Capstone Design Report

Project Title: Red Light Detection System

Team #20

Team Name: Red Light Rangers

Team Members: Mathias Dawit, Liam Kearns, Ben Moskos, Megan Murphy

Electrical and Computer Engineering Department

College of Engineering

Ohio State University

ECE 4905

Date: 4/30/2023

Document Change Notice

Date	Change
12/15/2022	Initial Release
4/30/2023	Capstone 2 Revision

Executive Summary

The area surrounding red lights is one of the most dangerous sections of the road. Distracted driving has greatly increased the danger presented by red lights. In 2020 there were 116,000 car accidents at and around red lights with nearly 1,000 of these accidents being fatal [2]. Many of these accidents are caused by distracted driving, whether that is a distraction by electronic devices or another distraction, it is our goal to cut down on these accidents.

There are many stakeholders invested in the safety of traffic light areas, from the municipality and city workers to the drivers and insurance companies. For the municipality and insurance companies it comes down to financial incentives, however when it comes to drivers, pedestrians and response workers it comes down to the safety and wellness of everyone involved.

To start attempting to tackle this issue we calculated the estimated reaction and stopping time required by a driver when a red light is activated. We completed these calculations for a variety of common speeds and found a stopping distance in feet and car lengths as show in [Table. 1]. After calculating the stopping distance for different situations that the device could be used in, we got to work on requirement specifications. The requirements that we specified were detection distance, signal processing time, audible output, and operating speed range of the device, these specifications were presented in [Table. 2].

The final design came after screening five different designs. Concept A was a solar powered camera and alert system connected to onboard sensors to measure speed and acceleration. Concept B was DC powered into the car port from a wire and processed information from the onboard computer. Concept C was a camera and alert system reading car information via Bluetooth to the car's onboard computer. Concept D was a camera and alert system taking real-time data from traffic management to alert the user when approaching a red-light. Lastly, concept E was a roof-mounted and solar powered device connected via Bluetooth. The five concepts were evaluated on their prospected size, cost, ease of use, reliability, safety, and aesthetics. These evaluations are seen in [Table. 3]. To further weigh out our options we decided to utilize a weighted concept matrix and redo the scoring, this showed that concept B came away with the highest score, with concepts A and C also being possible viable options [Table 4]. Further work was completed on concept B, this led to a relatively simple hardware installation into a test vehicle.

Work progressed throughout the semester on both software and hardware fronts. Major breakthroughs were made in terms of detecting the traffic lights themselves and obtaining the speed of the vehicle. Full integration of systems was completed by the first week of April, this allowed all tests to be completed by April 13th. While all systems were completed and integrated there were still issues with the filter. Different lighting conditions lead to inconsistent results that will hopefully be able to be resolved by the next group working on this project.

Overall, the project reached all the major milestones that were set by the team but in the future, there are some small bugs that will need ironed out to be implemented fully.

Table of Contents

List of Figures and Tables

Chapter 1: Problem Definition

Introduction

The area around red lights are one of the most dangerous sections of the road, many of these accidents are caused by distracted drivers. These drivers are often more focused on roadside distractions or mobile electronic devices and may not notice the change of a traffic signal in front of them.

There are many stakeholders invested in solving this problem. Drivers on the road, Pedestrians, the municipality, insurance companies and emergency/recovery workers. Firstly, there are the drivers themselves that are physically and financially invested in the road being a safer place. Pedestrians are also invested in a safer intersection area whether they are on foot or another form of transportation. The municipality that controls the traffic system around these intersections will also be heavily invested in creating a safer and more efficient road system for everyone, as this is one of their priorities. Insurance companies will be looking to mitigate potential traffic accidents in order to save money on payouts for repairs. Finally, there are emergency and recovery workers that are already often very busy that will not be required to respond to as many avoidable traffic accidents, this will also be safer for recovery workers. Recovery workers have a fatality rate that is 15 times higher than the average worker in the USA [1], cutting down on accidents could help with the safety of these workers.

There are a few primary needs for our project. We want to create a device that is affordable and accessible for a large portion of potential users. We want a simple design that avoids many potential failure points to be as reliable as possible, in addition to this we want to maintain the safety of the car by avoiding any visual obstructions while giving an audible and visual alert to the driver if their vehicle may be approaching a red light at a speed deemed to be too fast to stop.

To keep the scope of this project reasonable we are focused on creating a functional prototype that can detect red lights and give an alert by the end of the semester, we want this device to work using computer vision to ensure that the device can work on as many traffic intersections as possible. If this device was focused on an IOT structure it would be extremely expensive and time consuming to implement, so a versatile design will be used.

Background

In 2020 there were nearly 116,000 accidents involving red traffic lights in the United States alone. Of these accidents, 928 were fatal. An analysis of 19 traffic cameras in four states found an average of 3.2 red light runners per hour [2]. While these instances depend on the behavior of the driver, it is not uncommon for an individual to accidentally run a red light. The driver may be distracted by external stimuli including pedestrians, drivers, and road hazards. There are some existing ways to mitigate this problem include turning off your phone and driving at the speed limit, but this is often neglected by drivers. There are no existing products on the market that detect red lights and alert the driver if they are going to run it.

While traffic light mounted cameras exist to detect if a driver has run a red light, this does not prevent a distracted driver from stopping before they run the red light. Technology exists in

modern cars that can detect lane departure and brakes to avoid collisions. This technology uses sensors and camera vision to avoid hazards. In a few years, this technology may advance to the point where it can detect red lights, but it currently can't.

The ideal system has specified constraints and requirements. System requirements are guided mainly by the stopping distance of the vehicle. While these distances range based on the tire quality, brake quality, road conditions, and vehicle weight, these estimated distances are represented by Table 1 [3].

