Spring 2019

Safe Pass

Alycia Riese
anr66@zips.uakron.edu

Julia Hariharan
jjh127@zips.uakron.edu

Greg Synek
gms63@zips.uakron.edu

Jonathan Hall
jwh81@zips.uakron.edu

Safe Pass

Final Project Report

Design Team 09

Jonathan W. Hall (JWH)

Julia J. Hariharan (JJH)

Alycia N. Riese (ANR)

Gregory M. Synek (GMS)

Faculty Advisor

Dr. Malik Elbuluk

Date Submitted
5/3/19

Table of Contents

List of Figures

List of Tables

Abstract

The purpose of this project is to design a sensor to be mounted on Class IV and higher vehicles to detect on-coming traffic. If traffic has been detected, the system is to warn drivers behind the stopped vehicle that passing is unsafe. The vehicle detection is to be implemented using a LiDAR detection method along with signal processing. A wireless transceiver is to transmit from the front radar module to the rear warning indicator module when the conditions are unsafe for passing. The project goals are to increase road safety and maintain traffic flow. The report details the challenges due to the wireless link and the original radar approach.

[JWH]

1. Problem Statement

1.1 Need

With an increase in delivery services there is a growing amount of stopped vehicles on the road, and therefore a greater need to safely maneuver around them. Stopped vehicles on the side of road that can be the source of dangerous and risky situations. Passing other vehicles when blind spots are present creates an inherent risk for all drivers on the road. When drivers are unaware of oncoming traffic, it can often result in a dangerous situation or slow down the flow of approaching vehicles. Offering an alert to notify drivers that attempting a pass may be unsafe provides additional safety measures and may help prevent collisions.

[JWH, GMS]

1.2 Objective

The goal of this project is to design and create a system that will allow drivers to pass stopped vehicles on the side of the road safely. This system will detect the presence of oncoming traffic from the front of the stopped vehicle. After the traffic has been detected, the system will then alert the driver to stop, letting them know it is unsafe to pass. If there are no warning lights, the driver can determine whether they believe it is safe to pass the stopped vehicle. The system will never prompt the driver to pass the stopped vehicle, because there could be other objects or cars that are undetected. This system will enable drivers to be safer while driving on the road, possibly preventing more accidents.

[JWH, JJH, ANR]

1.3 Background

1.3.1 Patent Search

There are no current patents that have the system required to accomplish the goal of this project; however, there are aspects of existing patents that could be incorporated into the project design. The technology used for car blind spot detection has been developed over the last couple decades, but the principles behind vehicle detection are the same today.

[JWH]

The patent "System and method for combined blind spot detection and rear crossing path collision warning" describes a system that can detect objects in the blind spot of a vehicle and provide a warning. The system uses sensors that can be programmed to detect objects at different ranges depending on the application. The ability to program the detection range of the sensor would be an effective method for detecting oncoming traffic in different speed zones, which would require different ranges of detection. In addition to changing the range of detection, the sensors in this system can be programmed to eliminate stationary objects when detected. This patent has a similar goal to the project goal, because it will detect moving objects and neglect stationary objects [1].

[JWH]

The patent "Device for determining distance between vehicles" is for a device that can accurately measure the distance between two moving vehicles. However, this device only detects other vehicles in the same lane as the car with the device. The device uses light receivers to detect and measure the distances of lane lines or moving objects. This technology could be useful in determining the traffic gap distances and evaluating the distances for unsafe passing conditions. To integrate this device into a stopped vehicle danger sensor system, the device will need to detect traffic in the opposite lane [2].

[JWH]

The patent "Smart blind spot detector" describes a radar system to be used to detect obstacles in a vehicle’s blind spot. It works by directing a radio signal toward the area of the blind spot and measuring the signal again if it is reflected backward. If the signal is shifted upward when it is reflected backward, then that means an obstacle is approaching the blind spot. If it is shifted downward or not reflected at all, then the obstacle is moving away or not present. This could be a useful technology to detect if a vehicle or another obstacle is approaching the stopped vehicle danger sensor system. This will help the project because it will help determine if the object is moving closer or further away [3].

[ANR]

"Truck light warning system" provides a description of a system that uses visual indicators to alert drivers that they may be located within a truck's blind spot regions or be tailgating a truck too closely. The system uses front, side, and rear facing sensors using laser or microwave technologies, a link to the engine control module, and a link to the driver's instrument control panel. Digital displays are used as the primary alert to drivers on the road and an icon on the truck driver's dashboard shows that a vehicle may be in one of four blind spot locations on a semi-truck with trailer. The project proposal does contain aspects of this patent, but it differs in how drivers may be alerted of potential problems and focuses on being an assistive technology for commuters more than an aid for a truck driver [4].

[GMS]

1.3.2 Article Search

The article "Overtaking vehicle detection using implicit optical flow", explores the possibility of replacing radar and laser detection of objects and passing vehicles with optical detection. Optical detection systems are not as effective in object detection as laser sensors, but camera systems are oftentimes cheap and relatively easy to operate. Capturing images at a substantial rate can be difficult if the whole image is captured and processed. To increase the speed and response of the system, the article suggests capturing images in small windows which focus on the area of interest on the road. To further increase the processing time, the camera only processes the "edge pixels", the pixels which make up the outlines of objects. The technology described in this article could be useful in effectively and reliably detecting on-coming traffic for stopped vehicles. The concept of processing parts of an image instead of an entire image is an efficient method for detecting the objects of interest on the road [5].

[JWH]

In the article "A radar-based blind spot detection and warning system for driver assistance", the authors discuss the topics of blind spot detection for day and nighttime conditions using system architecture, radar system structure and algorithms, intermediate frequency (IF) signal processor, motive target detector and blind spot area calibration method and system control strategy. Another topic of discussion at this conference was how to maintain higher detection rates and lower failure rates of object detection using signal processing hardware/software. The technology for this system was implemented using a DSP-embedded platform. The ideas proposed at this conference were relevant to the proposed project because they go into detail on technologies that perform advanced object detection. Advanced object detection is necessary for accurate alerts when on-coming traffic is present. One of the most applicable ideas of this article was the use of a DSP-embedded platform to create the detection system. This is a relatively cheap and user-friendly platform which could be implemented into the design project [6].

[JJH]

Many sensing modules are available now for vehicle detection, such as radar, LiDAR, and cameras. With the advancement of computer processors and GPUs, computer vision is approaching real-time implementation. The article "Looking at Vehicles on the Road" aims to compare radar, LiDAR, and camera implementations for detecting obstacles on the road. Radar works best for a narrow field of view, but it struggles at cross traffic intersections. Also, it detects other things and could mistake them for a vehicle. Lidar provides a much wider field of view than radar. It can track obstacles over multiple lanes, but it does not work very well in the rain and is expensive. Vision based detection uses ambient light and images to track and detect obstacles. This solution can detect obstacles using multiple cameras places around the vehicle. It is intuitive to understand but it is sensitive to light and weather [7].

[ANR]

The article "Vision in and out of vehicles" describes the need for driver assistance technology to help make driving safer. To obtain the most results from object detection, multiple methods of detection must be used. No one method will work in all situations. One of the tests in the article placed a camera on the front of the vehicle that tracked the lane and how it curved. Meanwhile, another camera was tracking the driver's eye position. The experiment wanted to know what the driver looked at the most based on the curve of the road. This could be useful to us for the project because it will help us understand how to detect obstacles and what the driver is mainly looking at while driving. This information could be used to help us get the drivers attention when an obstacle is present [8].

[ANR]

At the 2017 IEEE International Conference on Image Processing the paper titled "Appearance and motion based deep learning architecture for moving object detection in moving camera," describes some of the latest techniques for object recognition. The purpose of this conference was to improve previously used methods for object detection by moving cameras. Background subtraction from an image has been a commonly used method for object detection in the past. However, this method is flawed because it is susceptible to interference in a moving camera. The article proposes a method to detect a moving object with technology that adapts to environments which enables the detection of objects regardless of dynamic background. This technology identifies both appearance and motion features of an object. The system is composed of two networks combined to detect moving objects by utilizing the appearance and motion of the target object.

[GMS]

In moving object detection there are two methodical approaches: object-centric and background centric. The object-centric method may be slightly more accurate, but it is very expensive and complicated, thus the costs outweigh the benefits. The background centric method is much more cost-effective and practical because it is a relatively simple model. In this model, the moving object is considered the foreground and the rest of the area is the background. The system will analyze the background to form a model and use this analysis to form a response.

The concepts of this article are applicable to this project because it provides a cost-effective and efficient way to detect moving objects. This conference was also helpful because it considered the flaws of past technology concerning motion and object detection and noted the drawbacks of alternative options. The information provided in this article points out the issues in previous technology concerning this topic and offers ideas on how current moving object detection can be improved [9].

[JJH]

The article "A Visual Blindspot Monitoring System for Safe Lane Changes" describes a system to detect objects within blind spots using live video feeds. Drivers can then be alerted of an object present in an unsafe location before switching lanes. This is a different methodology for blind spot monitoring compared to patents listed above that make use of microwave and laser-based technologies. Visual motion patterns are discovered and run through algorithms to improve the accuracy of the system without necessarily being able to identify specific objects that could pose a safety risk. A described advantage to visual detection is that radar and other active broadcast options have a more limited range for low power consumption and can be deflected or absorbed in certain situations. Combining infrared cameras alongside traditional visual range image sensors offers the ability to function day and night with potentially fewer problems to handle. The article goes on to detail different methods of image processing techniques that may be applicable to the proposed senior design project should it be determined that a visual detection system fits the project goals best [10].

[GMS]

Although the proposed system does not yet exist, there are relevant technologies and concepts that will be useful in the development of the desired system. The on-coming traffic detector will use a reliable sensor to identify and evaluate the traffic conditions.

[JWH]

1.4 Marketing Requirements

One of the first steps in a design project is to create a list of specific requirements from the marketing perspective. Below are the criteria for the proposed system:

1. The system should detect on-coming traffic in front of stopped vehicle.
2. The system should determine and notify drivers not to pass stopped vehicle when on-coming traffic is detected.
3. The system should have reliable sensors to detect motor vehicles.
4. The system should have sensors that are operable in all environmental conditions.
5. The system alert should be easily understood by drivers.

6. The system can be installed on any vehicle.

[JWH, GMS, JJH]

1.5 Objective Tree

After realizing the marketing requirements, it is important to organize the criteria categorically. The main objectives are listed at the top of the objective tree, and below the objectives are expounded. The objective tree is shown below for the Safe Pass System.

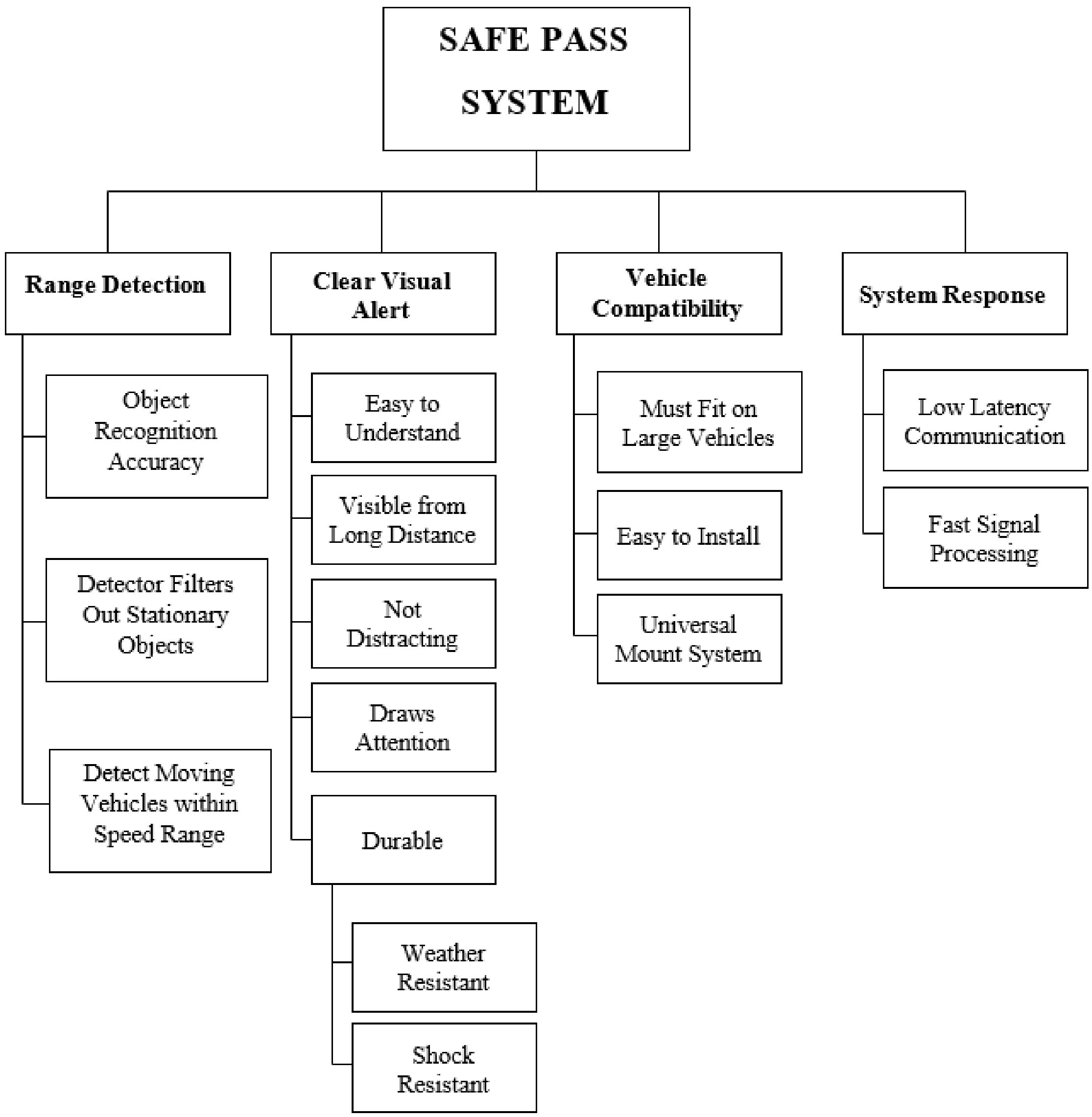


Figure 1: Safe Pass Objective Tree

[JWH]

2. Design Requirements Specification

The design requirements are listed below with their justifications. Each design requirement is correlated to at least one marketing requirement.

Table 1: Engineering Design Requirements

Marketing Requirement(s)	Engineering Requirements	Justification
1,3	1. The system shall detect objects up to 356 ft. away.	Based on a maximum speed of 55 mph the distance required to safely stop is 356 ft.
3	2. The system shall detect objects moving less than 55mph.	Delivery and Transportation vehicles should not stop on any road faster than 55mph.
1	3. The system shall give a response within 500 milliseconds of detection.	This time should give ample time for traffic to stop, but the actual response will be determined by chosen parts.
1	4. The system shall detect vehicles as small as a motorcycle.	The system is intended to detect all motor vehicles, ranging from trucks to motorcycles.
6	5. The system shall run off a 24V/12V power source.	24 V supply is the standard voltage in new truck batteries and 12 V is in older vehicles.
1,3	6. The system shall detect vehicles inside a single 12-foot wide lane.	The standard width of road lanes is 12ft.
6	7. The system shall wirelessly communicate to warning indicator.	A wireless connection to the warning indicator will allow for an easy installation on the vehicle.
2	8. The warning indicator shall be visible from 356 feet.	This is the required distance for the vehicles behind the stopped vehicle to safely stop.
6	9. The system shall not consume more than 100W.	The maximum power consumed by a headlight is 100W, the system will consume less power than one headlight.
5	10. The warning indicator shall have adaptive brightness based on surrounding lighting conditions.	The warning indicator should be brighter in the daytime and darker in the nighttime.

[JWH]

Marketing Requirements (Reiterated):

1. The system should detect on-coming traffic in front of stopped vehicle.
2. The system should determine and notify drivers not to pass stopped vehicle when on-coming traffic is detected.
3. The system should have reliable sensors to detect motor vehicles.
4. The system should have sensors that are operable in all environmental conditions.
5. The system alert should be easily understood by drivers.
6. The system can be installed on any vehicle.

3. Accepted Technical Design

3.1 Hardware Block Diagrams

Below are the hardware block diagrams with the modules and their descriptions. The block 0 diagram shows the inputs and outputs of the system in relation to the modules.

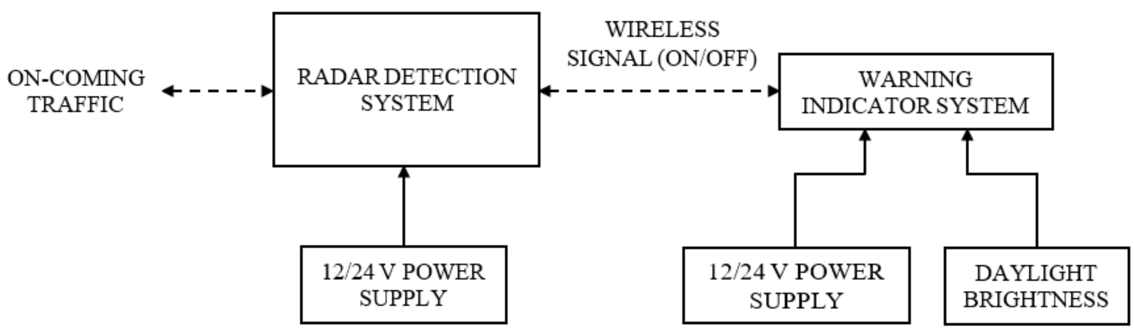


Figure 2: System Level 0 Block Diagram

[JWH, GMS]

For the block 1 diagram the blocks from the previous block diagram are expanded with more detail. From the design requirements, the input power supply shall be a 12 V or 24 V supply because the system will run off the vehicle battery.

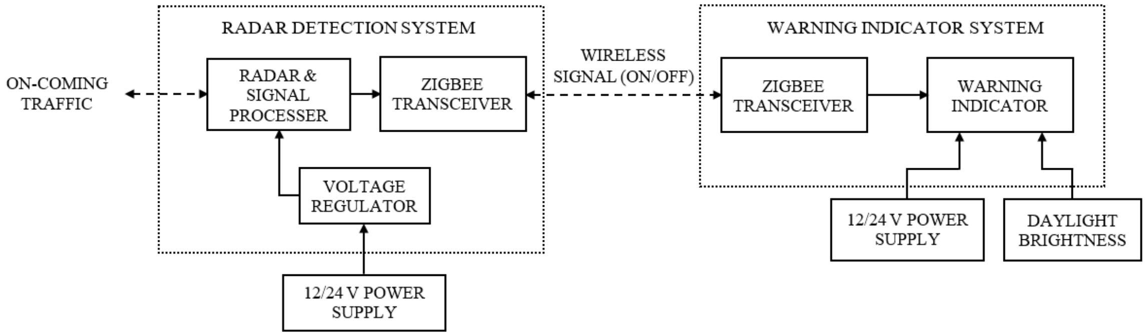


Figure 3: System Level 1 Block Diagram

[JWH]

For the wireless link, the Level 2 Block Diagram below was used for the initial circuit design.

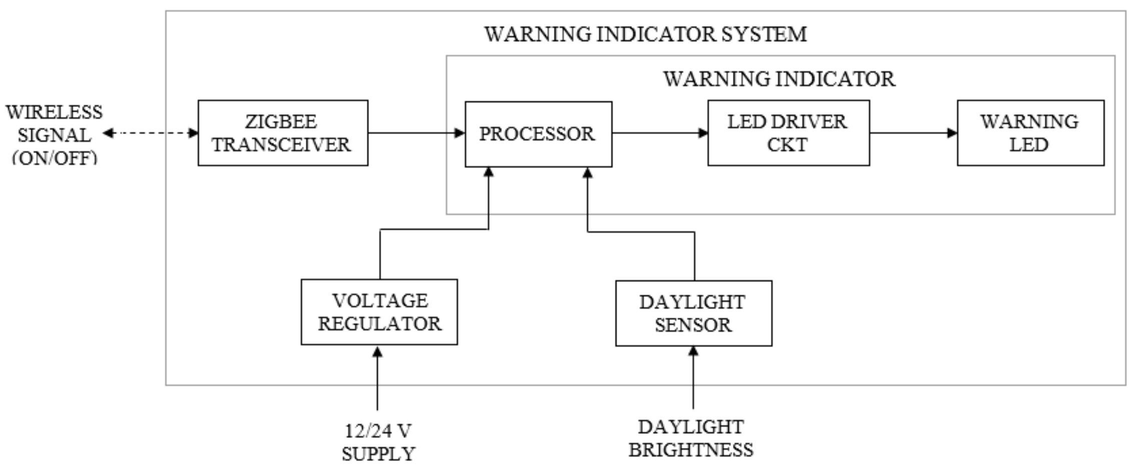


Figure 4: Warning Indicator System Level 2 Block Diagram

[JWH]

Table 2: Warning Indicator Module Description

Module	Warning Indicator
Designer	Jonathan Hall
Inputs	Warning Indicator On/Off Signal, Supply Voltage, Daylight Intensity
Outputs	Message LED Board, Indicator State Signal (On/Off)
Description	The warning indicator system will receive a signal from the radar detection system with a command to turn the warning indicator on (traffic detected) or off (no traffic detected). When the warning indicator receives the signal, the processor will generate a pulse width modulated signal at the output to drive the LED warning indicator. The light intensity of the indicator will change based on the daylight, the daylight intensity will be determined using a photoresistor. When the warning indicator has been activated, the warning indicator module will send a verification signal to the radar module to confirm the action has been executed. The warning indicator LED will be connected to a 12/24 V supply, and the Indicator will be controlled by the microprocessor.

