Final Report: Vehicle to Vehicle Communication (V2V) Device

ECE 445, Fall 2019

Team 37

TA: Christopher Horn

Alejandro Gonzalez, Harsh Harpalani, Tiger Chung

December 11, 2019

Abstract
In this project, we proposed, verified, designed, implemented, and tested a vehicle to vehicle communication device. Our proposed device collects vehicle data and uses RF to exchange this information with other such devices on the road. Furthermore, our device hosts and broadcasts a web server that is constantly refreshed with the latest device data, resulting in an interface that can be accessed by users and used as an API by other smart devices.

Table of Contents

1. Introduction……………………………………………………………………………..4
    - 1.1 Objective…………………………………………………………………....4
    - 1.2 Background.………………………………………………………………...4
    - 1.3 High-level Requirements List.……………………………………………...5
2. Design……………………………………………………………………………………..6
    - 2.1 Power Module………………………………………………………………....7
        - 2.1.1 Voltage Regulator………………………………………………………7
    - 2.2 Sensors Module……………………………………………………………..7
        - 2.2.1 Global Positioning System………………………………………………..7
        - 2.2.2 Inertial Measurement Unit………………………………………………..8
    - 2.3 Controls Module…………………………………………………………………8
        - 2.3.1 Microcontroller…………………………………………………………..8
    - 2.4 Communication Module………………………………………………………..10
        - 2.4.1 Transceiver………………………………………………………………...10
3. Design Verification……………………………………………………………………11
    - 3.1 Power Module Verification………………………………………………..11
    - 3.2 Sensor Module Verification………………………………………………11
        - 3.2.1 GPS Verification………………………………………………………...11
        - 3.2.2 IMU Heading Verification………….…………………………………...13
        - 3.2.3 IMU Acceleration…………………………………………………….....13
    - 3.3 Microcontroller Verification……………………………………………….....13
4. Costs………………………………………………………………………………………...14
    - 4.1 Parts…………………………………………………………………………14
    - 4.2 Labor…………………………………………………………………………14
5. Conclusion……………………………………………………………………………15
    - 5.1 Accomplishments……………………………………………………………15
    - 5.2 Challenges……………………………………………………………….....15
    - 5.3 Ethical Considerations………………………………………………………16
    - 5.4 Future Work…………………………………………………………………16

References ……………………………………………………………..……………...18

Appendix A Requirement and Verification Tables……………………………………...….19

1 Introduction

1.1 Objective

Currently, even the most advanced commercial smart vehicles collect information about nearby object through sensors located around the vehicle's exterior. However, these sensors are very limited in accuracy for distances past their field of view. Since cars travel at such fast speeds and have to continuously reassess their environment for potential risks, their sensors are often hard pressed to deliver reliable, short-range data in these short time frames. Consequently, smart systems are incapable of making complicated decisions that require long-range data, and often make erroneous life-or-death decisions due to faulty data. With the rise in popularity of autonomous driving systems, reliably collecting data over long distances is becoming more and more crucial.

To resolve this issue, we propose making a device that would attach to every vehicle on the road. This device would broadcast information about the vehicle (dimensions, speed, acceleration, position, heading, etc.) and receive information about surrounding similarly equipped vehicles.

This strategy takes the guesswork out of smart vehicles trying to figure out what other vehicles are doing around them. Instead, vehicles can directly receive this information from other vehicles on the road and use more resources to monitor and track non-vehicular objects. The obtained data could then be fed into an intelligent system (whether an autonomous car or just a smart system in a modern car) and used to make complex decisions (i.e. crash detection at intersections or highways, avoidance of erratic drivers, traffic decisions, etc).

1.2 Background

Motor vehicle crashes are a leading cause of death with an estimated 37,133 lives lost on U.S. roads in 2017 [1]. With a network of our devices, drivers and autonomous vehicles will be aware of the other vehicles on the road to make more informed decisions if there are high traffic locations or accidents on the road. It is estimated that the use of V2V technology could result in a near 80% reduction in non-impaired crashes [2].

Current V2V concepts cover significantly more ground (by nearly 2 magnitudes) than sensor based detection, and are much more accurate in keeping track of vehicular surroundings when compared to sensors [2]. However, most of these devices are still in production and have not been proven effective. Furthermore, most are constrained to 300 meters while our model will at least reach 500 meters.

1.3 High-level Requirements List

- Sensors must reliably collect and have microcontroller store sensor data at a rate of 10 measurements per second.
- Device must be able to receive information from another device within 500 meters at 10 kbps.
- Device must provide an interface or API to export and display data externally.

2 Design

The device requires four major subsections to operate: power module, sensor module, microcontroller, and the communications module. First, the device must be powered by the 12 V auxiliary outlet from the vehicle. And so, a system of voltage regulators or regulator must be implemented to lower the voltage to 3.3 V, the input voltage for our components on our device. The device must also include an array of sensors that will collect data regarding the vehicle. This data will be used by other nodes in the network. A control unit will also be necessary to process the data from the sensors and send that data to a transceiver. Finally, a communication module will be responsible for connecting the devices together, allowing them to transmit and receive data from other in-range devices. This pipeline is displayed in Figure 1.