Table 1. Stopping Distance Table

Speed	Perception/Reaction Distance	Braking Distance	Overall Stopping Distance	Equal to Approx Number of Car Lengths (@15 feet)
30 mph	12 feet	45 feet	57 feet	4
40 mph	17 feet	80 feet	97 feet	7
50 mph	20 feet	125 feet	145 feet	10
60 mph	24 feet	180 feet	204 feet	14

Chapter 2: Preliminary Design

Project Constraints and Requirements

System computing must be factored into this stopping distance. The median reaction time for a person to react to a visual stimulus is 275 milli-seconds. The distances in the Perception/Reaction Distance column are distances covered in 275 milli-seconds at those respective speeds. All computer operations must be completed within 100 milliseconds to modestly affect stopping distance total.

The system will enter standby mode when the user is driving consistently above 60 mph. Assuming most roads over 55 mph do not have stop lights, we can say that the system does not need to operate at a speed greater than 60 mph. At the 55-mph threshold, the camera must be able to view and interpret data from 200 feet away. Once computing time is added to total stopping distance, interpreting distance may increase by ~20 feet. So, say 225 feet to be safe.

The system must accurately report red lights and not report them if inconclusive. For example, if the windshield is covered in rain, it will not be able to view as accurately. If it sees a red light shining through a rain drop that resembles a red circle, it should not alert the driver. The system should ideally operate correctly 100% of the time. It should deactivate the system when rain is detected or when the view is obscured.

To ensure the system does not create a dangerous situation for the user, alert audio is limited to 70 dB. This will get the users' attention but will not distract them from driving safely. The system requirements have been summarized in Table 2.

Table 2. Requirement Specifications

Requirement	Units	Range	Ideal
Camera perception distance	ft	200-300	225
Time to process input signal and produce output signal	s	<0.5	<0.1
Audibility of output signal	dB	60-80	70
System operation speeds	mph	5-60	N/A

Design Ideas/Concepts

The final design came after screening five different designs. Concept A was a solar powered camera and alert system connected to onboard sensors to measure speed and acceleration. Concept B was DC powered into the car port from a wire and processed information from the onboard computer. Concept C was a camera and alert system reading car information via Bluetooth to the car's onboard computer. Concept D was a camera and alert system taking real-time data from traffic management to alert the user when approaching a red-light. Lastly, concept E was a roof-mounted and solar powered device connected via Bluetooth.

The five concepts were evaluated on their prospected size, cost, ease of use, reliability, safety, and aesthetics. A + indicated a positive, a - indicated a negative, and 0 was neutral.

Table 3. Concept Matrix

	Concept A	Concept B	Concept C	Concept D	Concept E
Size	+	0	+	+	-
Cost	+	-	0	+	0
Ease of Use	+	+	+	-	-
Reliability	0	+	0	-	0
Safety	0	+	+	+	+
Aesthetics	0	-	0	0	-

Using the concept matrix and what was determined to be of value for the system requirements and user needs, concept D and E have been eliminated. Concept D is not a universally applicable concept because not every city has easily accessible traffic data. Concept E scored the lowest so therefore it has been removed as an option. Concept A and C are the most promising options with B at close second. We will adapt options from all three concepts to upgrade our design.

After doing a general evaluation, the concept went through another design concept scoring with score 1-5. 1 is a bad design aspect that should be avoided in the design. 5 is a good design aspect. While concept D and E were eliminated from the first round of criterion, they were evaluated again to ensure the concepts were not overlooked.

Table 4. Weighted Concept Matrix

	Weight	Concept A	Concept B	Concept C	Concept D	Concept E
Size	15%	4	3	3	3	2
Cost	15%	4	3	2	4	2
Ease of Use	20%	4	4	3	2	1
Reliability	25%	1	5	4	2	1
Safety	10%	3	2	3	4	3
Aesthetics	5%	4	2	3	3	1
Risk to Implement	10%	2	3	3	2	3
Score		2.95	3.55	3.1	2.7	1.7
Continue?		Improve	Yes	Yes	No	No

When scoring the different concepts, a new criterion was added. The risk of implementing takes into consideration the risk that is involved with the use of the product. For the products involving the use of systems external to our product (I.e., OBD-II port), a level of risk is involved with compatibility and functionality. If a car doesn't have a function port or the port is non-existent in an older model, the product will be obsolete. The scoring produced similar results to the screening matrix. Concept A, B, and C were given the green light to continue development. However, only one concept will survive and that will most likely be a combination of the three.

Chapter 3: Detailed Design

System Diagram

In our system design, there are three main components. The first component is the car and the information needed from the car. The car will provide the power to the actual sensing system. The sensing system comes from the environmental information component of the system design. The environment is where the car is in relation to red lights. The camera will be attached to the windshield and connected to servo motors. This allows the camera angle to be adjusted to pick up red lights in the cars area. This system needs to have an alert system. The system needs to be able to detect the red light and alert the driver. The last component is the processing information part. The original design will be attempted without a laptop, and solely rely on the Raspberry Pi board to control the system. Due to constraints on the processing capabilities on the Raspberry Pi, most likely a laptop will need to be connected to assist in the red-light detection. The last part of the processing information section is the control system that will be used to filter and detect red lights in the images captured by the camera.

