Vehicle to Vehicle Communication (V2V) Device

ECE 445 Design Document -- Fall 2019

Alejandro Gonzalez, Harsh Harpalani, Tiger Chung
Team 37
TA: Christopher Horn
October 3, 2019

1 Introduction

1.1 Objective

Currently, the way smart vehicles work is that they have sensors all over the car to detect nearby objects and information about them. However, these sensors tend to be limited in accuracy past very short distances. And since cars travel at such fast speeds and have to continuously identify changing information in their surroundings, sensors often do not have enough range to identify the necessary information for complicated decisions (crash detection at intersections or highways, avoidance of erratic drivers, traffic decisions, etc.). With the rise in popularity of autonomous driving systems, reliably collecting data over long distances is becoming more and more crucial.

To get over 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 way, the guesswork is taken 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 spend more resources monitoring/tracking non-vehicular objects. This data could then be fed into an intelligent system (whether an autonomous car or just a smart system in a modern car) and could be used to make complicated decisions such as those described earlier.

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 [2]. 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 using V2V technology, there could be a near 80% reduction in non-impaired crashes [3].

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 [3]. However, most of these 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 10kbps.
- Device must provide an interface or API to export and display data externally.

1.4 Physical Design

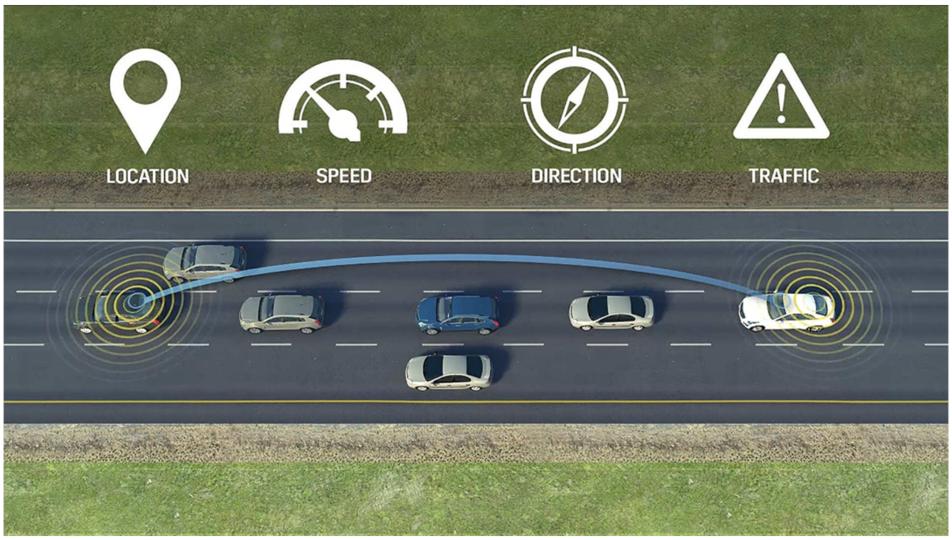

Figure 0. Design Concept

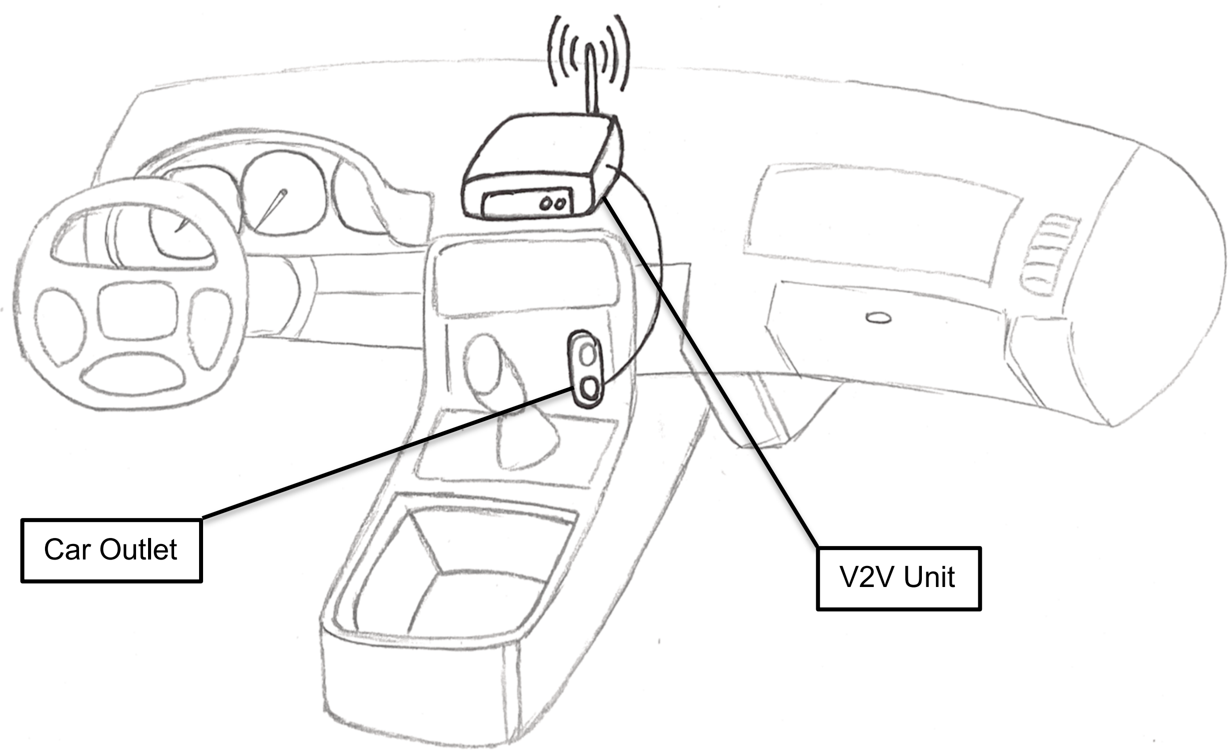

Figure 1. Unit Design and Placement

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 12V auxiliary outlet from the vehicle, so a system of voltage regulators or regulator must be implemented to lower the voltage to 3.3V, 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, such that other nodes will be able to use the data in processing. A control unit will then be necessary to process data from received and transmitted. Finally, a communication module will be responsible for connecting the devices together, allowing them to transmit and receive data from other in-range devices.

2.1 Block Diagram

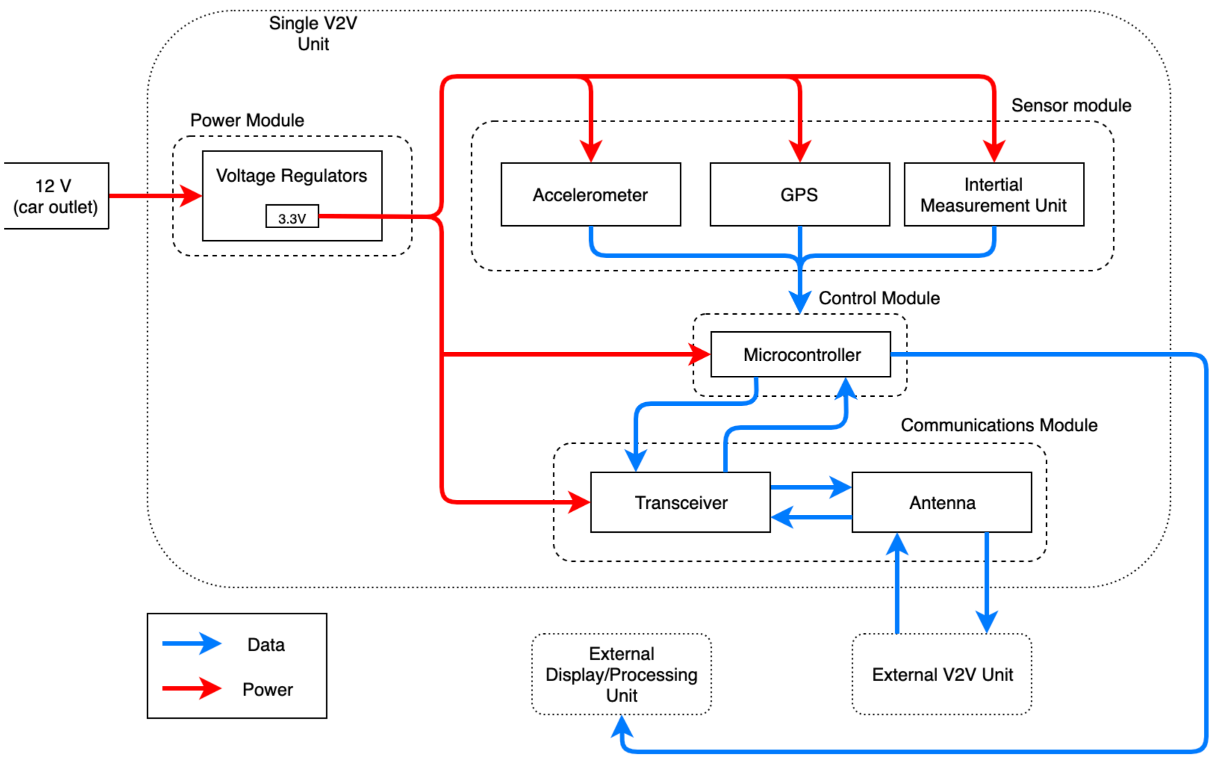

Figure 2. Block diagram

2.2 Physical Design

2.2.1 Power Supply

A power supply is necessary to power the components of the device. 12V is drawn from the vehicle and is then regulated to 3.3V which is then distributed through the rest of the system. This module will be required to continually power the system at the appropriate voltage.