These modules and components should come together on a printed circuit board, small enough to fit in a vehicle as shown in Figure 2.

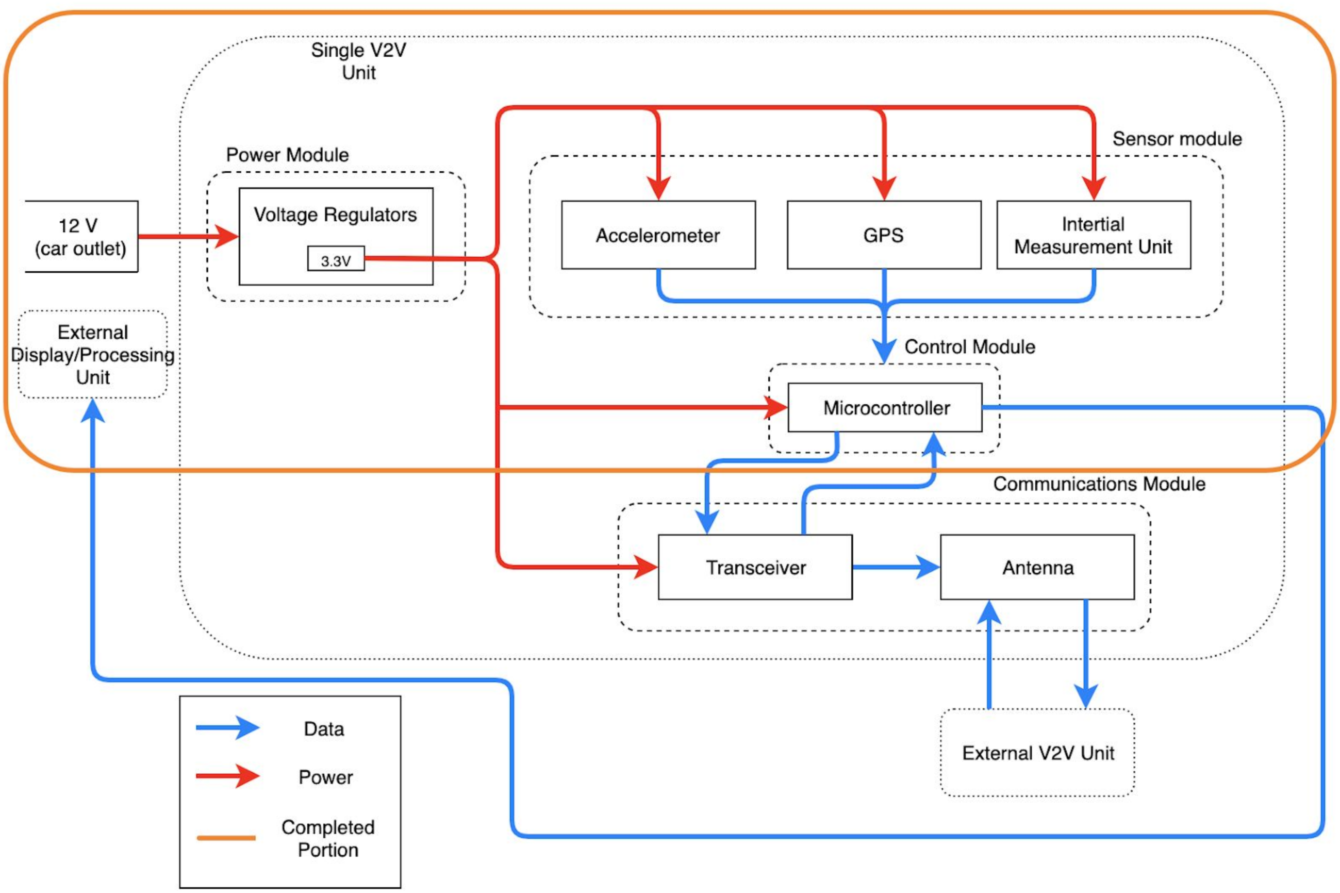

Figure 1. Block Diagram of V2V Device

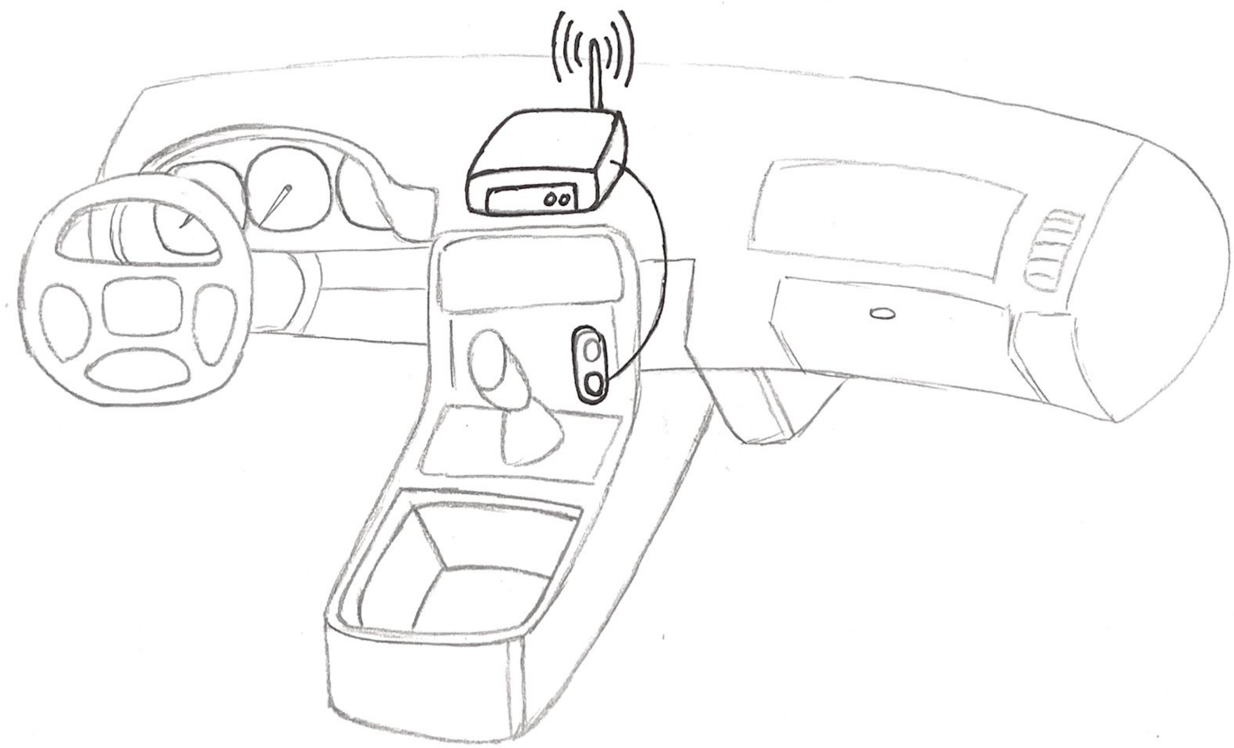

Figure 2. V2V Unit Design and Placement

2.1 Power Module

The power module is responsible for powering the other subsystems, demonstrated in Figure 2. This module draws 12 V from the vehicle and regulates it to 3.3 V, so the appropriate voltage can be continually distributed throughout the device.

2.1.1 Voltage Regulator

In order to distribute a consistent supply of power, a voltage regulator will convert 12 V from the vehicle to 3.3 V DC. Since in a vehicle temperatures can become extreme, the UA78M33CKC regulator withstands these temperatures with low power dissipation as to not overheat.

2.2 Sensors Module

The sensors module is responsible for collecting data from the vehicle and sending it to the control module. The sensors will track the vehicle's location, velocity, acceleration, and direction of heading.

2.2.1 Global Positioning System

The Global Positioning System (GPS) standalone chip is responsible for collecting the vehicle's position and speed. The GlobalTop FGPMMOPA6H GPS Module provides a positional accuracy within 6 meters, while at maximum sending data at 10 Hz. It communicates with the microcontroller through the UART transmitter and receiver of the module.

2.2.2 Inertial Measurement Unit

The Inertial Measurement Unit (IMU) is responsible for recording the vehicle's acceleration and direction. The MPU-9250 breakout board measures both acceleration and direction, while capable of sending data at 4000 Hz and 8000 Hz respectively. The breakout board allows for convenient access to the I2C ports, so it can communicate with the microcontroller.

2.3 Controls Module

The control module is responsible for managing data that is received and outputted from the device. The controls communicate the most recent information from the sensors and sends it to an external display.

2.3.1 Microcontroller

We selected the ESP32-WROOM-32D (specifically, an ESP32 KeeYees Development Board) as the microcontroller to handle the communication between the various components of our project, as well as the processing of all the data. The ESP-32 is a popular, cost-effective microcontroller that provides support for Arduino libraries, has many GPIO pins, offers built-in wireless communication (wifi and bluetooth) modules, and contains a powerful dual-core microprocessor. It also offers a heaping 520 kb of onboard SRAM so our device will not be constrained by the number of other devices' data it can store.

As shown in Figure 3, we utilized the ESP32's built in wifi capabilities by hosting a server on it that we constantly updated with the most current device data. From there, we broadcast this data through a wifi network that any internet-capable device can connect to. The resulting webpage is shown in Figure 4.