Table 3: ZigBee Wireless Link Module Description

Module	ZigBee Wireless Link
Designer	Jonathan Hall
Inputs	On/Off Signal, Supply Voltage, Indicator State Receiver
Outputs	Warning Indicator On/Off Signal
Description	The wireless link module communicates between radar and warning indicator. When traffic is detected by radar module the wireless link will transmit a signal to the warning indicator to turn on. When the on-coming traffic leaves the passing-zone the wireless link will notify the warning indicator to turn off.

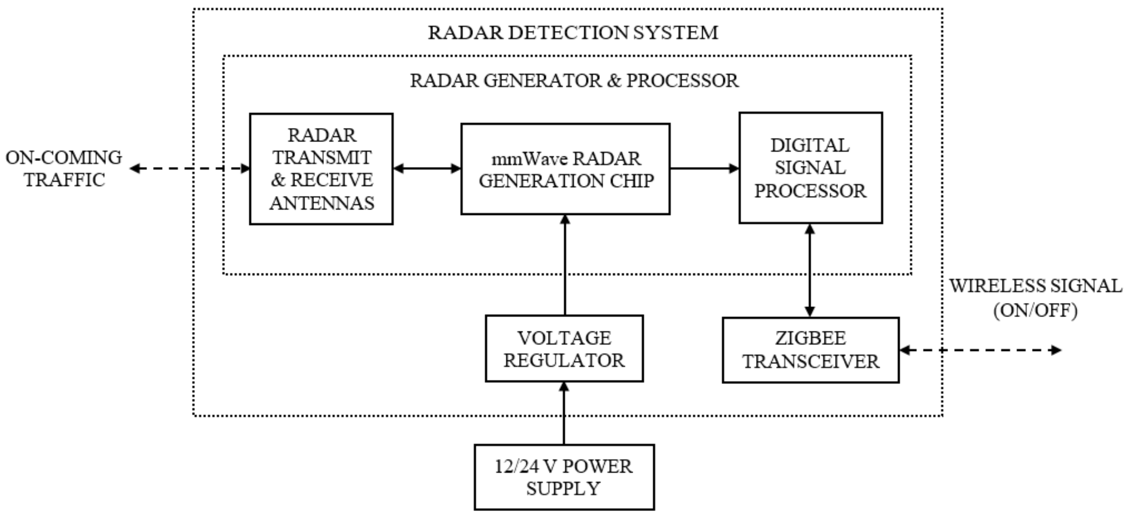


Figure 5: Radar Detection System Level 2 Block Diagram

[JWH]

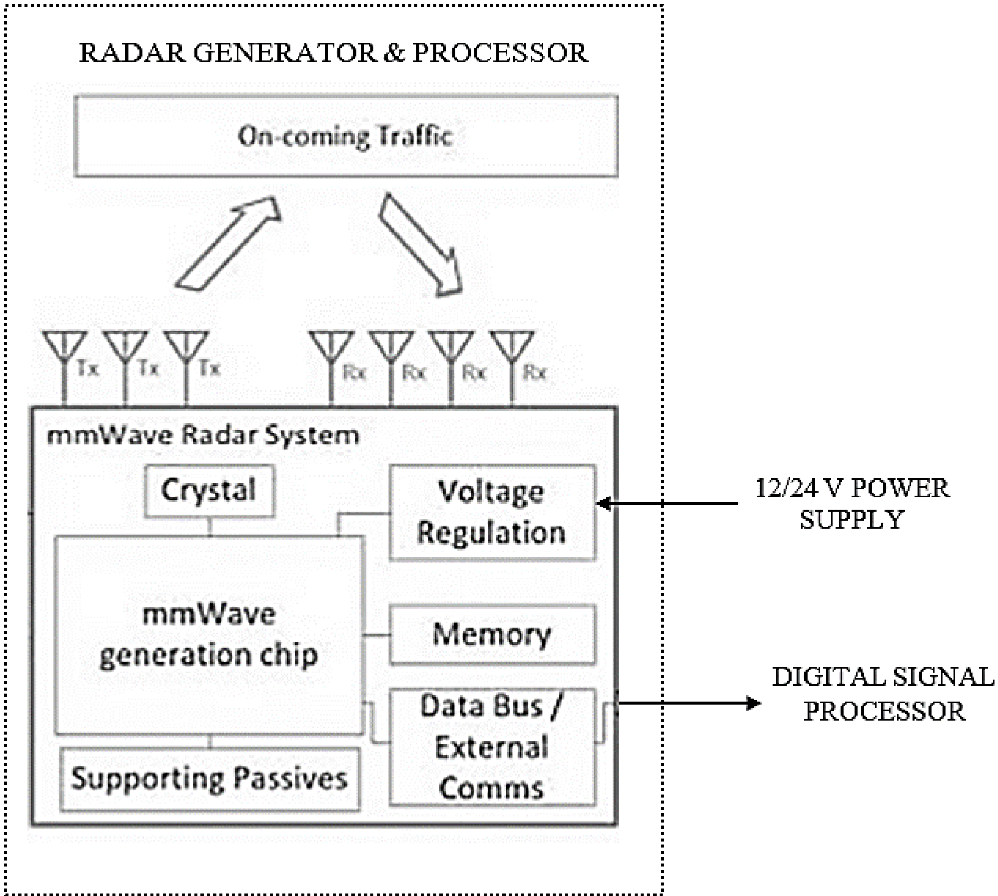


Figure 6: Radar Generation Chip Level 3 Block Diagram

[GMS]

Table 4: Radar System Module Description

Module	Radar System
Designer	Greg Synek
Inputs	Reflected Radar Signal, Supply Voltage
Outputs	Transmitted Radar Signal, Digital Signal
Description	This module will transmit and receive a radar signal to detect the presence of on-coming traffic. The processor will then send a signal to the wireless link when traffic is detected.

Table 5: Receive Antenna Module Description

Module	Receive Antenna
Designer	Julia Hariharan
Inputs	Received Signals, Input Impedance
Outputs	Output Signal with Associated Impedance
Description	While the sensor is responsible for encoding the signal information and into a radio frequency (RF), the antenna will wirelessly transmit the signal.

3.2 Software Flow Diagram

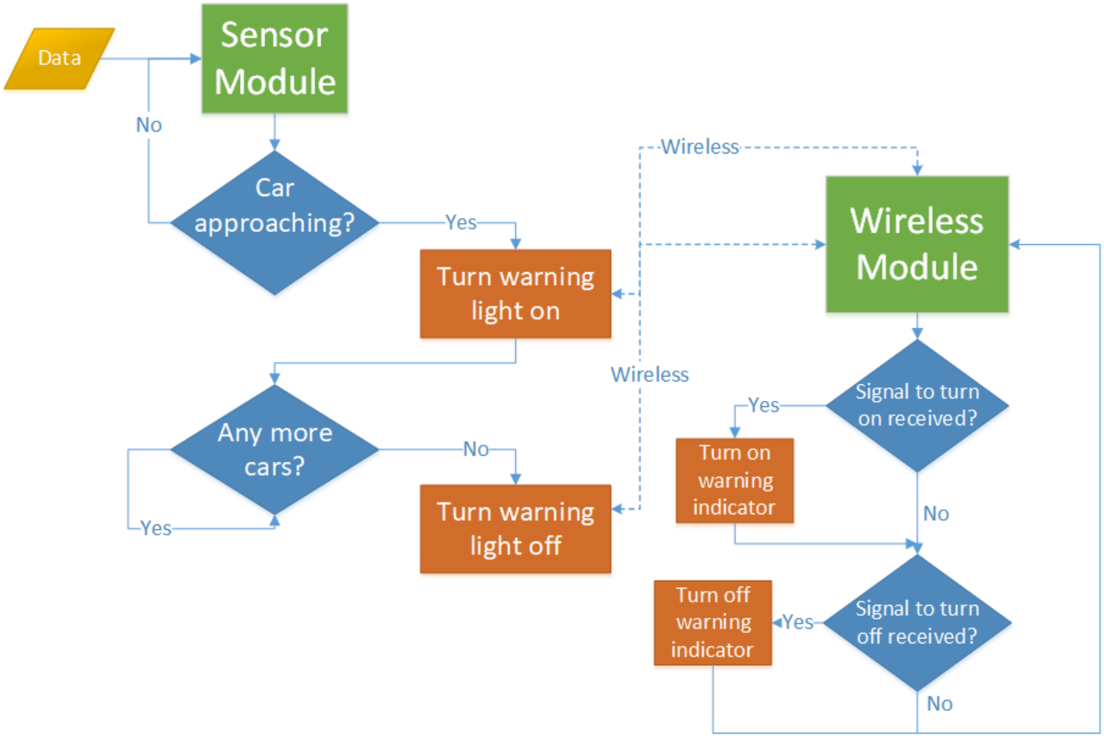


Figure 7: Software Flow Diagram

[ANR]

Table 6: System Control Software Flow Chart Description

Module	System Control Software
Designer	Alycia Riese
Inputs	Raw data from mmWave sensor
Outputs	Warning Indicator Light
Description	Figure 5 describes how the system will interact with software. The sensor module will receive data from the sensors. It will then process the data and determine if a car is approaching. If a car is approaching, the sensor module will send a signal to the wireless module to turn the warning light on. After that, the sensor module will continue to check if there are more cars approaching. If there are no more cars, the module will send a signal to turn the warning light off. The wireless module will be waiting to receive signals. If the signal to turn the warning light on is received, the wireless module will turn on the indicator. If the signal to turn the warning light off is received, the wireless module will turn off the indicator. Then, the wireless module will continue to wait for signals.

Below is a block diagram that shows the means of communication employed between each of the subsystems.

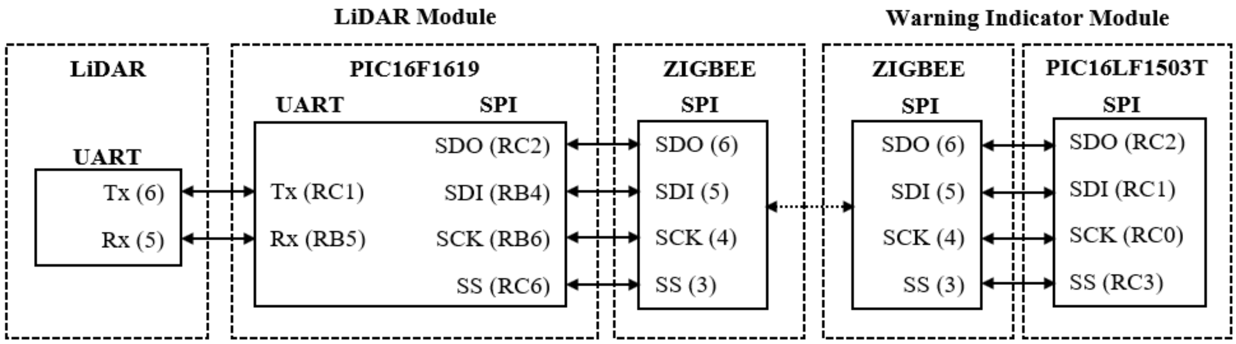


Figure 8: Subsystems Interfacing Path Diagram

3.3 Design Specifications and Operation

3.3.1 Sensor Range Calculation

The calculations below are used to determine the appropriate distance to warn drivers behind the stopped vehicle, and to detect the on-coming traffic.

$$Speed\left(\frac{ft}{s}\right) = \left(\frac{55\ miles}{hour}\right)\left(\frac{1\ hour}{60\ minutes}\right)\left(\frac{1\ minute}{60\ seconds}\right)\left(\frac{5280\ feet}{1\ mile}\right) = \mathbf{80.66\ ft/s}$$

The general rule for spacing between stopped cars is approximately 3 car lengths or 50 feet, which is represented as the clearance space in the table below.

$$Detection\ Range = Stopping\ Distance(ft) + Clearance\ Space(ft) + Processing\ Delay(s) \times Speed\left(\frac{ft}{s}\right)$$

$$Detection\ Range = (265\ ft) + (50\ ft) + (0.5s) \times \left(80.66\frac{ft}{s}\right) = \mathbf{355.3\ ft}$$

Table 7: Vehicle Stopping Distance Calculations

Vehicle Speed (mph)	Stopping Distance (ft)	Clearance Space (ft)	Processing Delay (ms)	Total Distance (ft)
55	265	50	500	355.3
50	229	50	500	315.7
45	196	50	500	279.0

40	164	50	500	243.3
35	136	50	500	211.7
30	109	50	500	181.0
25	85	50	500	153.3

[JWH]

3.3.2 Wireless Link Requirements

The distance required to transmit from radar module to the warning indicator module is up to 60ft (the standard length of a semi-truck). A ZigBee transceiver module will be used to transmit a "turn on signal" from the radar module to the warning indicator when traffic is detected. The range of the ZigBee wireless technology is up to 400ft. ZigBee is a low bandwidth, low power, and far range communication technology.

[JWH]

3.3.3 Input Voltage Conversion

Following the information provided in the TPS54231 datasheet, the following parameters provide a 3.3V output for 5V to 28V input. Parameters have been calculated in accordance with the datasheet using the following figure as a reference [15].

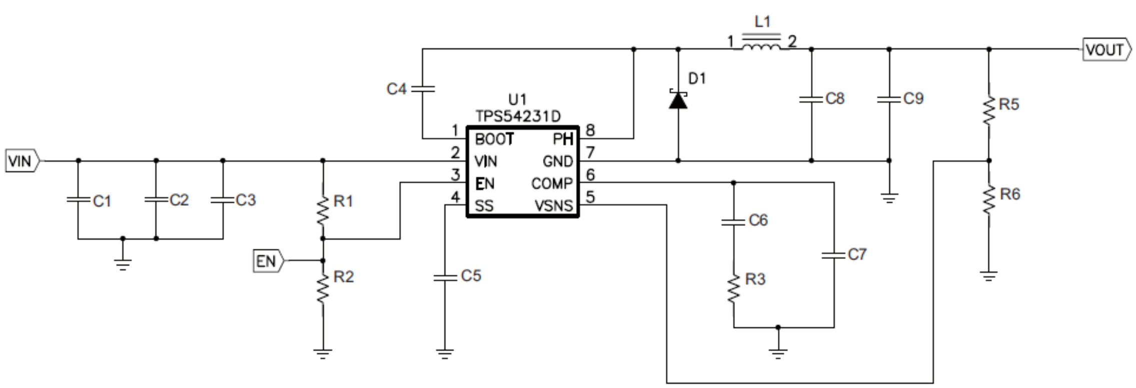


Figure 9: Datasheet Schematic for Buck Converter (TPS54231)

$$R_1 = \frac{V_{start} - V_{stop}}{3\mu A} = \frac{\sim 1V}{3\mu A} = 332k$$

$$R_2 = \frac{V_{EN}}{\frac{V_{start} - V_{EN}}{R1} + 1\mu A} = \frac{1.25V}{\frac{7V - 1.25V}{332k} + 1\mu A} = 68.1k$$

$$T_{SS}(ms) = \frac{C_{SS} * V_{ref}}{I_{SS}} = \frac{0.015uF - 0.8V}{2\mu A} = 6ms$$

$$R_6 = \frac{R_5 * V_{ref}}{V_{out} - V_{ref}} = \frac{10.2k * 0.8V}{3.3V - 0.8V} = 3.24k$$

$$V_{out} = V_{ref} * \left(\frac{R5}{R6} + 1\right) = 0.8V * \left(\frac{10.2k}{3.24k} + 1\right) = 3.3V$$

$$\Delta V_{IN} = \frac{I_{o(\max)} * 0.25}{C_t * f_{sw}} + \left(I_{o(\max)} * ESR_{max}\right) = \frac{2 * 0.25}{9.41uF * 570kHz} + (2 * 100m\Omega) = 300mV$$

$$I_{Cin(RMS)} = \frac{I_{o(\max)}}{2} = 1A$$

$$L_{min} \geq \frac{V_{o(\max)} * \left(V_{in(\max)} - V_{o(\max)}\right)}{V_{in(\max)} * K_{IND} * I_o * f_{sw}} \geq \frac{3.3 * (28V - 3.3V)}{28V * 0.3 * 2A * 570kHz} \geq 8.5\mu H$$

$$I_{LPP} = \frac{V_o * (V_{in(\max)} - V_o)}{V_{in(\max)} * L_{out} * f_{sw} * 0.8} = \frac{3.3 * (28 - 3.3)}{28 * 10uH * 570kHz * 0.8} = 0.64A$$

$$I_{L(RMS)} = \sqrt{I^2 out(\max) + \frac{1}{12} * I^2 LPP} = \sqrt{4 + \frac{1}{12} * 0.4096} = 2.008A$$

$$I_{L,peak} = I_{o(\max)} + \frac{I_{LPP}}{2} = 2A + \frac{0.64}{2} = 2.32A$$

$$C_{o(\min)} \geq \frac{1}{2 * \pi * R_o * F_{CO(\max)}} \geq \frac{1}{2 * \pi * 0.1 * 25kHz} \geq 64uF$$

Datasheet pages referenced for all calculations are linked in the Appendix.

[GMS]

3.3.4 Radar Generator Method and Device

To determine the appropriate method of vehicle detection, the different technologies, features, and functions. The different detection methods are compared in the table below.

Component	Infineon RASIC™ Front-End IC's for Automotive	Texas Instruments AWR series mmWave ICs	Various 24 GHz Radars	FLIR
Range	820 ft	820 ft	98 ft	Line of sight
Frequency	77-79 GHz	77-81 GHz	24 GHz	N/A
Type	FMCW	FMCW	FMCW / FSK /Doppler /Monopulse	IR Focal plane array
Comments	Only available directly, not via distributors	Has better documentation available	Does not meet range needs	Low resolution, sensor saturation problems, processing power concerns

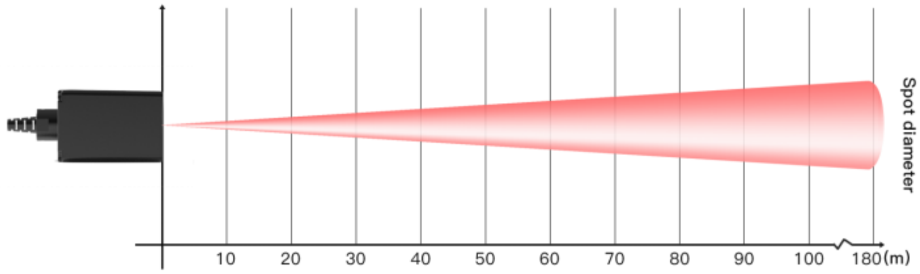
To achieve the required distance of 356 feet, there are only two options that can be used to detect vehicles at the needed distance. Therefore, only the mmWave Radar Chip from Texas Instruments and the Inifineon RASICS Front-End IC's are acceptable options. Both methods have similar ranges and frequency ranges; however, the Texas Instruments AWR chips have a larger frequency range. Initially, the Texas Instruments mmWave radar generation chip was chosen because more documentation was available. This approach was abandoned and replaced with Lidar for ease of software development.

[GMS]

3.3.5 LiDAR Detection Device

After pursuing this implementation, it was determined to be an unfeasible option and LiDAR was used instead. The range of the chosen LiDAR needed to reach up the design requirement level of 356 feet of detection within a 12-foot wide lane. The TF03 was chosen because it has a range of detection up to 180 m and within the desired width of detection.

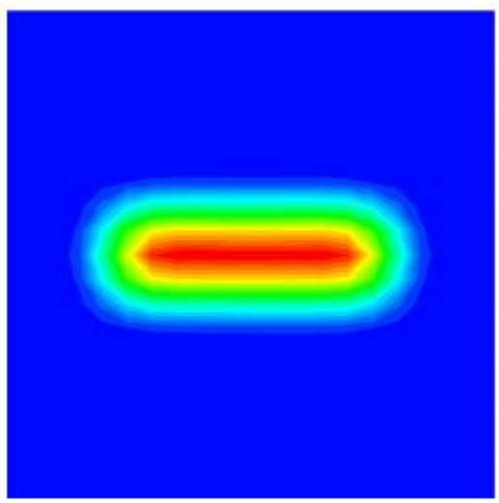
[JWH]



The above illustration shows the detection range of the lidar. The diameter of the detection range increases based on distance from the module.

[ANR]

From the datasheet for the TF03, the LiDAR detection area is shown in the figure below.



Spot 100m away	Size: 100cm*28cm

Figure 10: LiDAR Spot Size at 100m Distance

At a distance of 100 m detection the LiDAR spot is shown above, this distance is 328.084 feet. The LiDAR area of spot of detection is found to be about 100 cm wide and 28 cm tall. The 100 cm width can be converted to feet and is found to be 3.28 which is within the desired 12 ft wide lane requirement. However, to further compare the spot size at the required distance, the spot size is shown below at a distance of 150 m.