Requirements	Verification
(1) Voltage regulator provides 3.3V +/- 5% from a 12V source	1. Attach the input of the power supply to 12VDC. 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 3.13V to 3.46V.
(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 3.3V. 5. Otherwise check data sheet for temperature operating conditions.

2.2.1a Voltage Regulator

A 12V to 3.3V DC to DC Buck converter power supply will be used to distribute power to the system. A Buck converter will be used because it is our main power supply and all of our devices will draw load from it, so we chose a reliable high capacity power supply. Since in a vehicle temperature can become extreme, we chose a high tolerant power supply with low power dissipation, so it won't overheat.

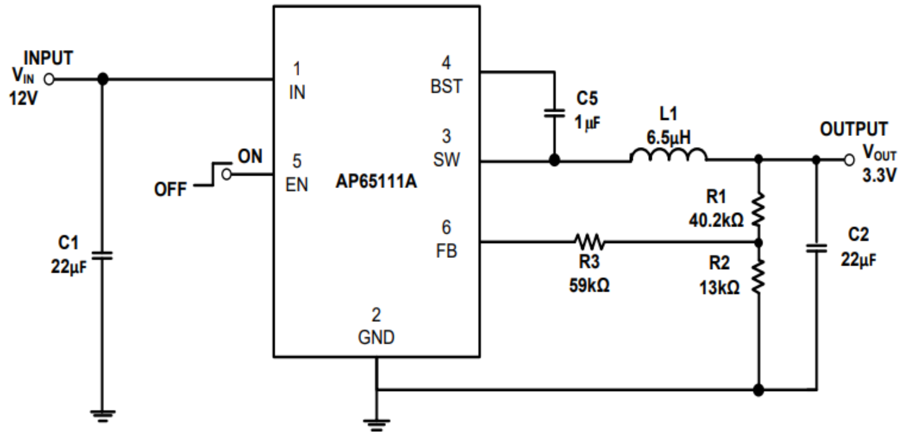

Figure 3. Voltage Regulator Schematic [6]

2.2.2 Sensors Module

An array of sensors will be responsible for collecting data and sending it to the control unit. The sensors will track the vehicle's speed, acceleration, direction, and position.

Requirements	Verification
(1) Accelerometer must be able to operate at 3.3V	1. Attach 3.3V to accelerometer. 2. Confirm the device powers on, and outputs readings.
(2) Accelerometer must be able to read at least 2g.	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 3.3V	1. Power GPS unit with 3.3V. 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 3.3V	1. Power IMU with 3.3V. 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.

2.2.2a Accelerometer

The accelerometer will measure the vehicle's acceleration. We need an accelerometer that can record the average acceleration rates of cars, which are commonly between 3 and 4 m/s^2. We chose the KXTJ3-1057 since it operates between 1.71V to 3.6V, fitting into our 3.3V pipeline. The sensor has extended configurable g-ranges, from ±2g to ±16g, which covers all cars' possible accelerations. This range is enough for us to deduce a dangerous amount of acceleration.

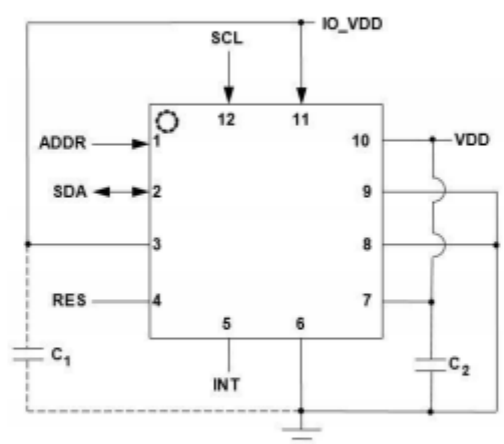

Figure 4. Accelerometer Schematic [7]

2.2.2b GPS device

The GPS breakout will return the GPS location of the vehicle. The GPS breakout has to operate close to 3.3V, to meet the voltage requirement. It requires a positional accuracy within 10 meters to fit our vehicle's needs. This accuracy is vital in the ability to determine the vehicle's speed and coordinates. We chose the FGPMMOPA6H, because it operates between 3 to 5.3V and has an accuracy within 8 meters.

