TAFFIC SENSING BICYCLE LIGHT

By

Anchu Zhu

Jiawen Chen

Yijun Gan

Final Report for ECE 445, Senior Design, Spring 2019

TA: Antony Caton

1 May 2019

Project No. 27

Abstract

Our project is to design and build a rear-facing bicycle light. It is able to detect vehicles' distance and velocity using mmWave sensor. And based on the safety distance and road condition, it can alert the car drivers with flashing LED light and loud speakers. Therefore, it can prevent accidents caused by approaching vehicles hitting the back of the bicycle due to drivers' awareness. Meanwhile, the cyclist will be noticed by a OLED display that there is a approaching car behind. It further reduces the possibility of fatality ocurrance.

Contents

1. Introduction

1.1 objective

Despite the increased awareness and improved visibility of bike lanes, there were still more than 800 cyclists' deaths in accidents involving motor vehicles in the United States in 2016. According to the National Highway Traffic Safety Administration, there were 58% of pedalcyclist fatalities that did not occur at intersection locations. It was almost twice as many as compared to those that occurred at intersections, which was 30%. [1] This means more accidents happened during straight ahead road, which is when the cars are hitting from the back of the bicycle because the drivers were unaware of cyclist being in their way. To decrease the chances of this kind of accident from happening, having a device located at the back of the bicycle that is able to sense the incoming traffic from the back and alert drivers of the presence of cyclists would be helpful.

The device should have three modes of warning based on distance and relative speed of approaching vehicles. Under normal condition, mode one can be controlled by the user on their preference. When the car excesses certain speed at around 25 to 30 m range, a white high intensity strobe light is on. When the car reaches up specific speed at around 12 to 16 m, audible alarm is turned on to alert the drivers. Thus, the driver will be warned to not crash onto the cyclist.

1.2 Background

There are several different types of devices in the market related to bicycle safety. The bicycle lights that are available on the market are the devices that can only flash red light when the bicyclists hit the brake or just flashing red light when the device is on. In general, their functionalities are singular and not dynamic enough to alert the driver effectively. In the case of preventing accidents, multiple use of auditory and visualized equipments would be more efficient. Moreover, the capability of operating various modes would be suitable under different traffic situations.

Having such a traffic sensing bicycle light is able to not only heighten the bicyclists' awareness of potential accidents caused by approaching vehicles coming up behind but also encourage the drivers to pay extra attention to the bicyclists. And there would be three alarm modes for how close and how fast is the vehicle approaching to the cyclist. When both the relative distance and the relative velocity hits a certain threshold the device will start flashing white light. If the car continues getting closer, it will start playing a loud horn sound to warn the driver. Other times, it will flash red light to alert the driver behind. The parameters such as road condition and safety distance can be adjusted via bluetooth on an Android devices. The user can also have an evaluation report of the route they have ride. And they can compare which route is safer for the same destination. In a way, it lowers the chance of accidents from happening.

1.3 High-Level Requirements

- The device must be able to distinguish between vehicles coming up from behind, vehicles parked, vehicles passing by, and vehicles passing in the opposite direction.

- The device must be able to have three modes of operation to alert approaching drivers under different situation: a red flashing strobe light, white flashing strobe light, and loud horn.
- The fully charged battery must provide power for at least two and a half hours running.

2 Design

The traffic sensing bicycle light includes five subsystems: a power unit, a control unit, a sensor unit, output units, and a user interface on Android. The power unit consists of two rechargeable Lithium-ion batteries, a charging circuit via power jack or mini usb, and two voltage regulators. The control unit contains a microcontroller to handle a moderate amount of I/O pins, and a bluetooth receiver to communicate with the user interface Android app. The output units have two led lights, a pair of speakers, and a oled display. The user interface on the Android app is able to change the input parameter of safety distance and road condition, and evaluate the trip based on the travel distance and triggered times of warning mode and danger mode. The sensor unit uses a mmWave sensor evaluation module which involves a AWR1642 chip and on-board antenna, for detecting the position of objects.