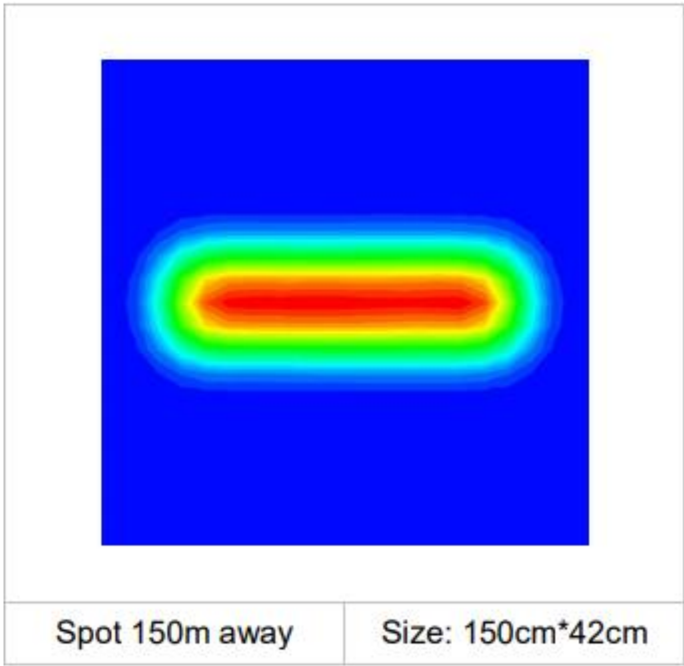


Figure 11: LiDAR Spot Size at 150m [19]

The LiDAR spot size is now determined to be 150 cm wide which is 4.9 feet; therefore, the detection range is still within the desired 12-foot wide lane requirement. This distance exceeds the design requirement for detection as 150 m is equivalent to 492 ft.

[JWH]

3.4 Schematic Design

3.4.1 Radar Module and Signal Processing Schematic Design

For the radar and signal processing unit, the schematic was designed to implement the block diagram shown in Figure 4. After researching several different devices and methods of implementing the proposed design the AWR1642 was selected. This device is an integrated single chip radar sensor with capability of operation between 76-81GHz band and 4GHz bandwidth. Because of the large frequency band, higher performance can be achieved in a small footprint. Some other advantages of choosing this device include: low power operation, high accuracy, and its performance against other comparable technologies. Some useful features included in this device are: integrated PLL, Transmitter, Receiver, and precision A2D converters. There are four receive channels and two transmit channels. The design of the schematics pertaining to this device are shown below.

[JJH]

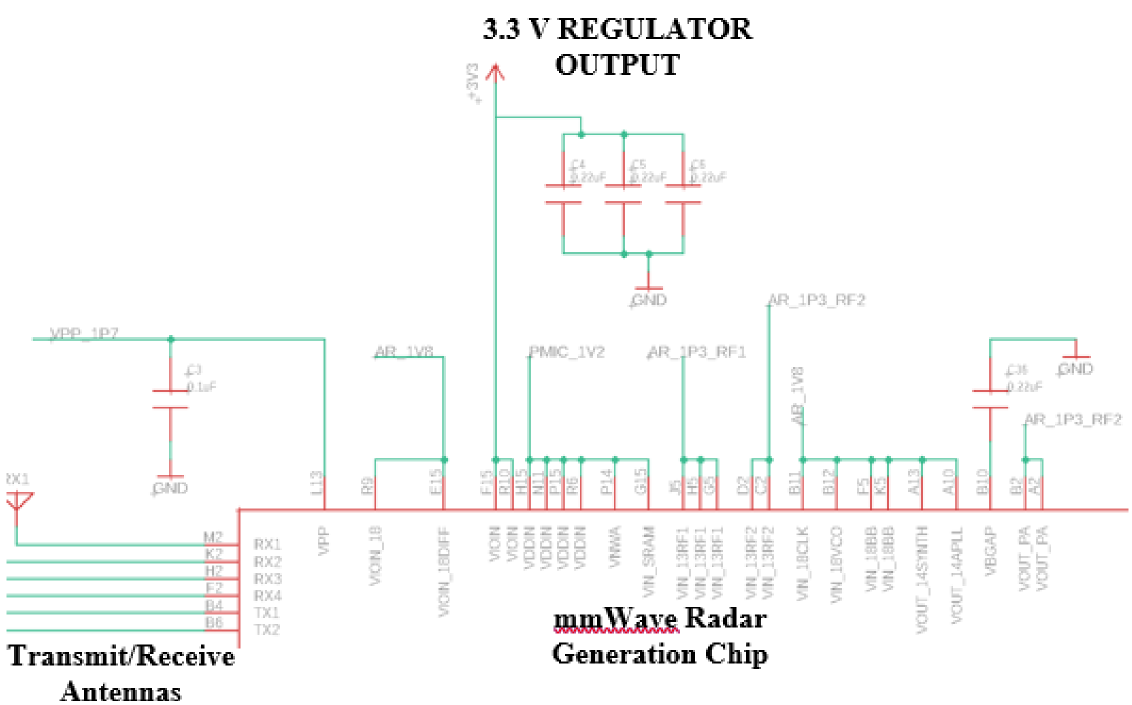


Figure 8: Eagle Schematic for AWR1642 (1)

The top of the schematic symbol contains voltage references and input voltages for different internal sections of the device. In accordance with ANSI symbol standards, pin names describe the functional role of each input with an abbreviation that offers the purpose and type of input. Net names outside of the symbol are intentionally the same as those used in the datasheet so that another group of engineers could easily glance at the schematic and figure out where all connections are made.

Table 8: Pin Labels and Descriptions

Pin Name	**Description**
VPP	Voltage supply to internal fuse
VION_18, VION_18DIFF	1.8V to I/O Pins
VIOIN	3.3V to I/O Pins
VDDIN	1.2V chip supply
VNWA, VIN_SRAM	1.2 to SRAM
VN_13RF(X)	1.3V analog and RF voltage reference
VIN_18CLK	1.8V clock voltage source
VIN_18VCO	1.8V RF mixers
VIN_18BB	1.8V baseband voltage source
VOUT_14SYNTH	1.8V reference to internal module
VOUT_14PLL	Phase-locked loop voltage supply
VBGAP	Bandgap voltage reference
VOUT_PA	Power Amplifier voltage

For any questions regarding pinout, reference pages 14 through 24 of the AWR1642 datasheet [18].

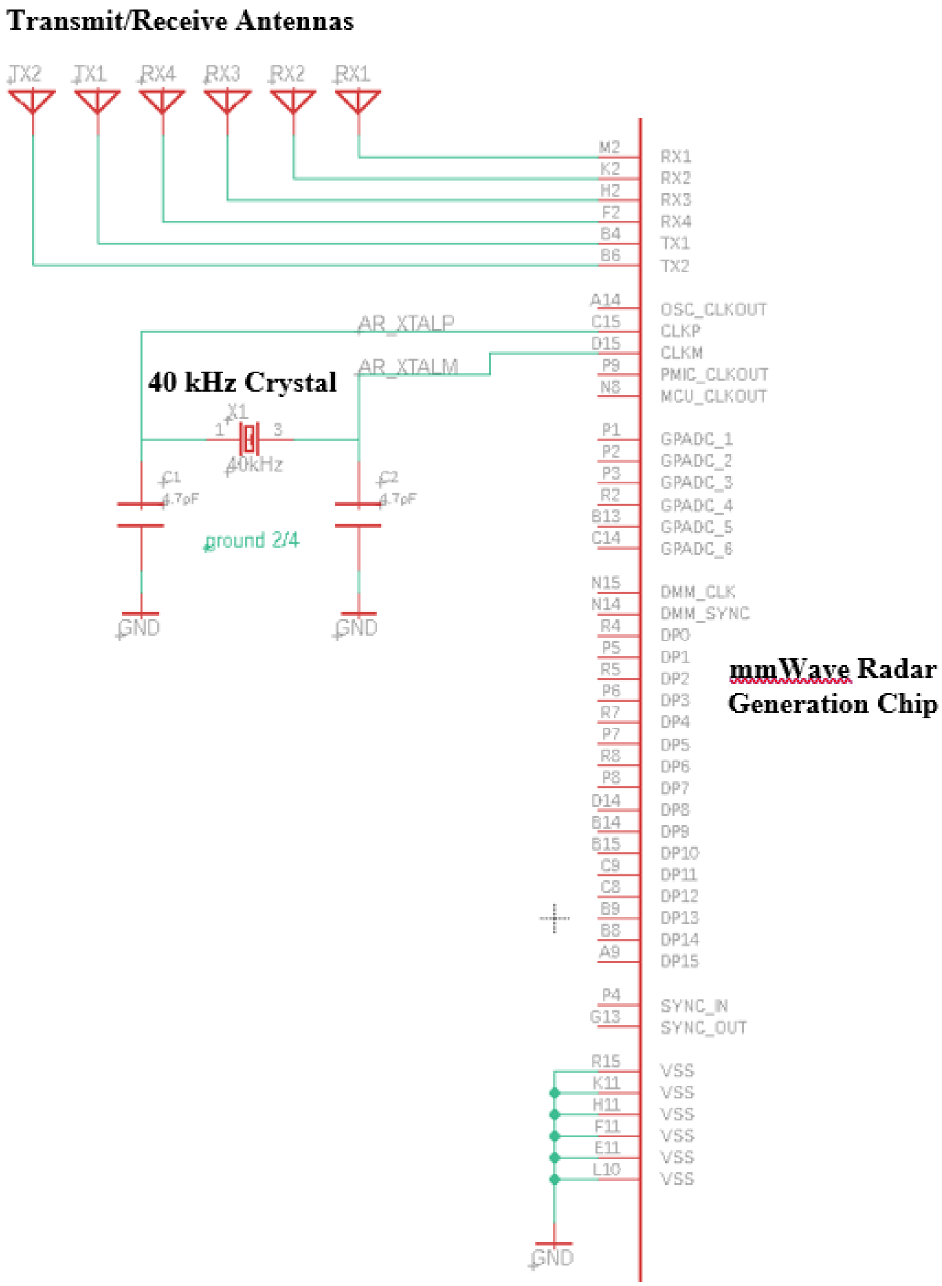


Figure 9: Eagle Schematic for AWR1642 (2)

All pins shown without connections are NC.

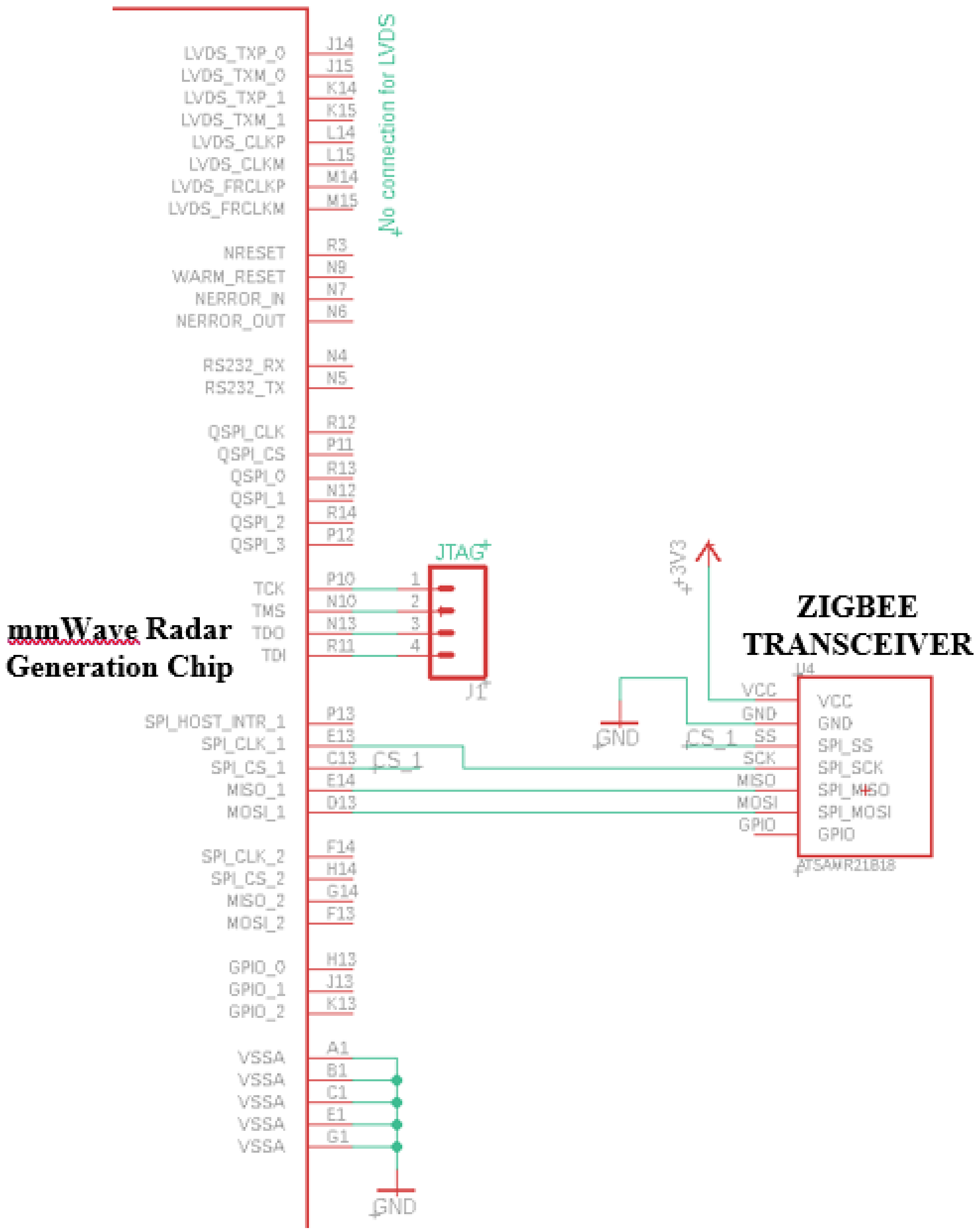


Figure 10: Eagle Schematic for AWR1642 (3)

Pins without connections shown are NC.

[GMS]

3.4.2 Wireless Link and Warning Indicator Schematic Design

For the warning indicator circuit, the schematic was designed to implement the level 2 block diagram which is shown in Figure 4. The warning indicator system will have a ZigBee Transceiver. After searching for an appropriate ZigBee module, the Atmel ATSAMR21B18-MZ210PA was selected because the chip is already equipped with a transmit/receive antenna, the temperature range is -40˚F to 257˚F, and the chip interface can be implemented with SPI. The microprocessor for this portion of the system needs to have SPI capabilities, have multiple analog to digital converters (ADCs) and have PWM outputs. These parameters are best fulfilled by the Microchip PIC16(L)F1503 microcontroller, this microcontroller has 8 ADCs, 4 PWMs, and the required SPI connections. The operating voltage of the microcontroller is 1.8 V to 3.6 V, and the ZigBee module has an operating voltage range of 2.7 V to 3.6 V. The warning indicator system includes a voltage regulator, the regulator must have an input of 12 or 24 Volts and the output must always be 3.3 V to supply power to the ZigBee module and the microprocessor. The voltage regulator schematic is shown in the next section 3.4.3. However, there is another voltage regulator that will be used when the input voltage is 24 V to step down to the appropriate 12 V for the warning LED. However, if the power supply voltage is already 12 V the circuitry will allow for the regulator to be ignored because the voltage regulation will be unnecessary. The microcontroller and the ZigBee modules are connected according to the data sheets [13, 15] (excerpts of the data sheets are in the appendix).

[JWH]

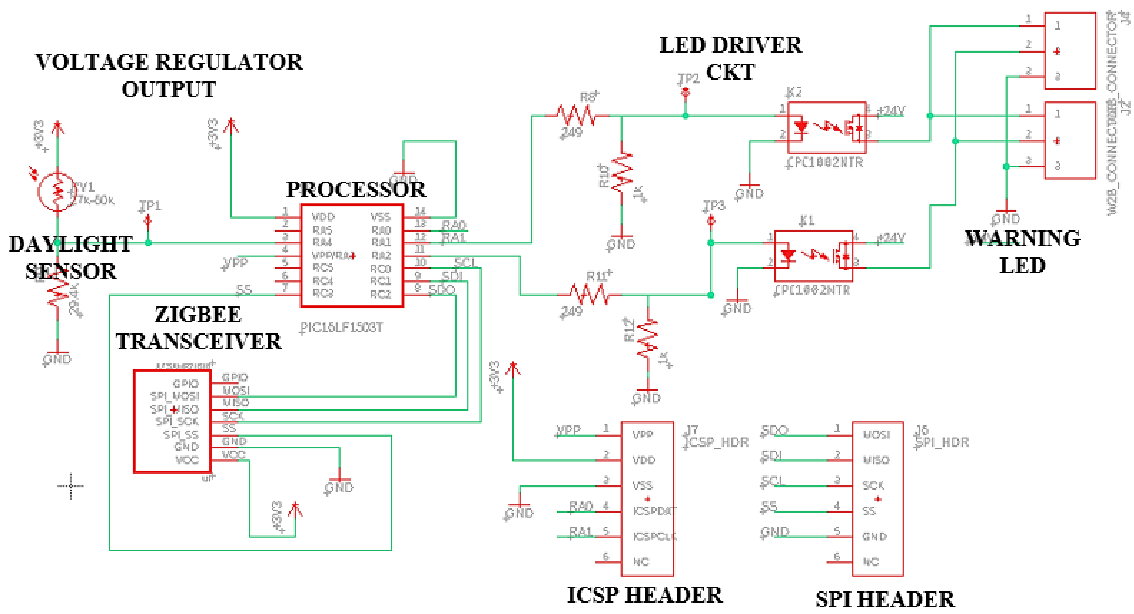


Figure 12: Warning Indicator System Schematic

For the daylight sensor, a photoresistor voltage division circuit will be used as shown in the circuit above. The chosen photoresistor has a resistance range from 27 kΩ to 60 kΩ. The voltage divider resistor value is 29.4 kΩ, which is almost the lower range of the photoresistor (27kΩ). When the photoresistor resistance is at a minimum, the output voltage will be approximately half the rail voltage. For the LED Driver Circuit, two relays are used one will turn on the LED "bright mode" during the daytime. The second relay will turn on the LED "low brightness mod" when it is dark out to reduce the glare. The relays are solid state relays to enable a fast switching operation. The ZigBee transceiver is wired to the microcontroller using the SPI capable ports for the clock (SCL), input data (SDI), output data (SDO) and the slave select (SS). The voltage regulator output voltage is used as the top rail voltage of 3.3 V for the warning indicator system. The voltage regulator schematic is shown in section 3.4.3, the regulator circuit is used for both the radar system module and the warning indicator system module.

To avoid another voltage regulator for the LED warning light a variable input voltage LED was chosen. The TopPower LT-TP-0111 LED was chosen because it has the desired input voltage range from 10 V to 30 V. Therefore, no voltage regulation is required to supply power to the LED, because the system will run off either a 12 V or 24 V battery. The chosen LED lights have a diameter of 4" which will meet the visibility requirement of the warning indicator. The LED is a common product for school bus brake lights, and the lights are appropriately rated for outdoor use in all weather conditions.

[JWH]

3.4.3 Buck Converter Schematic Design

After determining the system parameters, a buck converter was set up to take a 12V or 24V input and step it down to 3.3V. The front end and back end of the system will have the same buck converter design to implement a 3.3V rail. Component parameters have been calculated in accordance with information from the datasheet. A full walkthrough of this process is shown in section 3.3.3.

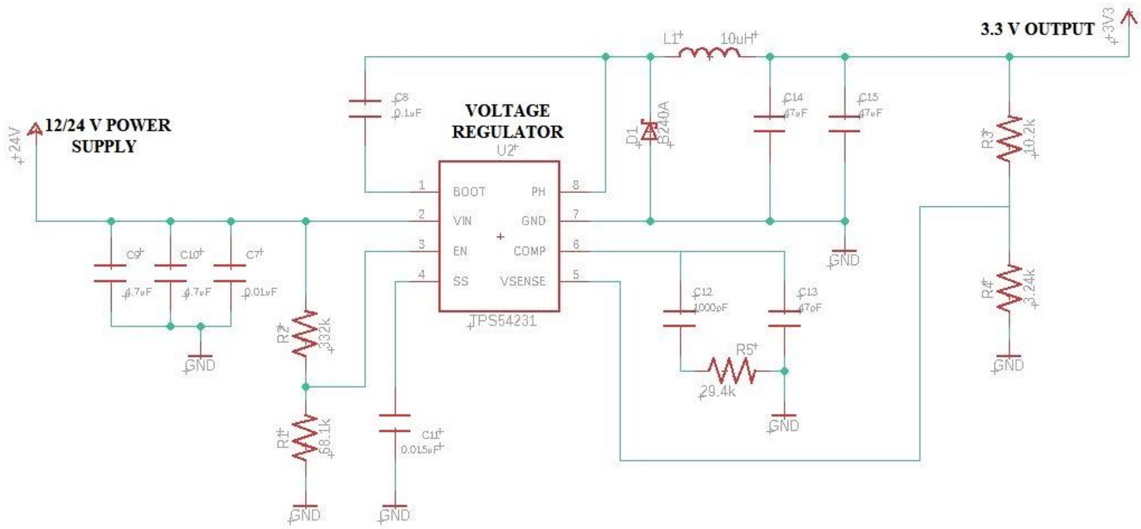


Figure 13: Voltage Regulator - Buck Converter 12/24 V Supply to 3.3 V Output

[GMS]

3.4.4 1.8V Low Drop Out Circuit

Powering the AWR1642 MMIC requires careful consideration of low noise voltage references and supplies. As such, an LDO regulator is required to ensure proper operation of the device. The TPS7A8101 satisfies the requirements and can provide up to 1A of current and makes use of a precision voltage reference with feedback to provide a clean output with appropriate capacitor selection. Less than 25uVrms of noise is acceptable [16].

