Road Interference Mapper (The RIM)

Team 35- Euiho Jung, Minh Le, Zhiyuan Zheng
ECE 445 Design Document - Fall 2019
TA: Kristina Miller

1. Introduction

1.1 Problem and Solution Overview

In many places, there are multiple roads you can take to get to your desired destination within a similar amount of time. Especially in many metropolitan cities where the road layout is a grid, there are countless paths to a given destination. However, in many places certain roads are littered with potholes and debris. These roads with many interferences would ideally be avoided but there is currently no way to tell whether or not a road contains a large number of potholes and debris. Even government surveyors have no real way to find and count potholes besides manually counting them.

To allow for better driving path planning and pothole detection, our product will allow government road surveyors to attach a device to their vehicle that will detect the number of interferences on the road and send that data to a server. Consumers can then view this data on their own smartphones using our app. By periodically sensing the average distance to the road underneath a car using ultrasonic sensors, any deviation in this distance within a certain threshold would signify the presence of potholes or debris. A microcontroller unit (MCU) will periodically process the distance data and, upon detection of an obstacle, send a signal to a smartphone app notifying the app of a detected interference. The smartphone app would then increment the number of interferences the surveyor has detected on the current road. This information would then be sent to a server where the displayed number of interferences on each nearby road would be an average of all of the surveyor data on a given road. All consumers using this app would be able to see the average number of interferences on each road that the surveyors detected. However, we believe the server portion of this system is out of the scope of this class. Because of the limited reading rate of ultrasonic sensors, our solution will only detect debris and potholes when the user is driving under 9.5 mph, which is why the detection portion of our system will only be available to government surveyors.

Other applications aim to detect potholes and debris, however, these applications utilize the sensors of a smartphone such as the accelerometer and the gyroscope to detect the impact with an obstacle. This method can only detect obstacles that were hit. Our solution will utilize an array of sensors to detect roadside obstacles. This will allow for more accurate results and more importantly, our solution will detect obstacles, such as potholes, that a driver may have driven over without collision.

1.2 Visual Aid

The RIM’s hardware components will be encased by a sheet of speaker grill cloth and attached to the bottom of a motor vehicle’s bumper using an adhesive.

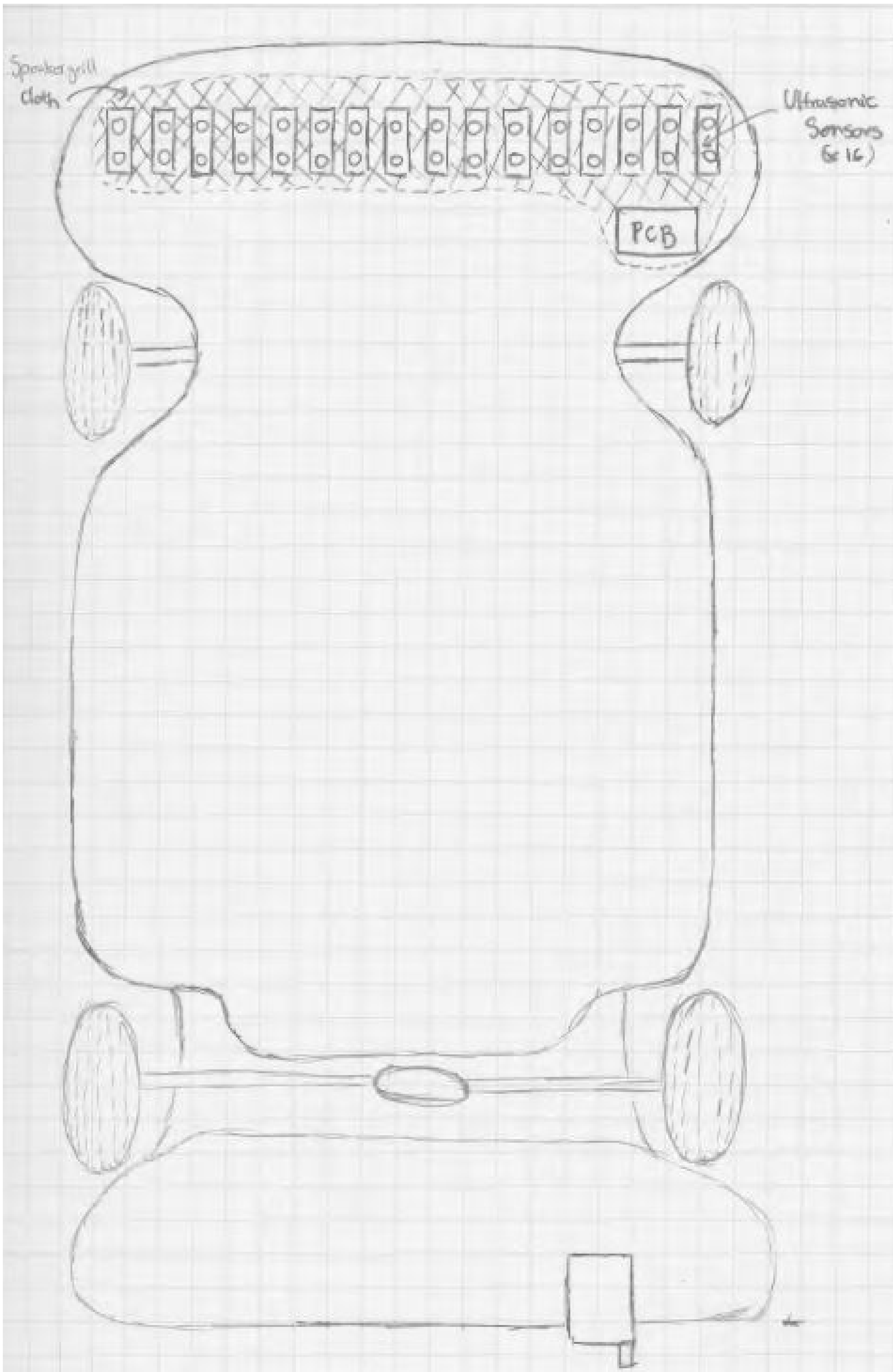


Figure 1.2.1 Car configuration bottom view

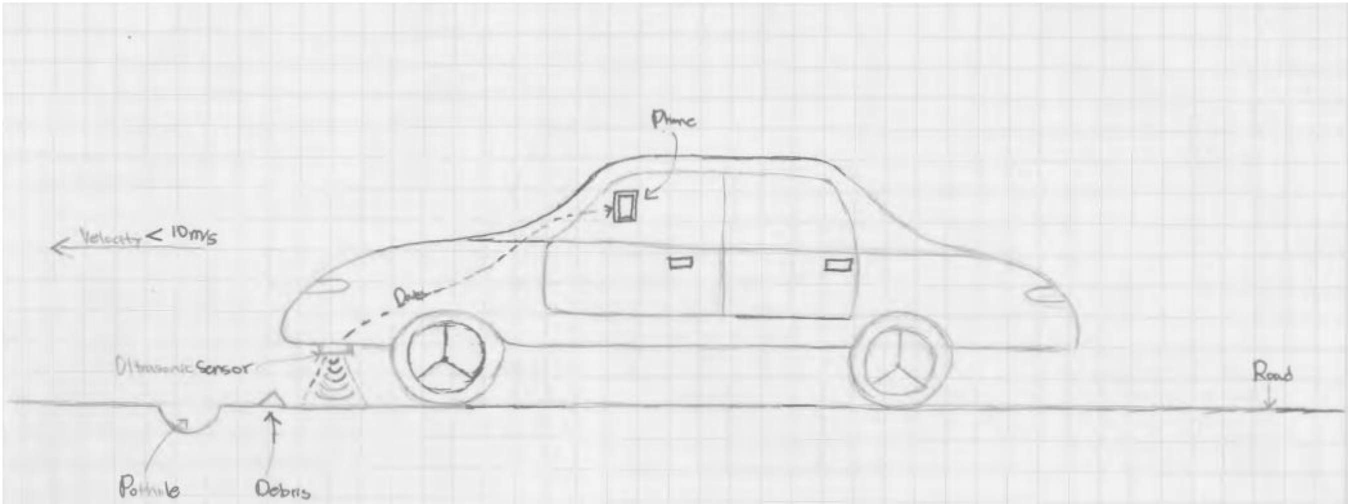


Figure 1.2.2 Car configuration side view

1.3 High-level Requirements

- Sensors must be able to detect potholes and debris of at least 0.25 m in length, 0.25 m in width, and 0.04 m deep.
- The RIM counts potholes and debris of within +/-2 potholes and debris
- The RIM can detect potholes and debris at speeds up to 9.5 mph