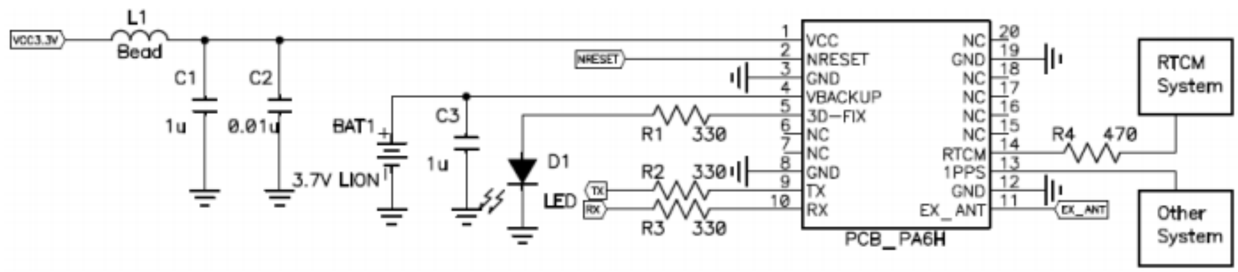

Figure 5. GPS Schematic [8]

2.2.2c Inertial Measurement Units

The inertial measurement unit (IMU) needs to determine the direction the vehicle is heading towards and must work within the operating voltage of a vehicle. We chose the LSM6GSOXTR, because it operates between 2.5V to 38V, and fits our operating requirements. The sensor has very low sensitivity to voltage and temperature changes. This sensor will be responsible for outputting a vehicle's heading.

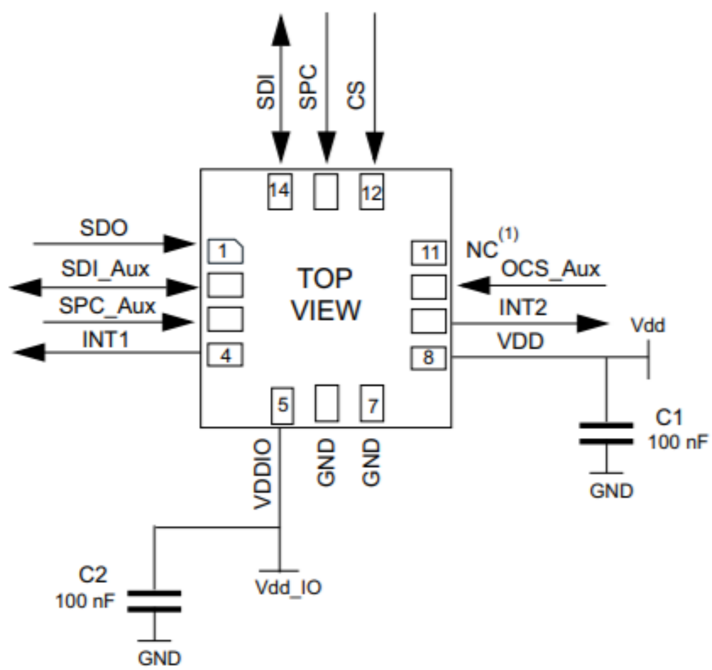

Figure 6. Inertial Measurement Unit Schematic [9]

2.2.3 Controls Module

The controls are responsible in managing data that is received and outputted from the device. The controls will be responsible for communicating the most recent information and sending the correct notification.

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.
(4) Microcontroller must have a variable clock to interface with varied frequencies that sensors, transceiver, and read/write operations will be running at. Needs to allow synchronization control to prevent reading/writing simultaneously	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. Will run program for requirement #1 with interrupt control. 5. Verify accuracy of stored values.

(5) Microcontroller needs to perform encryption/decryption of data fast enough as to not bottleneck operations (<1ms per byte)	1. Power/set-up microcontroller. 2. Run test script to encrypt/decrypt 4kb of data. 3. Ensure results and verify that per byte, encryption/decryption time is less than 1 ms.

2.2.3a Microcontroller

The microcontroller, chosen to be ESP32-WROOM-32D, will be responsible for handling data, storing, receiving, processing, and outputting the correct signals. We chose this microcontroller for several reasons. Firstly, it operates between 2.8V to 3.63V so it fits into our voltage preference of 3.3V and is a cost-effective microcontroller. It has a programmable clock so we can use it for synchronization purposes. Another consideration we had was that the microcontroller has a trigonometric math unit, which accelerates trigonometric calculations (used for distance calculations) by 400% as opposed to regular library functions. The microcontroller also has 520 kb of onboard SRAM, so we can load a few large programs onto it (for operation and verification purposes), as well as store real-time sensor/device data. The ESP microcontroller supports Arduino libraries which makes programming much more user-friendly for us. Lastly, the microcontroller contains an AES hardware accelerator so our encryption/decryption will likely be faster by an order of 2 magnitudes.