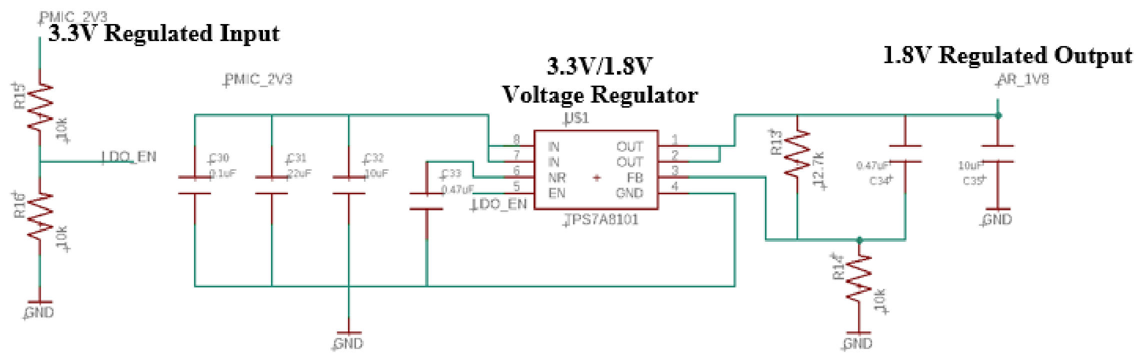


Figure 14: 1.8 V LDO Regulator Circuit

[GMS]

3.4.5 1.3V LDO Regulator

Isolated voltage references are required on the AWR1642. The TPS7A8801 provides independent 1.3V outputs that can be used repeatedly throughout the front-end schematic. Again, low noise is a stringent requirement for the MMIC. An output with less than 5uVrms noise is achievable under best case conditions. This is one of only a handful of devices available on the open market that can allow the LVDS lines and ADCs of the MMIC to function correctly without incurring noise that impedes measurement accuracy [17].

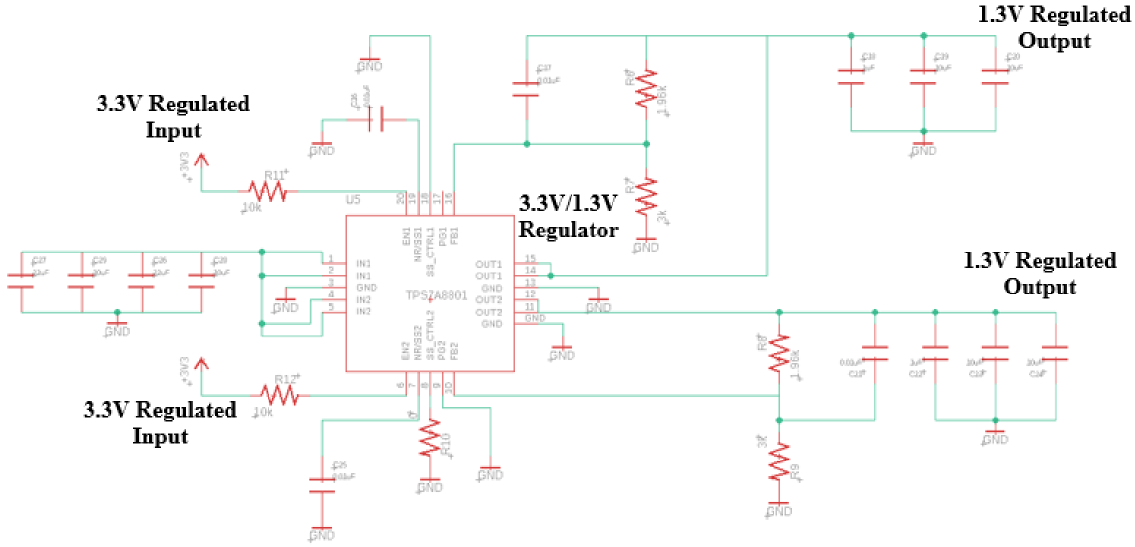


Figure 15: 1.3V LDO Regulator Circuit

[GMS, JJH]

3.4.5 Front End Schematic for Lidar Module

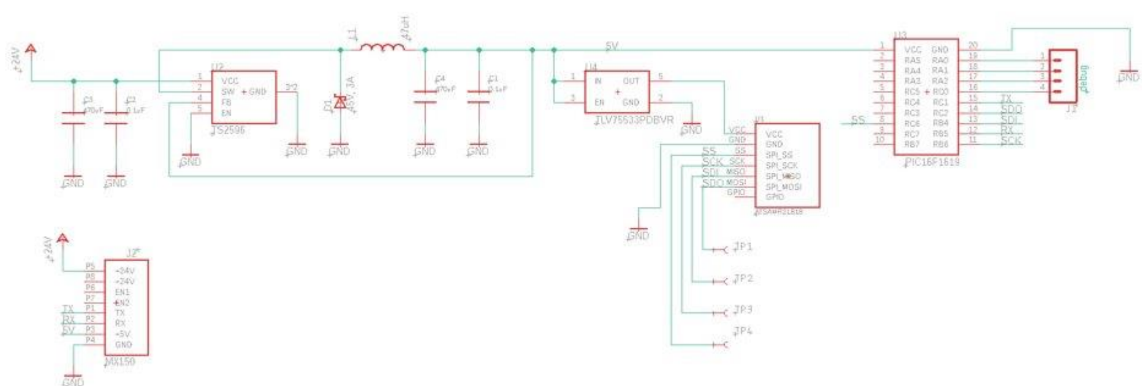


Figure 16: Schematic for Front-End PCB with Lidar Module connection

A new PCB was designed to make use of a Lidar module. A buck converter handles input voltage from 8V to 30V and currents in excess of 3A. A 3.3V linear regulator takes the 5V output of the buck converter and supplies power to the Zigbee module. A PIC16F1619 handles all UART data received from the Lidar module and calculates the distance at which objects are being detected. Upon detection, a signal is sent to the back-end PCB to turn on the warning indicator. An automotive connector handles all inputs and outputs on the board so that it can be sealed off from environmental conditions if necessary.

3.5 Software Design Module Pseudocode

There are three main sections of code for the system: the radar module code, the ZigBee module code, and the warning indicator module code. The radar module code will use the API in the mmWave Real Time Operating System (RTOS), provided by TI. This code will be executed on the AWR1642. The code for the ZigBee module will use the ZigBee transmitter to send information to the warning indicator. This code will also be executed on the AWR1642. The warning indicator code will handle receiving the incoming data from the ZigBee module on the main module and make decisions about the data received. This code will be executed on the chosen PIC16 chip. This module will have its own ZigBee transmitter and receiver in order to communicate with the main module.

3.5.1 Radar Module Pseudocode

```
// pseudocode for radar chip module
// this is a basic idea about the task that will be running on the chip to continuosly collect and
// send radar data
int main()
{
    mainData_t data;

    // Pass in the config file
    inputConfigFile();

    // This will create task that will initialize the drivers and module,
    // and initialize the 3 main tasks
    mssInitTask();

    // Read raw data from sensors
    executeAPI();
    radarData_t sensor_data = sensorStart();

    // Put it through the 1D FFT
    rangeData_t range_data = rangeFFT(sensor_data);

    // Put it through the 2D FFT
    dopplerData_t doppler_data = dopplerFFT(range_data);

    // Detect objects
    data = detection();
    data = angleEstimation();

    // stop collecting so we can send
    sensorStop();

    // Send data out to serial for now
    sendSerial(data);

}

void inputConfigFile()
{
    char[50] path = "config/file/path..."
    // configuration options in file:
    // dfeDataOutputMode, channelCfg, adcCfg, adcbufCfg, profileCfg, chirpCfg, bpmCfg, lowPower, frameCfg
    // advFrameCfg, subFrameCfg, guiMonitor, cfarCfg, peakGrouping, multiObjBeamForming, calibDcRangeSig
    // extendedMaxVelocity, clutterRemoval, compRangeBiasAndRxChanPhase, measureRangeBiasAndRxChanPhase,
}
```

```
// pseudocode for radar chip module
// This is a basic idea about the task that will be running on the chip to continuously
// collect and send radar data.


Pass in configuration file. The configuration file will tell the processor which
options we want and will set initial values

Initialize the main task. This task will kick off 3 separate tasks in the API
that will start up the relevant modules and drivers for the radar chip.

Read raw data from radar sensors.

Put collected data through the one-dimentional FFT.

Put data returned from one-dimentional FFT through a two-dimentional FFT.

Call internal API functions to determine if objects are in the detection zone.

Stop collecting data so that we can send it to relevant places.

For now, we will send via serial for debugging.
```

The above code is pseudocode for the radar module. The radar module code must be started in a task, which will run in an infinite loop looking for data and processing data. To start the task, a configuration file must be passed into the code. The code will read the configuration file and determine what options are set. Some of the options that can be set are: channel configuration, chirp configuration, low power mode, and data output mode. Once these options are set, the main task can be started. This task will begin by collecting raw data from the sensors. Then it will put the raw data through two FFT functions, which will determine range and objects detected. Finally, the task will send the data out for evaluation.

[ANR]

3.5.2 ZigBee Module Pseudocode

```
// pseudocode for zigbee module
int main()
{
    // choose initial config settings for spi
    SPI_MasterModeParams_t = {bitRate, numSlaves, t2cDelay, c2tDelay, wDelay};
    SPI_SlaveModeParams_t = {dmaCfg, chipSelect};

    // configure the handle pointing to device channel and message for spi.
    // this will open the spi channel
    SPI_handle handle;
    char[1] message;

    // initialize the spi pin
    SPI_init();

    // determine what warning to send
    if (warning_on == true)
    {
        message = 1;
    }

    else if (warning_on == false)
    {
        message = 0;
    }

    // transfer the message then close the spi channel
    SPI_transfer(handle, message);
    SPI_close();
}
```

```
// pseudocode for Zigbee module on main chip.

Choose initial config settings for SPI. We must choose SPI Master mode paramaeters
and SPI Slave mode parameters.

Configure handle pointing to the correct device channel. TI's API requres an
internal data type to be initialized in order to be able to reference the same
SPI channel. This useful because the chip has multiple SPI channels, and they
both can be configured with separate values.

Initialize the pin we are using for SPI.

Recieve data from radar module.

// The message is transmitted using the SPI handle configured above.
IF radar module data determined there is a vehicle approaching
THEN send "on" message to the warning indicator

IF radar module data determined there is no vahicles approaching
THEN send "off" message to the warning indicator

Wait for acknowledgement from warning indicator so that we know indicator has
received message.

Close the SPI channel to indicate that we have no more messages to send.
```

The above code is pseudocode for the ZigBee module in the system. Before starting, the ZigBee module must be passed parameters for master mode and slave mode. The master mode parameters consist of: bit rate, number of slaves, and delay. The slave mode parameters consist of: direct memory access configuration, and chip select. Once these parameters are configured, the module can start. This is also handled in a task. The task starts by opening the handle for the SPI channel selected. Then, a message is configured to be sent. This message will be a simple on/off to send to the warning indicator. If the warning indicator should be on, the message is a 1. If it should be off, the message is a 0. Finally, the message is transferred using the handle configured earlier. Then the channel must be closed.

[ANR]

3.5.3 Warning Indicator Module Pseudocode

```
int main()
{
    // the array of pins to turn on for the warning indicator
    char led_pin_array[50];
    char message[10];
    // create an infinite loop to continuously check for data
    for(;;)
    {
        // we receive a parity bit, the data, and a stop bit
        message = SPI_receive();

        // get the data bit
        char warning_on = message[1];

        if(warning_on == 1)
        {
            // turn on each led in the pin array
            for(int i = 0; i <= led_pin_array.size(); i++)
            {
                led_pin_array[i] = 1;
            }
        }

        send_ACK();

        if(warning_on == 0)
        {
            // turn off each led in the pin array
            for(int i = 0; i <= led_pin_array.size(); i++)
            {
                led_pin_array[i] = 0;
            }
        }
    }
}
```

```
// pseudocode for warning indicator module

Set up an array of pins to toggle for the warning indicator. The idea is to
only turn on the selected pins when we receive an "on" message.

// Start infinite loop
FOR(;;)

Receive a message from the SPI channel via ZigBee

Extract data from the message. The message is sent with a parity bit and a stop bit.
We must ignore these bits because they are not part of the data message. The parity
bit is used to make sure the message reaches the destination intact. The stop bit
indicates that me sender is done sending messages for now.

IF we receive an "on" message
THEN iterate through all pins in our pin array and set them to "on"

IF we receive an "off" message
THEN iterate through all pins in our pin array and set them to "off"

Send acknowledgment to indicate that we have received the message and
successfully changed the warning indicator status.
```

The above code is pseudocode for the warning indicator module. The warning indicator will consist of an array of LEDs. In the code, it will choose which LEDs to turn on and off by defining them in a character array. An infinite loop will need to be started to continuously check for data. The message that is received from SPI will consist of a parity bit and a stop bit. These bits are not related to the actual message contents, so only check the message bits. If the message received was a 1, loop through each LED in the LED array and turn them on. If the message was a 0, loop through each LED in the LED array and turn them off. Then send an acknowledgement back to the main module to let it know that the LEDs were changed.

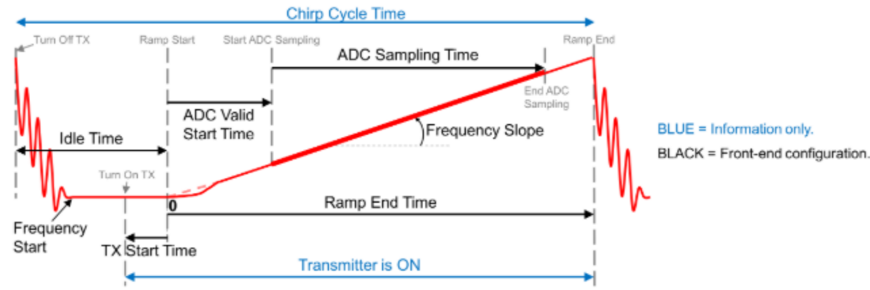
[ANR]

3.5.4 Notes on Chirp Sequencing and Ranging

Outputs and Chirp Design

Detectable Object Range

Max Range for Typical Detectable Object (m)	134.83	Min RCS Detectable at Max Range (m^2)	7.66



Chirp Configuration Parameters

Frequency Start (GHz)	76	# of Chirp Loops	508
Frequency Slope (MHz/us)	3.48	Frame Periodicity (ms)	100
Frequency Slope Constant	72	Idle Time (us)	2
Sampling Rate (ksps)	3862	ADC Valid Start Time (us)	5.20
# of Samples per Chirp	124	Ramp End Time	38.30

Information Only Parameters

Bandwidth (MHz)	133.15	# of Range FFT Bins	128
Beat Frequency (MHz)	3.48	# of Doppler FFT Bins	512
Chirp Cycle Time (us)	40.30	Range Interbin Resolution (cm)	130.78
Chirp Repetition Period (us)	40	Velocity Interbin Resolution (m/s)	0.10
Active Frame Time (ms)	20.32	Radar Cube Size (KB)	762

```
No errors are found.
```

Figure 16: Texas Instruments Software for mmWave Radar Chirp Calculations

After speaking with a Texas Instruments Field Applications Engineer, the above chirp configuration was verified to be a viable option on the AWR1642. The theoretical range of the above configuration is approximately 134m or rather 440 feet, well beyond the 355-foot design requirement. Parameters were configured using information gathered from TI's mmWave Training Series materials.

[GMS]

3.6 Software Design Final Code

3.6.1 Front End PIC16F1619 Code

Pictured below is the main C code file for the Front End PCB.

```c
/*
 * File:   main.c
 * Author: DT09
 *
 * Created on March 6, 2019, 3:09 PM
 */

#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include "mcc_generated_files/pin_manager.h"
#include "spi.h"
#include "mcc_generated_files/eusart.h"
#include "mcc_generated_files/mcc.h"

/*
 *
 */
int main(int argc, char** argv)
{
    // Initialize system and clock
    SYSTEM_Initialize();
    OSCILLATOR_Initialize();

    // Initialize the pin manager
    PIN_MANAGER_Initialize();
    __delay_ms(10);


    // Initialize UART
    EUSART_Initialize();
    __delay_ms(10);

    int x = 0;
    uint8_t data[8];
    memset(data, 0, sizeof(data));

    TRISAbits.TRISA3 = 0;
    TRISAbits.TRISA4 = 0;
    TRISAbits.TRISA5 = 0; // set pin as output
```

Figure 17: Detection Module Main Function (1 of 2)

```
    // Begin infinite loop
    while(1)
    {
        data[x] = EUSART_Read();

        uint8_t header = 0x59;
        uint8_t  zero = 0x00;

        // 32 cm
        //uint16_t feet = 0x0020;

        // 20 ft
        uint16_t feet = 0x0299;

        // 365 ft
        //uint16_t feet = 0x2A62;
        x++;

        if (x == 8)
        {
            x = 0;

            for (int n = 0; n < 8; n++)
            {
                if ((data[n] != header) && (data[n] != zero))
                {
                    uint16_t distance = (data[n+1] << 8) | data[n];

                    // checksum
                    if((data[n - 1] == zero) && (data[n] != zero))
                    {
                        LATAbits.LATA5 = 0;
                        PORTAbits.RA4 = 0;
                    }


                    else if (distance < feet)
                    {

                        LATAbits.LATA5 = 1;
                        PORTAbits.RA4 = 1;
                    }

                    else
                    {
                        LATAbits.LATA5 = 0;
                        PORTAbits.RA4 = 0;
                    }
                }
            }
        }
    }

    return (EXIT_SUCCESS);
}
```

Figure 18: Detection Module Main Function (2 of 2)

This code handles the initialization of the entire system, as well as the EUSART (UART) module. After the initializations, an unsigned 8-bit character array is defined. This array will hold 9 bytes; enough to contain the entire data packet received from the lidar. The array is set to zero using the memset C library function. The Tri-State registers for the pins LATA5 and RA4 are set to 0, indicating that these pins should act as outputs. The LATA5 pin controls an LED on the microcontroller, and the RA4 pin is a standard GPIO pin. Once the infinite loop begins, the data array is filled with the 9 bytes received from the lidar. When the bytes are received, the data is filtered so that only the distance high and distance low values are examined. The distance high value is then bit-shifted by 8 to position it before the distance low, so the distance can be evaluated correctly. If the distance detected is less than the detection value set, this means there is an object in the detection range. The LATA5 pin and the RA4 pin are set to high. This tells the warning indicator to turn on the LED.

Table 9: TF03 Serial Data Frame Format

Data bit	Definition	Description
Byte0	Frame header	0x59
Byte1	Frame header	0x59
Byte2	DIST_L	DIST low 8 bits
Byte3	DIST_H	DIST high 8 bits
Byte4	Reserved	/
Byte5	Reserved	/
Byte6	Reserved	/
Byte7	Reserved	/
Byte8	Checksum	Low 8 bits of Checksum, Checksum = Byte0 + Byte2 +... + Byte7

The data received from the lidar is transmitted 9 bytes at a time. The first two bytes are the header bytes, indicating the start of a new data packet. Byte 2 and byte 3 are the distance low and distance high bytes respectively. These indicate how far away the detected object is from the lidar module. Bytes 4 through 7 are reserved, and always transmitted as 0. The final byte is the checksum byte; it is used to ensure the entire packet was received correctly.

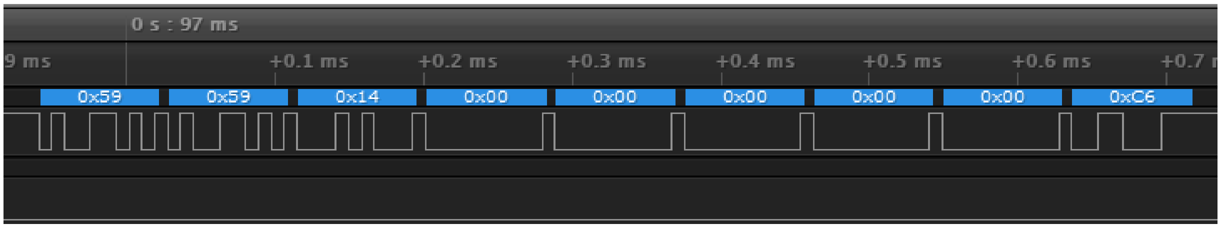


Figure 19: LiDAR Received Data Logic Analyzer Waveform

Pictured above is an example of a data packet received from the lidar. In this case, the distance value was 0x14 with no data in the distance high value. This means that the detected object was approximately 20cm away from the lidar.