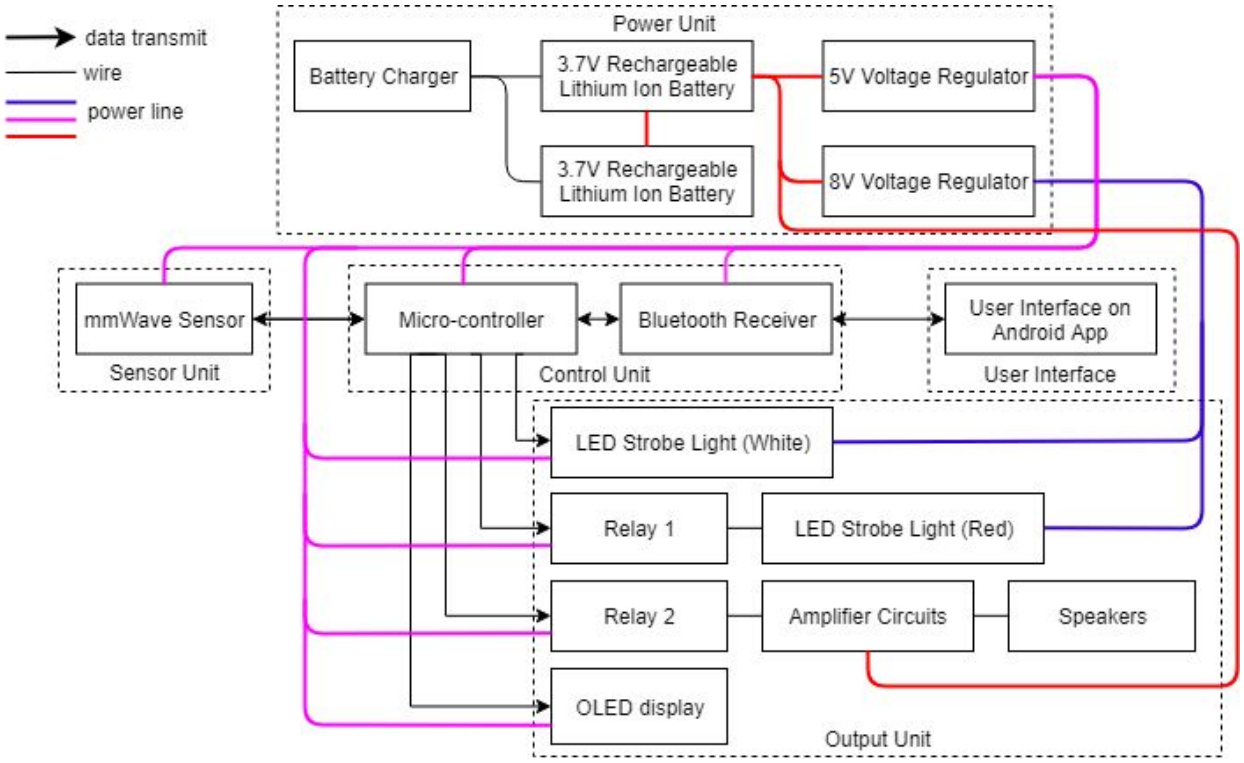

Figure 1. Block Diagram

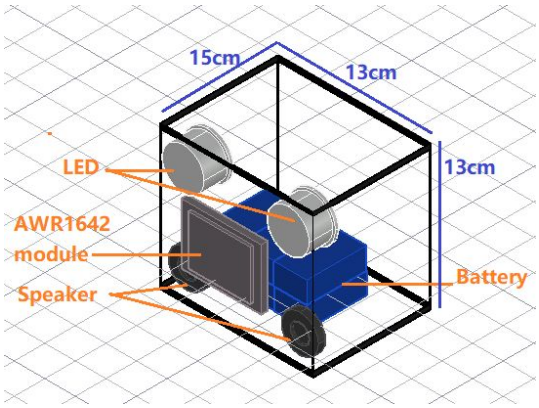

Figure 2. Physical Design

2.1 Sensor Module

The sensor module we are using is directly from AWR1642 evaluation module, which contains the AWR1642 single chip and a complete antenna design shown in the figure below. The AWR1642 single chip mmWave sensor module utilize frequency-modulated continuous-wave radars to determine the distance and velocity of the objects.

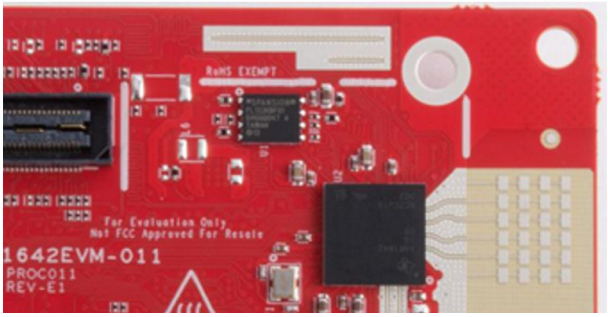

Figure 3. Slot Antenna Array Design with Coupon Structure for a Probe by TI [8].

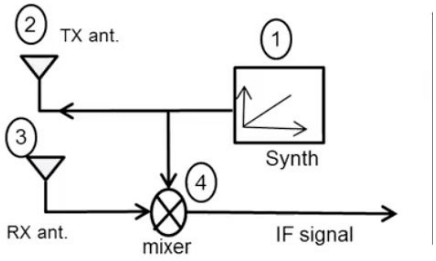

Figure 4. The Process of How a FMCW Radar Works with 1 RX Antenna and 1 TX Antenna [8].

2.1.1 Theory & Calculations

First, a synthesizer generates a chirp wave and then sends out via the transmitter antenna to send out a chirp. And then receive the chirp back and put it into a mixer, which takes the difference of their instantaneous frequency and the difference of their instantaneous phase difference to generate an intermediate frequency signal.

The sensor module is the input of the device. The sensor module contains a single chip AWR1642 which contains a on-chip microcontroller unit ARM Cortex-R4F and a integrated digital signal processing unit TI

C674X. And the distance will between two objects will be calculated using the following equation with its resolution determined by the bandwidth of the chirp. This FMCW radar can also detect multiple objects' distance using the time difference on the reflected time.

For the case of multiple objects with the same distance, using the doppler fourier transform on their phase difference to determine their velocity difference to distinguish each objects. There is certainly trade off when it comes to maximum detection range and the detection resolution. Higher the chirp bandwidth is, better the resolution, but sacrificed the detecting range. And higher sampling rate also helps higher detection range.

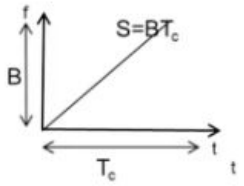

Figure. 5 The chirp of wave via the transmitting antenna where B is bandwidth, f is the frequency, Tc is the transmission time when it increases to that specific frequency B and the S is the slope of the chirp, which is B divided by Tc.

$$\Delta d > \frac{c}{2STc} \Rightarrow \Delta d > \frac{c}{2B} \tag{2.1}$$

Equation 2.1 shows that bandwidth of the chirp determine the resolution on how close two objects separated to be detected [6].

$$d_{max} = \frac{F_s c}{2S} \tag{2.2}$$

Equation 2.2 indicates that sampling rate of the ADC, which is Fs in equation 2.11 and the bandwidth of the chirp, which is expressed in its slope form, determines the maximum detecting range [6].

$$\omega = \frac{4\pi v T_c}{\lambda} \Rightarrow v = \frac{\lambda \omega}{4\pi T_c} \tag{2.3}$$

Equation 2.3 showcases that in FMCW, the velocity is measured by using consecutive chirp with phase difference [6]. Therefore, it is determined by the phase difference ? as other variables in the equation are constant.

$$V_{max} = \frac{\lambda}{4T_c} \tag{2.4}$$

Equation 2.4 implies maximum velocity requires lower transmission time, which means it requires closely spaced chirp [6].

$$V_{res} = \frac{\lambda}{2T_f} \tag{2.5}$$

Equation 2.5 conveys that resolution of the velocity is inversely proportional with the frame time, which means a higher velocity resolutions requires a higher frame rates [6].

In the case of angular estimation. FMCW Radar uses the method angle of arrival to give estimation of the angle relative to the orthogonal axis. And at least two antennas are required to use this method.

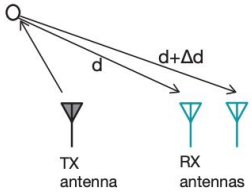

Figure. 6 How AoA Applies on Angle Estimation [6].

1. And the expression for the angle is shown in the following equation. One thing that needs to take notice here is that, the angle would be more accurate when it is closer to the orthogonal axis of the sensor because the theta is approaching zero, and the estimation would use the theta directly when it is very closed to 0 in AoA. And the accuracy would decrease when theta is closer to +/-90 degrees which is on the horizontal axis of the antenna when viewed on top [6].

$$\theta = sin^{-1}\left(\frac{\lambda \Delta\phi}{2\pi l}\right) \tag{2.6}$$

Equation 2.6 is the expression of the AoA estimation of the angle [6] where l is the spacing distance between Rx antenna and the phase difference is derived from the angle difference between wave from Tx antenna to the object to Rx1 antenna and Tx antenna to the object to Rx2 antenna shown in the Figure above.

