School of Engineering Science
Simon Fraser University
Burnaby, BC V5A 1S6
kalmoamm@sfu.ca

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 340 Design Specifications for Safe Direction

Dear Dr. Rawicz:

In the enclosed documents contains the set of technical guidelines for the design of Safe Direction- blind spot detection and parking assistance system. Safe Direction offers blind spot detection while cruising and parking assistance for the driver to ensure safety.

The design specification outlines the system overview of Safe Direction and engineering details of each module. This document is applied on the bases of proof-of-concept model only. Any further improvements will be included in future development.

If you have any questions or concerns, please do not hesitate to contact us by email at kalmoamm@sfu.ca or by phone at (778) 321-5551.

Sincerely,

Khalid Almoammar

Khalid Almoammar
Chief Executive Officer
Car Sense

Enclosure: Design Specification for Safe Direction

Design Specifications of the blind spot monitoring and parking-assisting device

Safe Direction™

Project Team:

Aziz Mikwar
Angel Tian
Shaham Shafiei
Karen Wang
Khalid Almoammar

Contact Person: Khalid Almoammar
kalmoamm@sfu.ca

Submitted to: Dr. Andrew Rawicz – ENSC 340
Dr. Steve Whitmore – ENSC 305
School of Engineering Science
Simon Fraser University

Date: March 14, 2014

Abstract

The document contains the design specification of the project Safe Direction by Car Sense. The goal of the project is to create a Parking Assistance and Blind spot Detection systems combined in one large system. The system is portable and can easily be installed by drivers. The system uses ultrasound sensors for objects detection and notifies the driver either by an alarm sound or LED lights. The Sensors send data wirelessly through RF. The design specification document demonstrates the design details of every aspect of the project, hardware, software, power and logic. The document is designed for the proof of concept model.

Table Of Contents

List Of Figures

List Of Tables

Design Specifications of the blind spot monitoring and parking-assisting device
Safe Direction™

1 Introduction

Car Sense's Safe Direction is an electronic driving aid system that can be installed on any car regardless of the model. Using Ultrasound sensors this system sends the appropriate signals to the main Arduino for processing and as needed warns the driver of any objects in their blind spot in addition to the sensors on the bumper which help the driver to avoid hitting other cars and objects while parking. This document lays out the design specifications of the Safe Direction system.

Note that this document refers to functional specification standards that were met in the design by putting the number of the standard in square bracket in the following form **[Rxx-p]** where **xx** is the functional requirement number, and **p** is the priority of the functional requirement as denoted by one of three values:

- **I.** The requirement applies to the proof-of-concept system only.
- **II.** The requirement applies to both the proof-of-concept system and the final production system.
- **III.** The requirement applies to the final production system only.

1.1 Scope

This document specifies the design of the Safe Direction blind spot and parking assistant system. The design specification includes all necessary requirements/mathematical calculations of the proof-of-concept system. However, detailed mathematical theories are not included in this document.

1.2 Intended Audience

This document is intended for the uses of Safe Direction associates and its members. This document should be used by the Test engineers to confirm the correctness of the system functionality. Design engineers should follow the documented specifications to ensure public safety.

2 Overall System Design

Safe direction consists of three main subsystems which are: Blind spot assistance, parking assistance and user interface unit. This section will provide an overview of the whole system design and cover the design of subsystems' common functionalities. Design details of each subsystem will be covered in later sections.

2.1 System Overview

Safe direction is a portable system **[R2_II]**, which can be installed easily by the user **[R4_III]** on various types of household automobiles **[R1_II]**. The system assists the driver by monitoring the driver's blind spot and giving a notification when an object is within the driver's blind spot. In addition it gives the driver parking assistance by giving a notification when an object is at close proximity of the vehicle's front or back side.

The following drawing shows an overview of the system's features:

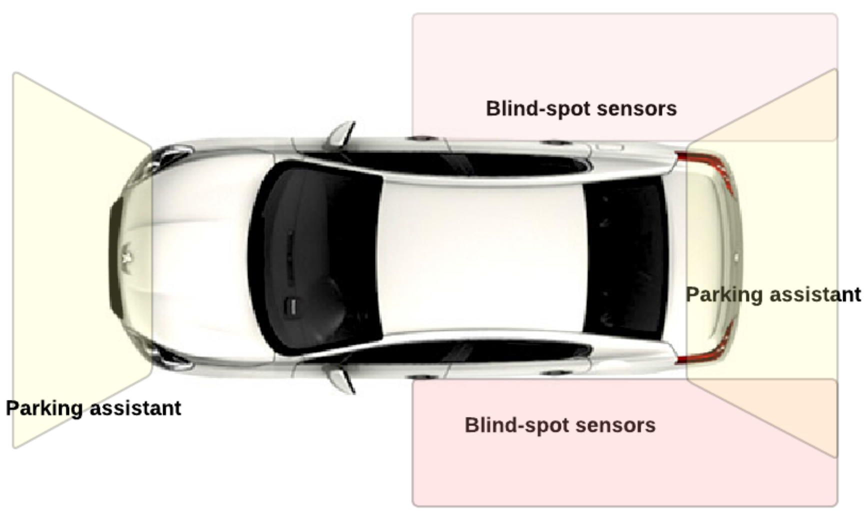

Figure 1: Drawing of the overview of the system

The system main components are: 12 ultrasonic sensors, 3 microcontrollers, LED lights and a buzzer. Those components are distributed over the three main subsystems mentioned earlier (i.e. parking assistance, blind spot monitoring and main processing unit). Parking assistance and blind spot monitoring are installed outside the vehicle and consist of: two Arduino Uno Microcontrollers, twelve ultrasonic sensors and two RF transmitters. Parking assistance and blind spot monitoring transmit the data collected from the sensors to the main processing unit which is installed inside the car and consists of: one Arduino Uno Microcontroller, one RF receiver, LEDs and buzzers to caution the driver when an object is detected depending on the data it received as shown in the following high level block diagram.

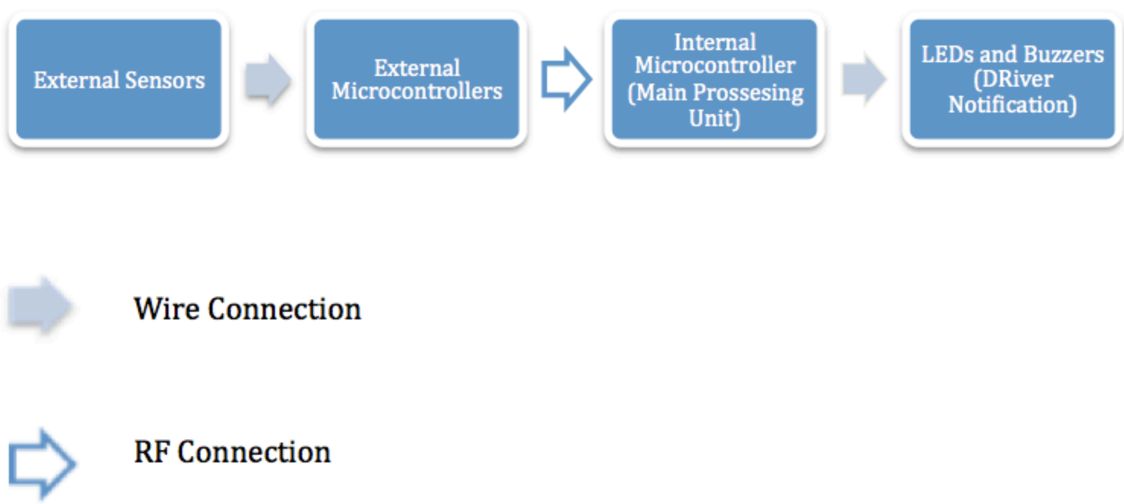

Figure 2: High Level Block Diagram

Details of the sensors distribution, data transmission and systems block diagram are given in later subsections.