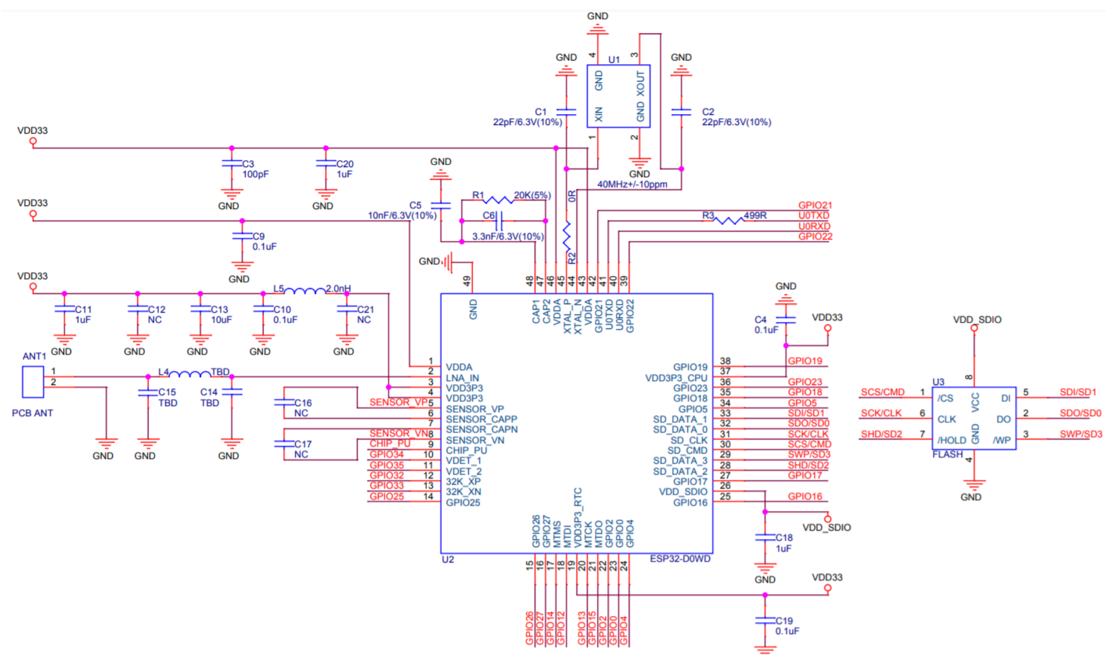

Figure 8. Microcontroller Schematic [10]