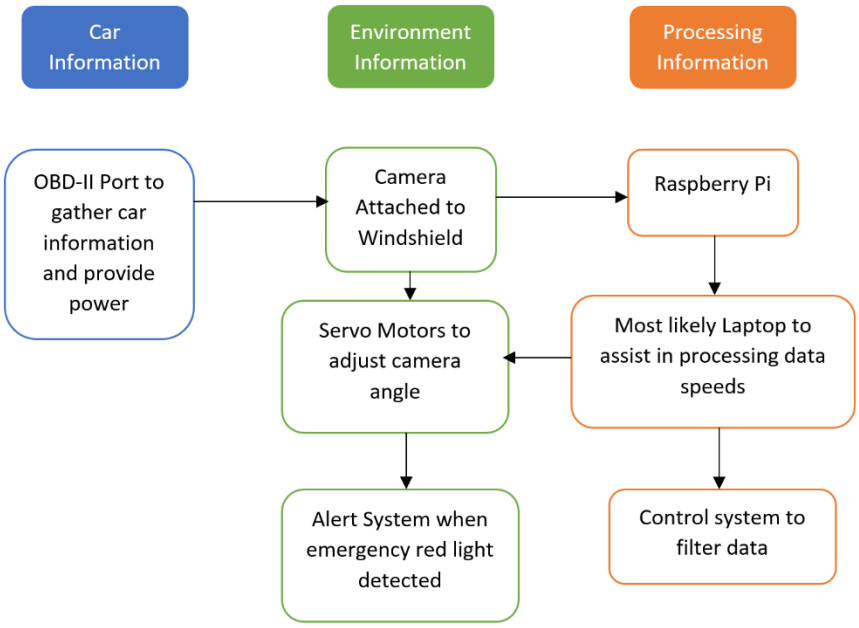


Figure 1. Flow Chart Diagram

Analysis and Evaluation

The most important part to pick for the system was the camera lens. The camera needed to have enough focal length to pick up a large enough field of vision. The Raspberry Pi 50mm camera lens provided a 50mm focal length to allow for a greater field of vision. There were 6mm, 10mm, and 16mm options for the camera lens, but the 50mm was the best choice of the three. If the camera lens has a very narrow field of vision, it could become problematic to detect red lights. The Raspberry Pi lens is $50[4]. The next part needed in the system is the camera module to attach the

lens. The Raspberry Pi High Quality Camera is compatible with both the 50mm lens and the Raspberry Pi already in the Capstone Component list. This camera allows live video to be captured using the lens and Raspberry Pi technology. The cost of the Raspberry Pi High Quality Camera is $50[5]. The next part was easy to select because OBD-2 connectors are fairly common and easily obtained because car mechanics use them frequently on cars. The OBD-2 connector has 16 pins to get the car information and power devices connected to it. The OBD-2 connector is around $20[6].

Mechanical Design and Schematics

Table 5. Part Information Table

Part	Product Name	Product ID	Cost
Camera Lens	16mm 10MP Telephoto Lens for Raspberry Pi HQ Camera- 10MP	4562	$50
Camera Module	Raspberry Pi HQ Camera Module	SEN-16760 ROHS	$50
Car Connector	16Pin OBD2 to USB Port Charger Adapter Cable Connector Diagnostic Tool	N/a	$20
Raspberry Pi 3	Raspberry Pi 3 Model A+	N/a	$0(Already in Component List)
Servo Motors	Servo Motor	N/a	$0 (Already in Component List)

The preliminary cost of the overall system design came out to $120 without shipping and processing costs.

Detail Design

In Figure 2, there is a diagram of the emergency red light detection system. The image is from the perspective of outside the car looking into the car. The car will be mounted looking out of the windshield towards the road with the HQ Camera Module behind the lens. Then, the camera module will be connected to the Raspberry Pi 3 board. Two servo motors will be on the sides of the system to adjust the angle of the camera. Away from the windshield will be the power source and OBD-2 connector cable. The OBD-2 port will be connected to the Raspberry Pi and to the car. The car connection is typically underneath the dash on the driver's side of the vehicle.

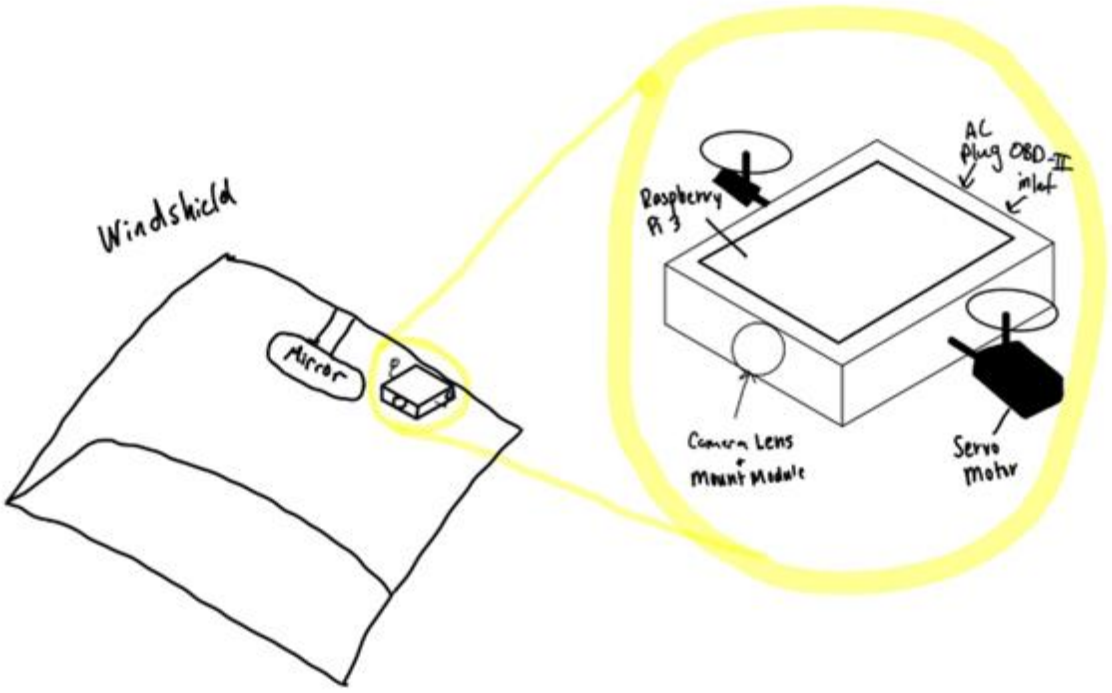


Figure 2. Full System Design Drawing

In figure 3, there is a diagram of the 16-pin connection cable used by the OBD-2 port. The pins are important to understand to make sure the proper information and power is connected to our red-light detection system.