2.2 Sensors

Sensors used for Safe Direction are HC-SR04 Ultrasonic Sensors which are shown in Figure 3.

Figure 3 : A picture of the ultrasonic sensor HC-SR04

HC-SRO4 starts measuring when its triggered and receives a high pulse of 5V for 10us, then the sensor will transmit 8 cycles of ultrasonic signal at 40 KHz and wait to receive the reflected signal at the receiver. Upon receiving of the reflected signal, the sensor's eco pin is set to high (5V) for a period of time that is proportional to the distance between the sensor and the reflection surface i.e. the detected object [1]:

HC-SRO4 Technical Specifications are shown in the following table [1]:

Table 1: Technical specifications of the ultrasonic sensor [1]

Power Supply	+5V DC
Quiescent Current	<2mA
Working Current	15mA
Effectual Angle	<15°
Ranging Distance	2cm – 400 cm
Resolution	0.3 cm
Measuring Angle	30 degree
Trigger Input Pulse width	10uS
Dimension	45mm x 20mm x 15mm

HC-SRO4 Sensors were chosen for Safe Direction since their technical features meet the demands for Safe Direction in terms of ranging distance **[R52_II]** and measuring angle **[R53_II]**. In addition, HC-SRO4 are cost efficient compared to high performance ultrasonic sensors which have better technical features which meets the product budget and aim to maintain a low production cost.

2.2.1 Sensors' Placement

2.2.1.1 Parking Assistance Sensors

As mentioned earlier, Safe Direction includes 4 ultrasonic sensors placed at the front of the car and 4 at the back of the car for parking assistance.

Figure 4: Parking assistance sensors distribution and coverage range

In order to determine the number of sensors needed at each side and their placement locations, we examine the sensor's range shown in figure 4.

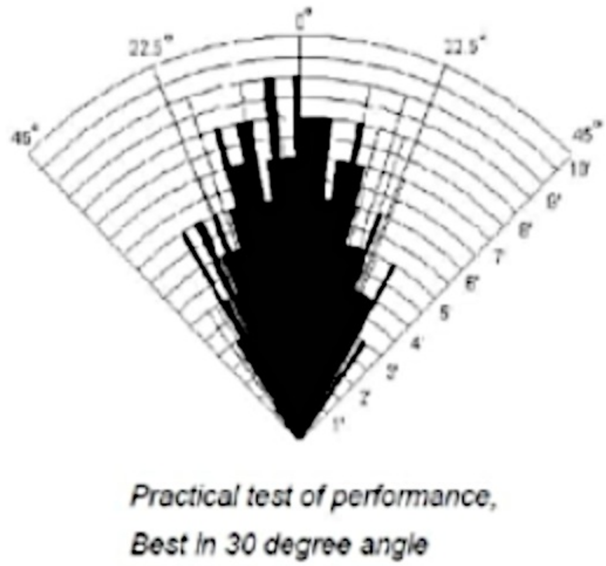

Figure 5: The converge range of an ultrasonic sensor [2]

As it can be seen from the figure and the technical specifications in table1, the sensors' angle of detection is best at 15 degrees to each side of the sensor's center which gives us a total of 30 degrees measuring angle. The figure also shows that the width which the sensor covers increases as we go further from it.

Based on this information, we find the relationship between the width (w) which the sensors covers and the distance from the sensor (d) as shown in the following figure:

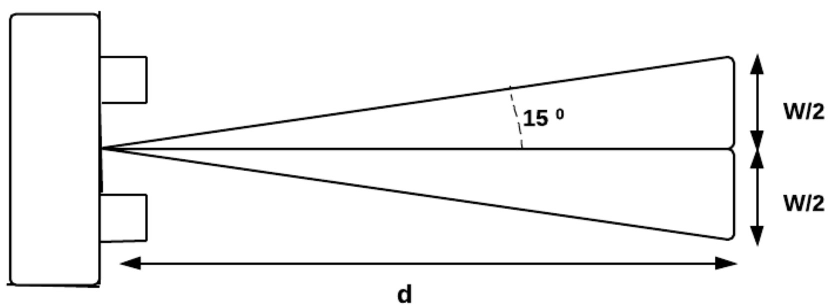

Figure 6: The relationship between the sensor-to-object distance and the width covered

$$\tan(15) = \frac{\frac{w}{2}}{d} \text{ (eq.1)}$$
$$w = (2d)(\tan(15)) \text{ (eq.2)}$$

As Safe Direction parking assisting aims to cover a parking space width, which is standardized at 2.4 m [3]. Then, the relationship between to total space covered and the number of sensors (n) is as follows:

$$n * w = 2.4\,m \quad \text{(eq.3)}$$

$$n = \frac{\mathbf{2.4\,m}}{\mathbf{2d\,tan\,(15)}} \quad \text{(eq.4)}$$

Examining formulas (2) and (4) we choose the number of sensors to be 4. Although it does not totally cover the 2.4 m width at distances closer than 1.12 m which means there will be uncovered gaps between each sensor's covered rang. In practice the total width does not need to be totally covered at close distances to the vehicle since the device assists the drive against hitting a car or a wall when parking. Such wide objects will not fall in the uncovered gaps.

The following table shows the uncovered gap width at several distances from the vehicle.

Distance From the vehicle	Uncovered Gaps Width
>112 cm	0
100 cm	8.55 cm
90 cm	15.69 cm
80 cm	22.84 cm
70 cm	29.98 cm
60 cm	37.13 cm
50 cm	44.27 cm
40 cm	51.42 cm
30 cm	58.56 cm
20 cm	65.71 cm
10 cm	72.85 cm

Table 2: A list of the distance from the vehicle against the uncover gaps

Therefore, Safe Direction parking sensors for both front and backsides will include 4 sensors evenly distributed over the cars width as shown in the following figure:

Figure 7: The sensors' distribution on the front bumper for parking assistance

2.2.1.1 Blind Spot Assistance Sensors

Since standard minimum lane width is 3.6m [4] and taking in consideration the previous standard width for parking space, then the distance between two adjacent cars on road shall be calculated as follows:

$$d = 3.6\ m - 2.4\ m = 1.2\ m$$

At that distance each sensor covers a width of 64.31 cm. By taking in consideration that blind spot sensors aim to detect cars within the blind spot as was shown in figure (system overview figure). Then, we chose to place two sensors at each side, which will cover a total length of 3m on the side of the car with a gap of 171.38 cm which does not fit a car and therefore will not cause the problem of not detecting a car within the driver's blind spot.

