	42	22.61767	●	○
Light 2	72	32.05817	○	●	42	22.60567	●	○
Light 3	72	32.06883	○	●	42	22.61117	●	○
Light 4	72	32.04917	○	●	42	22.61217	●	○

DSRC Connection Failed

Auto TX Stopped

Budget:

Quantity	Unit Price	Description	Shipping	Sub Total
2	$15.05	Atmega128 Microcontroller	$5.20	$30.10
2	$1.50	Female Right Angle Adaptor	$3.30	$3.00
3	$2.31	Buzzer	$0.00	$6.93
4	$0.77	Voltage Regulator	$0.00	$3.08
2	$37.95	Header Board for ATmega128	$6.60	$75.90
2	$1.50	Coin Cell Holder	$0.00	$3.00
2	$1.95	Coin Cell Battery	$0.00	$3.90
1	$79.50	AC12 OEM Board	$0.00	$79.50
1	$90	Mini-Mag Antenna	$0.00	$90
1	$18.75	Molex to Molex	$0.00	$18.75
1	$15	Molex PCBA Header	$0.00	$15
2	$38.67	Ethernet Controller	$0.00	$77.34
1	$34.95	Ethernet Controller	$14.93	$34.95
3	$1.95	Breakout Board for RF-24G Transceiver	$6.60	$5.85
4	$21.95	Transceiver	$6.19	$87.80
4	$0.95	Transceiver Connector	$0.00	$3.80
3	$16.99	DSRC Transceiver Antenna	$9.01	$50.97
3	$33.00	PCB Board	$0.00	$99.00

Subtotals:	$51.83	$688.87
Total Spent:	$740.70	
Over Budget By:	$140.70	

Cost per OBE:

Quantity	Unit Price	Description	Shipping	Sub Total
1	$37.95	Header Board for ATmega128	$6.60	$37.95
1	$1.50	Coin Cell Holder	$0	$1.50
1	$1.95	Coin Cell Battery	$0	$1.95
1	$79.50	AC12 OEM Board	$0	$79.50
1	$90	Mini-Mag Antenna	$0	$90
1	$18.75	Molex to Molex	$0	$18.75
1	$15	Molex PCBA Header	$0	$15
1	$38.95	Ethernet Controller	$0	$38.95
1	MSRP*	DSRC	$0.00	$0
1	$16.99	DSRC Transceiver Antenna	$9.01	$16.99
1	$33.00	PCB Board	$0.00	$33.00
1	$50.00	Miscellaneous Parts and Connectors	$0	$50.00

Total for OBE = DSRC MSRP* plus $399.20

* See Current Market Price for DSRC