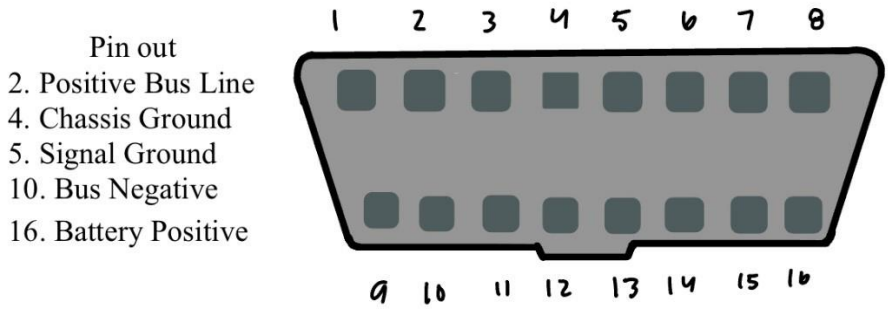


Figure 3. OBD-2 Port Pin Diagram

Electrical Design and Schematics

The red-light detection system consisting of all prefabricated parts limited the need for wiring diagrams and schematics. However, a block diagram was created to assist in understanding the operation process of the system. The block diagram is shown in Figure 4.

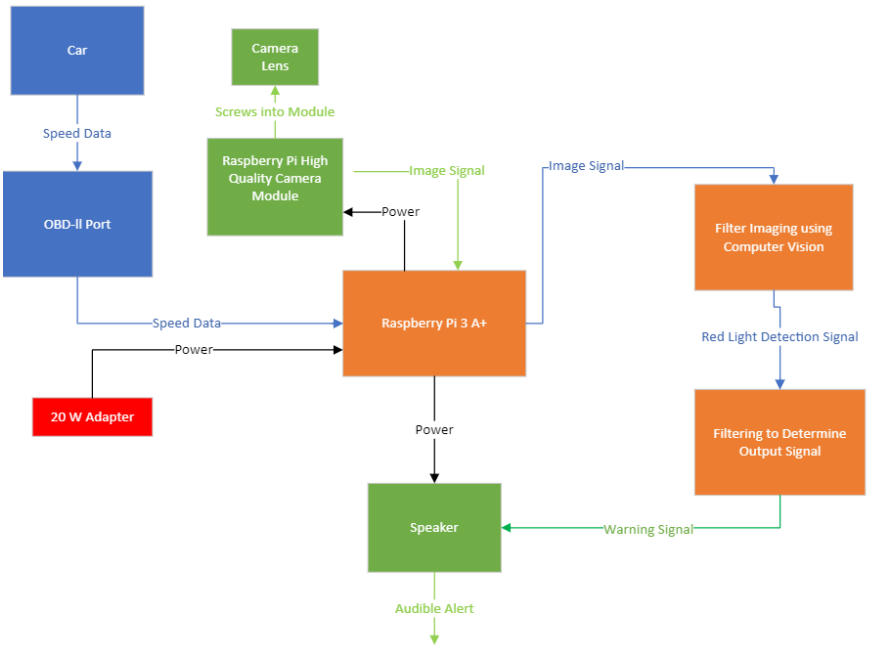


Figure 4. System Block Diagram

Chapter 4: Build

Assembly of device

The integration of our device build was relatively straightforward, this is because the pieces that we purchased are meant to interact and integrate with each other relatively simply. The camera that we purchased was specifically designed to use on a raspberry pi, making it a simple plug and play, this also applied to the lens that was used on the camera. Our OBD2 cable and speaker were USB devices, this meant that the hardware side was straightforward the whole way.

Figure 5. Camera and Raspberry Pi setup in the car

Figure 6. OBD-2 Reader set-up in the car

Integration of software

There were, however, some issues with integration despite this relatively simple setup. The speaker that we chose was a USB speaker that was powered by the raspberry pi. There is not a simple interface that allows for audio to be output from pi to the speaker, this led to some code needing to be written to output to the speaker. The code was relatively simple however it essentially required us to pre-load an audio file and play it back each time the alert was sent, alternatively to

just calling the file each time, which may have used some more processing power to keep the file ready to always use.

The original code that was 15riten on the laptop in the rapid prototyping process was utilizing a webcam as the camera, we wrote a large amount of the main code under this assumption that the code would be able to translate over to the raspberry pi. This was an incorrect assumption as the code for using a raspberry pi camera is quite different than a USB camera, this led to the software side of the team needing to rewrite a rather large amount of the code so that the new camera worked with the rest of the code.

The most pleasant surprise out of all was that the OBD2 code ended up being relatively easy to integrate, this was because the code ended up being less than 20 lines and the integration only required one line of code being changed to work with the rest of the program.

Changes made

All the major changes for this project were on the software side except for a couple things, the major change made on hardware side was the testing of multiple lenses, despite this we still did not find a lens that we were completely satisfied with. More camera and lens changes will be needed in the future. Other than that, all changes that were made were on the software side, this allowed us to keep costs down below the price range that we were operating in. On top of that, changes on software were able to be made instantly whereas a hardware change would take time to order or construct the piece or part that was needed.

Chapter 5: Test Plan and Results

Mechanical Tests

Purpose
The mechanical test plan will test our system to confirm that it will withstand the stresses of everyday driving while still functioning properly.

Scope
The scope of the mechanical test will examine how the system performs under certain stresses. These stresses include smooth driving conditions, potholes, emergency stop situations. The system should stay rigid on its mounts and not break free in any situation. The system should also remain relatively still and have a minimum vibration while driving to ensure a clear image is received by the camera.

Table 6. Requirement/Verification Cross- Reference Matrix

METHOD OF VERIFICATION	CLASSES OF VERIFICATION
N/A- NOT APPLICABLE	I - FIRST ARTICLE TEST
A - ANALYSIS	II - ENVIRONMENTAL QUALIFICATION
D – DEMONSTRATION	III - ACCEPTANCE TEST
I – INSPECTION	IV- NONE
T - TEST	

Table 7. Mechanical Verification Table

Section 3 Requirement	Verification Class				Section 4 Verification
	I	II	III	IV	
Alert Audibility	D		I		4.1.1
Camera Perception Distance	D				4.1.1
System Secureness	I		D		4.1.2 4.1.3

Tests on Red Light Camera
This section describes the scope of each test, overall procedure such as types of tests, main specification such as load, height, and dimension to be tested, and pass/fail criteria. The test procedures are available in Appendix A.