The following diagram shows the placement locations and length covered by the sensors:

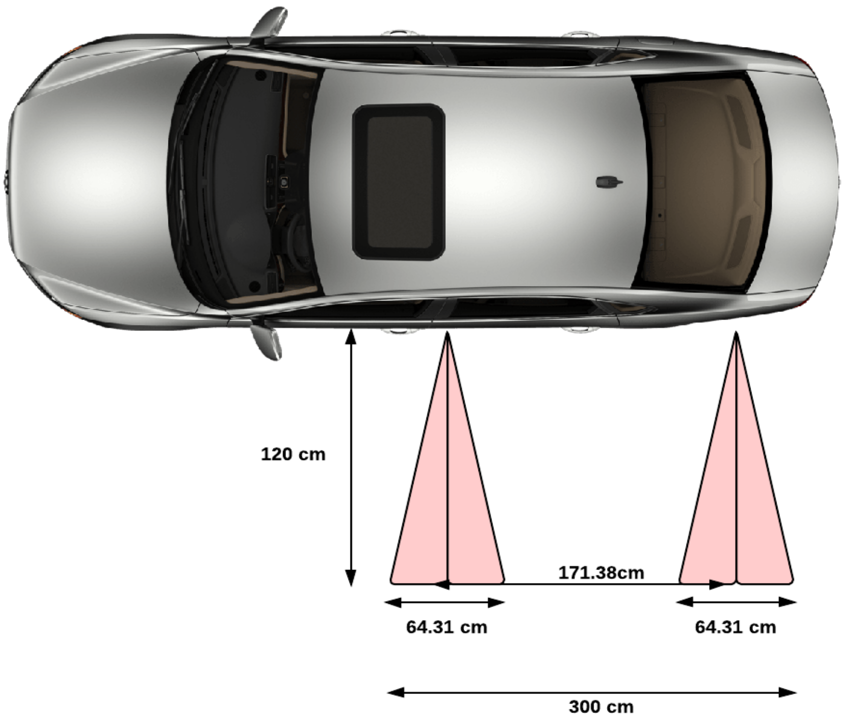

Figure 8: The sensors location and coverage range for the blind spot detection system

Therefore, the back blind spot sensor will be placed at 32.16 cm from the back of the car and the front blind sensor will be placed at a distance of 235.7 cm from the back sensor.

2.3 Data processing

2.3.1 Microcontroller Specifications

In order to process the data collected from the ultrasonic sensors and perform the desired tasks, Safe Direction uses the Arudino Uno microcontroller shown in the following figure:

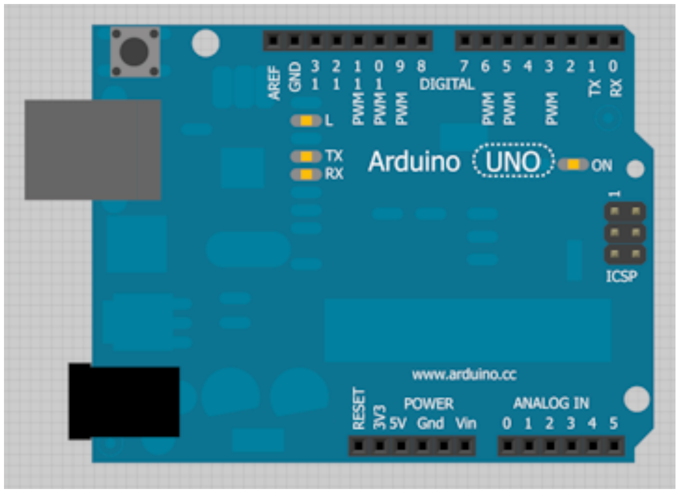

Figure 9: A picture of Arduino UNO board (microcontroller) used in the system [5]

The need for usage of microcontroller in Safe Direction arises from the need to wirelessly transmit the data gathered from the external sensors to the main processing unit which is installed inside the car **[R72_II]**.

Arudino Uno has the following technical specifications [6]:

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Table 3: Technical specifications of Arduino UNO [6]

2.3.2 Microcontrollers Distribution

A minimum of three microcontrollers is needed for Safe Direction, as we have two separate group of sensors installed. One group at the front of the car including: Four parking assisting sensors and two blind spot monitoring sensors (one on the right and one on the left) (Previous figures). Another similar group of sensors is installed on the back.

Each group needs to be connected to a microcontroller in order to wirelessly transmit the data to the main processing unit. Moreover, the main processing unit also needs to include a microcontroller in order to wirelessly receive the data transmitted from the sensors **[R72_II]**.

Data transmitting is demonstrated in the following block diagram:

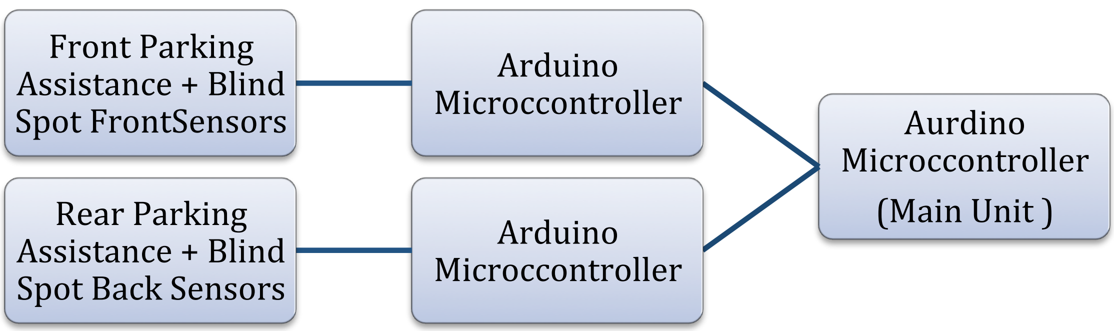

Figure 10: Data travelling path block diagram

2.3.3 Microcontroller Software

Safe Direction is programmed using Arduino IDE 1.0.5 software, which support C and C++ programming languages. Details of programming of each microcontroller will be given in their corresponding sections **[R74_III]**.