[ANR]

3.6.2 Warning Indicator (Back End) PIC16LF1503 Code

The configuration header code used for the PIC16LF1503 is shown in the figure below.

```
1  //config.h
2
3  // PIC16LF1503 Configuration Bit Settings
4
5  // 'C' source line config statements
6
7  // CONFIG1
8  #pragma config FOSC = INTOSC    // Oscillator Selection Bits
9                                  //(INTOSC oscillator: I/O function on CLKIN pin)
10 #pragma config WDTE = OFF       // Watchdog Timer Enable (WDT disabled)
11 #pragma config PWRTE = OFF      // Power-up Timer Enable (PWRT disabled)
12 #pragma config MCLRE = ON       // MCLR Pin Function Select
13                                 //(MCLR/VPP pin function is MCLR)
14 #pragma config CP = OFF         // Flash Program Memory Code Protection
15                                 //(Program memory code protection is disabled)
16 #pragma config BOREN = OFF      // Brown-out Reset Enable (Brown-out Reset disabled)
17 #pragma config CLKOUTEN = OFF   // Clock Out Enable (CLKOUT function is disabled.
18                                 // I/O or oscillator function on the CLKOUT pin)
19
20 // CONFIG2
21 #pragma config WRT = OFF        // Flash Memory Self-Write Protection (Write protection off)
22 #pragma config STVREN = OFF     // Stack Overflow/Underflow Reset Enable
23                                 // (Stack Overflow or Underflow will not cause a Reset)
24 #pragma config BORV = LO        // Brown-out Reset Voltage Selection
25                                 // (Brown-out Reset Voltage (Vbor), low trip point selected.)
26 #pragma config LPBOR = OFF      // Low-Power Brown Out Reset
27                                 // (Low-Power BOR is disabled)
28 #pragma config LVP = OFF        // Low-Voltage Programming Enable
29                                 // (High-voltage on MCLR/VPP must be used for programming)
30
31 // #pragma config statements should precede project file includes.
32 // Use project enums instead of #define for ON and OFF.
33
34 #include <xc.h>
```

Figure 20: PIC16LF1503 Configuration File

For the warning indicator the internal oscillator of the PIC controller is used as shown in line 8 of the code above.

```c
/*
 * File:   main.c
 * Author: Jonathan Hall
 * Created on March 28, 2019, 11:09 AM
 */

#include "xc.h"
#include "config.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int vehicle_detected = 0, Brightness = 1;

void ms_delay(int N)
{
    T1CON = 0x0073; // TON, Prescale 8
    TMR1 = 0; // Clear Timer1
    while(TMR1 < 62.5*N) // wait for TMR1 to count up.
        {
        }
}
```

Figure 21: Warning Indicator Microsecond Delay Function & Global Variables

The code above includes the configuration header, the standard library and Input output headers, and the string header. Two global integer variables are declared: "vehicle_detected" is initialized to 0 and "Brightness" is initialized to 1. The millisecond delay function is defined with an integer argument, N. Timer 1 is used for this function and the prescale value is set to 8. The timer is cleared initially and increments until the timer reaches the desired millisecond delay.

[JWH]

```
unsigned int readADC(int ch) {
    ADCON1bits.ADFM = 1; //right justify results, set to zero for left justification
    ADCON1bits.ADCS= 0b111; //TOSC*64 conversion speed
    ADCON0bits.CHS = ch;
    ADON = 1; //enable ADC
    ms_delay(2);
    ADCON0bits.GO = 1;
    while(ADCON0bits.GO == 1);
    unsigned int result = (ADRESH << 8) + ADRESL;
    return result;
}
```

Figure 22: ADC Function Defined

The code above declares the analog to digital converter function for a channel value that is passed as an integer argument. The oscillator is setup with a 64 multiplier to increase the speed. The Analog-to-Digital channel is set to the channel called in the main function. The ADC is then enabled and continues until the reading has completed. The ADC is used to determine the brightness of the surrounding environment by reading the voltage value from a photoresistor in a voltage divider circuit.

[JWH]

```
int main(void) {
    TRISA = 0;  //Set PORTA as an output
    TRISAbits.TRISA5 = 1;  //Set PORT RA5 as Input
    ANSELA = 0;
    PORTAbits.RA0 = 0;          // Set Warning Indicator Light Low
    PORTAbits.RA1 = 0;
    while(1){
        int adcVal = readADC(3);        //get ADC reading from RA4 / AIN3
        char buffer[5];
        itoa(adcVal, buffer);
        if(PORTAbits.RA5==1)
        {
            vehicle_detected = 1;
        }
        else
        {
            vehicle_detected = 0;
        }
```

Figure 23: Main Function (1 of 3)

The main function begins by setting all PORTA pins to outputs using TRISA. Pin RA5 is set to an input by setting its tri-state (TRISA) value to 1. Both the high and low brightness modes for the warning indicator are initially set low. In the while loop the ADC is set up and read and the value is declared as adcVal. The processor reads from the input RA5 to determine if a vehicle has been detected by the LiDAR system. If the pin RA5 is high, then the global variable "vehicle_detected" is set to 1. Otherwise, the global variable "vehicle_detected" is set to 0.

```
        if(adcVal>=0x5)                    //Determine Brightness
        {
            Brightness = 1;                    //Set Warning Indicator Brightness High
        }
        if(adcVal<0x5)                    //Determine Brightness
        {
            Brightness = 0;                //Set Warning Indicator Brightness Low
        }
        if(vehicle_detected == 1)
        {
            if(Brightness == 1)
            {                                  //High Brightness Mode
                PORTAbits.RA1 = 1;             // Set Warning Indicator Light High
                ms_delay(100);
                PORTAbits.RA1 = 0;             // Turn off Warning Indicator Light Off
                ms_delay(100);
                PORTAbits.RA1 = 1;             // Set Warning Indicator Light High
                ms_delay(30);
                PORTAbits.RA1 = 0;             // Turn off Warning Indicator Light Off
                ms_delay(30);
            }
```

Figure 24: Main Function (2 of 3)

If ADC value that is read (adcVal) is greater than the hexadecimal value of 5, then the global variable "Brightness" is set high (1). However, if the variable "adcVal" is less than the compared value, then the variable "Brightness" will be set low (0). Now if a vehicle has been detected, then it enters the next loop. If the brightness variable is found to be high, then the pin RA1 is toggled high which will turn on and off the relay for the high brightness mode.

```
            else                            //Low Brightness Mode
            {
                PORTAbits.RA0 = 1;          // Set Warning Indicator Light High
                ms_delay(100);
                PORTAbits.RA0 = 0;          // Turn off Warning Indicator Light Off
                ms_delay(100);
                PORTAbits.RA0 = 1;          // Set Warning Indicator Light High
                ms_delay(30);
                PORTAbits.RA0 = 0;          // Turn off Warning Indicator Light Off
                ms_delay(30);
            }
        }
        else
            {
            PORTAbits.RA0 = 0;              // Turn off Warning Indicator Light Off
            PORTAbits.RA1 = 0;              // Turn off Warning Indicator Light Off
            }
    }
}
```

Figure 25: Main Function (3 of 3)

If a vehicle has been detected and the brightness variable is found to be low, then the pin RA0 is toggled high which will turn on and off the relay for the low brightness mode. However, if no vehicle has been detected then the warning indicator will be off.

[JWH]

3.7 Antenna Design

3.7.1 Antenna Introduction

To obtain the required detection range with the radar generator chip, a custom antenna is to be designed for the front-end radar module. While the sensor is responsible for encoding the signal information and into a radio frequency (RF), the antenna will wirelessly transmit the signal. This is a part of the project that will need to be designed and simulated before building and testing. MATLAB’s Antenna Toolbox will be used to implement design specs. This software allows parameters such as the size, material, and shape of the antenna to be controlled and tested. The materials and size are limited by the chip. Upon testing, the far field patterns and resonant frequency can be viewed ensuring that the design lines up with the requirements for this project. Once the parameters of the antenna have been simulated and confirmed, the antenna will be implemented on the chip.

[JJH]

3.7.2 Antenna Design and Analysis: Choosing a Design

The first step in the antenna design project was choosing what type of antenna would most sense in terms of size, capability, and convenience. After analyzing every option, a microstrip patch antenna was chosen due to its practicality. The advantages of choosing a microstrip patch antenna were: the ability to easily adjust parameters, it was convenient to mount on a flat surface/PCB, and it required a less in-depth analysis than using an antenna array. The only disadvantage was that it would be slightly less accurate than utilizing an array. After this, a feeding method needed to be selected. The group decided on using an inset-fed antenna because the other options, such as microstrip line feed or a coaxial feed, were more either more difficult to implement or less accurate. The advantages of an inset feed were: the fact that it could be placed directly on the board and the feed allowed for impedance matching.

[JJH]

3.7.3 Antenna Design and Analysis: Design Requirements and Equations

The next step was to build the antenna using the simulation software in order to verify accuracy of the design. Automotive radars typically use antennas that operate between 77-81 GHz, so the operational design frequency was chosen to be 79 GHz and was used in the parameter calculations. The material used for the antenna was copper and the substrate (PCB) was RO4350b. The permittivity of RO4350b is 4.4, the PCB thickness is approximately 0.3 mm. Using these known values, the rest of the parameters could be calculated using the following equations:

$$Width = \frac{c}{2f_0\sqrt{\frac{\varepsilon_R + 1}{2}}} \quad \ldots (1)$$

$$\varepsilon_{eff} = \frac{\varepsilon_R + 1}{2} + \frac{\varepsilon_R - 1}{2}\left[\frac{1}{\sqrt{1 + 12\left(\frac{h}{W}\right)}}\right] \ldots (2)$$

$$Length = \frac{c}{2f_0\sqrt{\varepsilon_{eff}}} - 0.824 \times h\left(\frac{\left(\varepsilon_{eff} + 0.3\right)\left(\frac{W}{h} + 0.264\right)}{\left(\varepsilon_{eff} - 0.258\right)\left(\frac{W}{h} + 0.8\right)}\right) \ldots (3)$$

Where equation (1) is the antenna width, equation (2) is the effective dielectric, and equation (3) is the antenna length. Additionally, in these equations c is the speed of light, \epsilon sub r is the permittivity of the dielectric, and f sub o is the resonant frequency. Using the known values in the above equations resulted in the following parameters: $Antenna\ Width = 1.014\ mm$, $\varepsilon_{eff}(effective\ dielectric) = 3.1199$, and $Antenna\ Length = 0.5774\ mm$.

[JJH]

3.7.4 Antenna Design and Analysis: Simulation and Results

After all the parameters were found, the next step was to write code in MATLAB that would simulate the design to the specifications provided. The following code was used to create a simulation of the antenna and capture results:

```
% Design Team 09
%Antenna Design

load insetfeedpatchmesh
T=triangulation(t(1:3,:)',p');
figure
triplot(T)
axis equal
grid on
xlabel('x')
ylabel('y')
c=customAntennaMesh(p,t);
createFeed(c,[-0.045 0 0],[-0.0436 0 0])
show(c)

%Provide groundplane for patch antenna

r=reflector;
r.Exciter=c;
r.GroundPlaneLength=20e-2;
r.GroundPlaneWidth=15e-2;
r.Spacing=0.381e-3;
show(r);

%Assign Substrate to Patch Antenna

d=dielectric;
d.Name='RO4350b';
d.EpsilonR=4.4;
d.LossTangent=0.0009
r.Substrate=d;
show(r)



%Parameters of the Patch Antenna

insetpatch=patchMicrostripInsetfed;
insetpatch.Length=.0005774;
insetpatch.Width=0.001155;
insetpatch.Height=0.0003;
load insetfeedpatch

```

Figure 26: Matlab Antenna Design Simulation Code (1 of 2)

```
%Calculate S-Parameter of the Patch Antenna

freq=linspace(78.85e9,78.95e9,21);
s=sparameters(r,freq,50);
figure
rfplot(s,1,1);



%DBi Pattern
pattern(insetpatch,79e6,0,1:1:360)
```

Figure 27: Matlab Antenna Design Simulation Code (2 of 2)

This code yielded the following results:

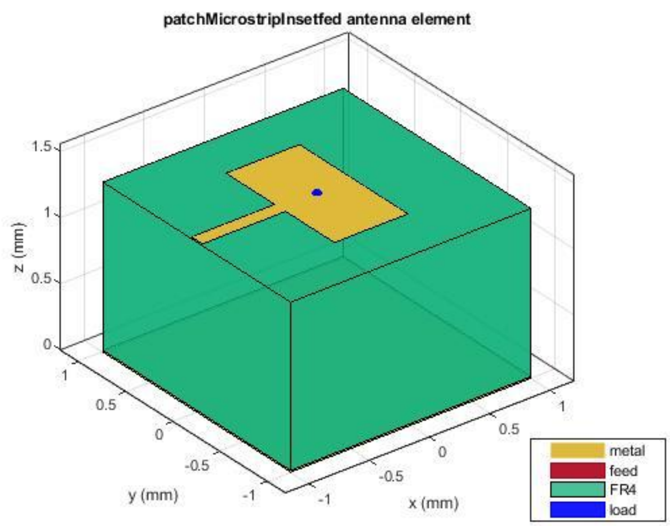


Figure 28: Image of Simulated Patch Antenna

The results above indicate what the antenna would behave like if implemented. The first figure shows what the antenna would look like mounted on a PCB along with the inset feed.

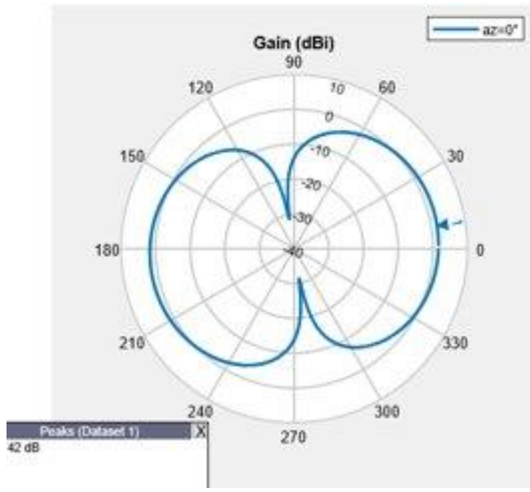


Figure 29: Image of Antenna Radiation Pattern

The second figure shows the radiation pattern of the antenna which indicates the antenna's directivity. For the purpose of long-range detection, this radiation pattern is satisfactory.

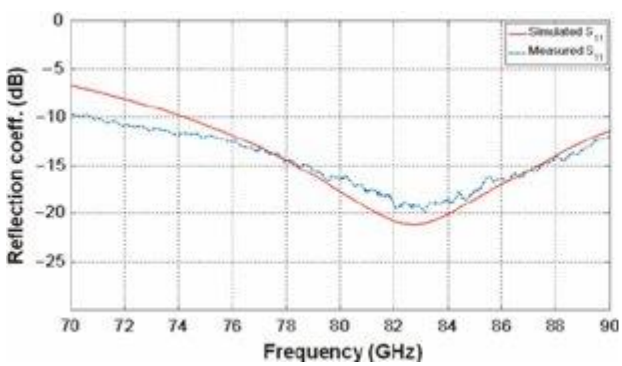


Figure 30: Image of S1, Pattern Simulated Without Noise (Red), with Noise (Blue)

The figure above shows the operational frequency of the antenna simulated with noise and without. With noise, the resonance is slightly less sharp, but still acceptable. The antenna operates at about 82 GHz which is slightly higher than calculated but should still be within an appropriate range for long distance detection.

[JJH]

3.7.5 Antenna Design and Analysis: Conclusion

While the simulation of the antenna operated approximately as expected, the group decided to go in a different direction with the project due to complications with such a high frequency design. An antenna of this operational frequency is necessary for Radar applications, because of this it was necessary to look for alternative options for detection. LiDAR detection was a reasonable alternative that does not require a high frequency antenna. This was the method the group selected for detection. Even though the microstrip patch antenna was not used, the theory behind it is very important to electrical engineers interested in electromagnetics and signal processing. It was a beneficial exercise in design and a worthwhile portion of the senior design experience.

[JJH]

3.8 Board Layout Design

3.8.1 Back End Warning Indicator PCB Layout

Below is the back-end PCB layout from Eagle, which was used in the implementation of the warning indicator. The antenna of the ZigBee Module is placed overhanging the edge to allow for the necessary clearance (required from the datasheet) to avoid interference. The power traces were set to 50 mil traces from the 24 V supply to the LED's and relays. The rest of the signal connections are 10 mil traces throughout the board. Vias are placed throughout the board to ensure a good connection from the top ground pour to the bottom side ground pour. The reference designators are the only labels remaining on the board layout, to find the values of each component reference the schematics above or the parts list below.

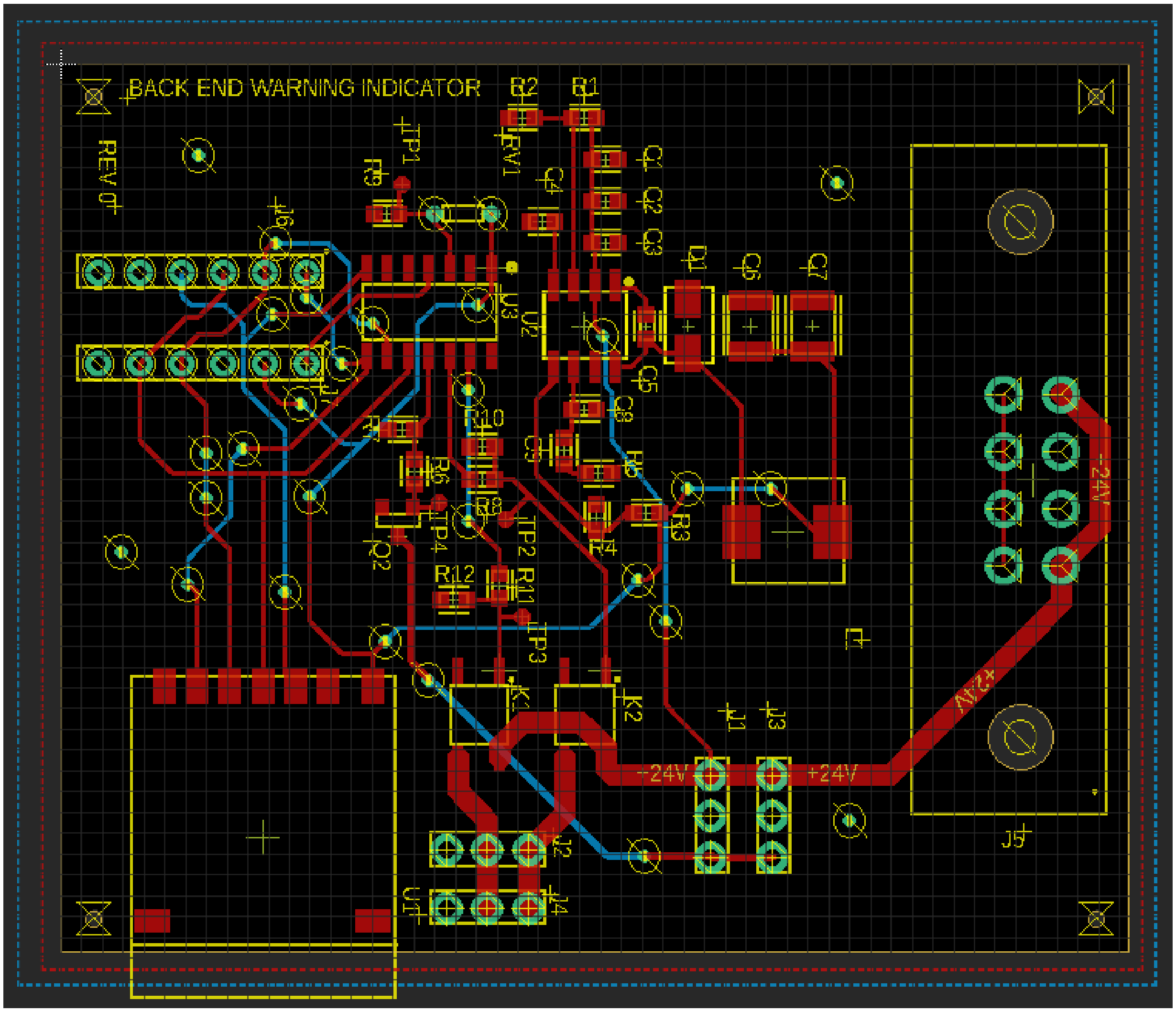


Figure 31: Warning Indiactor PCB Layout

3.8.2 Front End of SafePass with FMCW Radar

The front-end portion of the design contains four power stages, the mmWave radar section, and ZigBee transceiver. The radar antennas are strategically placed opposite the ZigBee module with separated ground planes and an isolation barrier between the two transmitters. All power management circuitry is located on the top layer of the board to minimize any EMI induced into lower layers that are responsible for powering the sensitive areas of the MMIC. Memory, mix-down staging, and raw ADC capture is intolerant to any low frequency noise generated from switching regulators and must be avoided through the use of linear regulators. To keep as much noise at bay as possible, the fifth layer is used to deliver power to areas of the chip requiring 1.8V and 1.2V.