1.1 Basic Functionality Test
This would test the functionality of all major physical components. The main components would be the raspberry pi, the high-quality camera with lens, and the speaker.

1.2 Brake Test
This would test that the system would stay rigidly fastened in an instance where the driver slams on their brakes. This is a situation that might arise from the use of our system. This would test a

horizontal force on the system in an emergency test case and cover all braking situations. While a top end test would be an emergency brake at a speed of 55 mph, this would potentially be unsafe. Therefore, this test will consist of more theoretical calculations.

1.3 Speedbump Test

The system must survive a test case where the car contacts a pothole or a large speed bump. If the system comes dislodged from its fastenings, it could damage or even compromise the system. This test on the system would cover all instances of vertical movements that may be experienced in standard paved road driving.

Changes to Mechanical Tests

The mechanical tests were not changed since all the tests passed.

Electrical Tests

Purpose

The purpose of the electrical tests done on the Red Light Camera system is to ensure the electrical components operate as intended. There are multiple different components requiring power to operate, and they need safety operate without endangering the driver, car, or passengers.

Scope

The scope of the electrical test will examine how the systems electrical components operate in the system. The key limitation is the power supply of the system can only come from in the car. The system should be operational during a typical drive for the driver. The system should be able to safety alert drivers to a red light.

Table 8. Requirement/Verification Cross-Reference Matrix

METHOD OF VERIFICATION	CLASSES OF VERIFICATION
N/A- NOT APPLICABLE	I - FIRST ARTICLE TEST
A - ANALYSIS	II - ENVIRONMENTAL QUALIFICATION
D – DEMONSTRATION	III - ACCEPTANCE TEST
I – INSPECTION	IV- NONE
T - TEST	

Table 9. Verification Table Electrical

Section 3 Requirement	Verification Class				Section 4 Verification
	I	II	III	IV	
Alert Audibility	D		I		4.2.3
Car Electrical Interface					4.2.1

Tests on Red Light Camera Electrical
This section describes tests run on the electrical components of the Red Light Camera system. There are tests on the power supply, operating duration, and alert system. The test procedures are available in Appendix B.

2.1 Power Supply Test
The Raspberry Pi, camera module, speaker, and OBD-2 port need power in order to run code and perform their functions. If the system cannot supply power, none of the intended operations will work.

2.2 Operating Duration Test
The system needs to operate without overheating during the duration of the drive. The average time spent commuting to work is one hour. The system will be run for one hour while monitoring the temperature of the system in different locations of the alert system. The temperatures will be monitored to see if they exceed the operating temperatures of the component's data sheet.

2.3 Alert System Test
The Red Light Camera system's key alert mechanism is an audible sound coming from a speaker. The speaker needs to be able to play an alert that is 70 decibels for the driver to hear and not be a hazard to the driver.

Changes to Electrical Tests
The electrical tests were not changed since all the tests passed.

Software Tests
Purpose
The purpose of this test is to demonstrate the ability of the system to accurately detect red lights under a variety of driving conditions.

Scope
The Raspberry Pi will need to be able to operate under many driving conditions. Some examples of conditions which will need to be tested for are time of day, precipitation/environmental conditions, and presence of other lights in the line of vision.

Table 10. Requirement10/Verification Cross-Reference Matrix

METHOD OF VERIFICATION	CLASSES OF VERIFICATION
N/A- NOT APPLICABLE	I - FIRST ARTICLE TEST
A - ANALYSIS	II - ENVIRONMENTAL QUALIFICATION
D – DEMONSTRATION	III - ACCEPTANCE TEST
I – INSPECTION	IV- NONE
T - TEST	

Table 11 Verification11 Table Software

Section 3 Requirement	Verification Class				Section 4 Verification
	I	II	III	IV	
Processing Time	D	D			4.3.3
Alert Accuracy					4.3.1 4.3.3
Speed Operating Range	D		I		4.3.2 4.3.3

Tests on Driving Conditions

The following tests are intended to show the system's performance under several driving conditions. These tests will require a driver, and a passenger to monitor the performance of the system during the test. In addition, a working prototype will be needed before these tests can be conducted. The setup for each test will require the raspberry pi to be connected to a power source in the car through a micro-USB cable. It will also require that the camera is placed in a stable position on the dashboard. Finally, it will require that the raspberry pi is connected to the OBD2 port with an OBD2-USB cable, and that the program for the system (coded in python) is bug-free and functional when the system is running.

3.1 Time of Day/Sky Color Test

This test will determine the system's functionality at different times of the day. This is necessary because at different times, the sky will be different colors, sometimes which may affect the visibility of the red light in its surroundings. The system will be tested during the morning, around noon, in the afternoon and at night. This should hopefully account for the different colors of the sky at these times. The test is passed if the system works at any time of the day, under any sky conditions. It fails if it cannot detect red lights in these varying conditions.

3.2 Traffic Test

Test for when behind other cars with taillights. These red taillights will likely cause the device to believe that it is viewing a traffic light and has a high likelihood of false positives.

3.3 Speed Beep Test

This test will determine whether the system makes noise when it is not moving. Ideally, the system will only alert the driver to a red light if the car is in motion. If the system makes noise when the car is not moving, such as when stopped in front of a red light, it would be annoying to listen to, which is why this test is necessary. Information on the motion of the car will be available to the system through the OBD2 port input, which streams the speed of the car to the processor to make decision on whether the car is in motion. The test is passed if the system only makes noise if a red light is detected, and the car is in motion. It fails if it makes noise when the car is motionless.

Changes to Software Tests

The software tests were not changed based on the results. The GRIP filter and code were updated to fix errors and problems with different tests. The errors came from differences in the light on any given day or the weather conditions.

Chapter 6: Redesign and Lesson Learned

Hardware Design Changes

There were many changes made to our device from the beginning of the semester to the end, initially we had plans for a servo motor to move the camera at different speeds however with further consideration and tests we decided that a static camera located higher up on the windshield would have all the benefits of a swiveling camera without the added complexity of a system that could move and aim at different heights. The team also decided that this additional level of complexity would likely lead to more failures, which we wanted to avoid as this was meant to be a safety improvement device. The new parts list has been updated from the original in chapter 3 and listed below.