And the maximum angular field of view is determined by the spacing between two antennas according to equation 2.7 where l is the spacing distance.

$$\theta_{max} = sin^{-1}\left(\frac{\lambda}{2l}\right) \tag{2.7}$$

According to the equation 2.7, maximum angular field of view of +/-90 degrees is achieved when the spacing is $\lambda/2$ [6].

2.1.2 Application & Implementation

For this project, we wrote a program that can identify objects that pose potential threats to the cyclist and output the signal accordingly to the microcontroller, based on a reference design from the Texas Instruments, the Short Range Radar [9]. And in the reference design, the x, y coordinates as well as the velocity, which expresses in doppler index form, are provided and stored in a data structure called detObj. And all of the detected objects points will be stored in a dynamic array filled with structure of detObj. And the header before that array, data object descriptor, has the information of how many objects detected on that frame. And we implement the following algorithm to achieve our goal. And both x, y coordinates are expressed in the view of the sensor, which explains why the position warning message shown in the OLED display is opposite to the coordinates, as the message refers to the cyclist's view.

The first step is to filter our all of the object points that pose the potential threats. To determine this, we have a linear equation, that derives from the stopping distance of a moving vehicle, to determine if the object in certain distance would pose as a potential danger when approaching in that speed.

$$D = d_c + 2v + \frac{v}{2 \cdot f \cdot g} \tag{2.8}$$

Equation 2.8 is the stopping distance equation from Geometric Design of University of Idaho [7] where D is the total distance between the car and the bike, g is gravitational constant, v is velocity of the vehicle relative to the bicycle, 2 is response time by the driver, f is frictional constant on the ground and d_c is safety distance between vehicle and the biker.

$$V(d) = \frac{(d - dc)}{(2 + \frac{1}{(2 \times f \times 9.8)})} \tag{2.9}$$

Equation 2.9 is the reverse function of equation 8, which is the one that is applied in the algorithm, which will be later shown in the flowchart.

As shown in the figure below, all of objects with approaching velocity that has a magnitude that is greater than V(d) would fall into the shaded area when d is greater than safety distance and that velocity smaller than V(d) when d is smaller than safety distance.

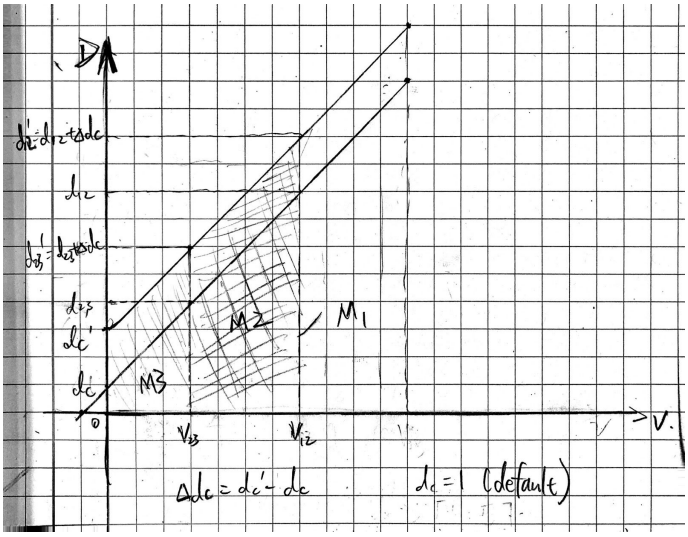

Figure. 7 Increasing the safety distance would certainly make the device easier trigger on relatively low velocity (more sensitive). And shifting the range by Δd_c as long as the ranges of the thresholds are between 0 and 40 m would make the alerting mode consistent.

And in order to adjust the velocity value to fit into the range with safety distance that is other than 1 m, which is the default safety distance, the range for each modes would also need to shift to have the reasonable velocity threshold. And the table below conveys that.

Table 1. The Table of Different Ranges of Velocity and Distance Threshold between Mode 2 and Mode 3 under Different Sets of Frictional Constant and Safety Distance

Input Parameters by user	Velocity threshold	Distance threshold
f = 0.75, d_c = 1 (default)	[-0.5 m/s, 5.32 m/s]	[0 m, 12 m]
f = 0.75, d_c = 3	[-1.451 m/s, 5.32 m/s]	[0 m, 14 m]
f = 0.45, d_c = 1	[-0.473 m/s, 5.20 m/s]	[0 m, 12m]
f = 0.45, d_c = 3	[-1.42 m/s, 5.20 m/s]	[0 m, 14 m]

The second step is to group the filtered object points into different objects by comparing their x, y coordinates values. And in order to do so, we iterate through the array filled with the filtered object points and put them into a 2D vector. Each row represents a new object and each column represents that same object but with slightly different x,y values, hence different object points but still indicating the same object. So from our observance, a valid object has at least two object points, so if there is a row where there is only one element, that means it is highly likely that it is a noise or a false detection.

The third step, is to check each 2D vector to see if they are in the potentially dangerous position whose x ranges from -3 m to 3 m as well as closed to the sensor. If so, that should be the top priority, and the fastest object points on that row would be taken as the velocity of the object.

The final step is to determine which mode that object is based on its position. And it is shown in the following table below. The text in the quotation mark is the message that would be shown on the OLED display, which is intended to installed on the front bar to give a warning to the cyclist.

Table 2. The Triggered Mode regarding to the Position of the Object

	y ∈ [0, 12+ d_diff]	y ∈ (12+ d_diff, 25+d_diff]
x ∈ [-3.0, -0.5)	“closed right”	“your right”
x ∈ [-0.5, 0.5]	“closed back”	“your back”
x ∈ (0.5, 3.0]	“closed left”	“your left”

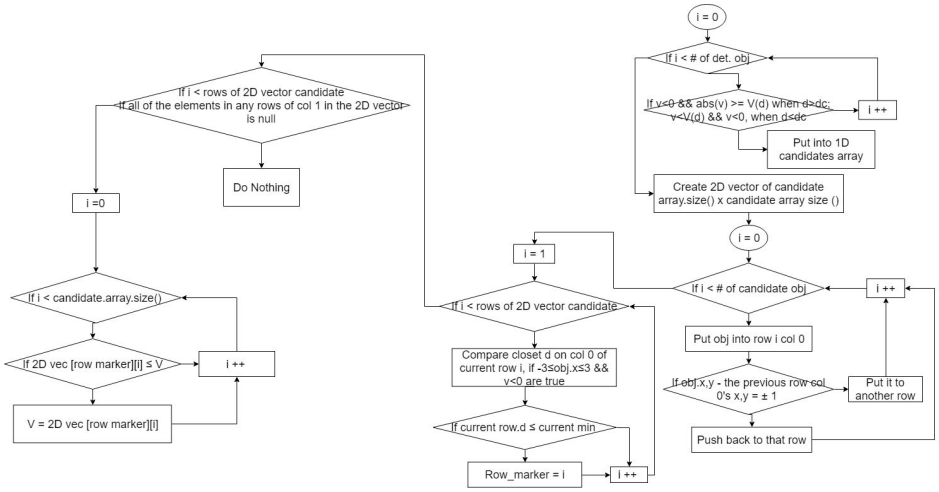

Figure. 8. Flow Chart of the Algorithm Used to Select the Top Priority Object that Poses the Potentially the Most Danger to the Cyclist.