The second layer of the board is a ground plane beneath the antennas for shielding purposes. Only the high frequency RF circuitry is grounded to this layer. Microvias are pattern stitched beneath the MMIC around the RF lines to provide a perfect 50ohm impedance match. Precise layout specifications are in accordance with Texas Instrument's layout validation guide. Notice that the signal paths themselves are setup in matched pairs so that higher noise rejection is possible using a combination of hardware and software processing techniques. Optionally, this setup also allows for ease of phase shift detection. Layers three and four are used as grounds for additional shielding purposes.

The bottom layer is a signal layer with ground pour that is used to route the main 3.3V supply voltage as well as the SPI bus and boot selector switches. Locating the ZigBee control lines on the bottom allows for signal integrity to remain intact while the MMIC is transmitting at full power on all TX lines. Verification of signal integrity was performed using an Altium Designer IBIS simulation in a sandboxed environment. Layer stackup impedances are a basic feature of Altium's software, thus eliminating the need to specifically simulate impedance matching characteristics.

Due to limitations in licensing, Eagle was used for the entire layout process of all PCBs created as part of this project. However, Altium Designer was utilized for the majority of design verification. Final ERC and DRC was run in Eagle using custom rulesets to account for difficulties in manufacturing. Minimum clearances were set to 4 mils with no minimum via size since errors often occur when blind, buried, and microvias are enabled.

The true minimum hole size was limited to 12mils to ensure no additional manufacturing costs.

Surface finish was selected to be silver immersion. Standard gold ENIG effectively destroys any useful RF performance at high frequencies, but it is still better than HASL. Silver immersion also has the added benefit of being extremely smooth, which is ideal for plating antenna pads as well as fine pitch BGA components.

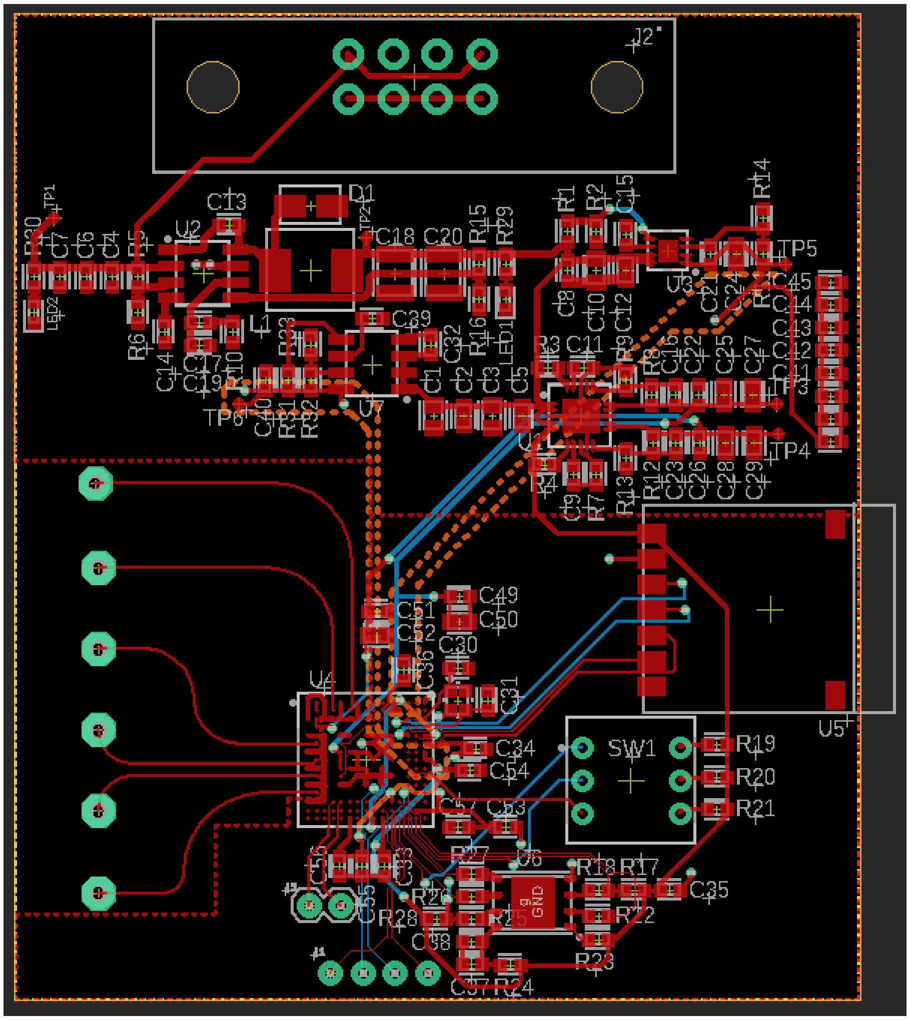


Figure 32: Front End PCB Layout

[GMS]

3.8.2 Final Front End of SafePass Using LiDAR

After a design change, the TF-03 LiDAR module was selected to replace the radar design. As a result, all that was then needed on the front end is a step-down converter, a microprocessor to accept the UART output of the LiDAR module, and a ZigBee transceiver to communicate with the back end of the system. The new board was designed with even greater robustness in mind given that automotive applications require high durability. A new buck converted with greater filtering was chosen. A Microchip

PIC16 series MCU was determined to be enough to handle the basic needs of the front end.

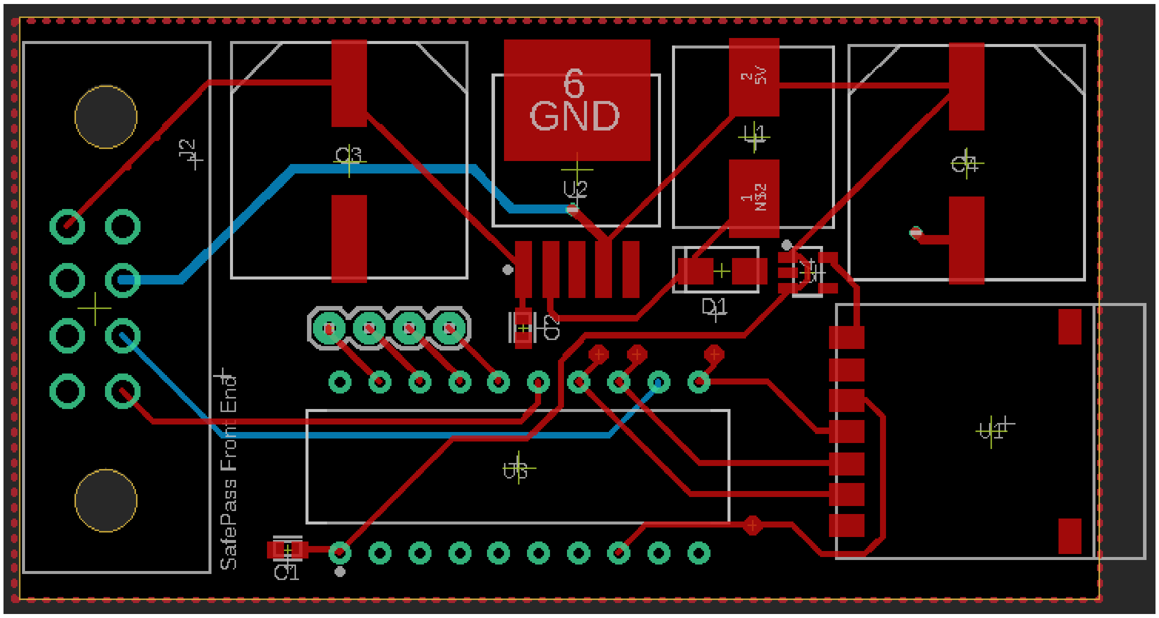


Figure 33: SafePass PCB for LiDAR Module

[GMS]

3.9 Physical implementation notes

In accordance with Texas Instruments' validation guide for RF PCB manufacturing, there are number of considerations that are to be considered when designing a circuit board. First and foremost, the antenna grid is heavily dependent on the substrate materials used for manufacture. Using FR4 with RO4835 is one of the recommended stack-up designs. Alternatively, RO3003 is also a secondary choice if the first combination is not available from a given manufacturer. Given that this design will be produced in small quantities, the higher manufacturing yield of RO4835 is not necessary, thus the RO3003 is preferable in this case given its lower electrical losses.

When connecting BGA pins from the AWR1642 to the PCB, transitions are expected to use a 50ohm impedance match. Signal pads are surrounded by a ground plane acting as a reference. Multiple ground planes can be connected together as needed with vias. The use of micro-vias and via-in-pad elements of design can make routing significantly easier, but it will limit the number of manufacturers that are capable of producing a board with extremely tight tolerances and small drill sizes. A number of board manufacturers use liquid photoimageable masks for etching, but the use of laser direct imaging for etching is recommended to preserve the necessary tolerances.

Selecting a solder mask will also affect the dielectric parameters of the board design. Transmission lines must not be covered by solder mask. Plating of the board also has significant effect on the RF characteristics. Standard ENIG (Gold) has too high of losses to be a viable option. Alternatively, immersion silver plating is an option with significantly better performance, but at the expense of allowing oxidation to occur. Fortunately, oxidation is completely invisible to the radar system and will not have any effect on overall performance while working on a prototype.

As a safety factor when working with a one-off design, the addition of extra heatsinking is a good design choice. While the AWR1642 is capable of operating without a heatsink depending on the chirp parameters being used and given optimal sinking to ground planes, an external heatsink will help protect the main component of the board without any drawbacks aside from physical form factor and minor cost addition.

[GMS]

3.10 Mechanical Sketch

The diagram depicts the operation of the system from the top view. The stopped vehicle has the LiDAR system mounted on the front of the vehicle. When an oncoming vehicle is detected within the 365-foot range the LiDAR system communicates wirelessly to the warning indicator system to turn on the light. The warning indicator is visible for 365 feet to approaching vehicles behind the stopped vehicle.

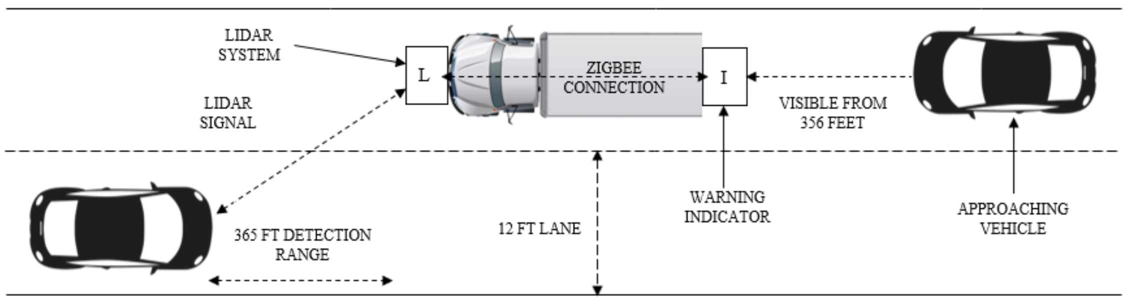


Figure 34: Operational Diagram (Top View)

[JWH]

The sketch below shows the location of the radar system and the warning indicator on the truck. The warning indicator will be attached on the driver-side of the truck, and the warning indicator module will be mounted on the back of the vehicle.

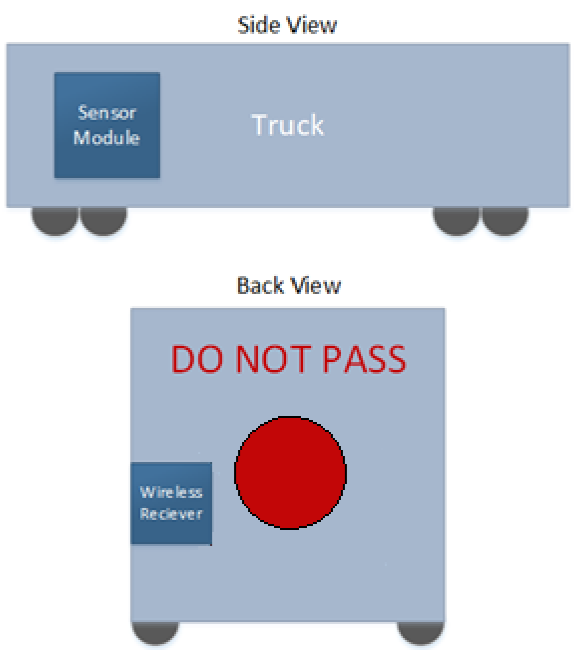


Figure 35: Mechanical Drawing of Mounted Sensor System

[ANR]

The figure below shows the mechanical sketch for the warning indicator PCB case. Both the base and lid for the case were designed in Solidworks.

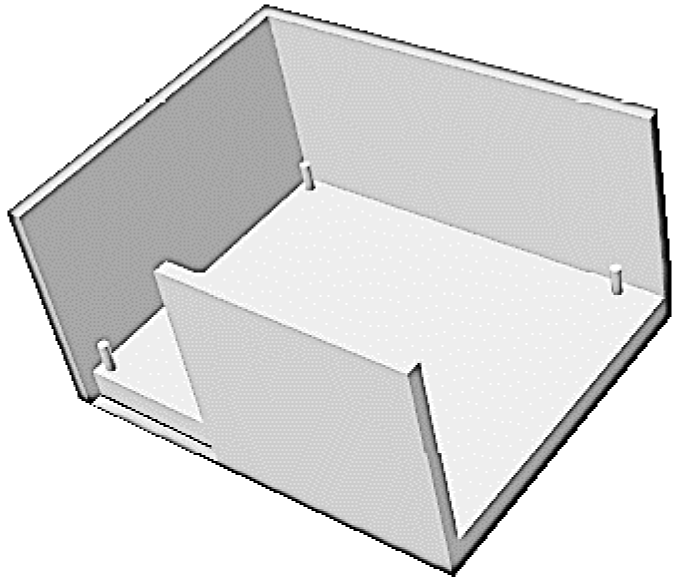


Figure 36: Warning Indicator PCB Case (Base)

The PCB sits on top of the 4 studs in the base of the PCB case and the Zigbee module will be on the side where the window is slot on the left is located. The front of the PCB case is left open to connect the power supply to the PCB.

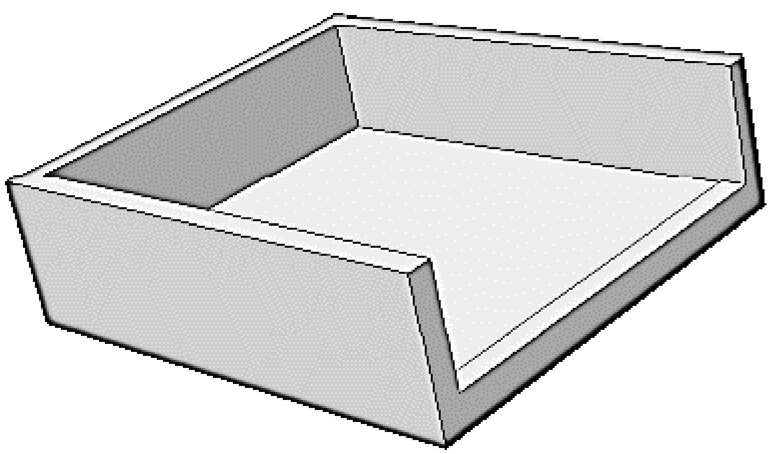


Figure 37: Warning Indicator PCB Case (Lid)

The PCB case lid will leave opening in the front for the power wires to connect to the PCB ports.

[JWH]

3.11 Parts List

Below is the completed list of all parts ordered, purchased, and used during the implementation of the project.

Table 10: List of Parts for Safe Pass System

Qty.	Refdes	Part Num.	Description	Cost	Total Cost
1	LED	LT-TP-0111	Warning LED - 10-30V Input	$36.00	$36.00
1	U1	ATSAMR21B18-MZ210PA	ZigBee Module	$10.72	$10.72
2	U2	TPS54231	BUCK-CONVERTERTPS54231-SOIC-8	$1.65	$3.30
1	C5	GCM188R71H104KA57J	CAP-0.1UF-0603	$0.32	$0.32
1	C3	885012206089	CAP-0.01UF-0603	$0.10	$0.10
1	C4	CL10B153KB8WPNC	CAP-0.015UF-0603	$0.10	$0.10
2	C1, C2	CL10B475KQ8NQNC	CAP-4.7UF-0603	$0.23	$0.46
1	C8	8.85012E+11	CAP-47PF-0603	$0.10	$0.10
2	C6, C7	GRM32ED71A476KE15L	CAP-47UF-1210	$1.26	$2.52
1	C9	885012206083	CAP-1000PF-0603	$0.10	$0.10
1	J5	348300801	CONN-MX150	$4.91	$4.91
2	K1, K2	CPC1002NTR	Solid State Relay	$1.73	$3.46
2	J6, J7	22-23-2061	6 pin Header Pin		
1	L1	ETQ-P3M100KVN	Inductor -10uH	$1.11	$1.11
1	Q2	IRLML2502TRPBF	NMOS Transistor	$0.62	$0.62
1	RV1	PDV-P8104	Photoresistor - 27-60k	$0.89	$0.89
1	U3	PIC16LF1503T-I/SL	Microcontroller Microchip	$0.82	$0.82
1	R4	RC0603FR-073K24L	Resistor - 3.24k	$0.10	$0.10
1	R3	AC0603FR-0710K2L	Resistor - 10.2k	$0.10	$0.10
1	R6	RT0603BRB0727RL	Resistors - 27	$0.72	$0.72
2	R5, R9	RT0603DRE0729K4L	Resistor - 29.4k	$0.12	$0.24
1	R2	RC0603FR-0768K1L	Resistor - 68.1k	$0.10	$0.10
1	R7	RC0603JR-07100KL	Resistors - 100k	$0.10	$0.10
1	R1	CR0603-FX-3323ELF	Resistor - 332k	$0.10	$0.10
1	D1	B240A-E3/5AT	Diode	$0.39	$0.39
4	J1, J2, J3, J4	1053072203	Wire to Board Connector	$0.85	$3.40
1	U2	TPS54231	BUCK-CONVERTERTPS54231-SOIC-8	$1.65	$1.65
1	C5	C410C104M5U5TA7200	CAP-0.1UF	$0.27	$0.27
1	C3	C322C103KDR5TA	CAP-0.01UF	$1.05	$1.05
1	C4	K153K10X7RF5UH5	CAP-0.015UF	$0.20	$0.20
2	C1, C2	C320C475K3N5TA91707301	CAP-4.7UF	$0.67	$1.34

1	C8	K470J15C0GF5TH5	CAP-47PF	$0.27	$0.27
2	C6, C7	FG22X5R1A476MNT06	CAP-47UF	$1.62	$3.24
1	C9	DE2E3KY102MA3BM02F	CAP-1000PF	$0.45	$0.45
1	L1	77F100K-TR-RC	Inductor -10uH	$0.31	$0.31
1	R4	MFR-25FBF52-3K24	Resistor - 3.24k	$0.10	$0.10
1	R3	MFR-25FBF52-10K2	Resistor - 10.2k	$0.10	$0.10
1	R5	MFR-25FBF52-29K4	Resistor - 29.4k	$0.10	$0.10
1	R2	MFR-25FBF52-68K1	Resistor - 68.1k	$0.10	$0.10
1	R1	MFR-25FBF52-332K	Resistor - 332k	$0.10	$0.10
1	D1	SB240S-E3/54	Diode	$0.53	$0.53
2	U1	ATSAMR21B18	ATSAMR21	$10.72	$21.44
2	R10, R12	RCA06031K00FKEAHP	1 kOhm Resistor	$0.18	$0.36
2	R8, R11	CRCW0603249RFKEAHP	249 Ohm Resistor	$0.18	$0.36
4		CRCW06030000Z0EAHP	0603 Short	$0.17	$0.68
1	C4	C0603C153J3GACTU	CAP-0.015UF-0603	$0.54	$0.54
1	C3	CC0603KRX7R9BB103	CAP-0.01UF-0603	$0.10	$0.10
2	R5, R9	RT0603DRE0729K4L	Resistor - 29.4k	$0.12	$0.24
1	R7	RC0603JR-07100KL	Resistors - 100k	$0.10	$0.10
4	J1, J2, J3, J4	1053072203	Wire to Board Connector	$0.85	$3.40
1	Q2	AO3414	NMOS Transistor	$0.46	$0.46
1	U3	PIC16LF1503T-I/SL	Microcontroller Microchip	$0.82	$0.82
1		TF03	TF03 Long Range LiDAR	$229.00	$229.00
1		DM164137	PIC16 Development Board	$21.60	$21.60
1		Safe Pass PCB	Warning Indicator PCB	$22.95	$22.95
1		Safe Pass PCB	Detection (Front End) PCB	$23.75	$23.75
2	C1, C2	GCM188R71H104KA57J	CAP CER 0.1UF 50V X7R 0603	$0.31	$0.62
2	C3, C4	EEE-FK1H471AM	CAP ALUM 470UF 20% 50V SMD	$1.96	$3.92
1	L1,	PA4342.473NLT	FIXED IND 47UH 3A 167 MOHM SMD	$1.44	$1.44
1	D1	TSSA3U45 R3G	DIODE SCHOTTKY 45V 3A DO214AC	$0.43	$0.43
1	U1	TLV75533PDBVR	IC REG LINEAR 3.3V 500MA SOT23-5	$0.65	$0.65
1	U2	ATSAMR21B18-MZ210PA	RF TXRX MOD 802.15.4 TRACE ANT	$10.72	$10.72
1	U3	PIC16F1619-I/P	IC MCU 8BIT 14KB FLASH 20DIP	$1.56	$1.56
1	U4	TS2596CM550 RNG	SWITCHING REGULATOR (BUCK)	$1.86	$1.86
1		B07M6WQ4B7	PICKit3 Programmer	$18.99	$18.99
				Total	**$446.58**

The total cost of the project was $446.58 on a $800 budget. The overall cost was below the original cost estimate around $600. The cost includes the custom PCBs, the corresponding parts to populate the PCBs, the LiDAR for detection, the PicKit 3 to program the warning indicator module, and a development board.