.25m was chosen for the dimension of potholes based on the most popular tire sizes of 2018.[1]
.25m is the average width of the top 10 passenger car tires.
.04m is the average minimum depth of a pothole, below which United Kingdom councils won't service or asses[2].
9.5mph is the maximum speed an ultrasonic sensor unit sampling at 17hz can move and still obtain readings every 0.25m (see risk analysis for proof).

2. Design

2.1 General Block Diagram

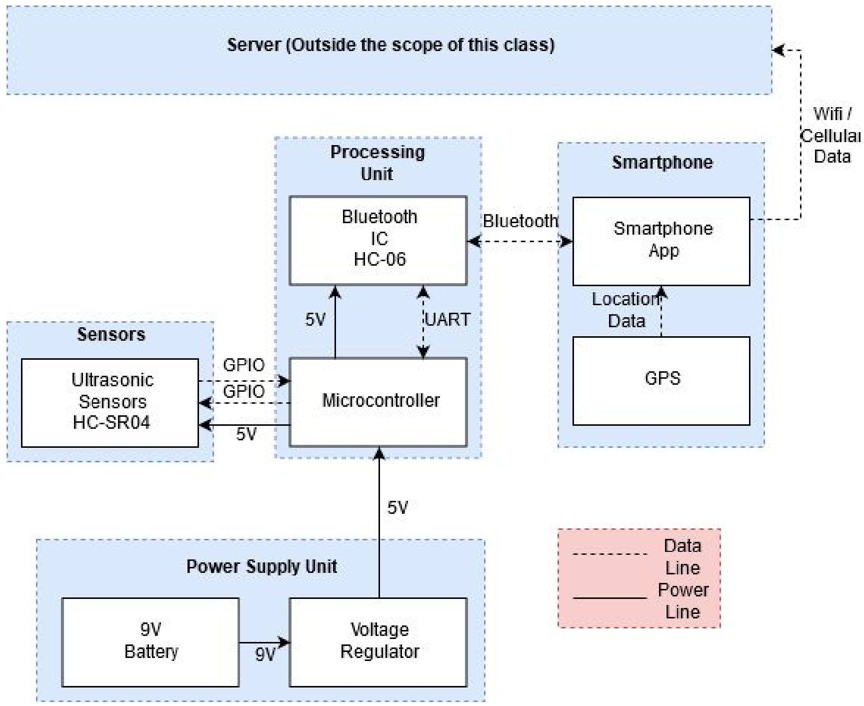


Figure 2.1. Block Diagram

The RIM requires five main electrical components for operation as shown in Figure 1. First, a 9V battery is needed to supply power to a processing unit at all times. Second, ultrasonic sensors are required to detect potholes and debris on roads at speeds within 9.5 mph. Third, we need a MCU to power the ultrasonic sensors and Bluetooth module at 5 V, and to process all data from the sensors and Bluetooth module. Fourth, a Bluetooth module will be used as a bridge of data to process it between the MCU and a phone. And fifth, a phone/software is needed to exchange data with Bluetooth module, detect any sudden bumps by using the on-phone accelerometer, and mark the location of potholes and debris for users.

The power supply system is on at all times to provide power to the MCU and the Bluetooth module. However, the sensors will be turned on, when the app is turned on by the user. Therefore, the data from both the sensors and the phone will be delivered to the MCU, such as the time difference between when the trigger signal is sent out from the sensors and when the echo signal is received to the sensors. When the calculated distance between the sensors and the ground in the MCU is over a specified threshold, the signal via Bluetooth module will be marked on the app at the location sent from the GPS. In addition to the data from the sensors, the accelerometer in the phone is used to detect any direct hit by the tires on the roads.

2.2 Physical Design

For the purpose of demonstrating The RIM's functionality in an indoor environment, a scaled down model of the product will be attached to the bottom of a rolling chair pictured in figure 2.2.1 and 2.2.2. The scaled down model will be rolled along an artificial terrain containing various obstacles and potholes. The sensors and PCB will be placed inside a rigid case with an opening on the side the ultrasonic sensors are facing. This opening will then be covered by speaker grill cloth, an acoustically transparent material.