Table 1212. Updated Part Information Table

Part	Product Name	Cost
Camera Lens	16mm 10MP Telephoto Lens for Raspberry Pi HQ Camera- 10MP	$50
Camera Module	Raspberry Pi HQ Camera Module	$50
Car Connector	16Pin OBD2 to USB Port Charger Adapter Cable Connector Diagnostic Tool	$21.00
Raspberry Pi 3	Raspberry Pi 3 Model A+	$0(Already in Component List)
Power Inverter	BESTEK 300W Power Inverter DC 12V to 110V AC Car Inverter with 4.2A Dual USB Car Adapter	$29.99

Software Design Changes

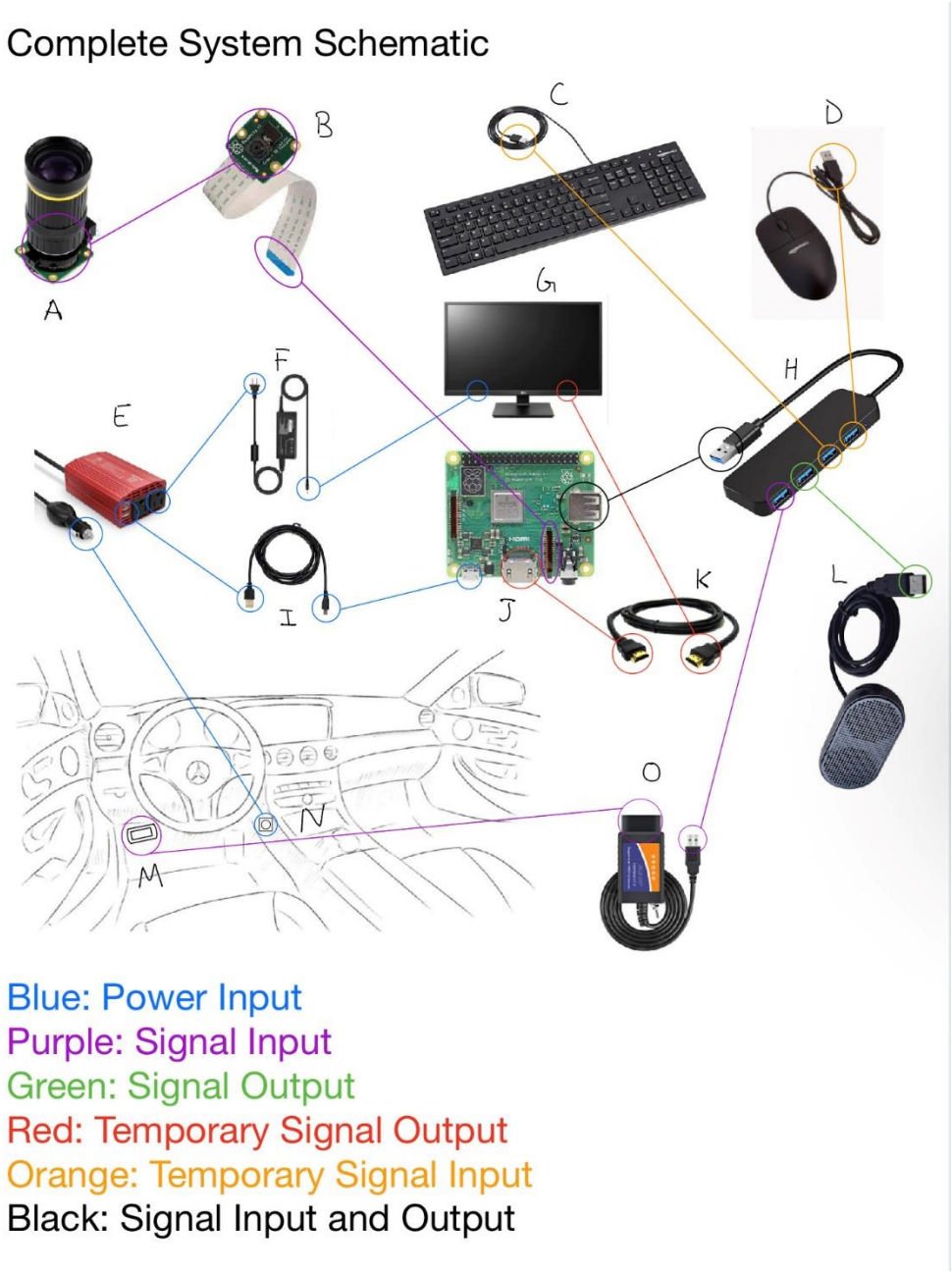
Originally the code that was written worked on a laptop with a Logitech webcam, however we needed to make the code work on a raspberry pi which took a decent number of changes to make it work. There were many iterations of code that were written, our very first piece of code simply opened a prerecorded video that was saved in the file system of the computer, this showed that we could access video and allowed us to work on viewing objects in the videos. Next, we wrote code that opened a window that read from the camera that was plugged into the laptop, this was proof of concept that we could access the camera and read information from it. Following these two breakthroughs we integrated the Graphically Represented Image Processing engine (GRIP) filter that we had been developing into the live feed from the webcam, this allowed us to test our filter on different lights in the classroom environment so we could rapidly edit the filter in a controlled environment. This previous part of the project was not cut and dry, it required many changes over the course of the entire semester and could be improved upon even further in the future. While we were working on the GRIP filter, we continued implementing more features in the code, next was the audio alert system, for this application we simply alerted the driver at a constant volume and repetition. However, we have the capabilities to alter the volume and repetition if it was determined that it would be beneficial. In the meantime, we decided that it could be viewed as confusing or

distracting for the driver if the device was constantly changing its output. The final individual feature added to the device was the ability to read information from the car, this allowed us to know the speed that the car was travelling at which allowed us to determine if we needed to alert the driver or not, obviously if the car was stopped then they don't need to be constantly warned. As each of these individual pieces were being completed, we integrated into the main code. The final code ended up being around 70 lines for the main and the GRIP filter ended up being 131 lines of code on a different python file.