[JWH]

4. Operation, Maintenance, and Repair Instructions

4.1 Operation of LiDAR System

4.2 Operation of Warning Indicator System

When the warning indicator PCB was ordered there were some errors found in the designed circuit. The In-Circuit Serial Programming (ICSP) connections were found to be incorrect. To repair the PCB, two jumper wires were soldered onto the PIC16LF1503. The ICSP header pin is shown in the figure below.

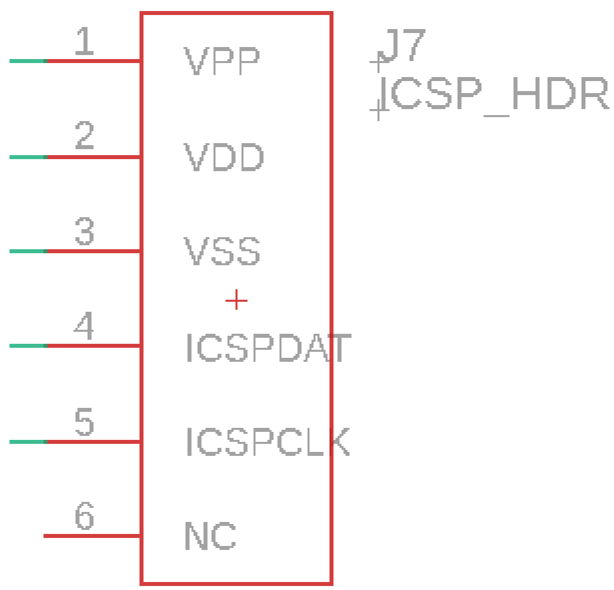


Figure 38: Warning Indicator PCB ICSP Header Pin

To correct the routing error, the pin ICSPDAT was wired to pin RA0, and ICSPCLK was wired to RA1. After correcting the routing error, a 10 kΩ pull-up resistor was soldered between the V_{DD} and V_{SS} pins. After these corrections were made the processor was able to be programmed and operated as needed.

[JWH]

4.3 Operation of ZigBee Link

The ZigBee link was intended to communicate between the warning indicator and the detection system. The processors communicate to the ZigBee modules using SPI. The SPI code is shown below to establish the wireless connection.

```
void spi_setup(){
        TRISCbits.TRISC2 = 0; //Data Out Line
        TRISCbits.TRISC1 = 1; //Data In Line
        TRISCbits.TRISC0 = 0; //Driving Clock Line
        TRISCbits.TRISC3 = 0; //CS Pin -- Active when pulled to ground
        APFCONbits.SDOSEL = 0; //SDO on RC5
        APFCONbits.NCO1SEL = 0; //SDI on RC4
        APFCONbits.SSSEL = 0; //SS on RC3
        SSP1CON3bits.BOEN = 0; //Override Buffer if necessary
        SSP1STATbits.SMP = 1; //Data sampled at rising edge of clk
        SSP1STATbits.CKE = 1; //clk idle high
        SSP1CON1 = 0b00010000; //Serial Port Pins Config, Idle CLK HIGH, clk = FSC/64
        PIE1bits.SSP1IE = 1; //Enable interrupt
        PIR1bits.SSP1IF = 0; //Clear interrupt flag
        SSP1CON1bits.SSPEN = 1; //Enable SPI
}

void spi_write(char data){
    SSP1BUF = data;
    while(SSP1STATbits.BF == 0);
    char readData =  SSP1BUF;
}

char spi_read(char data){
    SSP1BUF = data;
    while(SSP1STATbits.BF == 0);
    char readData =  SSP1BUF;
    return readData;
}
```

Figure 39: SPI Setup Functions

This code initializes the appropriate pins for SPI communication. Pin RC2 is initialized to the data out pin (SDO). Pin RC1 is initialized to the data in line (SDI). The clock pin set to RC0 for the SPI communication. The slave select pin is set to RC3 to choose the correct device for communication. Until the bits are full in the master (PIC16LF1503) the spi_setup function will continue to run. Once the SPI communication has been setup, the appropriate slave select is pulled low. The detection module (master) writes to the ZigBee (slave) using the function “spi_write.” The warning indicator (master) reads from the Zigbee module (slave) using "spi_read" function. Unfortunately, the communication was unable to be established wirelessly and a wire was used to complete the link.

[JWH]

5. Testing Procedures

5.1 Testing Procedures for LiDAR System

5.2 Testing Procedures for Warning Indicator System

After populating the warning indicator PCB, the functions of the board were tested. To test the buck converter a 12 V power supply was connected to the board and the resultant voltage was measured using a multimeter. The buck converter output voltage was measured to be about 3.35 V which is close to the desired 3.3 V. To ensure that the circuit operated at the required voltage range, the source voltage was increased to 24 V and the same output voltage resulted.

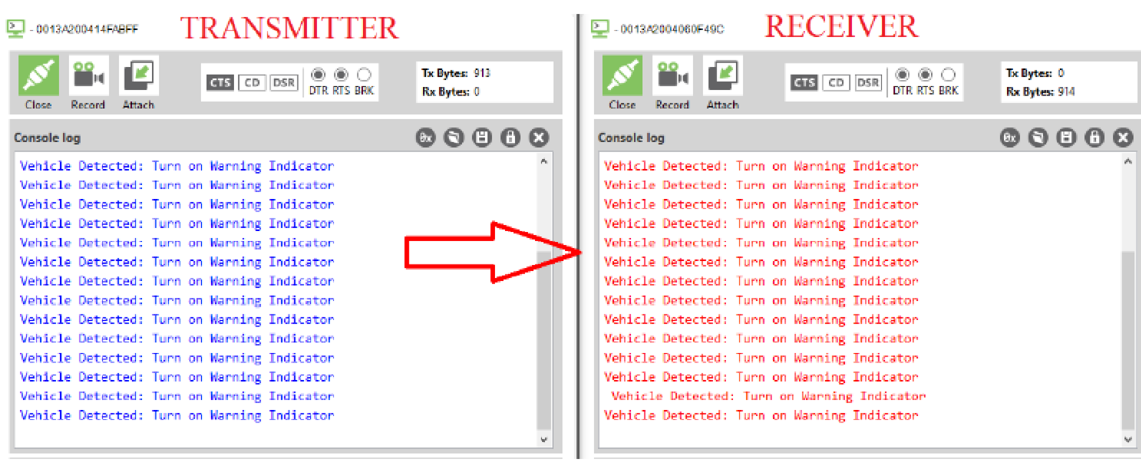
To test the effectiveness of the relay switching for the warning indicator light, a voltage was applied to turn on the relays. The required voltage to turn on the relays is about 1.8 V, this voltage was applied to the relay and successfully turned on the light with the respective brightness. The relays operated as expected and each relay controlled one of the brightness modes. The PIC16LF1503 output pins provided adequate voltages to the relays to turn them on when necessary; however, a small voltage divider was employed as shown in the previous schematics to drop the voltage to an appropriate level.

Once the software was developed, the analog to digital converter was implemented to read the voltage from the photoresistor voltage divider. Different light levels were used to find a good transition point the LED to turn on low brightness mode. The hexadecimal value that was used to determine the transition point from high to low brightness was 0x5. The adaptive brightness proved to be effective at transitioning from high to low brightness according to the surrounding lighting.

[JWH]

5.3 Testing Procedures for ZigBee Link

The wireless ZigBee link was implemented initially with a Xbee module to prove the wireless link could be established using this method of communication. The Xbee module used a software called XCTU to establish the wireless link between the two Xbee modules. The link established by first writing the Xbee MAC address of the receiver to the transmitter and vice versa. Below is a figure of the Xbee link and the information sent and received from the two modules.



The text in blue represents the message sent, and the red text is the received message. However, the Zigbee part that was chosen was not able to utilize this software. The Zigbee module used was intended to communicate from the detection processor to the warning indicator processor via SPI.

6. Financial Budget

The budget for this project was set to a maximum of $800. Last semester the estimated cost of the project was about $400 for the detection module system and about $130 for the warning indicator system. The cost of the project was projected to be a total of $530; however, due to the change in detection technology the actual cost of the project was significantly reduced. Below is the initial parts list for the Safe Pass Warning Indicator system.

Table 11: Warning Indicator Original Parts and Cost Estimate

Qty.	Reference Designator	Part Number	Description	Price (ea.)
2	C1, C2	CL10B475KQ8NQNC	CAP-4.7UF-0603	$0.23
1	C3	885012206089	CAP-0.01UF-0603	$0.10
1	C4	CL10B153KB8WPNC	CAP-0.015UF-0603	$0.10
1	C5	GCM188R71H104KA57J	CAP-0.1UF-0603	$0.32
2	C6, C7	GRM32ED71A476KE15L	CAP-47UF-1210	$1.26
1	C8	885012006021	CAP-47PF-0603	$0.10
1	C9	885012206083	CAP-1000PF-0603	$0.10
1	D1	B240A-E3/5AT	SCHOTTKY-B240A	$0.39
1	L1	ETQ-P3M100KVN	INDUCTOR-10UH-3.6A	$1.11
1	Q2	IRLML2502TRPBF	IRLML2502PBF	$0.62
1	R1	CR0603-FX-3323ELF	RES-332K-0603	$0.10
1	R2	RC0603FR-0768K1L	RES-68.1K-0603	$0.10
1	R3	AC0603FR-0710K2L	RES-10.2K-0603	$0.10
1	R4	RC0603FR-073K24L	RES-3.24K-0603	$0.10
1	R5	RT0603DRE0729K4L	RES-29.4K-0603	$0.12
1	R6	RT0603BRB0727RL	RES-27-0603	$0.72
1	R7	RC0603JR-07100KL	RES-100K-0603	$0.10
1	R9	RT0603DRE0729K4L	RES-29.4K-0603	$0.12
1	RV1	PDV-P8104	PDV-P8104	$0.89
1	U1	ATSAMR21B18-MZ210PA	ATSAMR21	$10.72
1	U2	TPS54231	BUCK-CONVERTERTPS54231-SOIC-8	$1.65
1	U3	PIC16LF1503T-I/SL	PIC16LF1503 Microcontroller	$0.82

1	U4	1053072203	Wire to Board Connector	$0.44
1	LED	LT-TP-0111	Warning Light	$36.00
			Total Component Estimated Cost	**$56.31**

As shown in the table above, the total cost of the parts and the warning light is $56.31.

The table below shows the estimated and actual costs for different portions of the project.

Table 12: Comparison of Estimated and Actual Costs

Project Part	Estimated	Actual
Warning Light	$36.00	$36.00
Warning Indicator PCB	$24.00	$22.95
Warning Indicator Parts	$70.00	$108.53
Detection	$350 (mmWave Radar)	$22.84 (LiDAR)
Detection PCB Parts	$50	$5.66
LiDAR Module	-	$250.60
Total	$530.00	$446.58

The overall actual cost was noticeably lower than the estimated cost and was a little more than half of the budget provided for this project. However, some changes were made to the project implementation in part due to the cost of the mmWave radar chip and PCB board.

[JWH]

7. Project Schedule

The following Gantt Charts below show the timeline for the development, design, and implementation of the Safe Pass system.

7.1 Midterm Design Gantt

	Task Name	Duration	Start	Finish	Resource Names
1	SDP1 Fall 2018				
2	Project Design				
3	**Preliminary report**	11 days	Thu 9/6/18	Sun 9/16/18	**ALL**
4	Cover page	11 days	Thu 9/6/18	Sun 9/16/18	JWH
5	T of C, L of T, L of F	11 days	Thu 9/6/18	Sun 9/16/18	JWH
6	Need	11 days	Thu 9/6/18	Sun 9/16/18	JWH,GMS
7	Objective	11 days	Thu 9/6/18	Sun 9/16/18	JJH,ANR
8	Background	11 days	Thu 9/6/18	Sun 9/16/18	GMS,JJH,JWH,ANR
9	Marketing Requirements	11 days	Thu 9/6/18	Sun 9/16/18	GMS,JJH,JWH
10	Objective Tree	11 days	Thu 9/6/18	Sun 9/16/18	JWH
11	**Block Diagrams Level 0, 1, ... w/ FR tables**	**11 days**	Thu 9/6/18	Sun 9/16/18	**ALL**
12	Digital Signal Processing Hardware Module				JJH
13	Radar Harware Module				GMS
14	Warning Indicator Hardware Module	11 days	Thu 9/6/18	Sun 9/16/18	JWH
15	Software Flow Chart	11 days	Thu 9/6/18	Sun 9/16/18	ANR
16	Mechanical Sketch	11 days	Thu 9/6/18	Sun 9/16/18	ANR
17	Team information	11 days	Thu 9/6/18	Sun 9/16/18	ALL
18	References	11 days	Thu 9/6/18	Sun 9/16/18	ALL
19	**Midterm Report**	**35 days**	**Thu 9/6/18**	**Wed 10/10/18**	**ALL**
20	Design Requirements Specification	14 days	Mon 9/17/18	Sun 9/30/18	JWH
21	Midterm Design Gantt Chart	14 days	Mon 9/17/18	Sun 9/30/18	JWH
22	**Design Calculations**	**24 days**	**Mon 9/17/18**	**Wed 10/10/18**	**GMS,JJH,JWH**
23	**Electrical Calculations**	**24 days**	**Mon 9/17/18**	**Wed 10/10/18**	**GMS,JJH,JWH**
24	Communication	24 days	Mon 9/17/18	Wed 10/10/18	JWH
25	Radar	24 days	Mon 9/17/18	Wed 10/10/18	GMS
26	Power, Voltage, Current	24 days	Mon 9/17/18	Wed 10/10/18	GMS
27	Antenna Calculations	24 days	Mon 9/17/18	Wed 10/10/18	JJH
28	**Block Diagrams Level 2 w/ FR tables & ToO**	**7 days**	**Mon 9/17/18**	**Sun 9/23/18**	**GMS,JJH,JWH**
29	DSP Hardware Module	7 days	Mon 9/17/18	Sun 9/23/18	JJH
30	Radar Hardware Module	7 days	Mon 9/17/18	Sun 9/23/18	GMS
31	Warning Indicator Hardware Module	7 days	Mon 9/17/18	Sun 9/23/18	JWH
32	Warning Indicator Pseudocode	7 days	Mon 9/17/18	Sun 9/23/18	ANR
33	Midterm Design Presentations Part 1	1 day	Thu 10/11/18	Thu 10/11/18	ALL
34	Project Poster	14 days	Mon 10/8/18	Sun 10/21/18	ALL
35	**Final Design Report**	**52 days**	**Mon 10/8/18**	**Wed 11/28/18**	**ALL**
36	Abstract	52 days	Mon 10/8/18	Wed 11/28/18	JJH
37	**Software Design**	**31 days**	**Mon 10/8/18**	**Wed 11/7/18**	**ANR**
38	**Modules 1...n**	**31 days**	**Mon 10/8/18**	**Wed 11/7/18**	**ANR**
39	Psuedo Code	31 days	Mon 10/8/18	Wed 11/7/18	ANR
40	**Hardware Design**	**31 days**	**Mon 10/8/18**	**Wed 11/7/18**	**GMS**
41	**Modules 1...n**	**31 days**	**Mon 10/8/18**	**Wed 11/7/18**	**GMS,JJH,JWH**
42	DSP and Antenna Schematics	31 days	Mon 10/8/18	Wed 11/7/18	JJH
43	Radar Module Schematics	31 days	Mon 10/8/18	Wed 11/7/18	GMS
44	Warning Indicator Schematics	31 days	Mon 10/8/18	Wed 11/7/18	JWH
45	**Parts Lists**	**52 days**	**Mon 10/8/18**	**Wed 11/28/18**	**GMS,JJH,JWH**
46	Parts list(s) for Schematics	52 days	Mon 10/8/18	Wed 11/28/18	GMS,JJH,JWH
47	Materials Budget list	52 days	Mon 10/8/18	Wed 11/28/18	GMS,JJH,JWH
48	Proposed Implementation Gantt Chart	52 days	Mon 10/8/18	Wed 11/28/18	JWH
49	Conclusions and Recommendations	52 days	Mon 10/8/18	Wed 11/28/18	JJH
50	Final Parts Request Form	56 days	Mon 10/8/18	Sun 12/2/18	JWH

Figure 40: Project Design and Develop Midterm Gantt Chart

During the beginning of the process, the project proposal was expounded to include the block 0, 1 and 2 diagrams for the hardware and the software flow diagram. During this portion of the project the design requirements were specified, revised and finalized to meet the appropriate marketing requirements. After completing the project description and objective the parts were chosen to meet these requirements. The 12/24 V to 3.3 V buck converter was chosen to fulfill the design requirement to have a variable source voltage. Originally the mmWave radar chip was chosen because it appeared to be the best option for detection and it provided an accurate and reliable method of detection. Around week 5, the wireless link was chosen to be a ZigBee module to interface between the detection module and the warning indicator module. After the components were chosen the remaining six weeks were used to draw the schematics in Eagle for the respective PCBs. The Gantt Chart shows the team members that worked on the different portions of the project. During the schematic developments, the pseudocode was developed for the detection and warning indicator modules.

[JWH]

7.2 Final Design Gantt Chart

Below is the original Gantt Chart for the implementation and assembly of the Safe Pass system. However, the implementation route was changed during February and a more accurate Gantt Chart was used as shown after.

ID	Task Mode	Task Name	Duration	Start	Finish
1		**SDPII Implementation 2018**	**103 days**	**Mon 1/14/19**	**Fri 4/26/19**
2		Revise Gantt Chart	11 days	**Mon 1/14/19**	Thu 1/24/19
3		**Implement Project Design**	**96 days**	**Mon 1/14/19**	**Fri 4/19/19**
4		**Hardware Implementation**	**88 days**	**Mon 1/14/19**	**Thu 4/11/19**
5		Design Antenna Grid	17 days	Tue 1/22/19	Thu 2/7/19
6		**Front End PCB Layout**	**32 days**	**Tue 1/22/19**	**Fri 2/22/19**
7		Voltage Converter Placement	7 days	Tue 1/22/19	Mon 1/28/19
8		MMIC Component Placement	7 days	Mon 1/28/19	Sun 2/3/19
9		Voltage Converter Routing & Verification	7 days	Sun 2/3/19	Sat 2/9/19
10		MMIC Routing & Verification	14 days	Sat 2/9/19	Fri 2/22/19
11		Antenna Grid Placement and Routing & Verification	1 day	Fri 2/22/19	Fri 2/22/19
12		**Backend PCB Layout**	**28 days**	**Tue 1/22/19**	**Mon 2/18/19**
13		Component Placement	7 days	Tue 1/22/19	Mon 1/28/19
14		Routing & Verification	14 days	Mon 1/28/19	Sun 2/10/19
15		Review Backend PCB	7 days	Mon 2/11/19	Sun 2/17/19
16		Purchase PCBs and Components	14 days	Fri 2/22/19	Thu 3/7/19
17		Test Hardware	50 days	Thu 3/7/19	Thu 4/25/19
18		*MIDTERM: Demonstrate Hardware (Fixed deadline)*	5 days	Mon 3/4/19	Fri 3/8/19
19		SDC & FA Hardware Approval	1 day	Fri 4/19/19	Fri 4/19/19
20		**Software Implementation**	**56 days**	**Mon 1/14/19**	**Sun 3/10/19**
21		**Develop Software**	**54 days**	**Mon 1/14/19**	**Fri 3/8/19**
22		Run demo from TI on development board	12 days	Mon 1/21/19	Fri 2/1/19
23		Use mmWave Studio and SDK to customize software	12 days	Mon 2/4/19	Fri 2/15/19
24		Start development of Zigbee code	12 days	Mon 2/11/19	Fri 2/22/19
25		Code for warning indicator	7 days	Mon 2/25/19	Sun 3/3/19
26		**Test Software**	**12 days**	**Mon 2/25/19**	**Fri 3/8/19**
27		Use our board to run software	12 days	Mon 2/25/19	Fri 3/8/19
28		Revise Software	42 days	Fri 3/15/19	Thu 4/25/19
29		*MIDTERM: Demonstrate Software (Fixed deadline)*	5 days	Mon 3/4/19	Fri 3/8/19
30		SDC & FA Software Approval	0 days	Fri 3/8/19	Fri 3/8/19
31		**System Integration**	**42 days**	**Sat 3/9/19**	**Fri 4/19/19**
32		Assemble Complete System	5 days	**Mon 3/11/19**	Fri 3/15/19
33		Test Complete System	42 days	Fri 3/15/19	Thu 4/25/19
34		Revise Complete System	42 days	Fri 3/15/19	Thu 4/25/19
35		*Demonstration of Complete System*	1 day	Fri 4/26/19	Fri 4/26/19
36		**Develop Final Report**	**99 days**	**Mon 1/14/19**	**Mon 4/22/19**
37		Write Final Report	99 days	Mon 1/14/19	Mon 4/22/19
38		Submit Final Report	0 days	Mon 4/22/19	Mon 4/22/19
39		Spring Recess	7 days	Sat 4/13/19	Fri 4/19/19
40		***Project Demonstration and Presentation***	1 day	Fri 4/26/19	Fri 4/26/19

Figure 41: Original Implementation Gantt Chart

At the beginning of the implementation process, the parts list was ordered for the warning indicator’s buck converter to test the implementation. Since, the schematics were completed for the PCB designs the layout began at the start of the implementation phase. The warning indicator PCB took the first four weeks and purchased after the design was reviewed and verified. However, during this time it was found to be an expensive route to purchase a PCB for the mmWave radar implementation. The attempts to program the mmWave radar development board yielded few results which resulted in a change in the implementation. In order to implement the project completely, the team decided to change to LiDAR during the fifth week to succeed on the project. The Gantt Chart below reflects the work done before the change occurred and the remaining steps to implement the LiDAR detection system.