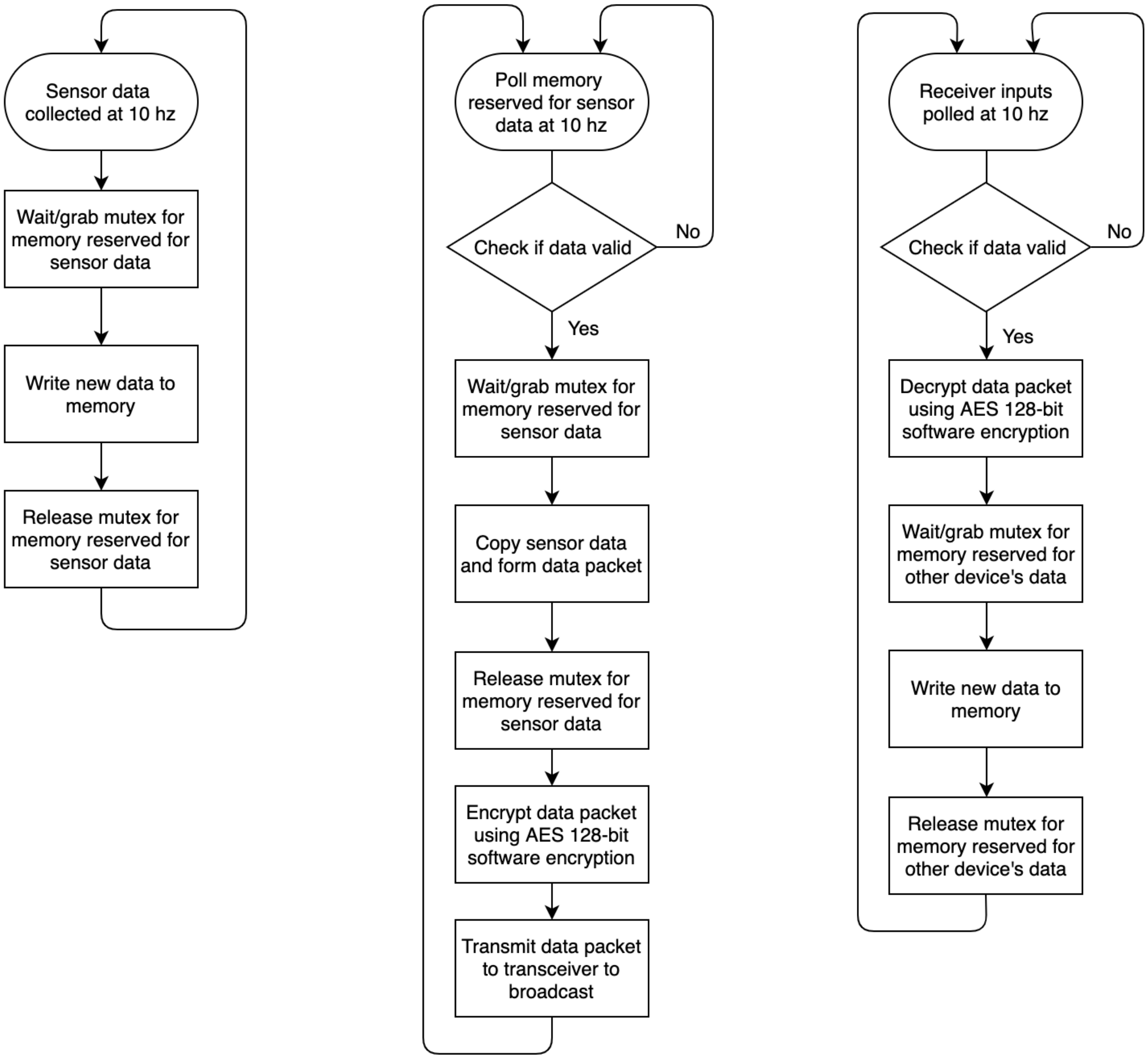

Figure 7. Microcontroller flowcharts. From left to right:
Collecting data from sensors, transmitting data of device, collecting receiver data of another device

2.2.3b Microcontroller Calculations

Velocity Calculations:

We will be using our GPS coordinates to calculate velocity. Specifically, we will find the velocity by measuring the average distance between GPS coordinates over a second's interval. If a second has 10 GPS measurements in the form of $[a, b, c, d, e, f, g, h, i, j]$, with each letter specifying a pair of coordinates, then our velocity for a given second would be:

$$average\left(\frac{deltaDist(a,f)}{deltaTime(a,f)} + \frac{deltaDist(b,g)}{deltaTime(b,g)} + \frac{deltaDist(c,h)}{deltaTime(c,h)} + \frac{deltaDist(d,i)}{deltaTime(d,i)} + \frac{deltaDist(e,j)}{deltaTime(e,j)}\right).$$

The *deltaTime* functional is trivial as each measurement of coordinates occurs at a 0.1s interval. The *deltaDist* function is what is known as the *haversine function*, as described below:

$$a = sin^2\left(\frac{\Delta\varphi}{2}\right) + cos(\varphi_1) \cdot cos(\varphi_2) \cdot sin^2\left(\frac{\Delta\lambda}{2}\right)$$
$$c = 2 \cdot atan2(\sqrt{a}, \sqrt{1-a})$$
$$distance = R \cdot c$$

$where\ \varphi\ is\ latitude, \lambda\ is\ longitude, R\ is\ earth's\ radius\ (mean\ radius = 6{,}371km).$

2.2.4 Communications Module

The communications module is required to transmit and receive data to other identical units. The data from the control unit will be sent to the transceiver to be broadcasted 900 MHz through a compatible antenna.

Requirements	Verification
(1) Transceiver must operate at 3.3V.	1. Connect transceiver with 3.3VDC. 2. Confirm that the device powers on and 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 10hz.	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 have microcontrollers switch between receiving/transmitting and have the two transceivers exchange data 5. Verify accuracy of results

2.2.4a Transceiver

The transceiver will be responsible for transmitting and receiving the data at a specific frequency, so we chose the ADF7021-VBCPZ. This transceiver operates between 2.3V to 3.6V, fitting into our voltage requirement. It has an operating current of 16.3mA for receiving and 13.8mA for transmitting and has a large frequency range giving us flexibility with bandwidth, distance, and channeling. This range includes the 900 MHz frequency we chose to use, and also has sufficient output power and receiver sensitivity for our design.