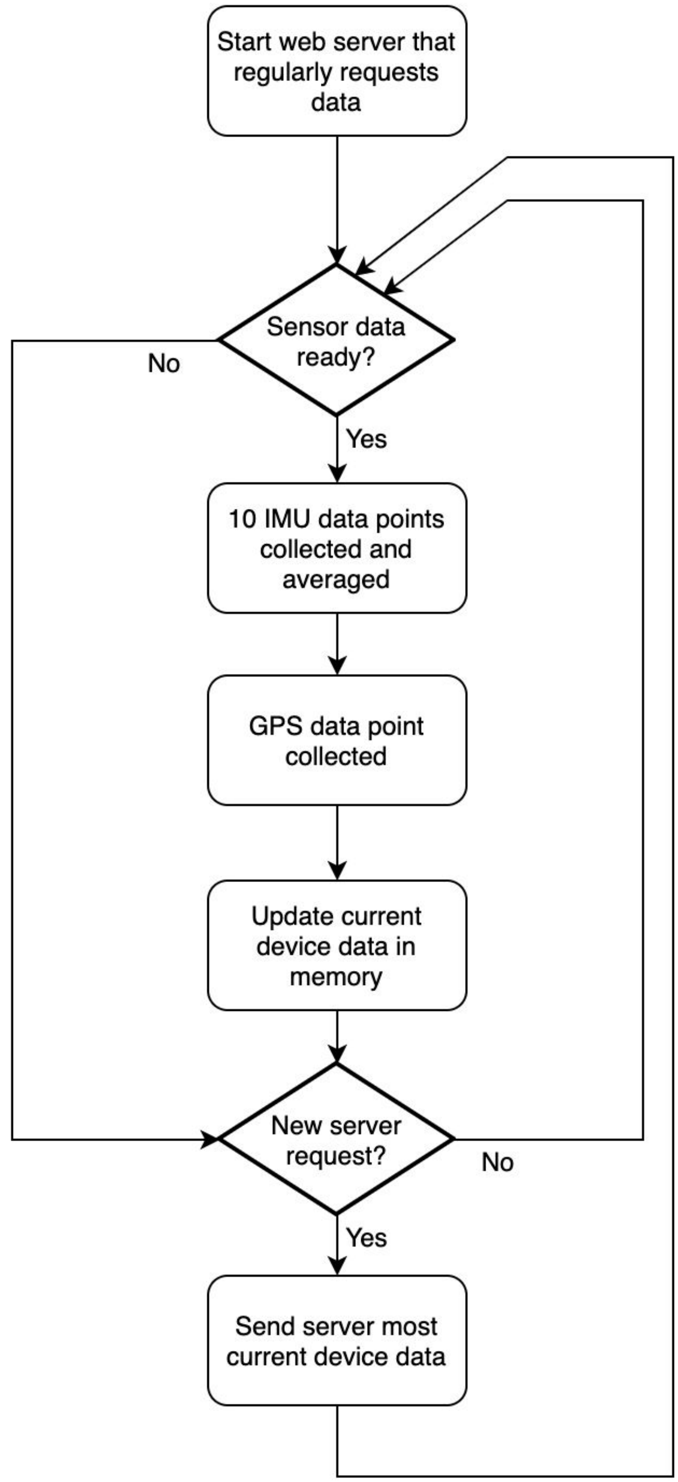

Figure 3. Microcontroller Flowchart of Data Acquisition

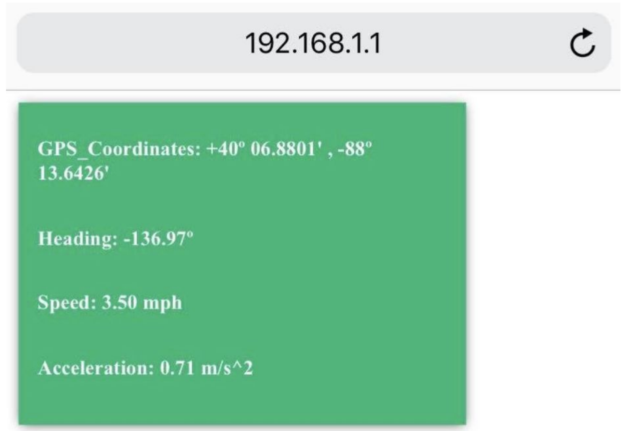

Figure 4. Wifi Broadcasted Device Interface

2.4 Communications Module

The communications module is responsible for transmitting data to and receiving data from other identical units. The data received from the control module will be broadcasted at 900 MHz through a compatible antenna.

2.4.1 Transceiver

The transceiver will transmit and receive data at our specified frequency. This transceiver must operate within our 3.3 V and an operating current under 1 A. It needs a large frequency range giving us flexibility with bandwidth, distance, and channeling and sufficient output power and receiver sensitivity for our design.

3 Design Verification

3.1 Power Module Verification

While inputting voltage from 4 V to 20 V, the power system generated 3.28 V to 3.32 V, which was sufficient voltage to power the device. The system was able to handle the 170 mA load of the other subsystems.

3.2 Sensor Module Verification

3.2.1 GPS Verification

After powering the GPS, and running appropriate code on the device, the GPS module outputs data once a fix is obtained. To verify this data, we had our device output its readings to a text file. We proceeded to travel in a vehicle and recorded our position, and speed. We compared the difference in locations and speed to what was read off of a mobile device and the speedometer of the vehicle. The GPS's location was more accurate than our mobile device GPS as shown in Figure 5 and Figure 6. The data for speed is plotted against the readings from a speedometer in Figure 7. The GPS data had some latency where the GPS data reached the value five to eight seconds after the speedometer had read the speed.