2.2 User Interface

The control unit will be accessed by Android devices via bluetooth. We created our own Android App using Android Studio, which provides interface for connecting bluetooth, entering threshold parameters and receiving evaluation number.

2.2.1 Bluetooth Communication

We used the Classical Bluetooth that is introduced by Android Developer Documentation. In our user interface, we allow users to turn on/off bluetooth, discover nearby bluetooth devices, enable other devices to discover, and start connection with one device. Once the phone is connected to our bluetooth device, the user can click “NEXT” to go on to the entering input interface. Using the Bluetooth APIs, we can scan for other Bluetooth devices, query the local Bluetooth adapter for paired Bluetooth devices, establish RFCOMM channels, connect to other devices through service discovery, transfer data to and from other devices, manage multiple connections. [1]

2.2.2 Input Entering

Since we are using equation 8 above to calculate the standard velocity V(d), we need users to enter two input threshold parameters f (road friction constant) and d (expected stopping distance). Once the user enter the corresponding parameters, the trip starts. Then we use the Google Play SDK to record the latitude and longitude of user's current location for calculating the travel distance for evaluation. Moreover, on the left of the "ENTER" button, we have an additional button "LIGHT ON/OFF" for user to turn on or off the red light for mode 1 at the back of the bicycle anytime.

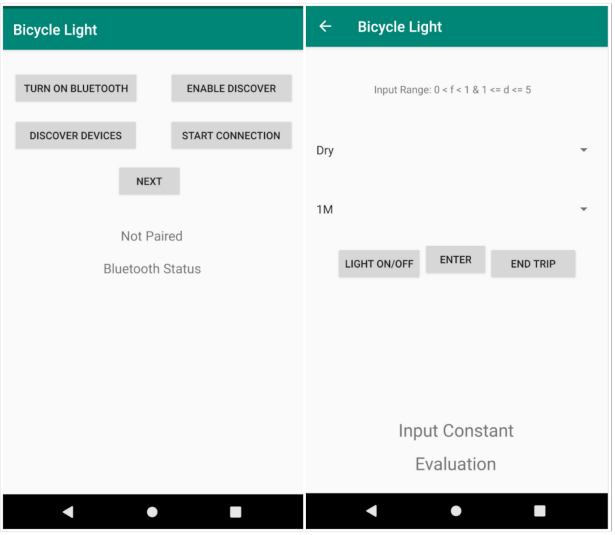

Figure 9. Bluetooth Connection and Input Entering User Interface

There are total 4 options for road friction constant, dry, wet, snowy and icy, and there are for options for expected stopping distance ,1m to 4m.

Table 3. The Static Coefficient of Friction and Integer Signal for Different Road Conditions

Road Conditions	Static Coefficient of Friction	Integer Signal to Microcontroller
Dry	0.7	1
Wet	0.4	2
Snowy	0.2	3
Icy	0.1	4

Table 4. The Integer Signal to Microcontroller for Expected Stopping Distance

Expected Stopping Distance	Integer Signal to Microcontroller
1M	1
2M	2
3M	3
4M	4

Moreover, expected stopping distances from 1m to 2m are more suitable for bicyclists on crowded city roads. On the other hand, expected stopping distances from 3m to 4m are more suitable for situations like driving on a highway or rural road.

2.2.3 Evaluation Number

When the user press the "END TRIP" button, we will record the latitude and longitude of user's current location again and calculate the travel distance based on these two pairs of latitude and longitude. Also, the Arduino module will collect how many times mode 2 or 3 triggered from the sensor and send back the number to the phone. Then we divide the number of mode 2 or 3 triggered by the travel distance to give users a helpful feedback, which tells them how dangerous the route they just took. The smaller the evaluation number is, the safer the route is.

2.3 Microcontroller

A microcontroller based on Atmega328P is designed for controlling multiple output devices as well as handling communications between the user interface and the microcontroller, and between the sensor module and the microcontroller. We have 3 output devices which take 3 digital signal pins. And another 7 pins, consisted of 3 analog pins and 4 digital pins are required for the communication between the sensor module and the microcontroller via GPIO. Two analog pins are needed for I2C communication between the OLED display and the sensor. Another two digital pins (Tx and Rx pins) are required for the bluetooth transceiver connection. Last but not least, one analog pin would be used to monitor the battery level and give out recharge message to the user via the OLED display when the battery level is at the edge of not being able to function. At the same time, doing the programming on the microcontroller is also convenient when using this chip due to the abundant resources online because Arduino Uno is built on this chip. And the Arduino coding platform is well implemented with many open sources libraries. At the same time, the Atmega328p is to implement into a working microcontroller with the example of Arduino Uno.

2.3.1 Interface with other devices

The microcontroller interacts with user interface via bluetooth, which illustrates in the section above. And from the receiving signals, we transfer the signals to the sensor module via GPIO using four digital pins. Each two of the four pins are used to represent one parameter. And the user can control the mode 1 LED strobe light that is connected on the microcontroller independently, which is explained in depth in the bluetooth section.

Table 5. Safety Distance, the Output from the Microcontroller to the Sensor Module via GPIO

Signals from the microcontroller (D6, D7)	Decoded value in the Sensor module (dc)
00	1 m
01	2 m
10	3 m
11	4 m

Table 6. Frictional Constant, the Output from the Microcontroller to the Sensor Module via GPIO

Signals from the microcontroller (D2, D4)	Decoded value in the Sensor module (f)
00	0.4
01	0.7
10	0.2
11	0.1

The microcontroller would control the mode 2 LED strobe light, the speaker as well as the OLED display according to the input signals from the sensor unit. And the following table illustrates how the input signals determine what output devices should be activated.