2.3.4 Wireless Communication

For wireless communication between the external and internal components of Safe Direction we use WRL-10535 RF-link transmitter and WRL-10533 RF-link receiver shown in the following figures:

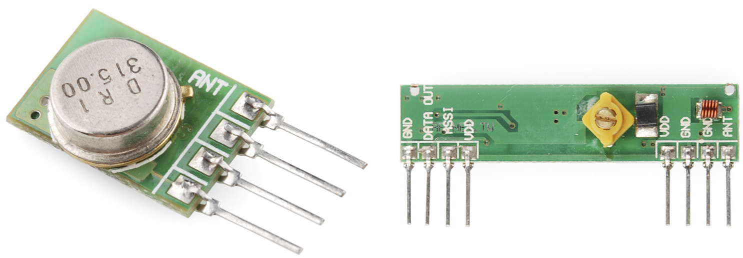

Figure 11: A picture of the RF-transmitter on the left, and the RF-receiver on thr right [7][8]

Radio Frequency transmits and receives through electromagnetic waves, which propagate for long distance. In addition it has the advantage of passing through non-shielding material [9]. Those properties of RF transmission make it useful for our system as it guarantees signal propagation through the vehicles parts.

315 MHz frequency band was chosen for our system as it is widely used for remote controls [9] and therefore, we will not have frequent interference from the cars devices while the car is in motion.

An additional important aspect for the choice of RF transmission is its low power consumption compared to other wireless transmission options such as Bluetooth.

WRL-10535 RF-link transmitter and WRL-10533 RF-link receiver technical Specifications are shown in the following tables:

Transmitter Technical Specifications [7]:

Frequency Range	315MHz
Modulate Mode	ASK
Circuit Shape	SAW
Date Rate	8Kbps
Supply Voltage:	1.5~12V
Output Power	14dBm
Working temperature	-20~+85℃
Solder temperature	230℃(10 seconds)

Figure 12: Technical specifications of the transmitter [7]

Receiver Technical Specifications [8]:

Frequency Range	315MHz
Modulate Mode	ASK
Circuit Shape	LC
Date Rate	4800 bps
Selectivity	-108 dBm
Channel Spacing	±500KHz
Supply Voltage	5V

Figure 13: Technical specifications of the receiver [8]

3 Parking Assistance Design

The parking assistance system is used to detect objects that are on front or behind the car in order to guide the driver when parking. The system consists of two duplicate units. As shown on fig.7, one unit is mounted on the front bumper of the car and the other unit is attached to the back bumper.
Each unit composes of four main parts:

1) Four Ultrasonic Sensors
2) Arduino UNO microcontroller
3) RF 315MHz transmitter
4) 9V battery adapter
5) Buzzer

3.1 Circuitry Design

The system internal connections are shown in fig.14, and the pins descriptions are shown in table4:

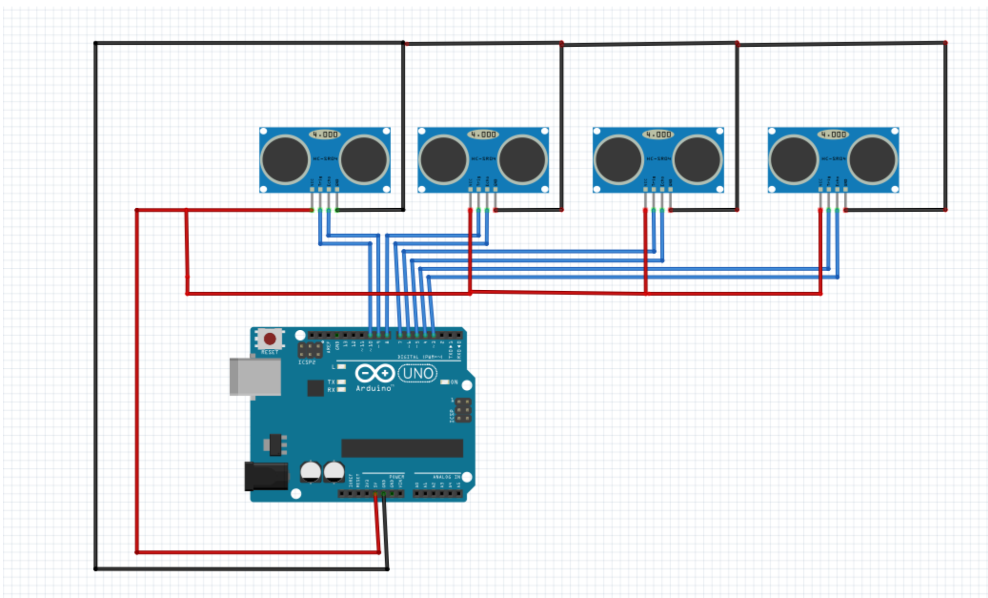

Figure 14: The electrical connections of the Parking Assistance system

Arduino Pin no.	Corresponding Pin in parts	Description
1	(Data pin) in RF transmitter	The distance in cm to an object is sent to the transmitter
2	(Trig pin) in sensor1	Arduino activate the sensor and let it start sending pulses
3	(Echo pin) in sensor1	The returned pulses are received and sent from transmitter to Arduino

4	(Trig pin) in sensor2	Same as Trig pin in sensor1
5	(Echo pin) in sensor2	Same as Echo pin in sensor1
6	(Trig pin) in sensor3	Same as Trig pin in sensor1
7	(Echo pin) in sensor3	Same as Echo pin in sensor1
8	(Trig pin) in sensor4	Same as Echo pin in sensor1
9	(Echo pin) in sensor4	Same as Echo pin in sensor1
GND	To all GNDs in all parts	All four parts in the system share a common ground
5V	To all input Vcc in all parts	Sensors and the transmitter are powered up by a common 5V input
External power Plug	9V battery adapter	Arduino is powered up by the 6 AA batteries

Table 4: Pin assignment and description of the parking assistance system

3.2 Parking Assistance Flow Chart

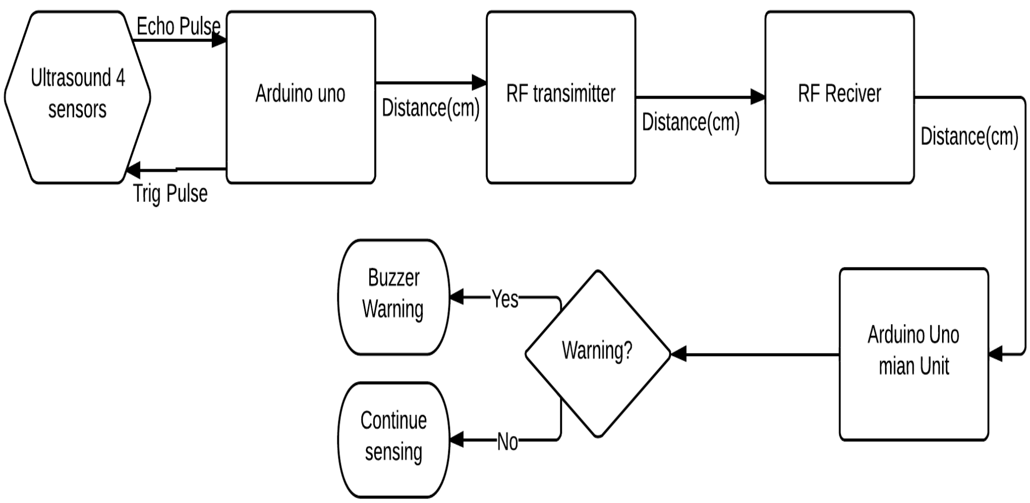

Figure 15: Signals flow chart for the parking assistance system

3.3 Parking Assistance Software Design

Setup function:

The system will initialize the input and output pins in the setup function. Furthermore it will set the required bit rate for RF-transmission. Nine digital pins are used in the system. Eight pins are for the sensors, two pins for each sensor. The echo pin is an input, and the trig pin is an output. The last digital pin is connected to the RF-transmitter, and it is an output pin.