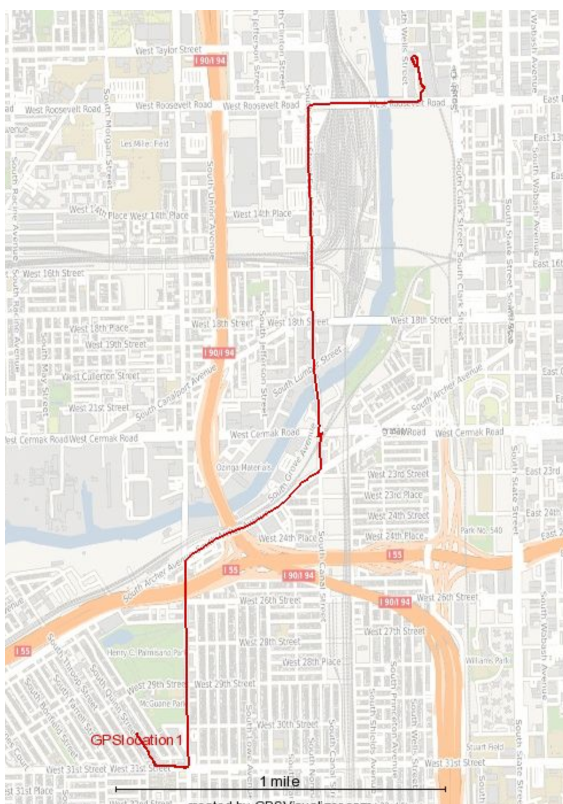

Figure 5. Map Generated by V2V Device

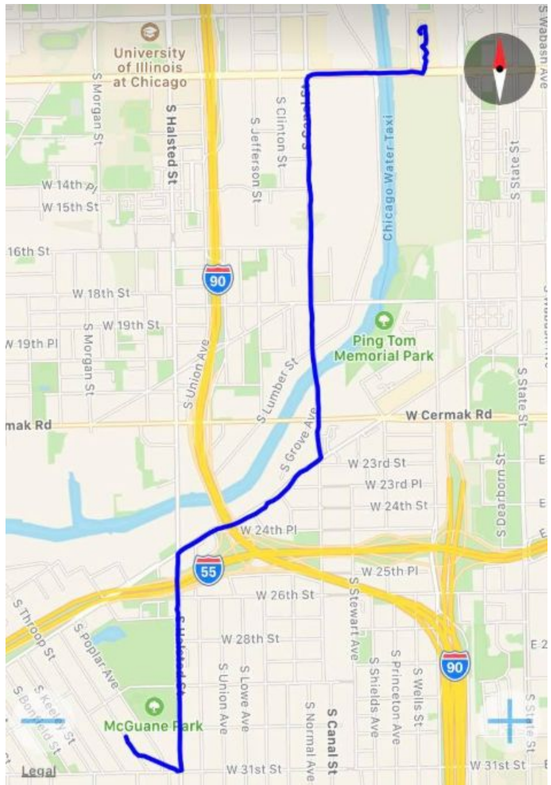

Figure 6. Map Generated by Mobile GPS

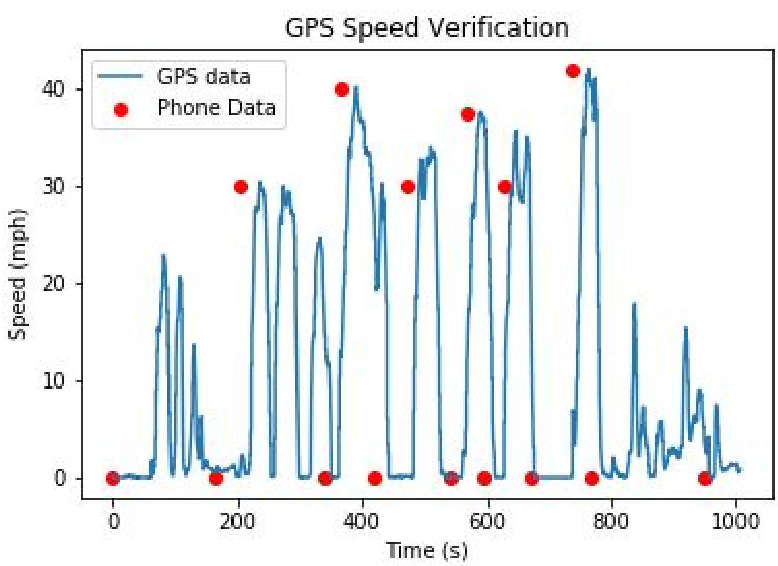

Figure 7. V2V Device Speed Measurements

3.2.2 IMU Heading Verification

We verified that our IMU functioned at the required voltage by operating it at 3.3V and measuring its accuracy. To verify that the IMU outputted a heading that was within 10 degrees of our actual heading, we compared it with our phone's magnetometer (whose datasheet described it to be accurate to a single degree). We put both parallel to one another on the same axis (a wooden plank) and then rotated them 3 degrees at a time. We recorded the measured values of both and plotted the difference, as shown in Figure 8.