Table 7. Output Devices Activated Regarding to the Input Signals to the Microcontroller from the Sensor Module

Signals sent into microcontroller (A1,A2,A3)	Decoded msg. in microcontroller (HEX)	Information displayed on the OLED display	Warning device that would be activated
001	0x11	“your right”	Mode 2 LED strobe
010	0x14	“your left”	Mode 2 LED strobe
011	0x12	“your back”	Mode 2 LED strobe
100	0x00 (Idle state)	N/A	N/A
101	0x19	“closed right”	Mode 3 Speaker
110	0x1C	“closed left”	Mode 3 Speaker
111	0x1A	“closed back”	Mode 3 Speaker
000	0x00 (default state)	“U R GOOD”/ “Please recharge”	N/A

The default state means the user is in safe riding environment, where no vehicles at the back are potential danger to the cyclist. And if the battery level dropped down to a certain battery level where it will not function properly very soon, the “U R GOOD” message would be replaced with flashing “Please recharge message”. The mechanism regarding to bring the battery level to 5 V scale using voltage regulator and the value of the on-edge battery level are explained in more details in the Power section.

2.4 Output Devices

2.4.1 LED Light

There are two relays used between the two output LED strobe lights and the control unit. The LED is turned on when the relay receives the signal from the control unit. The operating rated voltage of the LED strobe lights is 9 volts for 900 Lumens, which is equivalent to 60 watts incandescent light bulb. The

maximum current for the LED strobe light is 1100 mA. Since the power sources are only added up to 7.4 volts, a boost converter will be to used to bring up the voltage to desired level for the LED strobe lights.

2.4.2 Speakers

The speakers are connected to the signal directly from the microcontroller, ATmega328P. Therefore, a amplifier circuit is needed to increase the sound volume to rated output. The low voltage audio power amplifier, LM386, is used. With trimmable resistor, the level of sound volume can be adjusted for the better performance.

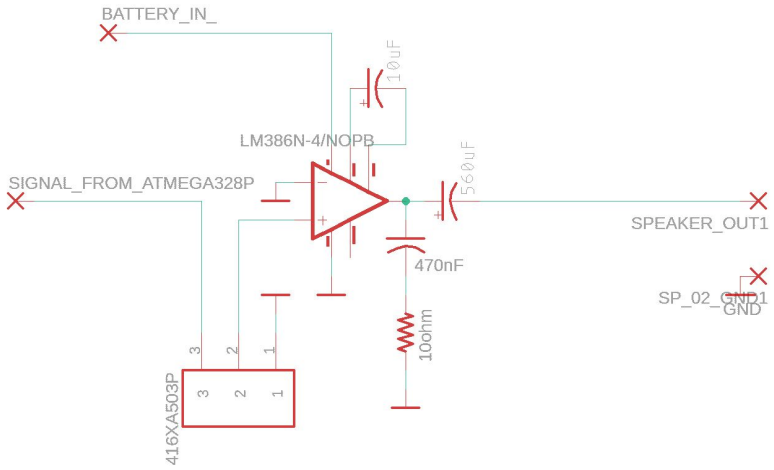

Figure 10. Amplifier Circuit

2.4.3 OLED Display

An 1.3 inch monochrome OLED display is connected to the microcontroller, ATmega328p, and installed on the bicycle bar for the cyclist to view the rear situation. The display's resolution is 128 by 64. And it communicates with the microcontroller through I2C. It is able to show the direction of the nearest potential danger car when warning mode or danger mode is triggered. And it can indicate the low battery level when the batteries are added up to 5.2 volts. Considering the motions of the cars, the frame rate of the display is determined to be 10 frames per second, meaning that the visualization on the back would be updated every 0.1 second.

2.5 Power Unit

The power unit includes two 3.7 volts 6 Ah lithium-ion rechargeable batteries, a charging circuit, and two voltage regulators. As stated in the top-level requirements, the battery must be able to provide power for the device for at least two and half hours continuously running. The sensor module takes up 0.8 A at average. The other components consumes up to 0.725 A in total. Considering the device continuously running for an hour, with 0.7 efficiency, it will consume about 2.18 Ah. Therefore, 6 Ah is able to provide 2.75 hours operating power.

2.5.1 Charging Circuit

The charging circuit is designed to provide more than 200 mA current for each battery for recharge purpose. The circuit can take 5 V input voltage from a power jack or mini usb. A set of switches can be used to select operating mode or charging mode.

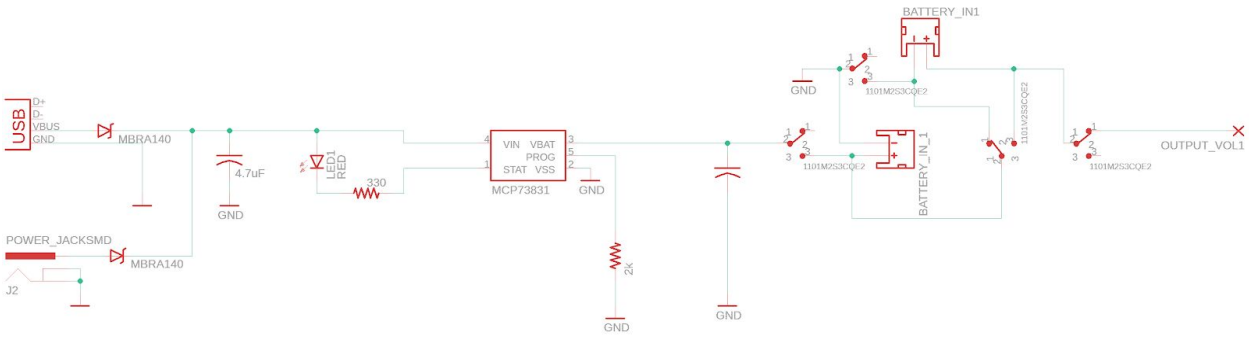

Figure 11. Charging Circuit Schematics

2.5.2 Voltage Regulator

There are two voltage regulators, one is 9 volts, the other is 5 volts. Since the battery level is added up to 7.4 volts, a switching voltage regulator is used to achieve 9 volts. With trimmable resistor, it is able to adjust the output voltage level in order to have a moderate brightness for the LED light. The other regulator maintains a constant 5 V voltage level. It is noticed that the sensor takes a great amount of current. A regulator with higher current tolerance is taken into consideration.