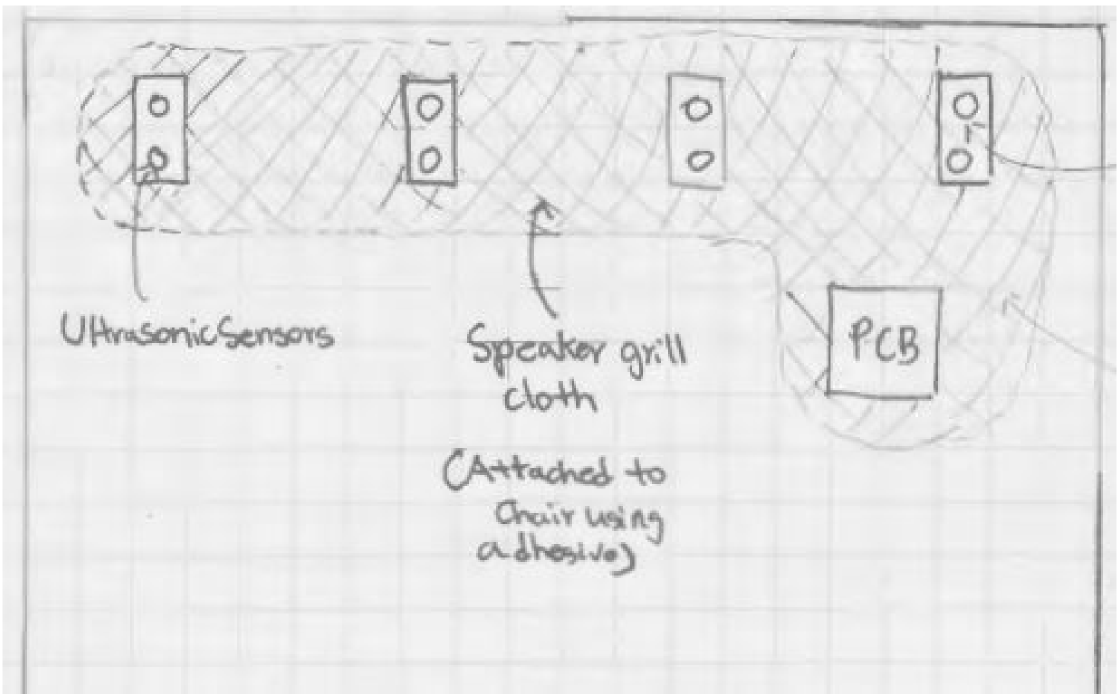


Figure 2.2.1 Demonstration configuration bottom view

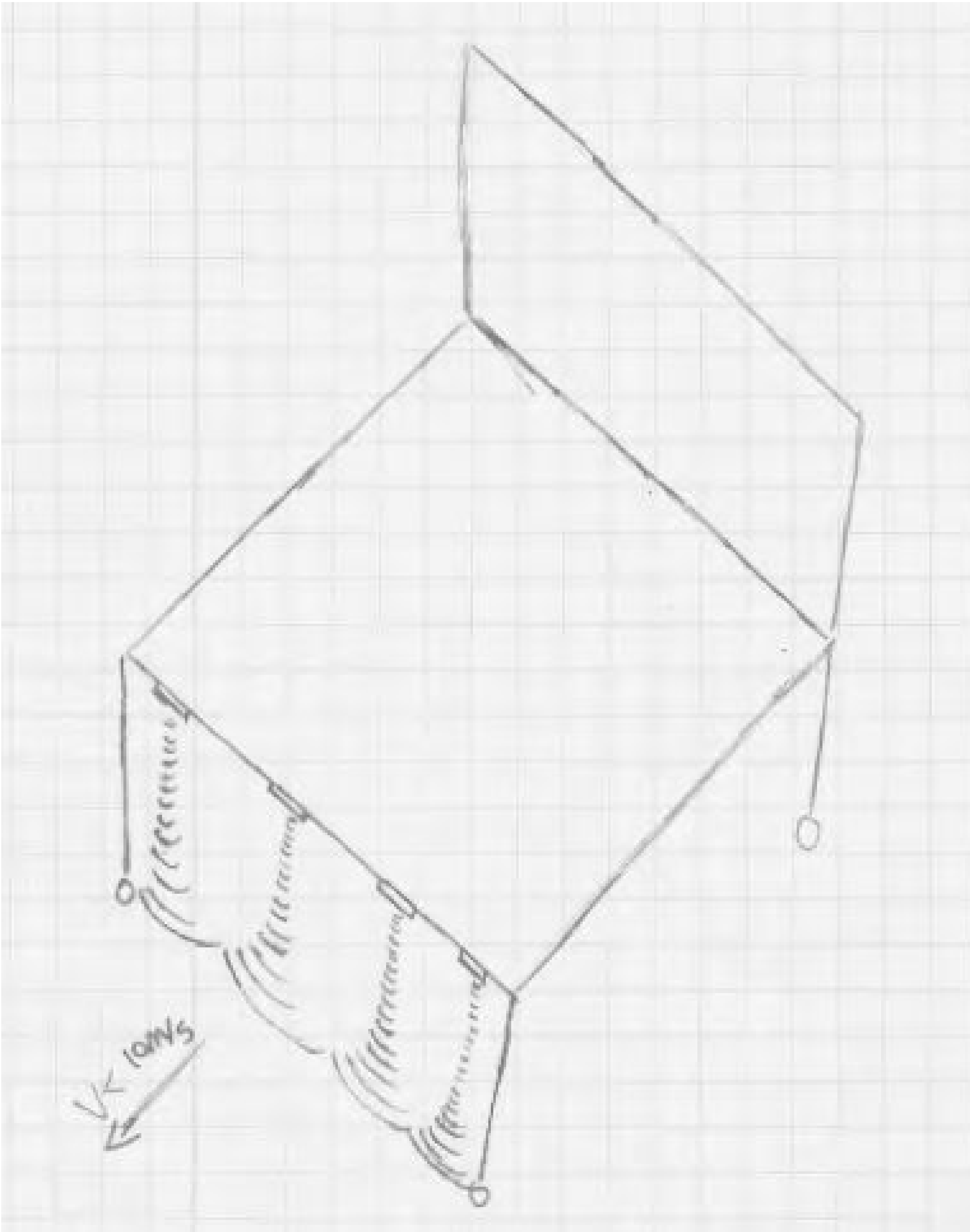


Figure 2.2.2 Demonstration configuration top view

2.3 Power

A 9 V battery supplies power to the processing unit through a voltage regulator, which in turn supplies power to the bluetooth module and ultrasonic sensors.

2.3.1 9 V Battery

A 9 V battery with a polarized snap connector at the top connects to the voltage regulator via a battery clip with bare leads.

2.3.2 Voltage Regulator (LM1117IMPX-5.0/NOPB)

As this voltage regulator is DC-DC converter and linear with high efficiency, it converts a 9 V input to a 5 V output which supplies power to the V_{in} pin on the microcontroller unit.

Requirements and Verification Table for Power System

Requirements	**Verifications**
1) The input of the LDO voltage regulator must sink 9 V+/-67%. 2) The output of the LDO voltage regulator must source 5 V +/-1%.	1) Using a digital multimeter: a) Connect the red port of the dmm to the input pin of the voltage regulator. b) Connect the black port of the dmm to the ground pin of the voltage regulator. c) Check the voltage reading is 9 V +/- 67%. 2) Using a digital multimeter: a) Connect the red port of the dmm to the output pin of the voltage regulator. b) Connect the black port of the dmm to ground pin of the voltage regulator. c) Check the voltage reading is 5 V +/- 1%.

Table 2.3.1. R&V Table for voltage regulator

2.4 Sensors

Ultrasonic sensors measure the distance between the vehicle they are mounted underneath and the surface below it. These measurements are sent to the MCU from which the sensors also draw operating power.

2.4.1 Ultrasonic Sensor

Ultrasonic sensors, chosen to be HC-SR04, send out trigger signals taking on the form of ultrasound(40kHz) and receive echo signals. The sensors input the time spent during the trigger signals are sent out and the time the echo signal is received to the MCU.

Requirements and Verification Table for Sensors

Requirements	Verifications
1) The distance between the ground and the underside of the vehicle the sensors are mounted underneath must be between 2 cm-400 cm. 2) The sensors should achieve a sampling rate of at least 10 Hz at a distance of 1m from the surface.	1) Measure distance to the ground from sensors with a meter stick or ruler to ensure values are within the specified 2 cm-400 cm range. 2) Verification for Item 2: d) Load test program on to an Arduino Uno to probe and read the data from a single ultrasonic sensor. e) Connect a laptop to monitor the sampling rate of the sensor. f) Connect one ultrasonic sensor to the Arduino Uno. g) Connect power to the Arduino Uno h) Observe the sample rate is at least 10 Hz using the test program with the surface 1m away.

Table 2.4.1. R&V Table for Ultrasonic Sensors

2.5 Processing Unit

The processing unit performs the required calculations, using the distance sensor data, to determine whether or not an interference is detected. This unit consists of a MCU(Microcontroller Unit)(ATMEGA328P-AU), a bootloader module (FT232RL), a Bluetooth module(HC-06), and a mini-B USB port. After performing the required calculations, the MCU will send a signal to the bluetooth module in order to communicate the existence of an interference to the smartphone app

2.5.1 MCU

The MCU will handle the reception of the ultrasonic sensor signal, as well as the calculation to determine whether there is an obstacle at the current position. The MCU will be coded via the bootloader module to handle the reception of the ultrasonic sensor signal, as well as the calculation to determine whether there is an obstacle at the current position. The MCU will exchange the required signals with ultrasonic sensors to measure the current distance to the road; each sensor is powered by the MCU at 5V. This will allow the MCU to depower the sensors when the smartphone app is not in use. After the MCU detects an obstacle or debris on the road, it will send a signal to the app through the offboard Bluetooth module signifying a road interference at the current position. The smartphone app will then increment the number of interferences on the current road. The MCU will also receive signals from the app through the offboard Bluetooth module to put the MCU to sleep or wake it up. Upon being put to sleep, the MCU will cut off power to the ultrasonic sensors and enter a low power state. In this state, the MCU will wait for another signal from the app to wake up.