Chapter 7: User Manual

Components and Setup

Pictures are worth a thousand words, so rather than try to explain the fairly complicated schematic involved in connecting all the system components, a visualization is provided below which illustrates all the connections and parts used.



Below is a list of the components illustrated above:

A. Raspberry Pi Camera Lens

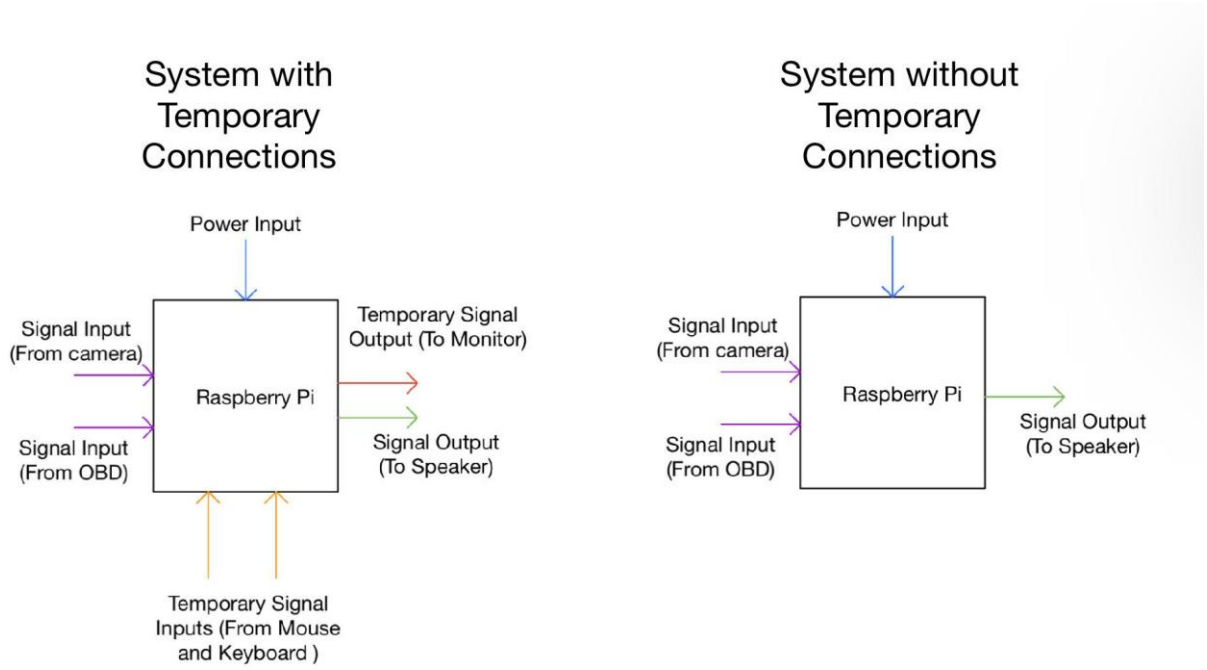
B. Raspberry Pi Camera
C. USB Keyboard
D. USB Mouse
E. Power Inverter
F. Power Adapter/Cable
G. Monitor
H. USB Extender
I. Micro USB Cable
J. Raspberry Pi
K. HDMI Cord
L. Audio Speaker
M. OBD-II Port
N. Car Auxillary Power Outlet
O. OBD-II to USB Adapter/Cable

The complete system schematic shows all the labeled components, and their connections. The connections are color coded to show the different types of connections. Connections drawn in blue represent the transfer of power from the automobile car power outlet (cigarette lighter) to the inverter, which supplies power to both the monitor and the raspberry pi. The connections in purple represent the two input signals involved in the system. The first originates from the OBD-II port underneath the steering wheel (which is the output of the OBD car computer), then passes to the OBD-II to USB adapter, which is then fed into the peripheral ports connected to the Raspberry Pi. The second input signal comes from the camera. It is fed directly into the Raspberry pi through a special camera port designed specifically for Raspberry Pi cameras. There is also one connection drawn in green. It is the speaker which outputs a noise when a red light is detected.

As for the connections in red and orange, they are temporary connections, meaning that they are not intended to be a part of the final project. These connections are used for programmers to see what is happening inside the Raspberry Pi, and to edit the code so that the software functions properly. The intention of this project is to create a system which can operate as soon as the car turns on. In other words, the program should run as soon as the Raspberry Pi is powered on. We were unable to get to this stage in our design and left it to the group who inherits this project next year. Once this functionality is accomplished, the temporary inputs (keyboard and mouse) and output (monitor) can be removed, but this should only be done once the Raspberry Pi is able to run on boot because otherwise there will be no way for programmers to edit the code.

There is also one connection drawn in black. Because we used a Raspberry Pi model 3 A+, there was only one USB port attached directly to the board. For this reason, we used a USB extender (item H) to give us more ports to work with. The extender allows for signals to pass through it into the Raspberry Pi (OBD-II, keyboard, and mouse), and also allows for signals to be output as well (audio speaker). The connection is drawn in black because it is both an import and output connection. If in the future a different model of Raspberry Pi is used (for example, model 3 B+), then the extender may not be necessary because the board may have enough USB connections directly connected to it. The decision to make this switch is left up to the group who inherits this project next year.

The schematic below can be simplified if the focus is just on the inputs and outputs to the Raspberry Pi. The image below shows a high-level overview of the system with (left) and without (right) the temporary connections.



Operation

To get the system to run, temporary connections must be present. As mentioned in the previous section, the eventual goal is to get the system to run on startup, but it has not yet been completed to this point. To get it to run, the programmer must log into the Raspberry Pi, navigate to the terminal, and run the program from there. Alternatively, it may also be run from inside a code editor, such as Geany or Thony, which are both pre-installed on Raspberry Pi's.