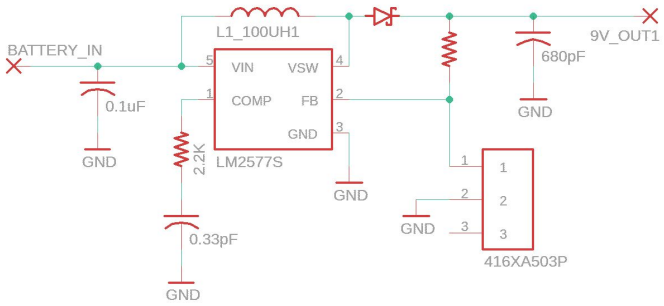

Figure 12. The Switching Voltage Regulator Schematic

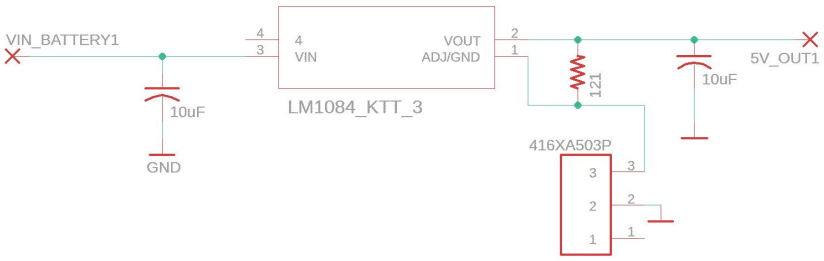

Figure 13. The 5 Volts Voltage Regulator Schematic

3. Design Verification

3.1 Sensor Design Verification

We initially do the testing in the hallway where there are metal doors along the hallway and other metal in the infrastructure in the building, which are all detected by the sensor and causing false alarming issues due to false signals sending to the microcontroller, which has lots of noises. And our original

algorithm in the design document doesn't solve the issue of false alarm due to noise object that also has a false velocity reading that might satisfy our threshold. And the sensor unit itself is powerful and sensitive and able to detect different kinds of metal object. Therefore, we came up with a more advanced algorithm that can first filter out all the object points that don't fall into the potential danger, and group the object points into their belonging object and eliminate noise by not counting any object that has only one object point and also prioritize one object among other detected objects, whose flow chart is in the design section. And in the screenshot below shows the device is able to determine the potential danger object and send out the signals correctly.

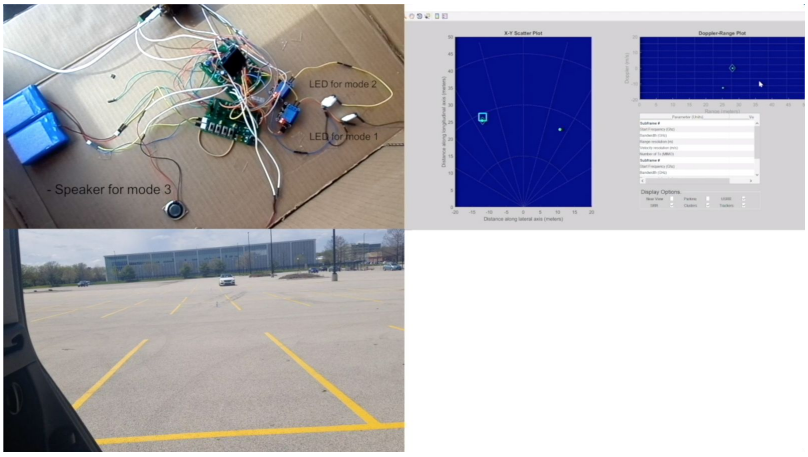

Figure 14. The object point labeled green means there are multiple object points on that approximate position. The red light of the relay that controlled LED mode 2 is on [3].

From the figure 14 above, we can see that the sensor can clearly tell the approaching vehicle is dangerous as its velocity is displayed on the doppler-range plot. And it is at about 25 m away from the sensor. Therefore mode 2 LED is correctly triggered. In real time, the display also shows "your left". But due to the quality of camera lens, and the lighting condition at that day, it is unable to see from the figure.

And when the vehicle is within 10 m away from the sensor, mode 3 triggers. We didn't put an status LED to indicate that, from the video we can see it does [3]. And in real time, the OLED display shows "closed left".

3.2 Bluetooth Verification

We implemented multiple functions of broadcasts in Android Studio, while writing Bluetooth connection, which can be sent when an event of interest occurs, such as found a Bluetooth device. Then, we tested our program by connecting our android phone to the HC-05 and check the print message in the log. If we successfully find HC-05 and start a connection with it, we can clearly see the corresponding message in the log saying that we successfully connect to it. We also used the same implementation for both receiving data and sending data.

```
/**
 * Broadcast Receiver for listing devices that are not yet paired
 * -Executed by btnDiscover() method.
 */
private BroadcastReceiver mBroadcastReceiver3 = (context, intent) → {
        final String action = intent.getAction();
        Log.d(TAG,  msg: "onReceive: ACTION FOUND.");

        if (action.equals(BluetoothDevice.ACTION_FOUND)){
            BluetoothDevice device = intent.getParcelableExtra (BluetoothDevice.EXTRA_DEVICE);
            mBTDevices.add(device);
            Log.d(TAG,  msg: "onReceive: " + device.getName() + ": " + device.getAddress());
            mDeviceListAdapter = new DeviceListAdapter(context, R.layout.device_adapter_view, mBTDevices);
            lvNewDevices.setAdapter(mDeviceListAdapter);
        }
};
```

Figure 15. One of the Broadcast Receiver Code that Print Information of Found Devices

3.3 Interface Verification

We used the bluetooth device to send a few cases to the device, and have the after-decoded received signals printed out on the console. From the figure below, it shows that the entered input are Dry as road condition, which has frictional coefficient of 0.7 and 4 m as safety distance.

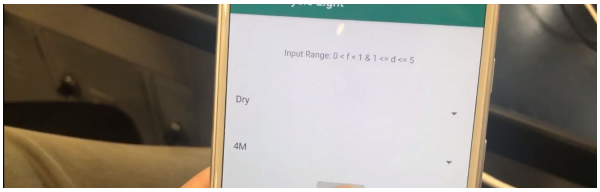

Figure 16. Input Values to the Device to Check If the Sensor Module Received the Correct Values [4].

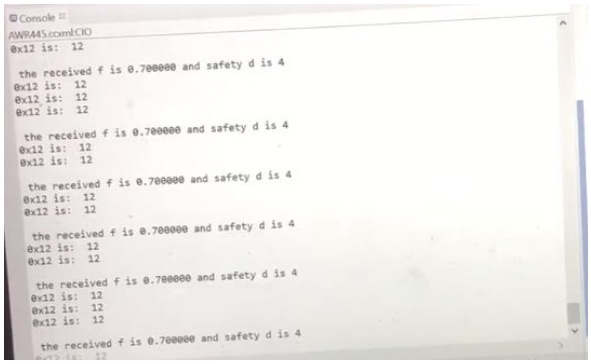

Figure 17. The Received Values from the Sensor Unit [4].