Requirement and Verification Table for MCU

Requirements	Verifications
1) The MCU should receive distance data from the ultrasonic sensors. 2) The interference detection algorithm on the MCU should detect the number of interferences on a given road within +/- 2 interferences 3) The Bluetooth chip should receive all signals from the MCU and send them to the smartphone app within 2 seconds +/- 5%. 4) The Bluetooth chip should receive all signals from the smartphone app and send them to the MCU within 2 seconds +/- 5%.	1) Verification for Item 1: a) Connect each ultrasonic sensor to the MCU circuit. b) Connect power to MCU and ultrasonic sensors. c) Create a test program to send and receive the required signals to get distance data from the ultrasonic sensors. d) Ensure accurate distance data is being received with a ruler. 2) Verification for Item 3: a) Load interference detection algorithm onto the MCU. b) Connect each ultrasonic sensor to the MCU circuit. c) Connect power to MCU circuit and ultrasonic sensors. d) Connect a laptop to MCU circuit to monitor results. e) Move ultrasonic sensor array over a known number of interferences and count how many are actually detected. f) Ensure the number of actually detected interferences are within +/- 2 interferences of the true value.

Requirement and Verification Table for MCU (con)

	3) Load a test program onto the arduino uno. Attach the Bluetooth chip to the correct pins on the Arduino Uno. Using a test smartphone app, attempt to send data from the arduino to the smartphone and measure the time it takes with a stopwatch. Ensure the time is within 2 seconds +/- 5%. 4) Load a test program onto the arduino uno. Attach the Bluetooth chip to the correct pins on the Arduino Uno. Using a test smartphone app, attempt to send data from the smartphone to the Arduino and measure the time it takes with a stopwatch. Ensure the time is within 2 seconds +/- 5%.

Table 2.5.1. R&V for MCU

2.5.1.1 Schematics

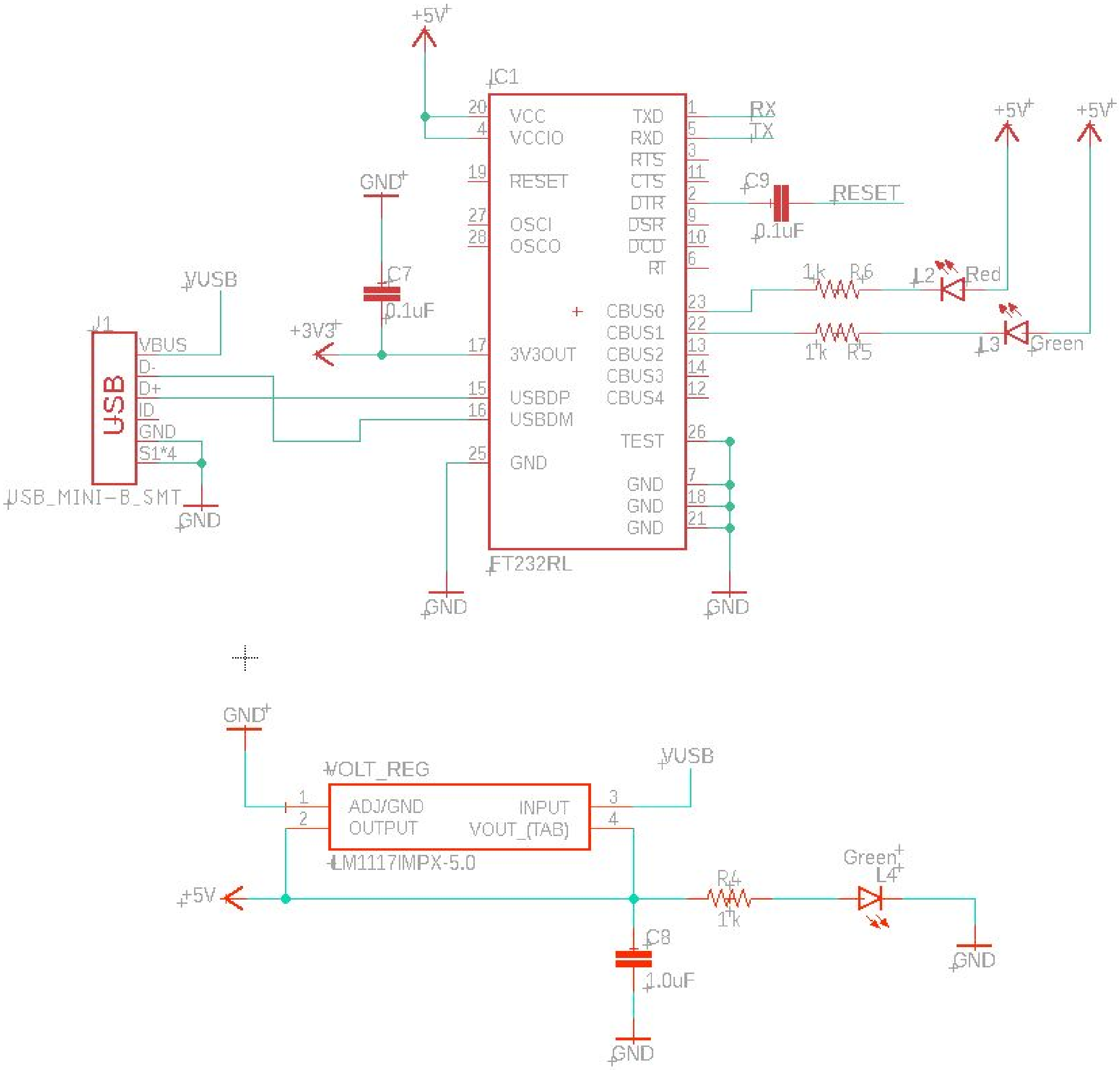


Figure 2.5.1.1. Schematic of MCU-1

As shown in Figure 2.5.1.1, by connecting a computer to the bootloader(FT232RL) via the mini-B usb port, our MCU can be coded in computer language. The voltage regulator regulate the 9V voltage supply to 5V which satisfies operating voltage of all the modules used in this project.