Loop Function:

The loop function contains the core code of the system. This function will keep looping while the Arduino is powered. The code contains a Ping() function that is called to send a pulse to the sensor and receive it back. Ping() calculates the duration by measuring the time it takes to send and receive a pulse. Also, it calculates the distance between the sensor and the detected object and returns the distance. The equation used for distance calculations is:

$$\text{Distance} = \text{Duration}/58.2\ (\text{cm})$$

The number "58.2" is the time required for a sound wave to travel 10m and come back, i.e. travel for 20m.

$$58.2\text{msec} = 20\text{m} / (343.5\text{m/s})$$

343.5m/s is the speed of sound in 20°c.
The sensors are activated in a sequential manner. Ping() will call each sensor once in the main-loop function. If the object-car distance is within the range specified in the system overview, the distance will be sent to the RF-transmitter (pin1). Otherwise, the system will disregard any object out of range.

4 Blind Spot Design

The Blind spot Detection system is used to detect objects that are at the blind spots of the car, see fig4. The system consists of two sensors on each side of the car. The sensors location and mounting are shown in fig5. The two front sensors are connected wirily to the front Arduino, and the two back sensors are connected to the back Arduino.

Each unit composes of four main parts:

1) Four Ultrasonic Sensors
   (Two sensors on each side of the car)
2) Two Arduino UNO microcontroller
   (The same ones used in Parking Assistance System front/back)
3) Two RF 315MHz transmitter
   Two Arduino UNO microcontrollers
   (The same ones used in Parking Assistance System- front/back)
4) Four AA batteries holder
   (The same one used in Parking Assistance System)
5) Eight LEDs

4.1 Blind Spot Circuitry Design

The system internal connections are shown in fig.6, and the pins descriptions are shown in table2:

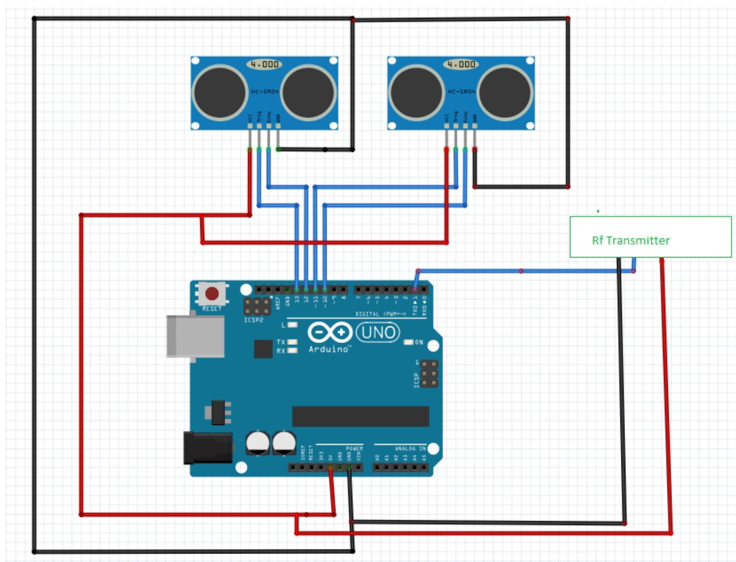

Figure 16: The electrical connections of the Blind Spot Detection System

Arduino Pin no.	Corresponding Pin in parts	Description
1	(Data pin) in RF transmitter	The distance in cm to an object is sent to the transmitter
10	(Trig pin) in sensor1	Arduino activate the sensor and let it start sending pulses
11	(Echo pin) in sensor1	The returned pulses are received and sent from

		transmitter to Arduino
12	(Trig pin) in sensor2	Same as Trig pin in sensor1
13	(Echo pin) in sensor2	Same as Echo pin in sensor1
GND	To all GNDs in all parts	All four parts in the system share a common ground
5V	To all input Vcc in all parts	Sensors and the transmitter are powered up by a common 5V input
External power Plug	9V battery adapter	Arduino is powered up by the 6 AA batteris

Table 5: Pin assignment and description of the Blind Spot Detection System

4.2 Blind Spot Flow Chart

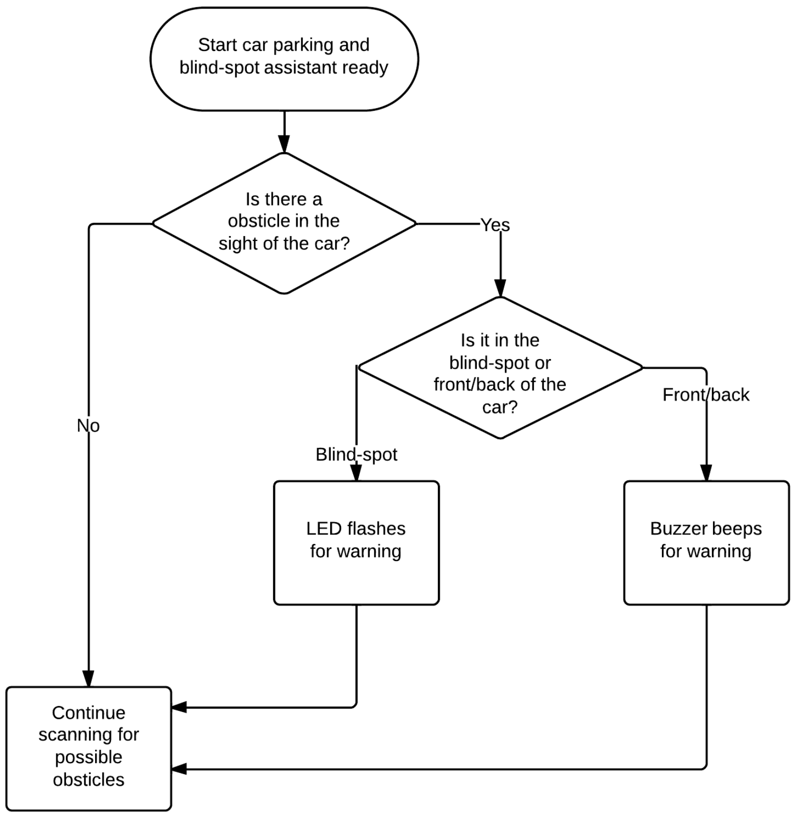

Figure 17: Signals flow chart for the Blind Spot Detection System

.