As we can see the sensor is receiving the correct values from the sensor. Therefore, the bluetooth communication as well as the 4 bits input from microcontroller to the sensor unit is working correctly.

And the message that is transferring back to the microcontroller is also shown on the console from the figure above, which is 0x12. This message will be encoded into three bits of information, 011 to the A1, A2, A3 analog pins. And the microcontroller will also use the same table to decode that three bits into 0x12 if the communication is working correctly.

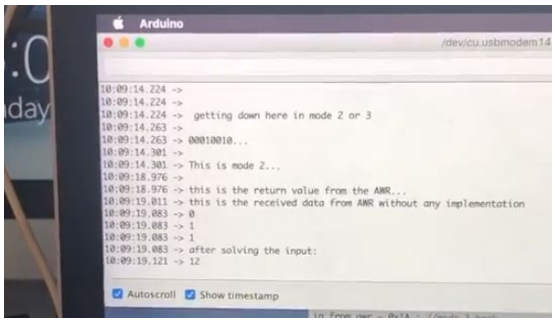

Figure 18. The Input Received from the Sensor Unit [4].

And as you can see from the above figure. It indeed received the 011 raw signals and decode it correctly into 0x12.

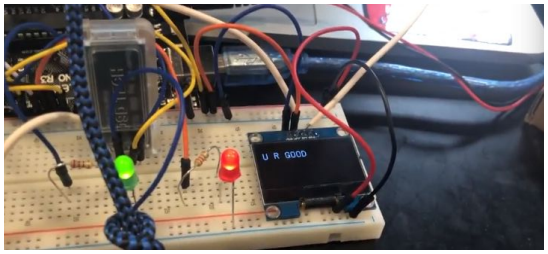

Figure 19. The red LED represents the signal status of mode 2 LED strobe light [5].

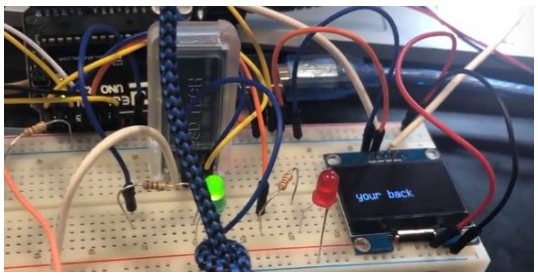

Figure 20. The OLED display shows the message according to the decoded signals 0x12 [5].

Due to the implementation in the microcontroller, our output device would activate before the OLED display shows the message, the delay between is about 0.8 second. And this would be addressed in further works. So far, the communication as well as the output control logic works correctly from this example. And the mode 3 output control as well as the logic verification l can be found in the demo video[3].

3.4 Output Devices Verification

As shown in the sensor verification and interface verification above, the flashing LED lights, speakers, and OLED display was working properly as designed. The operating voltage for LED light in this device was 7.8 volts. At first, rated voltage 9 volts was applied. It was too bright to open the eyes. Thus, we adjusted the voltage level to obtain a moderate brightness. The amplifier for the speakers has up to 200 gains. With trimmable resistor, we were able to acquire rated voltage level for the speakers in order to reach its rated sound volume, 90dB, which was considered loud enough to be heard by the car drivers.

3.5 Power Unit Verification

With 5 volts input voltage into the charging circuit, it provided 0.23 A for each battery in parallel. The switching voltage regulator was able to boost to 7.8 volts, and the 5 volt voltage regulator maintained a stable output voltage level.

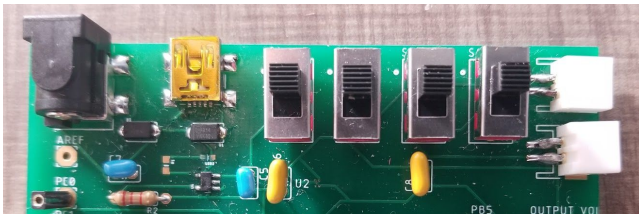

Figure 21. Charging Circuit PCB

4. Costs

4.1 Parts

Table . Parts Costs

Part	Manufacturer	Retail Cost ($)
3.7 V 6 Ah Lithium Ion Battery *2	Sparkfun	$59.90
5 V Voltage Regulator	Texas Instruments	$2.03
9 V Voltage Regulator	Texas Instruments	$9.46
Microcontroller	Microchip Technology	$2.14
16MHz Crystal	Abracon LLC	$0.27
Bluetooth Receiver	HiLetgo	$5.99
Relay *2	Sparkfun	$5.99
LED Strobe Light *2	Hontiey	$6.49
Speakers *2	CUI Inc.	$7.40
mmWave Sensor Evaluation Module	Texas Instruments	$299.00
Total		$392.68

4.2 Labor

The fixed development were estimated to be $20/hour. There were 15 hours/week for three people at average.

$$3 \cdot \frac{\$20}{hr} \cdot \frac{15hr}{wk} \cdot 12wk = \$10800$$

5. Conclusion

5.1 Accomplishments

We chose a sensor that is powerful despite limited online resources when it comes to debugging in the process and execute the idea make it work. We successfully implemented the algorithm which selects most dangerous object to the cyclist and responds accordingly using our own microcontroller. At the same time we made a user interface on Android for the user to interact with the device. More importantly, we designed our own circuit board for the microcontroller as well as the voltage management circuits to support different devices. After integrating all the components together, we showcase a working idea, all in the span of 2 months.

5.2 Uncertainties

In this project, there are still some uncertainties that we need to put into more work to find out the reason to move forward. One of the biggest issue of the final product that we put in the demonstration is that, the device is too big to fit into the back of a bicycle. The reason why each component is spreaded out from each other is that the battery is somehow affected by the circuits and if it is put near other components, the battery level drops drastically. Even there is nothing conducting between each parts. We are suspicious on the electric magnetic field that generated by the high current in the circuit, which might affect the battery level.

5.3 Ethical considerations

Our project aligns with the IEEE Code of Ethics because the device itself is safe and will not do any damage to others. Moreover, it is supposed to alert both bicyclists and car drivers and prevent potential dangers, which results in decreasing the bicycle accident involved with cars. Therefore, the project will " hold paramount the safety, health, and welfare of the public" and "disclose promptly factors that might endanger the public or the environment" [2].

5.4 Future work

5.4.1 Optimize the space of the device

We will try to make the device more compact so it can fit into a small container and put it at the back of the bike, so far the battery's voltage level is being interfered resulting in drastic voltage drop when the battery is put closed to other components. And further research and testing on the circuit is necessary to eliminate the uncertainties and move forward.