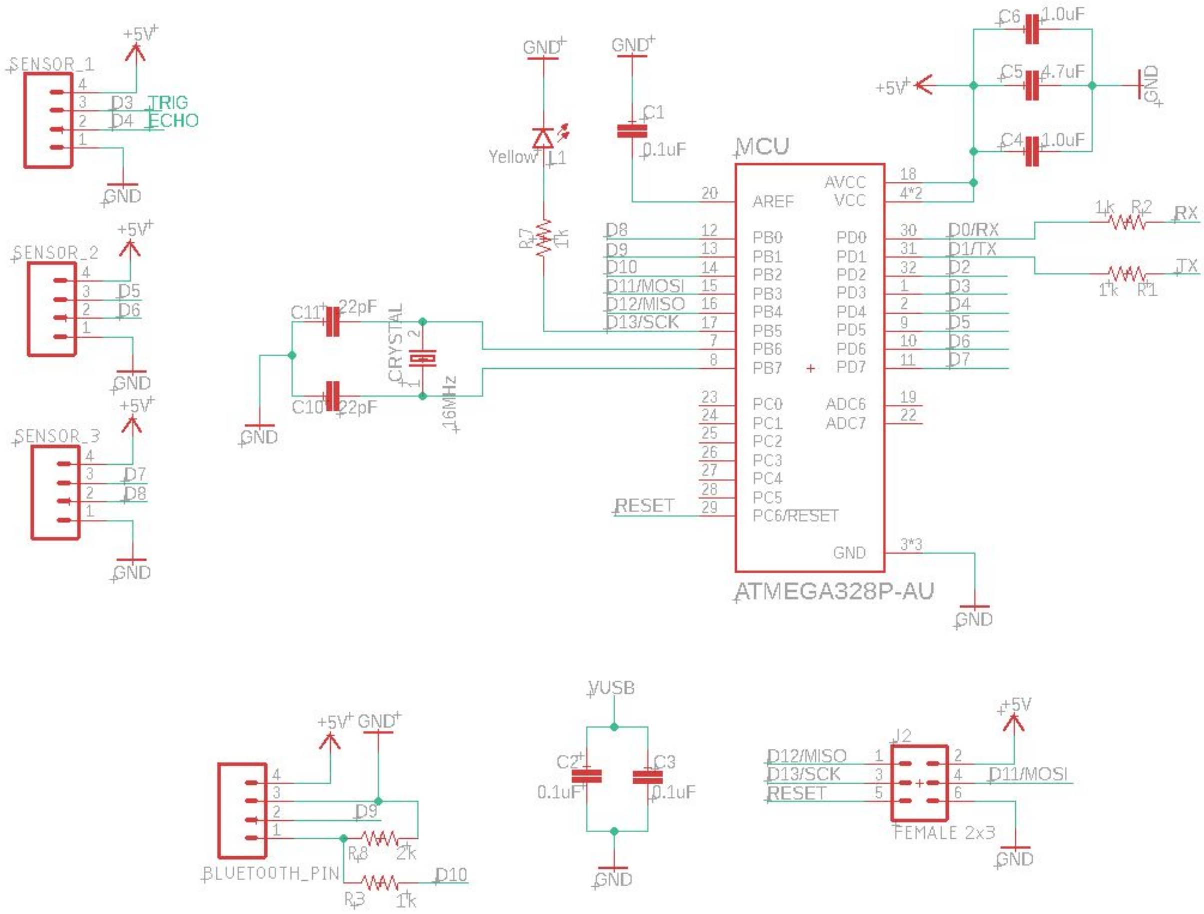


Figure 2.5.1.2. Schematic of MCU-2

As shown in Figure 2.5.1.2, The MCU is to communicate with ultrasonic sensors and the Bluetooth module. The crystal connected to pin 7 and pin 8 on MCU generates the 16MHz clock. The left three connectors are to connect sensors to the MCU and the Bluetooth pin connector is to connect the Bluetooth module to the MCU for data transmission via digital pins on MCU. Voltage divider is used for the Rx pin for the Bluetooth module as the Rx pin accept only 3.3V. The MCU read 3.3V as HIGH, Tx pin can be directly connected to the MCU. The bypass capacitor at the center of the bottom is to short the AC signals to ground.

2.5.2 Bluetooth Module

The HC-06 Bluetooth module will allow the MCU and the smartphone app to communicate. The Bluetooth module will send a signal to the app from the MCU upon the detection of a roadside obstacle. The Bluetooth module will also receive a signal from the app and transmit it to the MCU to wake up or put it to sleep depending on if the app is running. It will operate with the Bluetooth 2.0 protocol at an operating frequency of 2.4 GHz in the ISM frequency band[4].

2.6 Smartphone

The smartphone, chosen to be an Android smartphone, sends and receives information from the MCU through bluetooth. It will connect with the Bluetooth module chip in the processing unit to achieve this. It will operate with the Bluetooth 2.0 protocol at an operating frequency of 2.4 GHz in the ISM frequency band[4]. The smartphone module consists of a smartphone app and the GPS unit within the smartphone.

2.6.1 Smartphone Application

The smartphone application will act as an interface for which the user can see the number of potholes and debris on each road around them. The app will constantly display a map to the user, showing the current location of the user as well as markers on each nearby road displaying the number of potholes on that road. An open source map library, osmdroid[8], will be used to draw our maps and pothole and debris count markers. Fig 2.6.1 shows a rough draft of what the user interface will look like. The smartphone app will send a signal on boot up to the MCU using Bluetooth to wake up the MCU. First the app will check if the current road has changed using the GPS capabilities of the smartphone as well as the open source map library. If the road has changed, the app will update the number of potholes and debris on the current road the user is on. Then, the app will update the nearby roads along with the number of potholes and debris on each one. Next, the app will check if the MCU is sending a signal through Bluetooth, signifying the presence of an interference at the current location. If this signal is detected, the number of interferences on the current road will increment by one. After this the app goes back to checking the current road and continues the cycle until the closure of the app. The number of potholes and debris on each road would be sent up to a server where all the users in the network would be able to see all of the interferences detected in the system. This would also allow for greater accuracy in detecting the correct number of potholes on the road if an average of all users data was used. However, we believe the server side of this system is out of the scope of this class, and therefore will not be implemented. Finally, upon closing the app, a signal will be sent to the MCU through Bluetooth in order to put the MCU to sleep to save power. Fig 2.6.2 shows a flowchart describing the process of the app. Any of the states in Fig 2.6.2 will be taken to a pause state upon closing the app. This state will send the signal to the MCU to put it to sleep before fully closing the app.

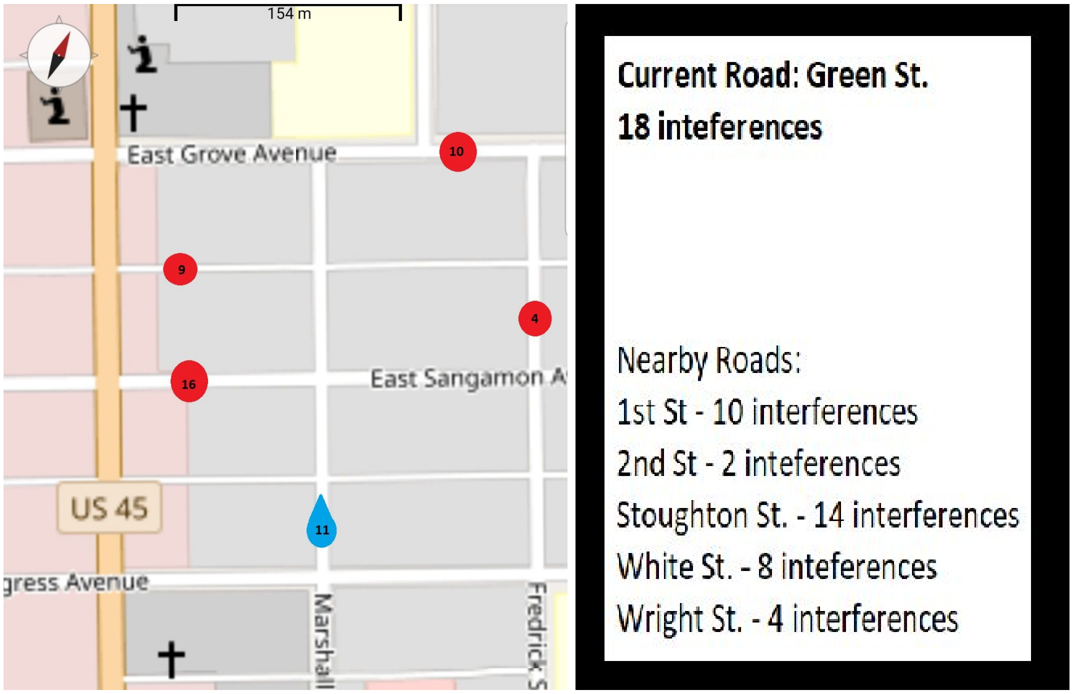


Figure 2.6.1. Left: Example UI Ideal Finished Product, Current location is the blue arrow, number of obstacles are the red circles[8]
Right: Example UI of Project Prototype, In demo the nearby roads will not be listed and the street name will be nothing.