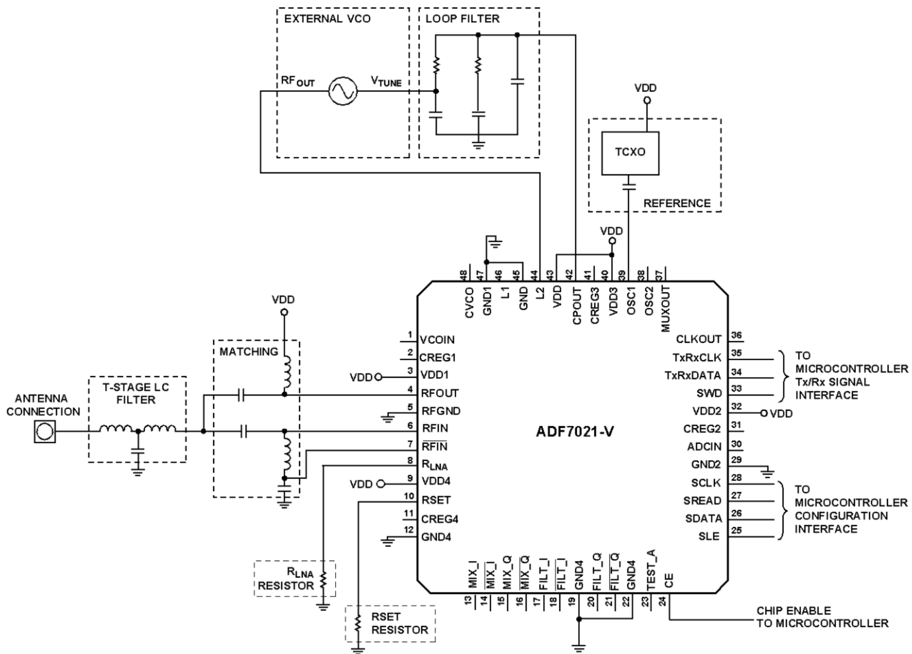

Figure 8. Transceiver Schematic [11]

2.2.4b Antenna

The antenna will assist in transmitting and receiving amplifying our signals. We will fabricate an antenna with a 3dB gain such that our distance requirement is met.

$$P_r = P_t + D_t + D_r + 20\log_{10}\left(\frac{\lambda}{4\pi d}\right)$$

Will be used to calculate the antenna specifications needed. P_r is the power available at the receiving antenna, and P_t is the power delivered. D_t and D_r are the directivities of the two antennas. The d is the distance between the antennas, what we will solve for, in order to fabricate our antenna.

2.3 Risk/Tolerance Analysis

For our project, one of the biggest risks will be the microcontroller. The microcontroller will have multiple jobs and must not bottleneck the entire device. First and foremost, the microcontroller will have to create a single data packet from sensor information and from the stored information about the other device. Simultaneously, it will have to constantly collect all new data packets that the receiver picks up and process it accordingly. It will have to decrypt the signal, process the data packet into individual measurements, and then store the relevant data. There will be synchronization issues that the microcontroller must handle (data packets being sent require reads, data packets being received require writes, and sensor data recorded requires writes). On the microcontroller, we will use hardware accelerator for our encryption/decryption, but this will add another layer of complexity to the synchronization issues we will have to overcome. We plan to program our microcontroller in C so we can handle the aforementioned risks and optimize synchronization for the different processes that need to occur.

2.4 Stretch Goals

One of our stretch goals is to incorporate a key exchange protocol for the encryption. Without this key exchange protocol, there exists the risk of an attacker “figuring out” the agreed upon key between the two devices and intercepting the information being sent.

Another one of our stretch goals is to shift our device from a P2P communication to a network-based communication. With this type of communication, each device would not only broadcast its own information, but also the information of all the devices it has information about.

This would allow data of an individual car to reach miles away in relatively dense areas. With so much data being 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 get 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, most likely 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 form this network of, we would also implement a dynamic channel allocation (DCA) system such that with a limited frequency range, we could communicate with a large number of nodes. The idea would be that one node would not be limited to broadcasting on a certain channel, but rather whichever channel would be available at the time of the broadcast. This would allow more nodes than existing channels on the network.

Another stretch goal is to process and interpret the data we collect. Currently the device is designed to collect information about itself and information from another node. However, we are not processing the data and converting it to useful information, so this would be our next step. For example, if an otherwise non-anomalous device suddenly experiences a sharp negative acceleration, we can assume with some certainty that the car got in an accident (or some car ahead was in an accident, depending on the level of acceleration and information from other sensors). From here, the car that crashed could give a brief emergency broadcast to let cars in the vicinity know what happened, and other cars could use this information accordingly. Another example would be if a vehicle behind is accelerating or moving at unsafe speeds, drivers in the vicinity could be warned to avoid that vehicle.

A final stretch goal for this project would be to interface with the car directly rather than use sensor data. The OBD-II port in vehicles export data about the vehicle such as acceleration, emissions, velocity, etc. This would allow for more information to be transmitted between the devices and would encompass a more holistic V2V communication network.

3 Cost and Schedule

3.1 Labor

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

3.2 Parts

Each singular device will contain these parts, so multiple units will need this assortment each.