[JWH]

ID	Task Mode	Task Name	Duration	Start	Finish
1		**SDPII Implementation 2018**	**103 days**	**Mon 1/14/19**	**Fri 4/26/19**
2		Revise Gantt Chart	11 days	**Mon 1/14/19**	Thu 1/24/19
3		**Implement Project Design**	**96 days**	**Mon 1/14/19**	**Fri 4/19/19**
4		**Hardware Implementation**	**88 days**	**Mon 1/14/19**	**Thu 4/11/19**
5		Design Antenna Grid	17 days	Tue 1/22/19	Thu 2/7/19
6		**Front End PCB Layout**	**32 days**	**Tue 1/22/19**	**Fri 2/22/19**
7		Voltage Converter Placement	7 days	Tue 1/22/19	Mon 1/28/19
8		MMIC Component Placement	7 days	Mon 1/28/19	Sun 2/3/19
9		Voltage Converter Routing & Verification	7 days	Sun 2/3/19	Sat 2/9/19
10		MMIC Routing & Verification	14 days	Sat 2/9/19	Fri 2/22/19
11		Antenna Layout and Design	22 days	Thu 2/7/19	Thu 2/28/19
12		**Front End LiDAR PCB Layout**	**22 days**	**Thu 2/28/19**	**Thu 3/21/19**
13		Lidar PCB Schematic Design	2 days	**Thu 2/28/19**	Fri 3/1/19
14		LiDAR PCB Layout	2 days	Fri 3/1/19	Sat 3/2/19
15		Purchase Lidar PCB and Components	7 days	Sat 3/2/19	Fri 3/8/19
16		Solder LiDAR Board	14 days	Fri 3/8/19	Thu 3/21/19
17		**Backend PCB Layout**	**28 days**	**Tue 1/22/19**	**Mon 2/18/19**
18		Component Placement	7 days	Tue 1/22/19	Mon 1/28/19
19		Routing & Verification	14 days	Mon 1/28/19	Sun 2/10/19
20		Review Backend PCB	7 days	Fri 2/8/19	Thu 2/14/19
21		Purchase Warning Indicator PCB and Parts	7 days	Fri 2/8/19	Thu 2/14/19
22		Solder Warning Indicator PCB and Components	14 days	Thu 2/28/19	Wed 3/13/19
23		Test Hardware	50 days	Thu 3/7/19	Thu 4/25/19
24		*MIDTERM: Demonstrate Hardware (Fixed deadline)*	5 days	Mon 3/4/19	Fri 3/8/19
25		SDC & FA Hardware Approval	1 day	Fri 4/19/19	Fri 4/19/19
26		**Software Implementation**	**56 days**	**Mon 1/14/19**	**Sun 3/10/19**
27		**Develop Software**	**54 days**	**Mon 1/14/19**	**Fri 3/8/19**
28		Run demo from TI on development board	12 days	Mon 1/21/19	Fri 2/1/19
29		Use mmWave Studio and SDK to customize software	12 days	Mon 2/4/19	Fri 2/15/19
30		Develop Software for LiDAR Detection	14 days	Thu 2/28/19	Wed 3/13/19
31		Demonstrate ZigBee Link	15 days	Thu 2/21/19	Thu 3/7/19
32		Start development of Zigbee code	12 days	Thu 2/28/19	Mon 3/11/19
33		Code for warning indicator	7 days	Mon 2/25/19	Sun 3/3/19
34		**Test Software**	**12 days**	**Mon 2/25/19**	**Fri 3/8/19**
35		Use our board to run software	12 days	Mon 2/25/19	Fri 3/8/19
36		Revise Software	42 days	Fri 3/15/19	Thu 4/25/19
37		*MIDTERM: Demonstrate Software (Fixed deadline)*	5 days	Mon 3/4/19	Fri 3/8/19
38		SDC & FA Software Approval	0 days	Fri 3/8/19	Fri 3/8/19
39		**System Integration**	**42 days**	**Sat 3/9/19**	**Fri 4/19/19**
40		Assemble Complete System	5 days	**Mon 3/11/19**	Fri 3/15/19
41		Test Complete System	42 days	Fri 3/15/19	Thu 4/25/19
42		Revise Complete System	42 days	Fri 3/15/19	Thu 4/25/19
43		*Demonstration of Complete System*	1 day	Fri 4/26/19	Fri 4/26/19
44		**Develop Final Report**	**99 days**	**Mon 1/14/19**	**Mon 4/22/19**
45		Write Final Report	99 days	Mon 1/14/19	Mon 4/22/19
46		Submit Final Report	0 days	Mon 4/22/19	Mon 4/22/19
47		Spring Recess	7 days	Sat 4/13/19	Fri 4/19/19
48		***Project Demonstration and Presentation***	1 day	Fri 4/26/19	Fri 4/26/19

Figure 42: Project Final Implementation Gantt Chart

After switching designs from mmWave radar to LiDAR, the project was divided into the detection module PCB, warning indicator PCB, the ZigBee wireless link and the software to receive data from the LiDAR system. During weeks five through eight the warning indicator module PCB was populated with the parts. From weeks six through eight, the detection module PCB schematic and board layout was designed and ordered along. Using the detection module processor, the LiDAR data was processed and analyzed to determine the distance to the object. For the warning indicator the software was developed for the warning indicator and implemented successfully. However, the efforts to establish the ZigBee wireless link yielded no results and were unsuccessful. Instead a wire was run from the detection module to the warning indicator to establish a link to communicate the presence of vehicles. The Gantt Chart reflects the progress and process of the project implementation.

[JWH]

8. Design Team Information

Jonathan W. Hall (Team Leader) – Electrical Engineer

Julia J. Hariharan (Engineering Data Manager) – Electrical Engineer

Alycia N. Riese (Software Lead) – Computer Engineer

Gregory M. Synek (Hardware Lead) – Electrical Engineer

9. Conclusions and Recommendations

9.1 Vehicle Detection Hardware Conclusions

Although a Lidar implementation was used for this project and is effective in most cases, a real-world implementation of such a system would still be far superior using mmWave radar. MmWave radar offers greater resolution, can accurately track motion of smaller objects in 3D space, and is effectively immune to all weather conditions that are safe to drive in. Output power can be increased to penetrate adverse weather conditions and additional filtering techniques can be applied to remove environmental noise. Using Lidar, there is no opportunity to achieve better performance if light pulses are being blocked by environmental conditions. Additionally, a radar-based approach is more practical for installation on modern vehicles considering many already have forward looking radar for features such as adaptive cruise control and frontal collision warnings. Adding an extra channel to existing radar solutions would be a marginal cost add-on compared to a costly lidar module.

[GMS]

9.2 Vehicle Detection Software Conclusions

The configuration of the system was difficult to initially understand and implement. Since the baud rate of the chosen Lidar was very high (115200), this required the clock speed of the microcontroller to be at the maximum of 32Mhz. This caused some timing issues with the Lidar data. The timing issued caused some bytes to be received more than once or not at all. When the Lidar was powered, it began to transmit data immediately. The code on the microcontroller had to account for this by specifically looking for the header bytes or the checksum bytes before performing calculations on the distance data. It then would compare the distance bytes with the specified detection range to determine of there was an object in line of sight of the Lidar. At farther distances, this caused the diameter of the Lidar's detection range to increase. When testing in a narrow hallway, the Lidar would detect the walls of the room at farther distances, indicating there was an object in view.

Since no commands were being sent to the Lidar module, the RX line of the Lidar was not used. This caused issues because when the code was running, the RX line would receive noise. This noise caused the Lidar to think it was being issued a command, so it would stop transmitting data and wait for the command. To solve this, the RX line was disconnected from the microcontroller and left floating.

[ANR]

9.3 Warning Indicator Hardware and Software Conclusions

The warning indicator system was able to receive a signal from the vehicle detection module and blink according to the brightness of the surrounding. However, during the implementation of the warning indicator some of the PCB routing was found to be incorrect. This problem was solved by soldering the appropriate jumper wires on the PCB. Some of the PCB component traces were placed at locations that made installation challenging. If the PCB was to be revised, then it would be advisable to verify the required space for each component and plan accordingly. The photoresistor through-holes on the PCB were found to be too narrow when the PCB was manufactured; therefore, it would be best to size the holes with an increased size to account for margin of errors. Finally, when the project was completed many of the pins on the PIC16LF1503 were used which limited some of the contingency plan options for the wireless link. To ensure the best outcome, it would have been better to choose a processor with more pins and functions (such as more SPI, UART, I2C ports…etc) to provide flexibility when encountering limitations or errors in the development process. However, the processor was adequate for the implementation of the warning indicator and it operated as required. Despite the difficulties encountered, the process of implementing the warning indicator software and hardware was an educational and rewarding experience. Also, to ensure that system operates in all weather conditions, it would be beneficial to design a custom case for all weather conditions.

[JWH]

9.4 Wireless Link Implementation Conclusions

Unfortunately, the wireless ZigBee link was unable to be established with the device that was used for this project. The chosen device lacked detailed documentation, this was an oversight in the decision process for choosing a wireless module. The lack of detailed documentation caused difficulty in understanding how to link the two modules. To further develop this project, it would be essential to find a reliable and well documented device to establish the wireless link. To implement the Safe Pass system on vehicles, it would be safer and more reliable to provide a hardwire connection from the detection module to the warning indicator module.

[JWH]

10. References

Works Cited

[1] Mark Cuddihy, Manoharprasad K. Rao, Eric L. Reed, "System and method for combined blind spot detection and rear crossing path collision warning." U.S. Patent 8,552,848 issued October 8, 2013.

[2] K. Kawaziri, H. Shimizu, "Device for determining distance between vehicles." U.S. Patent 5,929,784 issued July 27, 1999

[3] Pakett, Alan G, "Smart blind spot detector." U.S. Patent 8,111,826 issued August 25, 1993

[4] D. B. Johnson, "Truck light warning system." U.S. Patent 6,133,851 issued October 17, 2000.

[5] P. H. Batavia, D. E. Pomerleau and C. E. Thorpe, "Overtaking vehicle detection using implicit optical flow," Proceedings of Conference on Intelligent Transportation Systems, Boston, MA, 1997, pp. 729-734.

[6] G. Liu, L. Wang and S. Zou, "A radar-based blind spot detection and warning system for driver assistance," 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, 2017, pp. 2204-2208

[7] S. Sivaraman and M. M. Trivedi, "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis," in IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp. 1773-1795, Dec. 2013

[8] L. Fletcher, N. Apostoloff, L. Petersson and A. Zelinsky, "Vision in and out of vehicles," in IEEE Intelligent Systems, vol. 18, no. 3, pp. 12-17, May-Jun 2003.

[9] B. Heo, K. Yun and J. Y. Choi, "Appearance and motion based deep learning architecture for moving object detection in moving camera," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 2017, pp. 1827-1831.

[10] Saboune J., Arezoomand M., Martel L., Laganiere R. (2011) "A Visual Blindspot Monitoring System for Safe Lane Changes." In: Maino G., Foresti G.L. (eds) Image Analysis and Processing – ICIAP 2011. ICIAP 2011. Lecture Notes in Computer Science, vol 6979. Springer, Berlin, Heidelberg

[11] Vehicle Stopping Distance and Time. (2018). [ebook] National Association of City Transportation Officials. Available at: https://nacto.org/docs/usdg/vehicle_stopping_distance_and_time_upenn.pdf [Accessed 25 Sep. 2018].

[12] Safety in Numbers. (2015). [ebook] National Highway Traffic Safety Administration. Available at: https://one.nhtsa.gov/nhtsa/Safety1nNum3ers/august2015/S1N_Speeding-August2015_812008.pdf [Accessed 25 Sep. 2018].

[13] ZigBee Module "ATSAMR21 Smart Connect Module." Data Sheet. Available at: http://ww1.microchip.com/downloads/en/devicedoc/atmel-42486-atsamr21b18-mz210pa_datasheet.pdf [Accessed 23 November 2018].

[14] Microcontroller "PIC16(L)F1503 14-Pin Flash, 8-bit Microcontrollers." Data Sheet. Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/40001607D.pdf [Accessed 23 November 2018].

[15] Texas Instruments "TPS54231 2-A, 28-V Input, Step-Down DC-DC Converter with Eco-mode." Data Sheet. Available at: http://www.ti.com/lit/ds/symlink/tps54231.pdf [Accessed 26 November 2018].

[16] Texas Instruments "TPS7A8101 Low Noise, Wide-Bandwidth, High PSRR, Low-Dropout 1-A Linear Regulator." Data Sheet. Available at: http://www.ti.com/lit/ds/symlink/tps7a8101.pdf [Accessed 26 November 2018]

[17] Texas Instruments "TPS7A88 Dual, 1-A, Low Noise (3.8uVrms), LDO Voltage Regulator." Data Sheet. Available at: http://www.ti.com/lit/ds/symlink/tps7a88.pdf [Accessed 26 November 2018].

[18] Texas Instruments "AWR1642 Single-Chip 77- and 79-GHz FWCW Radar Sensor." Data Sheet. Available at: http://www.ti.com/lit/ds/symlink/awr1642.pdf [Accessed 26 November 2018].

[19] Benewake (Beijing) Co., Ltd "TF03 Single-Point Long-Distance LiDAR Product Specification." Available at: https://acroname.com/sites/default/files/assets/tf03_datasheet_v0.4_en.pdf [Accessed 27 April 2019]

11. Appendices

11.1 Data Sheets

11.1.1 Data Sheet Excerpt for ZigBee Module Pin Layout and Assignment

The following is obtained from the ZigBee Module data sheet [13]. The table (table 3-1 in the data sheet) shows the function of each pin, and the figure (figure 2-3 from the data sheet) shows the numbering of the pins on the physical device. The diagram shows the component's surface mount diagram with the first pin labelled.

Table 13: ZigBee Module Pin Layout from Data Sheet

Module pin	Function	uC pin	Pin function / Port Configuration
1	Vcc	4 / 24	Power supply pin
2	GND	3 / 6 / 11 / 14 / 28 / paddle	Ground pins
3	PA16	17	PWM1 SPI SS I2C SDA GPIO
4	PA17	18	PWM2 SPI SCK I2C SCK GPIO
5	PA18	19	PWM3 SPI MISO UART Tx GPIO
6	PA19	20	PWM4 SPI MOSI UART Rx GPIO
7	PA07	8	Analog In GPIO

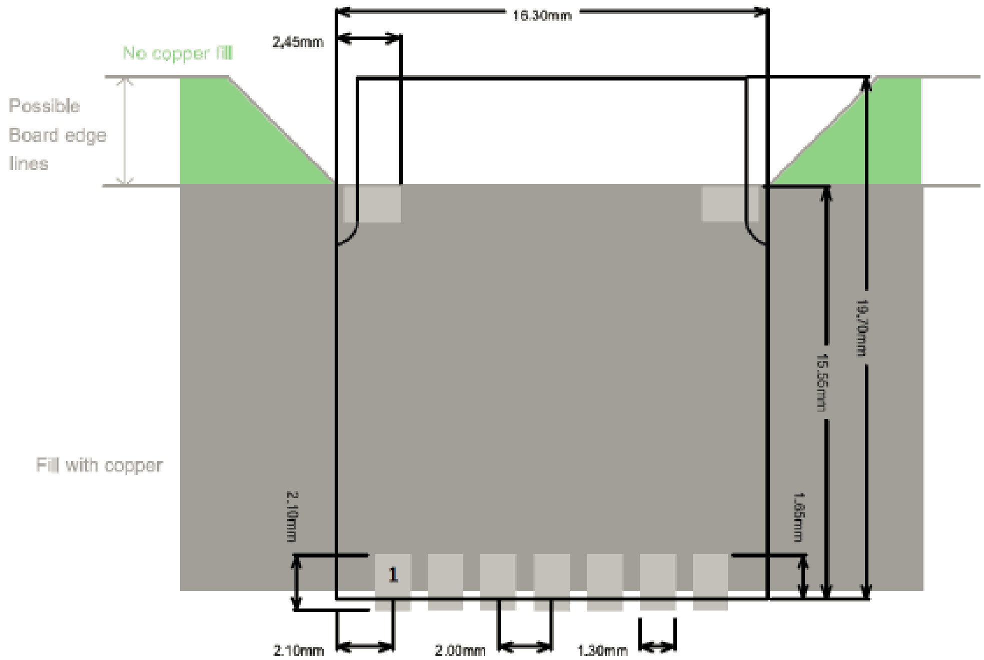


Figure 43: ZigBee Module Surface Mount Layout and Pin Numbers

[JWH]

11.1.2 Data Sheet Excerpt for Warning Indicator System Microcontroller

The following is the 14-pin diagram obtained from the data sheet for PIC16(L)F1503 [14].

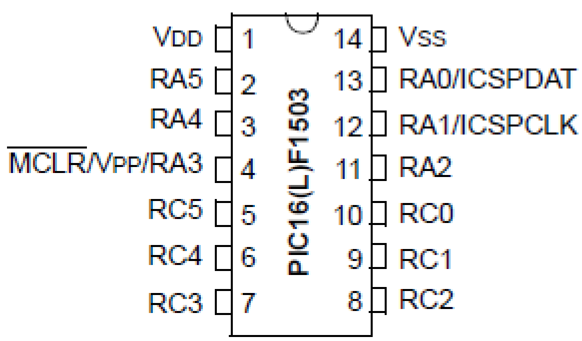


Figure 44: PIC16(L)F1503 14-Pin Layout

The following is a table (Table 1 of the data sheet) for the pin functions and assignments for the PIC16(L)F1503.

Table 14: 14-Pin Allocation for PIC16(L)F1503

I/O	14-Pin PDIP/SOIC/TSSOP	16-Pin QFN, UQFN	ADC	Reference	Comparator	Timer	CWG	NCO	CLC	PWM	MSSP	Interrupt	Pull-Up	Basic
RA0	13	12	AN0	DACOUT1	C1IN+	—	—	—	—	—	—	IOC	Y	ICSPDAT
RA1	12	11	AN1	V_{REF+}	C1IN0- C2IN0-	—	—	—	—	—	—	IOC	Y	ICSPCLK
RA2	11	10	AN2	DACOUT2	C1OUT	T0CKI	$\overline{\text{CWG1FLT}}$	—	CLC1	PWM3	—	INT IOC	Y	—
RA3	4	3	—	—	—	T1G(1)	—	—	CLC1IN0	—	$\overline{\text{SS}}$(1)	IOC	Y	$\overline{\text{MCLR}}$ V_{PP}
RA4	3	2	AN3	—	—	T1G	—	NCO1(1)	—	—	SDO(1)	IOC	Y	CLKOUT
RA5	2	1	—	—	—	T1CKI	—	NCO1CLK	CLC1IN1	—	—	IOC	Y	CLKIN
RC0	10	9	AN4	—	C2IN+	—	—	—	CLC2	—	SCL SCK	—	—	—
RC1	9	8	AN5	—	C1IN1- C2IN1-	—	—	NCO1	—	PWM4	SDA SDI	—	—	—
RC2	8	7	AN6	—	C1IN2- C2IN2-	—	—	—	—	—	SDO	—	—	—
RC3	7	6	AN7	—	C1IN3- C2IN3-	—	—	—	CLC2IN0	PWM2	$\overline{\text{SS}}$	—	—	—
RC4	6	5	—	—	C2OUT	—	CWG1B	—	CLC2IN1	—	—	—	—	—
RC5	5	4	—	—	—	—	CWG1A	—	CLC1(1)	PWM1	—	—	—	—
V_{DD}	1	16	—	—	—	—	—	—	—	—	—	—	—	V_{DD}
V_{SS}	14	13	—	—	—	—	—	—	—	—	—	—	—	V_{SS}

[JWH]

11.1.3 Buck Converter (TPS54231) Calculations

All formulas and calculations performed rely on data pulled from the TPS54231 Datasheet on pages 10 through 21.

[GMS]