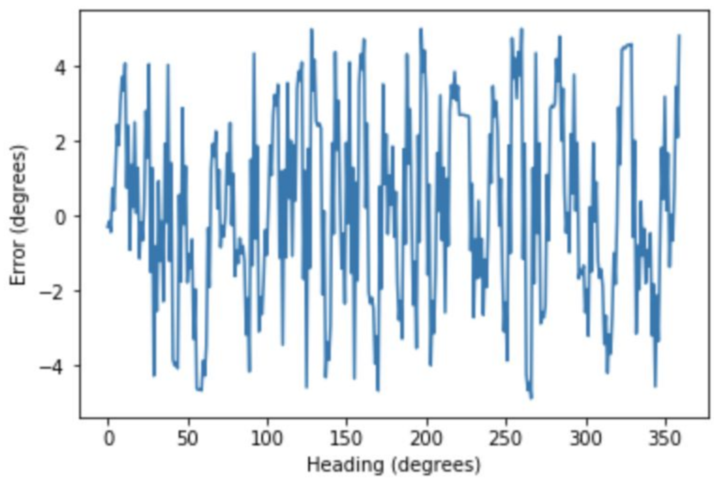

Figure 8. Error Graph of IMU

We found that an average of 10 measurements per reading, the IMU outputted a heading measurement that was at most 5 degrees off, thereby meeting our requirement.

3.2.3 IMU Acceleration Verification

We verified that our IMU functioned at the required voltage by operating it at 3.3V and confirmed that it worked. We did not have enough time to find a means to accelerate the device at consistent magnitudes and were therefore unable to test the IMU accelerometer's accuracy. We did manual testing by quickly accelerating/decelerating the device and checking its readings, that came in the range of -2.5g to +2.5g, thereby verifying that the sensor was capable of measuring 2g's.

3.3 Microcontroller Verification

Under normal operating conditions, the microcontroller's loop reads 1 kb of data from the sensors (the "ready to read" signals and data outputted) and writes 1 kb of data (through updates to the current data of the device). Therefore, we verified our microcontroller by running it under normal operating conditions and assuring that the outputted data (through the UI) was accurate.

4 Cost

4.1 Parts

Table 1: V2V Device Cost

Description	Manufacturer	Part #	Quantity	Cost per unit ($)
Voltage Regulator	Texas Instruments	UA78M33CKC	1	0.38
GPS	GlobalTop	FGPMMOPA6H	1	29.95
Inertial Measurement Unit	InvenSense (standalone), Sparkfun (breakout)	MPU-9250	1	14.95 (breakout)
Microcontroller	Espressif Systems	ESP32-WROOM-32D	1	4.50 (standalone), 10.99 (dev board)
Transceiver	Texas Instruments	CC1000PWR	1	11.80
			TOTAL	68.07

4.2 Labor

To quantify the labor cost for this project, we determined a salary of close to $40/hour. With three engineers working 10 hours per week for 16 weeks of the semester, the cost amounts to $16,000 per engineer. To amount a grand total, we add the total labor cost with the cost of the parts as shown in Table 1.

Labor Cost per Person = ($40/hour) * 2.5 * (10 hours/week) * (16 weeks/semester) = $16,000
Total Labor Cost = $16,000 * (3 partners) = $48,000
Grand Total Cost = Labor + (Parts * Number of devices) = $40,000 + (2 * 68.07) = $48,068.07

5 Conclusion

5.1 Accomplishments

Our project completed two of the three high-level requirements. First, our sensor module provides 10 reliable measurements per second. For each of these 10 measurements, 10 measurements of both acceleration and heading are averaged to minimize impact of outliers. Since our GPS module provides accurate location data at 10 hz, we did not need to perform any filtering or outlier detection on its outputs. The speed outputted is internally calculated using the built-in haversine function in the GPS module.

Additionally, we provide a method to externally view the device's data. Since almost everyone today has a smartphone and most modern vehicles have built in wireless capabilities, broadcasting the device's data through a wifi network is a convenient, user-friendly means to export it.

5.2 Challenges and Uncertainties

Although we accomplished a significant portion of our project, we were hard pressed on time due to revisions in our project. As a result, we did not have time to work on our communications module and did not get it functioning. Furthermore, we ran into driver issues with our IMU and and the transceiver. We did not have time to write our own drivers so we ordered a different IMU that had a supporter library to interface with our microcontroller.

Another challenge that we were surprised to run into was the power supply module. In our original design we had planned to use an efficient buck converter for lower power dissipation. However we were not able to achieve 3.3 V output with the switching converter. We confirmed that the switching converter was taking in an input and switching on a pin. We also confirmed the soldering was done accurately through continuity testing, reading the correct resistor value across each resistor. And finally we changed values in the feedback circuit to try to achieve a different output voltage, but nothing was conclusive. The final decision was to replace the buck converter circuit with a linear dropout regulator with an unsavory power dissipation because it successfully dropped 12 V to 3.3 V.