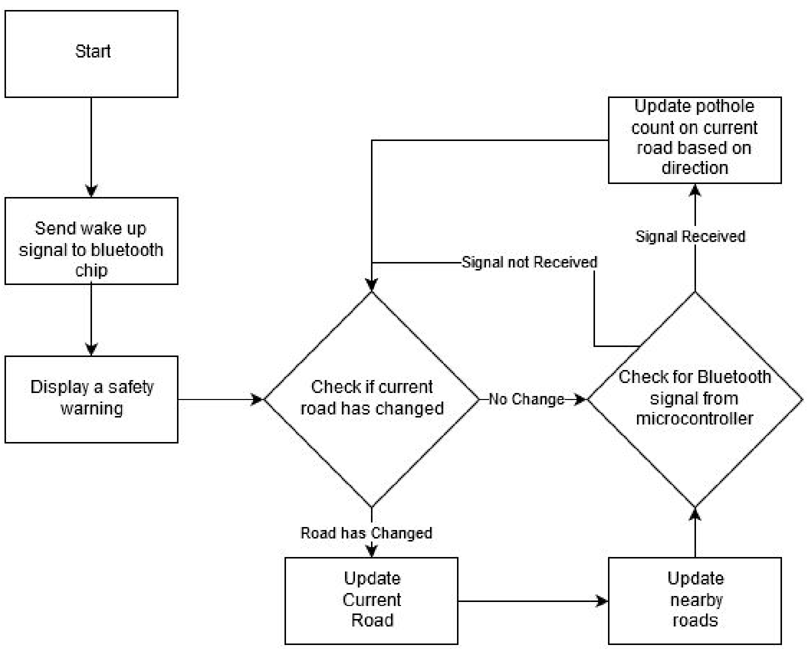


Figure 2.6.2. Flowchart for smartphone app

Requirements and Verification Table for Smartphone

Requirement	Verification
1) The smartphone app should process signals and update the number of interferences on the current road within 4 seconds of receiving the signal. 2) The smartphone app should send a signal to the MCU through bluetooth to enter sleep mode upon closure of the app. 3) The smartphone app should receive and recognize the Bluetooth signal from the MCU within 1 second. 4) The smartphone app should update the current road within 2 seconds of changing roads at speeds of up to 10 mph. 5) The GPS data must be able to tell the current road and nearby roads within 100m.	1) Create a test app that marks the location of an interference without using a Bluetooth signal. Measure the time from start to the updating of the interference count with a stopwatch. Verify the time is less than 4 seconds. 2) Create a test app that sends a bluetooth signal to the MCU upon closure. Observe that a signal has been received on a connected PC. 3) After implementing all of the parts of the main loop of the app, display which box in the flowchart we are currently in. Then, send a bluetooth signal from the microcontroller and measure how long it takes to go to the signal detected state. Verify the time is less than 1 second. 4) Create a test app that displays the current road. In a real car, drive around and change roads with the app open. Have a passenger measure the time it takes for the current road to change with a stopwatch and ensure the time is below 2 seconds. 5) Create app to tell the name of the current road and names of the nearby roads within 100m in a known area. Verify the results are correct.

Table 2.6.1. R&V for Smartphone Subsystem

2.6.2 Smartphone GPS Unit

We will use the Android Smartphone GPS module to track the user's current location as well as the current road they are on. The GPS module will also give the required data to find the names of nearby roads within 100m of the user’s current location.

2.7 Tolerance Analysis

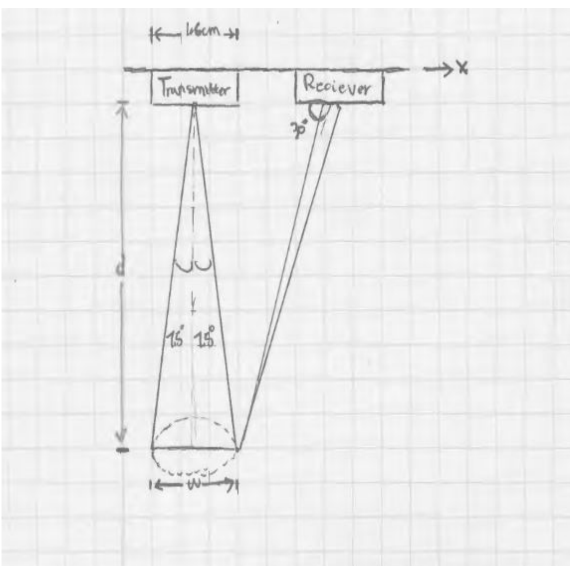
The operation of The RIM depends on the accuracy of its ultrasonic sensors which are affected by the sensors' measurement spread and sampling frequency as well as the lifetime of the battery from which The RIM draws power.

2.7.1 Ultrasonic Sensor Spread

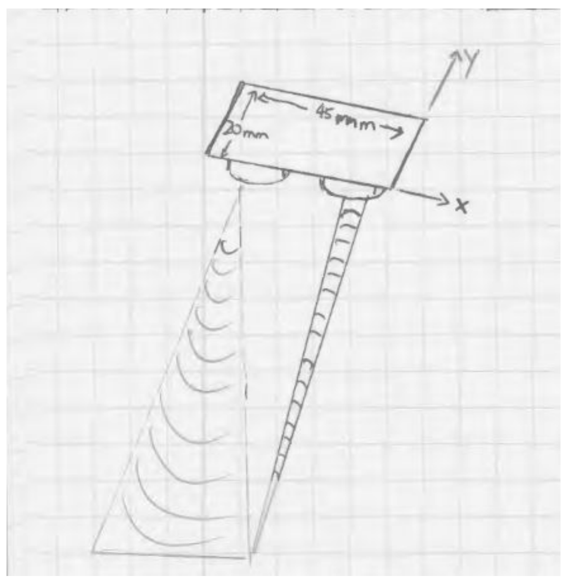
For The Rim to count all interferences underneath the vehicle it is mounted on, its sensors' must be able to detect them. Given a transmission signal angle of θ=15°(illustrated in fig 2.7.1 and fig 2.7.2) and a sensor to ground distance of d=0.4m (measured height of demonstration chair), a single ultrasonic sensor unit will cover a horizontal spread of w=10cm on an even surface (eq. 2.7.1).