4.3 Blind Spot Software Design

Since the sensors in the blind spot detection system are connected to the same Arduino used in Parking Assistance System, the setup and loop functions contains code for both systems together.

Setup function:

The system will set the input and output pins for the sensors attached either at the front of the car or back. Eight digital pins are used in the system, two pins for each sensor. The echo pin is an input, and the trig pin is an output. The last digital Since the RF-transmitter is shared between blind spot detection system and parking Assistance System, the transmitter pin is already sat.

Loop Function:

The four sensors will be called using the Ping() function mentioned in the parking assistance section. The exact same procedure will be followed for object detection and sending the distance to the transmitter. However, since the main Arduino need to distinguish between left and right sensor to light up the right LEDs, the distance sent to transmitter must be associated with a character that will state the position of the detected object. An "L" or "R" characters are sent alongside the distance. 'L' refers to left, and 'R' refers to right. The sensors are activated in a sequential manner. Ping() will call each sensor once in the main-loop function. If the object-car distance is within the Max and Min range specified in the system overview, the distance will be sent to the RF-transmitter with the required character (pin1). Otherwise, the system will disregard any object out of range.

5 Main Processing and User Interface Unit Design

5.1 Main Processing Unit

Main processing units is the brain of the product. It consist of 5 main parts

1. Arduino UNO microcontroller
2. Receivers

3. Stand by switch
4. 2 LEDs array
5. 1 Buzzers

5.1.1 Main Processing Unit Logic Design

The following flow chart demonstrates main unit logic:

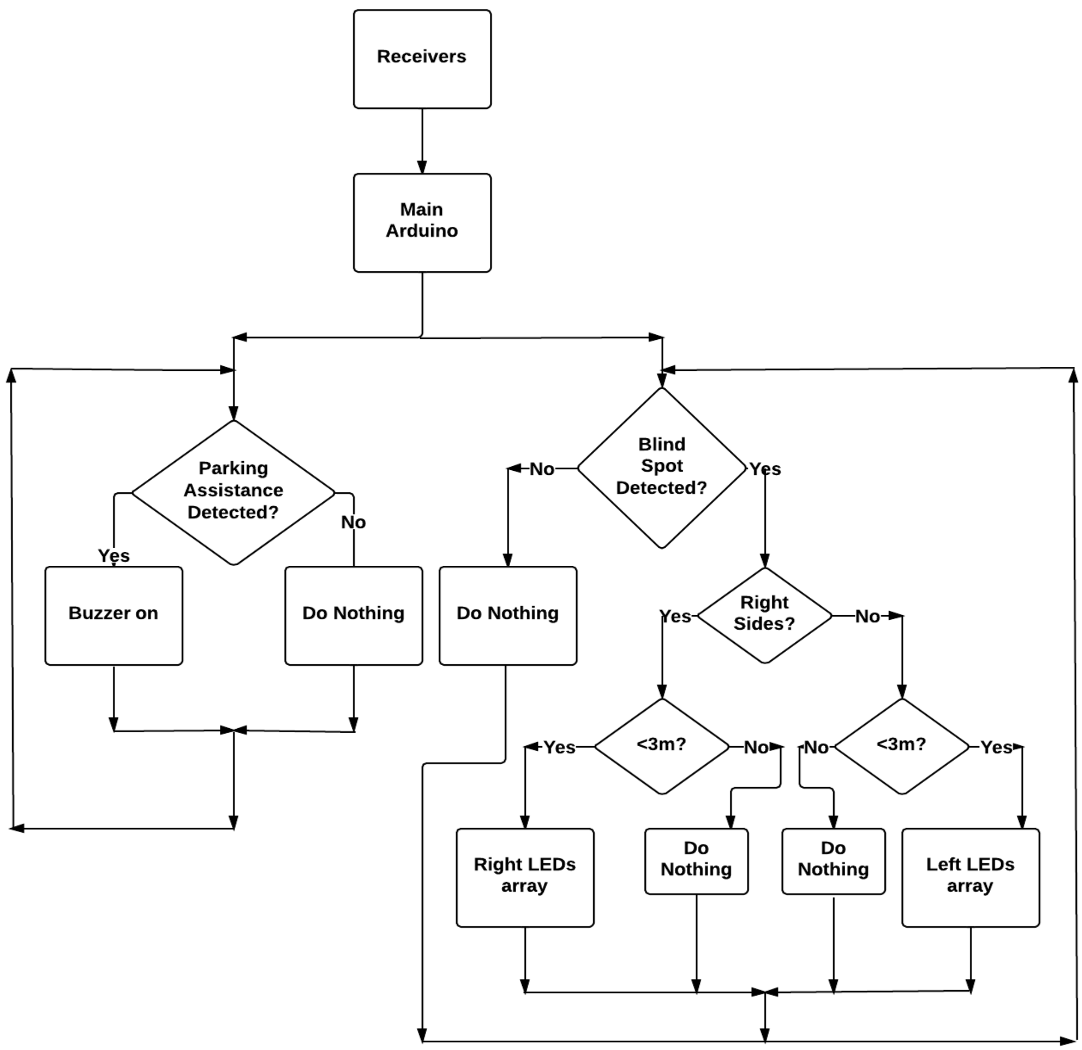

Figure 18: A flow chart of the main logic unit

In addition, the main unit processes a protocol for setting the system into sleep mode where power consumption is reduced. In this feature, the driver can turn off a switch so that main Arduino sends a signal to both front and back Arduinos and put them into sleeping mode.

5.2 User interface

The user interface indicates what our product is made of and what every element does. Our blind spot system and parking assistance system consist of 3 parts. They are LEDs, buzzers, and microcontrollers. LEDs and Buzzers are the notification elements. Microcontroller is the brain of our product. It receives the data from all the sensors and sends a message to the buzzers and the LEDs to notify the users.

5.2.1 User interface Hardware

The hardware consists of the buzzers and LEDs. The role of buzzers and LEDs are to notify the users. The difference between them is that buzzers are for parking assistance and LEDs are for blind spot detection. In other words, buzzers work when obstacles appear in front, and back of the car. LEDs work when there are obstacles on the sides of the car.

5.3 Main Unit and User Interface Software Design

Setup function:

The system will initialize the input and output pins in the setup function. Also, will set the required bit rate for data reception (must be equivalent to bit rate of transmission). Four digital pins are used in the system. The first pin is pin 7 for the data received which is connected to the RF-receiver and is an input pin. The second and third pins are output pins and are connected to the right and left LEDs (pins 12 and 13, respectively). The fourth and last pin is for the buzzer and is an output pin (pin 11).