There was also a challenge of soldering all the components to the printed circuit board because of the need to debug. When there were malfunctions across devices the printed circuit board did not offer any header pins or places to monitor signals. Running into this problem early, we chose to purchase breakout boards, so that debugging would be more convenient. The microcontroller remained on the development board due to the advantage of quickly being able to upload code using the FTDI port. There was also a challenge of getting the microcontroller onto a printed circuit board and uploading code, which was not completed due to the time constraint.

5.3 Ethics and Safety

There are few potential safety hazards within our project. The only concern of ours is the power consumption of our device. Our circuit draws a current load of about 200 mA which can dissipate 2400 Watts. This can lead to overheating in the power supply circuit. This can damage our parts and can reach temperatures that can cause harm if exposed to for extended periods of time. To mitigate the power dissipation in the power circuit, a more efficient power module can be chosen.

There are numerous safety concerns however with the use of our device. Since one of the sole purposes of vehicle to vehicle communications is to lower risks and increase safety in all types of vehicles, safety is the essence of our project. Especially since failures in this type of technology can lead to fatal accidents in the real world. The most major safety hazard in the use of this device would be the latency of data accumulation and transmission. We want to minimize latency to <50ms. Meaning that from the moment a signal is broadcast from the transmitter, the receiver on another device (<500m away) will be able to process the packet in less than 50ms. This is important because we need about 10 data points to calculate velocity and 0.5s (50ms*10 = 500 ms = 0.5s) is twice the average human reaction time to visual stimuli [3]. We can mitigate the latency of data accumulation with the use of the OBD-II port from the car. The OBD-II port has direct values of the speed of the car with no GPS calculations which would reduce the latency of the data accumulation. And to mitigate the transmission latency, frequency choice, antenna, interference, power out, can all help to mitigate the latency of transmitting data.

Our device does not directly violate any ethical concerns directly. However our device might be used for malicious purposes. The device has the ability to track and record the location of the vehicles. This can violate Section 1.6 of the ACM code of ethics, respect privacy [4]. To combat the misuse of our device, it is possible to implement some encoding to make it impossible to record the position of any vehicle, unless the user has the authorization to access the data.

5.4 Future Work

For our future work on this project, our highest priority is to finish integrating the communications module. Once we have a working transceiver that can communicate with the microcontroller, we will have a functional V2V device that we can reproduce. Next, we will migrate all our components from a perf board to a PCB, giving our device a much cleaner look and a more sturdy feel.

From there, we will focus on optimizing and increasing functionality of our V2V device. The first step would be to encrypt data and incorporate a key-exchange protocol. Without encryption, there is a risk that an attacker will be able to aggregate large amounts of data about vehicles on the road. Next, we would change the type of communication between devices; as of now, the devices are configured to communicate and exchange information with one another in a P2P fashion. To significantly increase the data available to each device, we would configure our devices to communicate in a network-based manner. That is, each device would not only broadcast its own information but also the information of every device it contains data of, thereby serving as a network extender.

This would allow data of an individual car to reach miles away in relatively dense areas. With so much data received and transmitted by each car, vehicles can make much more complex decisions. For example, a group of vehicles moving slower than usual would imply excess traffic to another vehicle that has yet to reach within direct communication range of the vehicles in traffic. We would generate unique IDs for each equipped device, so information is unique and updated as it is broadcasted to other devices. This network of nodes would require a distributed system implementation, point-to-point, in the sense that there is likely no server for all of the nodes to refer to. The goal would be that all nodes are clients and servers for this communication.

To expand on this idea of a network of nodes to form a network, we would also implement a dynamic channel allocation (DCA) system such that with a limited frequency range, we could communicate with a great amount of nodes. The idea is that one node would not be limited to broadcasting on a certain channel but whichever channel would be available at the time of the broadcast. This would allow more nodes than existing channels on the network.

A final extension of our project would be to interface with the car through the OBD-II port rather than using sensor data. The OBD-II port in vehicles exports data about the vehicle such as acceleration, emissions, velocity, etc., allowing for more detailed and accurate information to be transmitted through the V2V communication network.

References

[1] *Transportation.gov*, 2019. [Online]. Available: https://www.transportation.gov/sites/dot.gov/files/docs/policy-initiatives/automated-vehicles/320711/preparing-future-transportation-automated-vehicle-30.pdf. [Accessed: 19- Sep- 2019].

[2] T. Kevan, "V2V Technology: A Work in Progress - Digital Engineering", *Digital Engineering*, 2019. [Online]. Available: https://www.digitalengineering247.com/article/v2v-technology-work-progress/. [Accessed: 19- Sep- 2019].

[3] D. Yuhas, “Speedy Science: How Fast Can You React?,” Scientific American, 24-May-2012. [Online]. Available: https://www.scientificamerican.com/article/bring-science-home-reaction-time/. [Accessed: 02-Oct-2019].

[4] "The Code affirms an obligation of computing professionals to use their skills for the benefit of society.,” *Acm.org*, 2019. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 19- Sep- 2019].