$$w = 2sin(\frac{\theta}{2})\frac{d}{cos(\frac{\theta}{2})}$$

Sensor spread equation 2.7.1

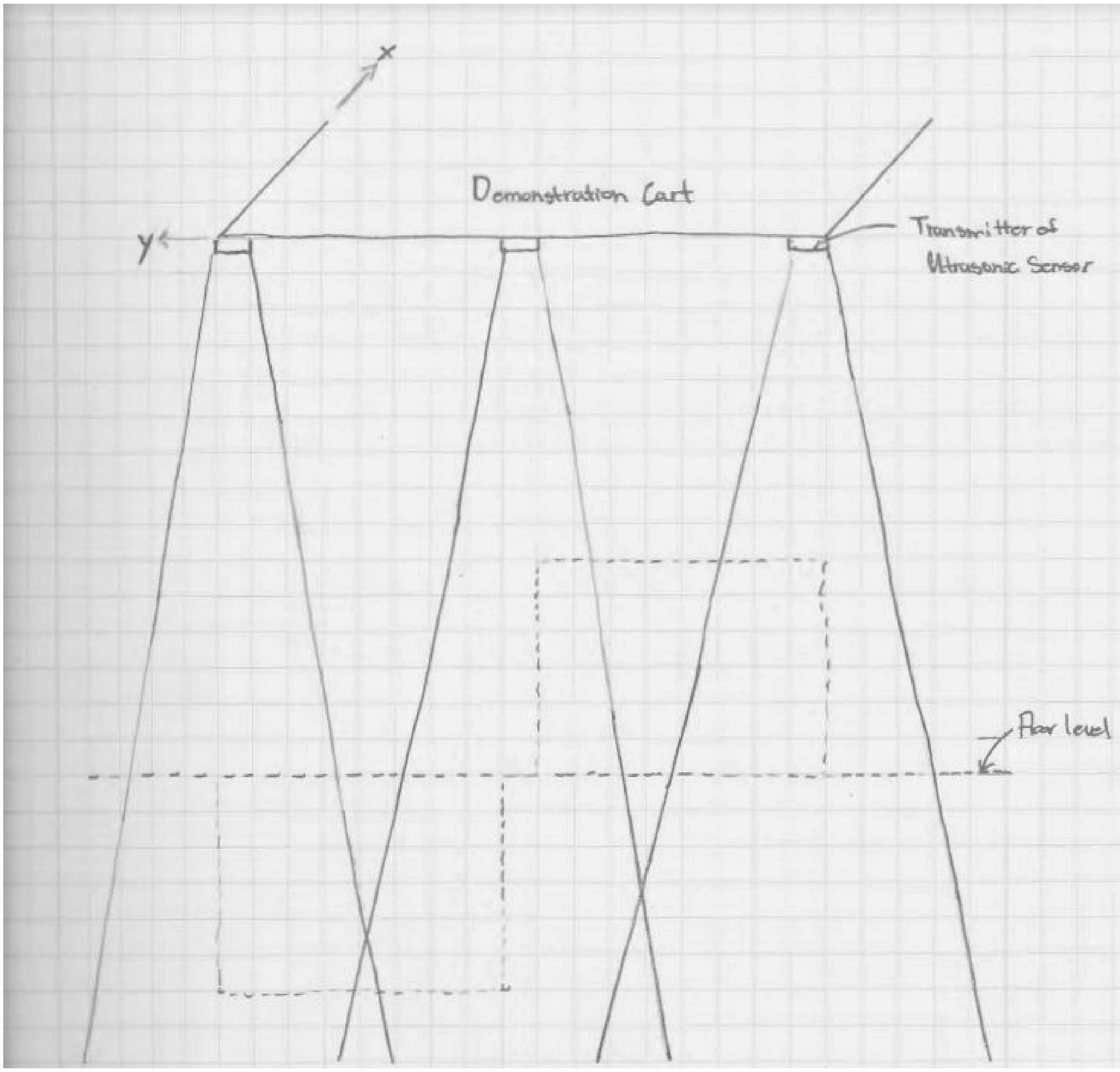


Ultrasonic sensor side view Figure 2.7.1



Ultrasonic sensor top view Figure 2.7.2

Three sensors will be configured to maximize the spread of the sensor by placing one at the leftmost edge of the demonstration cart, one at the rightmost edge of the demonstration cart, and one directly in the middle. In this configuration (figure 2.7.3), any interference with a diameter of 0.25 underneath the chair, whether it's above or below the floor level in elevation, will be detected. When The RIM is applied to a real vehicle, more sensors will have to be added to account for the increased width.



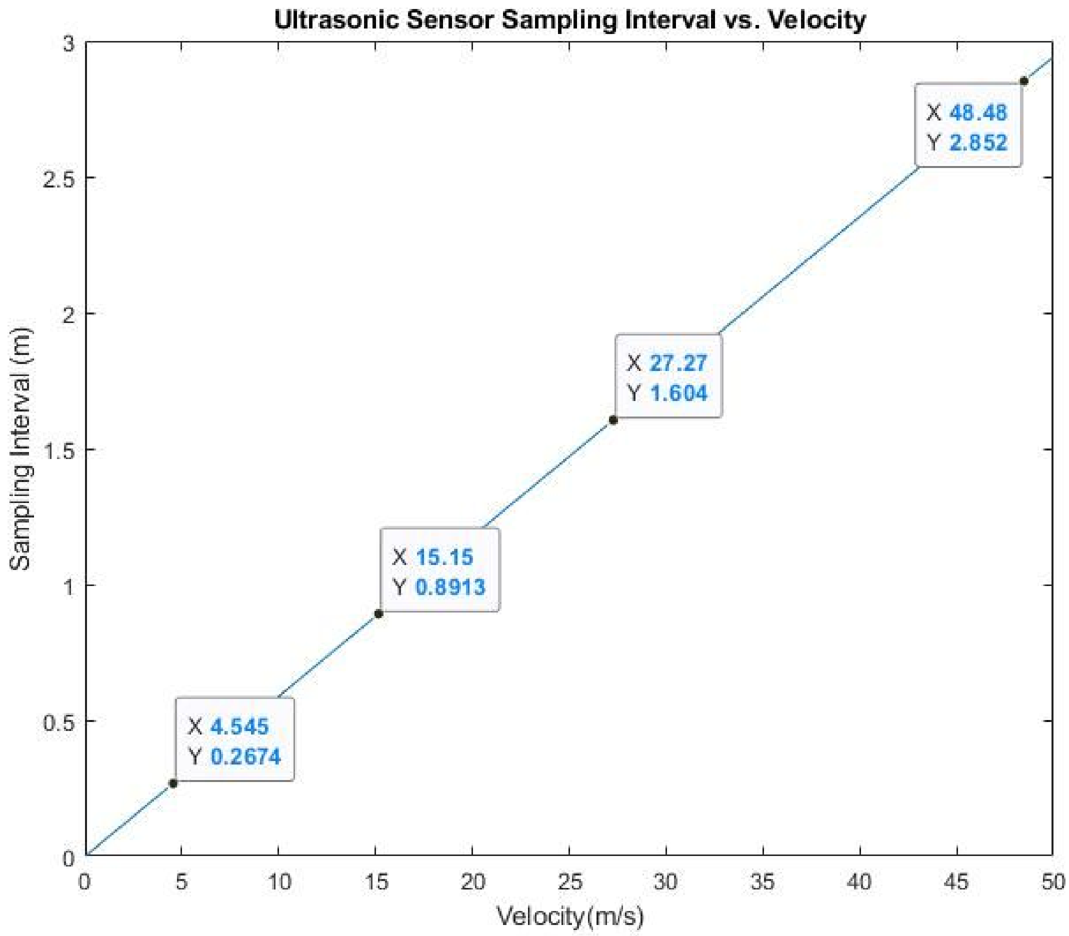
Sensor Configuration Figure 2.7.3

2.7.2 Ultrasonic Sensor Sampling Rate

The HC-SR04 has a suggested measurement cycle of 60ms [3] which corresponds to a sampling frequency of 17Hz. If we aim to obtain a ground-distance reading every 0.25m, an ultrasonic sensor sampling at 17hz will only be able to detect interferences under a car moving up to a velocity of 4.25 m/s (eq. 2.7.2). If a pothole or piece of debris has a width that is under the 0.25m threshold while the car is moving 4.25m/s, it will go undetected. It is hard to precisely quantify the size of an interference that will damage a car, but the limitations of detection for The RIM based on the velocity of the vehicle it is attached to illustrated in figure 2.7.2 will illustrates its operational constraints.

$$SamplingInterval(m) = \frac{Velocity\frac{m}{s}}{SamplingRate} = \frac{Velocity\frac{m}{s}}{17hz}$$

Sampling Interval vs Velocity Equation 2.7.2



Sampling Interval vs Velocity Graph Figure 2.7.2

2.7.3 Battery Lifetime

$$\text{Battery Lifetime} = \frac{\text{electric charge(mAH)}}{\text{current draw of PCB(mA)}}$$

Battery Lifetime Equation 2.7.3

Operating current draws of each module are 12mA (USB UART IC), 14mA (MCU), 10mA (voltage regulator), Bluetooth module (8mA), 15mA (ultrasonic sensor), respectively. Therefore, the operating total current draw is 62mA. As the 9V battery capacity in continuous discharge at 50mA current draw is 400mAH, the battery lifetime is about 8 hours.