Loop Function:

The loop function contains the core code of the system. This function will keep looping while the Arduino is powered. The code will wait for data (distance + ID character). The code will recognize the ID character and will act according to the following rules:

1) The received ID is for front or back sensors (parking assistance system), then sends a signal to the buzzer (pin 11) if the distance is within the range [5cm, 200cm]

2) The received ID is for right side sensors (Blind Spot Detection system), then sends a signal to the right LEDs (pin 12) if the distance is within the range [5cm, 300cm]

3) The received ID is for left side sensors (Blind Spot Detection system), then send a signal to the left LEDs (pin 13) if the distance is within the range [5cm, 300cm]

6 Electrical Design

The overall system will have three power supplies:

1) 6 AA battery holder (9V) for the front Arduino on front bumper **[R6_II]**
2) 6 AA battery holder (9V) for the back Arduino on back bumper **[R6_II]**
3) 12V car plug for the main Arduino on the dashboard **[R7_III]**

Devices	Power Supply	Current Consumption (max)(mA)
Front bumper Devices: Arduino + 6 sensors + RF-transmitter	6 AA battery holder (9V)	98.35
Back bumper Devices: Arduino + 6 sensors + RF-transmitter	6 AA battery holder (9V)	98.35
Main Unit on dashboard: Arduino + Receiver + 8 LEDs + Buzzer	12V car plug	70.1

Table 6: A table of the overall current consumption and power supplies

The main power issue resides in the front/back power supply since they are battery powered. The main unit power supply is not an issue since it is connected directly to the car battery. According to Energizer's website (batteries producer), the capacity of an AA battery at 100mA current consumption is 2500mAh [10], see fig.19.

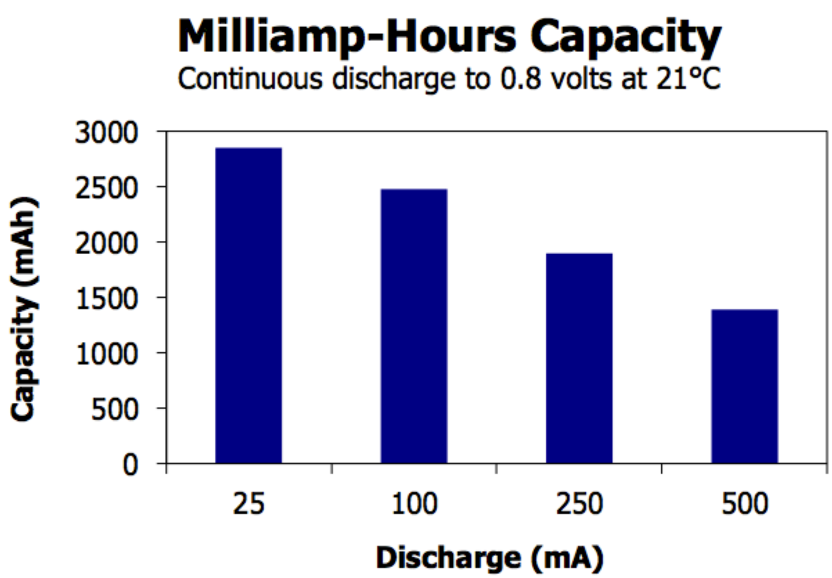

Figure 19: A diagram of the capacity (mAh) of AA battery against current discharge (mA)

Therefore, the lifetime of the 6 AA battery supplier, used in powering front/back Arduinos, is around 24 hours. To save energy, the code contains functions that will put Arduino to sleep mode when the sensors have not been used for 5 min. Arduino draw 34.4 mA at sleep mode. According to JRC Scientific and Policy report regarding Driving and Parking patterns of European car drivers, the average daily travel time is 1.5 - 2 hours [11]. Consequently, The front and back bumpers units will be active at 100mA for 1.5-2 hours if the system didn't go to sleep during this time. For the remaining 22 hours, the system will be asleep and consuming 34.4mA. At the previous current consumption, the battery lives for 80 hours. The following calculations will demonstrate the 6 AA battery holder lifetime at the front and back bumpers units:

(Assuming the battery doesn't go to sleep during travelling and travelling time is 2 hours)

The average AA battery's capacity 2700mAh.

	Day1	Day2	Day3	Sum
	2 hours*100mA = 200mA	2 hours*100mA = 200mA	2 hours*100mA = 200mA	600mA
	22hours*34.4mA = 756mA	22hours*34.4mA = 756mA	22hours*34.4mA = 756mA	2268mA
Total current consumption	956 mA	956 mA	956 mA	2868 mA

Table 7: Calculations of current consumption of the whole system

The calculation shows that the 6 AA battery supply will hold for approximately three days. The calculations are an approximation for the ideal situation. Consequently, the batteries will only be used for proof of concept model since changing 12 batteries every three days is not practical. For the final product, the car battery should power the front and back bumpers units.

7 Mechanical Design

Safe Direction mechanical design includes the design and placement of the external microcontrollers and their circuitry **[R19_III]**.. In addition, it includes the casing of the internal main processing unit and user interface.

7.1 External Microntrollers

Each of the front and back microcontroller circuitry will be implemented using a PCB board connected to each microcontroller. The casing for each external microcontroller will be a box containing the microcontroller with its PCB board as shown in the following figure:

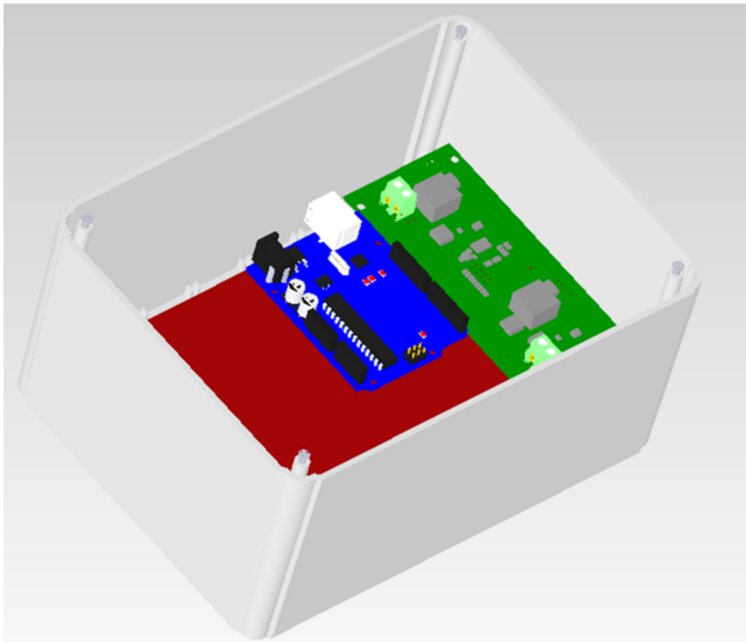

Figure 20: A design of the box carrying the PCB board and the microcontroller [12]

The box dimensions are 15cm x 10cm x 5cm. Box containing front side Arduino will be attached beneath the front bumper so that it does not affect the vehicle's outlook while the box containing the back side Arduino will be attached beneath the back bumper.

3M tapes are to be used for the casing attachment to the car.

7.2 Sensors

Sensors are placed at the locations explained earlier under the overall system design section. 3 M tapes are also used for sensors attachment to the car, as they do not damage the cars body.