5.4.2 Design required components for our sensor unit

On-chip antenna design, as well as a complete set of digital signal processing circuits and circuit protection design would be needed to support a working sensor unit that is commercially distributable and scalable since the AWR1642 Evaluation module, which is what the sensor unit consists of, costs about 300 USD and it is illegal to resell. To lower the cost into a more friendly price is crucial to make this device from prototype to a product.

5.4.3 Further development on trip evaluation system

Now we have a rough evaluation system and we also roughly know the maximum possible accumulated triggering times. But how we divide into reasonable spectrum that represents a certain kind of route condition for cyclist might need to further investigate. More testing is required to see the relationship between triggering times and route environment so that we can give out relatively correct suggestion on route evaluation.

5.4.4 Optimize the microcontroller's delay

Optimization on the microcontroller program is needed in order to shorten the delay. Right now, the way it implements is to have the output device activated before the message shown on the OLED display. And this causes the delay on the microcontroller. Using another way that has two devices activated at the same time using a buffer might shorten the delay. But further testing is needed to verify this theory.

References

[1] Android, "Developer Guide". [Online]. Availiable: https://developer.android.com/guide/topics/connectivity/bluetooth

[2] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 29- Feb- 2016].

[3] Gan, J. (2019). *445 Field Test Demo*. [online] YouTube. Available at: https://youtu.be/359lTHV69bs [Accessed 2 May 2019].

[4] Gan, J. (2019). *Communication testing between sensor unit and Microcontroller*. [online] YouTube. Available at: https://youtu.be/q2rFvLImO2Q [Accessed 2 May 2019].

[5] GAN, J. (2019). *Complete I/O test on breadboard with OLED Display and signal LEDs*. [online] YouTube. Available at: https://youtu.be/2ZqYb13EUiw [Accessed 2 May 2019].

[6] The fundamentals of millimeter wave sensors, Texas Instruments, Inc., 2017. [Online]. Available: http://www.ti.com/lit/wp/spyy005/spyy005.pdf

[7] Geometric Design. (2019). [online] Available at: https://www.webpages.uidaho.Unedu/niatt_labmanual/chapters/geometricdesign/theoryandconcepts/brakingdistance.htm [Accessed 22 Feb. 2019].

[8] TI mmWave Radar sensor RF PCB Design, Manufacturing and Validation Guide, Texas Instruments, Inc., 2018. [Online]. Available: http://www.ti.com/lit/an/spracg5/spracg5.pdf

[9] Short-range radar (SRR) reference design, Texas Instruments, Inc., 2018. [Online]. Available: http://www.ti.com/tool/TIDEP-0092

Appendix A Requirement and Verification Table

Table of System Requirements and Verification

Requirement	Verification	Verification Status
AWR1642BOOST Sensor 1. Sensor can work in the desired configuration: maximum range of 40 meters, range solution of 41 centimeters, maximum velocity of 70 km/h, and velocity resolution of 1 km/h	AWR1642BOOST Sensor 1. a. Input the desired configuration into mmWave Estimator provided by TI to see the requirement for bandwidth, frequency range, transmit and receive gain b. Connect the module via USB to the PC and use mmWave visualizer to check if the module can meet the requirement on single car detection	
Software on AWR1642BOOST 1. Objects are correctly mapped out in the plot on mmWave Visualizer 2. The device is able to activate different output device 3. The device correctly records the last time for mode 2 or 3.	Software on AWR1642BOOST 1. a. Load a test program into the board after bootloading and use TI's mmWave Visualizer to check if any objects are detected 2. a. Load our software program with easily accessible parameter on velocity so that the testing vehicle can test on outdoor with the bicycle being stationary with individual safety distance offset), and check if each output device has been activated correctly. i. If the device activates different output device accordingly, the input default parameters can test on field with safety distance to prevent possible collisions during. ii. If not, check the SPI connection first, then check the mmWave sensor performance using mmWave Visualizer. 3. a. Put an object at the range that activates mode 2, while manually record how long	

	the object stays in the range. Then, compare the manually recorded number to the number from the IWR1642.	
Microcontroller ATmega328P 1. Can both receive and transmit over UART at a speed 2. Can send signals to activate output devices 3. Cortex R4F can receive the output signal from the IWR1642 and passback the testing signals via GPIO	Microcontroller ATmega328P 1. a. Use a test program that contains time.h output the variable that stores the input received from the UART to the console b. Type in a set of data of parameters from the device and send it to the microcontroller c. Check the output on the console see if it matches to what was input from the mobile device 2. a. Use a test program that sends a active low signal to the pins where the output devices such as the strobe LED and speaker are connected. b. Check if the corresponding activated device is on. 3. a. on Arduino, test if the communication is working properly.	
LED Strobe Lights 1. Can blinks when relay is active 2. Provide bright enough light that the driver is able to see from at least 20 meters away	LED Strobe Lights 1. a. Connect the LED lights in series with the relays b. Operate the relay in a desire way, check if the LED blinks as it is set up 2. a. Turn on the LED light during the day b. Check if the LED light is able to be seen from 20 meters away	
Speakers 1. Can play sound when relay is active 2. Provide loud enough that the driver is able to hear from at least 10 meters away	Speakers 1. a. Connect the speakers in series with the relays b. Operate the relay in a desire way, check if the speakers operate as setup 2. a. Play the sound and place it around a car b. Check if the drive is able to hear the sound from 10 meters away	

OLED Display 1. Show the animation from the sample code correctly 2. The display updates according to the frame rate that we set and sensing data.	OELD Display 1. a. Connect SCL and SDA pins to I2C pins on ATmega328p and power the display with 5 volts input power, and then run the sample code 2. a. Set a low frame rate at 2 frames per second and have one object moving in front of the antenna and check if the display updates. i. If it updates correctly, we set to 10 frames per second and try the test again ii. If not, check the I2C connection between the sensor and the board and also check the assigned starting address on the interface, to see if further modifications are needed	
Bluetooth 1. Can use bluetooth connected devices to modify the parameters in microcontroller and also read the current parameters of the device once connected	Bluetooth 1. a. Use a test program that reflects the status to the Bluetooth Device and take input to turn on the LED which is connected to the ATmega328p b. Connect the bluetooth device to the microcontroller and Check if it could change the status of the LED and reads the current status of the LED	
Power Supply Unit 1. Be able to recharge the batteries in parallel at 4 volts when in charging mode 2. Maintain a constant 5 volt voltage level 3. Output a desired voltage level to turn on the LED strobe lights	Power Supply Unit 1. a. Input 5 volts voltage source via power jack or mini usb b. check if there is 4 volts between each battery 2. a. Input battery power source b. Check with multimeter if there is 5 volts output 3. a. Input battery power source b. Connect the output voltage to the LED c. Check if the LED is bright enough	