[5] "IEEE Code of Ethics", *Ieee.org*, 2019. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 16- Sep- 2019].

Appendix A Requirement and Verification Table

Table 2: Power Subsystem

Requirements	Verification
(1) Voltage regulator provides 3.3 V ± 5% from a 12 V source.	1. Attach the input of the power supply to 12 VDC. 2. Using a voltmeter, attach the leads to the output of the power supply. 3. Measure the output and confirm that the output is within 3.13 V to 3.46 V.
(2) Voltage regulator must be able to handle at least a 1 amp load.	1. Attach the power supply to an ammeter. 2. Power the power supply with 12 VDC. 3. Draw a load of 1 amp. 4. Confirm that power supply still supplies 3.3V.
(3) Voltage regulator can operate at temperatures -40°C to +85°C.	1. Power the device. 2. Heat the device to 80°C. 3. Measure the output voltage. 4. Confirm that output is 3.3 V. 5. Otherwise check data sheet for temperature operating conditions.

Table 3: Sensor Subsystem

Requirements	Verification
(1) Accelerometer must be able to operate at 3.3 V.	1. Attach 3.3 V to accelerometer. 2. Confirm the device powers on, and outputs readings.
(2) Accelerometer must be able to read at least 2 g.	1. If possible, power the accelerometer. 2. Apply 2g of acceleration to the device. 3. Confirm the measurement is read. 4. Otherwise check data sheet.
(3) Accelerometer reads at ±10% the actual reading.	1. Power up accelerometer. 2. Input a test acceleration of 1g, by letting the device fall a short distance. 3. Record the reading from the accelerometer and confirm it's within required range.
(4) GPS unit must operate at 3.3 V.	1. Power GPS unit with 3.3 V. 2. Confirm that the unit is powered and outputs GPS coordinates.
(5) GPS unit must output coordinates within 10 meters of its actual location.	1. Go to an intersection or location with known coordinates. 2. Power up GPS unit. 3. Record the GPS output and calculate error.
(6) Inertial Measurement Unit (IMU) must operate at 3.3 V.	1. Power IMU with 3.3 V. 2. Confirm IMU powers on and outputs a direction.
(7) IMU outputs a direction within ±10 degrees of the actual direction.	1. Power the IMU. 2. Measure the IMU reading. 3. Align a compass to the IMU direction and measure the compass reading. 4. Calculate the difference between the readings, should get within 10 degrees.

Table 4: Controls Subsystem

Requirements	Verification Steps:
(1) Microcontroller contains 10 general input/output pins and processing power to interface with sensors and transceiver.	1. Power/set-up microcontroller. 2. Power/set-up sensors and connect to microcontroller. 3. Power/set-up two transceivers and connect both to microcontroller, such that values sent/received can be stored. 4. Run script on microcontroller to input data from sensors and one transceiver, while simultaneously broadcasting data to the other microcontroller. 5. Will ensure that data received/sent from both microcontrollers is valid and performance is not hindered (by checking memory of microcontroller).
(2) Microcontroller must reliably write to memory at 1+ kbps to account for inputs from sensor data and from the other device's data.	1. Power/set-up microcontroller. 2. Power/set-up sensors and connect to microcontroller. 3. Run script (similar to requirement #1) to input data to memory at 2 kbps from sensors. 4. Verify accuracy of stored values.
(3) Microcontroller must reliably read from memory at 1+ kbps for transmission.	1. Power/set-up microcontroller. 2. Run script (similar to requirement #2) to copy data from one memory location to another at 2 kbps. 3. Verify accuracy of stored values.

Table 5: Communications Subsystem

Requirements	Verification
(1) Transceiver must operate at 3.3V.	1. Connect transceiver with 3.3 VDC. 2. Confirm that the device operates.
(2) Transceiver must be able to transmit and receive at 900 MHz.	1. Power transceiver. 2. Program transceiver to transmit at 900 if necessary. 3. Input signal to transmit to transceiver. 4. Record signal received on a receiver. 5. Confirm the signal is the same.
(2) Transceiver must be able to communicate with similar modules at 500 meters.	1. Connect modules to devices. 2. From the same location, test communications by receiving the GPS location of each device. 3. Walk one device away until communications terminate. 4. Measure the max distance, and check that it is more than 500 meters.
(3) Transceiver must be able to receive and transmit data at 10 kbps.	1. Power/set-up microcontroller. 2. Power/set-up sensors and connect to microcontroller. 3. Power/set-up two transceivers and connect both to microcontroller, so that values sent/received can be stored. 4. Run a program to send data at 10 kbps between the transceivers, while recording data sent/received in microcontroller memory. 5. Verify accuracy of results.
(4) Transceiver must be able to switch between transmitting and receiving at 10 Hz.	1. Power microcontroller. 2. Power sensors and connect to it. 3. Power/set-up two transceivers and connect both to microcontroller, so that values sent/received can be stored. 4. Run a program to have microcontrollers switch between receiving/transmitting and have the two transceivers exchange data. 5. Verify accuracy of results.