7.3 Main Processing Unit

Main processing unit casing is similar to the casing shown in figure 20 but will include LEDs and buzzer installed on the outer surface of the box in order for them visual to driver as shown in figure 21. The box will be placed on top of the car's dashboard where it will be within the drivers' sight.

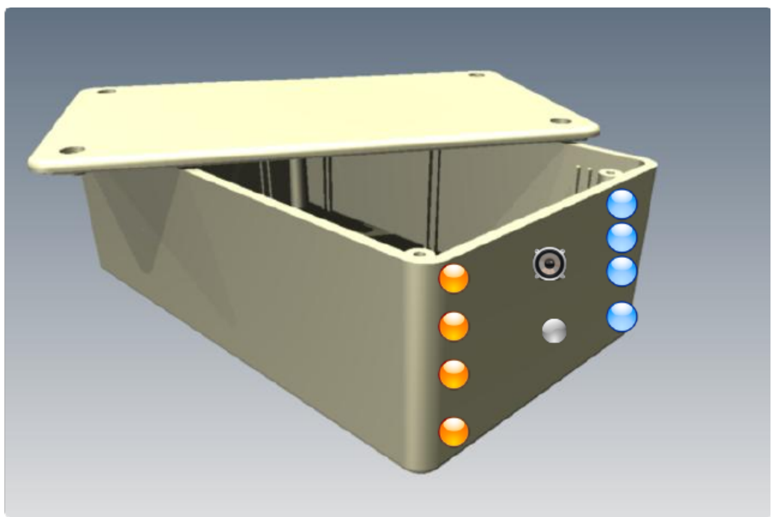

Figure 21: Main Unit Casing

8 Test Plan

8.1 Individual Testing

8.1.1 Ultrasonic sensor

The ultrasonic sensor will be tested on the given specifications for the accuracy in sensing range and width with respect to object size and distance based on the calculated values **[R52_II] [R53_II]**. Testing on the ultrasonic sensor will include but not limited to:

I. Accuracy of the sensing range and width
II. Interference between sensors
III. Signal delay for multiple sensors
IV. Testing sensors for false detection
V. Measuring the maximum accurate sensing distance on different obstacle size

8.1.2 Battery

After combine all the individual modules together, the system will be tested on the power consumption while using battery as the supply. Testing on the power consumption will include but not limited to:

I. The maximum continuous operation time duration
II. The best trade between duration and battery size
III. The consumption level while in sleep mode

8.1.3 Wireless module - Transmitter and receiver:

The wireless modules will be tested under the given specification and the accuracy and limitation of the signal distance, interference level, and accuracy of transmitted signal. Testing on the wireless module will include but not limited to:

I. Accuracy of the signal transferred
II. Interference with other products near by
III. The accuracy of the signal transfer under different weather conditions

8.1.4 Buzzer

The buzzer evaluation will be based on the strength of the notification and adjusting it to the most comfortable level for the user **[R68_III]**. Testing on the buzzer will include but not limited to:

I. Frequency of the buzzer sound
II. Accuracy of the buzzer

8.1.5 LEDs

The LED lights will be tested on the functionality and the warning notifications provided **[R66_II] [R67_III]**.

8.2 Arduino UNO+ Transmitter and receiver

The Arduino board and transmitter/ receiver will be used to test the wireless communication for the system. Considering the RF transmitter and receiver is the cheapest wireless communication method out there, the liability of the product needs to be extensively tested in the following areas:

I. Accuracy of the signal transferred
II. The signal transferred interval
III. The delay between signals send

8.3 Arduino UNO+ Buzzer

The Arduino and buzzer will test with variations of buzzer response time wirelessly and the level of notification that it provides.

8.4 Arduino UNO+ Ultrasound sensor

The Arduino board and ultrasonic sensor tests the programmable limitation of the sensor and code. It will be tested under conditions to ensure that the microcontroller will be able to handle multiple sensor data and analysis it in a timely manner. The testing will include:

I. Ensure all data transferred for multiple sensors are accurate
II. Test the interference between multiple sensor and microcontroller

8.5 Power consumption testing

Test and evaluate the power consumption of the system, evaluating the battery power usage and the accuracy of the signal under different power levels left in the battery. The test will include:

I. Buzzer noise level in lower battery condition
II. Single transfer accuracy in lower battery condition
III. Warning system for low battery

8.6 User interface Testing

I. Recognizable warning system
II. Comfortable level of notification

8.7 Unit Testing

Combining all individual parts and testing on real environments:

I. Ease of installation
II. Accuracy of the sensor
III. The possible interference
IV. Wireless delay between microcontrollers
V. Waterproof level

9 Conclusions

The design specifications will be built up from the functional specifications previously proposed and will enable the Car Sense team to conform to the functional specifications that have already been produced. The planning along with the calculations in this document will guide our team to implement a working proof of concept of the Safe Direction system. The test plan outlined in this document will let us know of any issues or problems that might rise up during the final stages. This will help us address any problems with the functionalities of the system and to ultimately deliver a working proof of concept on time by the designated deadline.

10 References

[1] Cytron Technologies Sdn. Bhd. (May 2013). Product User's Manual – HC-SR04 Ultrasonic Sensors. https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit

[2] Elec Freaks. Ultrasonic Ranging Module HC – SR04. http://www.micropik.com/PDF/HCSR04.pdf

[3] Ezekiel Data and Mike Furuya. (2nd Quarter 2010)Parking Dimensions. http://www.canadianparking.ca/files/ParkingDimensions_eng.pdf

[4] 1st Tram Consulting Ltd.(October 2006). Local Road Standards. http://www.cvrd.bc.ca/DocumentCenter/Home/View/1189

[5] M. Grinberg, miguelgrinberg blog, Image: http://blog.miguelgrinberg.com/static/images/arduino-robot-11.png

[6] Arduino uno http://arduino.cc/en/Main/arduinoBoardUno#.Ux-fi0JdWCZ

[7] Wireless Hi Power Transmitter Module (RF ASK) http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/General/TWS-BS-6_315MHz_ASK_RF_Transmitter_Module_Data_Sheet.pdf

[8] Wireless Hi Sensitivity Receiver Module (RF ASK) http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/General/RWS-374-3_315MHz_ASK_RF_Receiver_Module_Data_Sheet.pdf

[9] Martin Gotschlich, (2010). Infineon Technologies AG, 2010. Remote Controls – Radio Frequency or Infrared https://www.infineon.com/dgdl/RF2ir+WhitePaper+V1.0.pdf?folderId=db3a3043191a246301192dd3ee2c2ae4&fileId=db3a30432b57a660012b5c16272c2e81

[10] Product Datsheet ENERGIZER E91
http://data.energizer.com/PDFs/E91.pdf

[11] G. Pasaoglu, D. Fiorello, "Driving and parking patterns of European car drivers", European Commission, Milan, Italy, 2012.
Keren Reference:

[12] Edge Project server, public images:
http://edge.rit.edu/content/P12472/public/PCB_Prelim_InBox.png