Description	Manufacturer	Part #	Quantity	Cost per unit ($)
Voltage Regulator	Diodes Incorporated	AP65111AWU-7DICT-ND	1-2	0.44
Accelerometer	Kionix	KXTJ3-1057	1	1.69
GPS device	GlobalTop	FGPMMOPA6H	1	39.95
Inertial Measurement Unit	STMicroelectronics	LSM6GSOXTR	1	4.76
Microcontroller	Espressif Systems	ESP32-WROOM-32D	1	3.80
Transceiver	Analog Devices Inc.	ADF7021-VBCPZ	1	7.90
TOTAL				58.54

Grand Total Cost = Labor + (Parts * Number of devices) = $48,117.08

3.3 Schedule

Week	Alejandro	Harsh	Tiger
9/30	• Finalize design • Order parts	• Research and design	• Research and design
10/7	• Test sensors, transceiver	• Test microcontroller	• Design PCB
10/14	• Establish sensor to controller and controller to communications connections	• Configure subsystem connections	• Test subsystems together

10/21	• Develop prototype • Test in a vehicle	• Develop prototype	• Test PCB
10/28	• Integrate and test other features	• Make changes	• Make Changes
11/4	• Refine prototype • Test in a vehicle	• Refine prototype	• Make changes • Order new PCB
11/11	• Finalize the device • Work on reach goal(s) • Prepare for mock demo	• Finalize device • Prepare for demo	• Finalize device • Practice demo
11/18 (Mock Demo)	• Work on final paper • Adjust to mock demo suggestions	• Work on final paper • Practice demo	• Work on final paper, Practice demo
11/25 (Fall Break)	• Work on final paper • Adjust to mock demo suggestions	• Work on final paper • Practice final demo	• Work on final paper Practice final demo
12/2 (Final Demo)	• Work on final paper, final presentation	• Work on final paper, final presentation	• Work on final paper, final presentation
12/9 (Final Presentation)	• Work on final paper	• Work on final paper	• Work on final paper

4 Ethics and Safety

Safety is the essence of our project. One of the sole purposes of vehicle to vehicle communications is to lower risks and increase safety in all types of vehicles, especially since failures in this type of technology can lead to fatal accidents in the real world. Our goal in this prototype is to minimize packet loss to less than 1%. Meaning that 99/100 data packets that are transmitted are successfully received. Furthermore, 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 about twice the average human reaction time to visual stimuli [12] (giving users sufficient time to react to notifications).

Although the device has protective functions embedded in its components that prevent damage from overvoltage surges and undervoltage conditions, there remains some potential safety hazards in the device [5]. In the case of undervoltage and reverse polarity, the device will be protected but will not function as expected. This may lead to safety concerns for systems that rely on this device, such as other nodes or drivers.

Unexpected temperatures and high electromagnetic fields can also be potentially harmful to our system, not only to the PCB fabrication, but also to the sensors that we use [5]. To prevent this, we will monitor temperature usage regularly and make sure our device is capable of withstanding above average levels of magnetic fields that vehicle components are exposed to. Furthermore, we are also ensuring that all components of our device pass AEC-Q200 standards for ruggedness and durability of vehicular components.

Naturally, with any sort of network where information is exchanged, one of the key ethical concerns is that of privacy. Since we are creating this technology, we are responsible for its usage, as specified by IEEE Code of Ethics, #5: “To improve the understanding of technology; its appropriate application, and potential consequences” [1]. There runs the risk of malicious forces using the data that these devices collect for alternative means. For this reason, we plan to encrypt all data that goes through our network with industry standard AES-128 bit encryption. Furthermore, we will limit our processing to only handle the data we allow, so any other transmitted information would not be processed into readable data.

References

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

[2]*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].

[3]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].

[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]"Power Supplies.", Omron, 2019. [Online]. http://www.ia.omron.com/product/cautions/22/safety_precautions.html. [Accessed: 19- Sep- 2019].

[6]Diodes Incorporated, "TSOT26 Light Load Improved 1.5A Synch DC/DC Buck Converter," AP65111A datasheet, June 2017.

[7]Kionix, "±2g/4g/8g/16g Tri-axis Digital Accelerometer Specifications," KXTJ3-1057 datasheet, [Revised Oct. 2018].

[8]GlobalTop Technology Inc., "FGPMMOPA6H GPS Standalone Module Data Sheet," FGPMMOPA6H datasheet, Jan. 2012.

[9]STMicroelectronics, "iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope," LSM6DSOX datasheet, Dec. 2018, [Revised Oct. 2018].

[10]Espressif Systems, "ESP32-WROOM-32D Datasheet," ESP32-WROOM-32D datasheet, Nov. 2017, [Revised Sept. 2019].

[11]Analog Devices, "High Performance, Narrow-Band Transceiver IC," ADF7021-VBCPZ datasheet, April 2010, [Revised Sept. 2014].

[12]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].