3. Cost and Schedule

3.1 Cost Analysis

Description	Manufacturer	Part #	Quantity	Unit Price($)
MCU Unit	Microchip Technology	ATMEGA328P-AU	1	2.14/ea
Bootloader	FTDI	FT232RL-REEL	1	4.50/ea
Bluetooth Module	DSD Tech	HC-06	1	7.99/ea
Ultrasonic Sensor	Adafruit Industries LLC	HC-SR04 (4007)	4	3.95/ea
PCB	PCBway		1	5.00/10 items
Adhesive	3M		1	9.29/ea
9V Battery	Energizer		1	3.10/ea
9V Battery clip	Gonioa	2.1mmx5.5mm	1	0.58/ea
9V Battery Adapter	Duttek		1	6.99/ea
Mini-B USB port	CUI	490-UJ2-MBH-1-SMT	1	0.49/ea
Speaker Grill Cloth	The Wire Zone		1(6ftx3ft)	10.95/ea
Plastic Box	Saim		1	1.80/ea
Voltage Regulator	Texas Instruments	926-LM1117IMPX50NOPB	1	1.10/ea
Schottky Diode	Vishay Semiconductor Diodes Division	SS1P3L-M3/84AGICT-ND	1	0.45/ea
Resonator	Murata Electronics	CSTCR6M00G53Z-R0	1	0.46/ea
1k Ohm Resistor	EDGELEC	E10P004	1	0.80/10 items
100nF Capacitor	BOJACK	B07X5	1	0.80/10 items
1uF Capacitor	LATTECH	B074LZWRV5	1	0.70/5 items
Green LED	Kingbright	APT2012SGC	2	0.37/ea
Yellow LED	Kingbright	APT2012YC	1	0.37/ea
Red LED	Kingbright	APT2012SRCPRV	1	0.41/ea
Total Cost				**74.09**

Table 3.1.1. Cost Analysis of Components

Team Member	**Hourly Wage**	**Weekly Hours**	**Number of Weeks**	**Cost per member**
Michael	$35	15 hours	14 weeks	$7,350
Minh	$35	15 hours	14 weeks	$7,350
Ethan	$35	15 hours	14 weeks	$7,350
Total Cost				**$22,050**

Table 3.1.2. Cost Analysis of labor

3.2 Schedule

Week	Michael's task	Minh's task	Ethan's Task
10/7	Test ultrasonic sensor spread and sampling rate	Create rough draft of detection algorithm	Create schematic and board for PCB
10/14	Calibrate sensors; Research how to interface sensors with an app we will create	Interface bluetooth with microcontroller	Research how to interface the phone's accelerometer and gps with an app we will create
10/21	Mount hardware on PCB	Interface app with bluetooth module	Create demo environment
10/28	Test power and sensor modules	Test smartphone app and detection algorithm with simulated data	Test processing module
11/4	Interface hardware with software	Interface hardware with software	Interface hardware with software
11/11	Test product and make necessary adjustments	Test product and make necessary adjustments	Test product and make necessary adjustments
11/18	Make adjustments based on mock demo feedback	Make adjustments based on mock demo feedback	Make adjustments based on mock demo feedback

4. Ethics and Safety

There are a few potential safety issues with our project. The sensor mount could fall off of the vehicle and potentially damage the vehicle or other vehicles if it becomes completely unattached. Extensive stress testing and a strong adhesive or mountain apparatus will remedy this potential hazard. Also, in accordance with Illinois state laws, the sensor mount will not at all obstruct the view of the front license plate[9].

Another potential hazard is that weather conditions such as rain could potentially damage circuits and sensors and cause shorts. This could lead to high temperatures, melting, and even fires in the circuits. To prevent this hazard, all of the electrical components will be enclosed in a robust weatherproof casing in compliance with IP42 standards[5].

As with all batteries, the 9V battery used to power the system could overheat or melt due to an electrical short. First, the circuit will be carefully designed as to prevent shorts. Then every component of the circuit will be extensively tested to ensure there are no potential shorts in the circuitry of the system.

As our system is a decentralized network, there will always be the possibility of persons attempting to maliciously falsify data. For example, one could place debris, detect it with our system, then remove the debris. This could be done numerous times to trick a user into avoiding that road if they believe there are too many interferences on that road. This would go against #9 of the IEEE code of ethics, ”to avoid injuring others, their property, reputation, or employment by false or malicious action” [3]. A possible solution to this problem would be to implement a voting system for detected interferences. The likelihood of an interference shown on the user interface could be displayed based on how many users in the system also detected said interference. Interferences with many votes would be clearly distinguishable from those with very few. However, we will not be implementing the server portion of the system due to the scope of the class. Therefore, this problem will be largely ignored and instead the purpose and effectiveness of our app will be stated upon opening the app in compliance with #3 of the IEEE code of ethics, “to be honest and realistic in stating claims or estimates based on available data;”[3] to warn users about the possibility of false interferences.

Finally, our app could be a potential distraction for the driver and may cause accidents. To mitigate this, our app will remind the users to pay attention to their surroundings while using the app. This will dissuade users from putting too much focus on the app when driving.

Our solutions to potential safety and ethics concerns follow #1 of the IEEE code of ethics, “to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment;” [3]. There are many potential risks in a product that is to be used while operating an automobile, but we feel that our mitigations allow the benefits to outweigh the potential problems with such a system.

5. References

[1] https://tireguides.com, 2019. [Online]. Available: https://tireguides.com/TireTips/TireDocument/11 [Accessed: 12- Oct- 2019]

[2] "Council response times to potholes," *RAC Foundation RSS2*, 18-Jan-2019. [Online]. Available: https://www.racfoundation.org/research/mobility/council-response-times-to-potholes [Accessed: 13-Oct-2019].

[3] "Arduino Nano", Store.arduino.cc, 2019. [Online]. Available: https://store.arduino.cc/usa/arduino-nano. [Accessed: 19- Sep- 2019].

[4] *Cdn.sparkfun.com*, 2019. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf. [Accessed: 19- Sep- 2019].

[5] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available:http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 19- Sep- 2019].

[6] *Olimex.com*, 2019. [Online]. Available: https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf. [Accessed: 19- Sep- 2019].

[7] dsmt.com, 2019. [Online]. Available: http://www.dsmt.com/resources/ip-rating-chart/ [Accessed: 29- Sep- 2019].

[8] wiki.openstreetmap.org, "Test the accuracy of GPS devices", 2019. [Online]. Available: https://wiki.openstreetmap.org/wiki/Test_the_accuracy_of_GPS_devices. [Accessed: 02- Oct- 2019].

[9] github.com, "Osmdroid", 2019. [Online]. Available; https://github.com/osmdroid/osmdroid. [Accessed: 30- Sep- 2019]

[10] https://www.cyberdriveillinois.com, 2019. [Online]. Available: https://www.cyberdriveillinois.com/publications/pdf_publications/vsd7001.pdf [Accessed: 06- Oct- 2019].