Troubleshooting

It should be noted that when we developed this project, we ran into problems at first with getting the program to run because a lot of the Python libraries that were used in our code are not pre-installed on the Raspberry Pi. If this code is copied onto another Raspberry Pi, it is likely that it will not run on the first attempt because these libraries are not present. To fix this problem, the group that inherits this project needs to figure out which libraries are not installed on their Raspberry Pi, and then install them.

Other problems which may arise include computer freezing when the program is running. The processor on the Raspberry Pi is relatively small, so it may sometimes struggle to run large programs. This should be kept in mind if more code is added. Our system was able to run fine for the most part, but if more code is added, freezing could become a bigger issue. If the processor does freeze, try rebooting the system and running the program again.

There are several other problems which may arise as well, which are too numerous to list in this report. However, several online examples helped us figure out how to troubleshoot, and the internet is a great place to look for answers to technical problems.

Chapter 8: Project Management

Project Management

Ben Moskos was assigned to the team lead at the beginning of this semester. He did a lot of the groundwork when it came to managing the administrative parts of the project, including part ordering, file system management and weekly reminders for upcoming assignments and deadlines. We coordinated schedules and assignments by using iMessage group chat and OneDrive respectively.

Schedule

The schedule for the project was updated weekly to keep the team on track to complete it by the end of the semester. We decided to use a Gannt chart to keep track of when each task needs to be completed. See Appendix A for this. Our weekly schedule consisted of meeting in class from 12:45pm-2:05pm twice a week with occasional absences for other conflicts, Mathias would also sometimes work on the project from home on large programming days. We would have at least one meeting a week on Tuesday or Thursday from 11am-12:40pm, this day would change based on the schedules of all members, on some weeks we would meet at that time on both days in the week when the class was particularly busy. In addition to this Liam and Mathias met multiple times virtually on weekends to work on code. The overall weekly schedule is also shown in appendix A2.

Project Risks, Problems, and Mitigation Strategies

Our project may be affected by a few factors which could delay workflow, affect the design of the device, or impact the final product in other ways. The major problems that we could run into include the following:

1. Physical Limitations: The device could run into problems with how it interacts with its physical surroundings. That is, the sensors available to us might not be able to accurately detect the electromagnetic waves / photons coming from the traffic lights. It could be that the signal reaching the device is just not powerful enough to induce a reaction that is discernable from other light sources in the vicinity, such as brake lights from nearby cars. It may also be that the sensors are not angled in the right direction to be able to detect the EM waves/photons, even if the power from the signal is strong enough to reach the sensors. The device would need to be designed to constantly adjust its orientation relative to the red light, or it may need to be designed so that it can receive a signal from any direction (an omnidirectional receiver). In addition to the major challenges which will come from being able to physically detect the signal, that is the pulse coming from the red light, there will be challenges in designing the device to be able to operate in a multitude of conditions, including rain, snow, fog, foliage, trucks and other cars and objects which may block the line of view. A possible way to mitigate this is if the device and the traffic light were fitted with omnidirectional radio-frequency transceivers, as radio waves can travel much farther without attenuating than visible light waves, which is why they are the main frequency used for communication between devices. The signal could also be clearly discerned from surrounding sources of EM radiation if the traffic light is designed to emit a pulse that is uniquely identifiable as being from the traffic light as well. This is a design we have considered for its physical advantages as an alternative to the object-detection, machine learning system, which places more of the processing on the software side, which is explained below.

2. Software Failures:

The alternative to handling signal processing with the physical system described above is to do the signal processing on the software side, using machine learning and object-recognition software. The advantage of this system is that no transmitter needs to be fitted onto the traffic light for communicating the signal. This may mitigate the expenses of purchasing additional hardware and the complications which come from installing antennae and transceivers on traffic lights. The disadvantage is that it will rely on visible light sensors to detect the signal from the red light, and the challenges of doing so were described in the physical limitations section above. The other challenge is that the software involved is rather complicated, and detecting the right signal is not as straightforward as it was in the other design. Instead of having a uniquely identifiable pulse from the traffic light to process, the system will instead need to be able to discern the traffic light from all other objects in its line of sight. This will require some sophisticated machine-learning and object recognition software to be installed on the device. One final note is that the problem of orienting the sensors to face the red-light will still be an issue, as it was for the antennae in the other design. This can be solved by using omnidirectional photoreceptors on the device.

The magnitude of the problems identified on both the software and physical sides will become more apparent as we work through the details of our design. Depending on which presents a bigger challenge to the functionality of our project, we may have to revise our design to meet the main requirements of the red-light detection system.

3. Financial Limitations:

Another obstacle to our project is the financial feasibility of our design. If our design is only technically possible, then it is only theoretically possible. We need to create a design that is cost-effective. Expensive hardware is not an option. Neither is software requiring a license. Ideally, we can purchase all hardware components for under $100 and utilize open source and free software for this project. Depending on what is possible, and what resources are available, we may have to modify our design to be financially reasonable.

4. Time Constraints:

As this is a capstone project, not a project for industry or profit, we only have one semester to design the system. There is only so much which we can realistically do in this time given that we all have other classes, schedules, and busy lives. This means that we need to be realistic in designing the system; we should not get too focused on unimportant details or try to make it perfect. Realistically, it won't be up to industry standards, so we shouldn't try to achieve this unachievable goal. A simple, practical design that is acceptable for a capstone course is all we should try to achieve for this project.

5. Technical Abilities of the Team:

Although it is good to aim high, we need to be realistic about our abilities in this project. We should be focused on creating a project that is within the technical abilities of the team. We have chosen some advanced topics for our capstone project. While it is possible to learn how to work with the necessary technologies 'on-the-fly', unless some of us have some experience in our previous courses, it would be wise not to stretch too far beyond what we already know. Our final design will, in the end, be a result of the knowledge we can apply to a project. The features and

design of the device should reflect what we know how to do, otherwise we will not be able to make good progress on the project and complete it by the end of the semester.

Problems and What to do Differently

There were many issues encountered this semester on the software and hardware side, we had many snags on the software side with buggy and non-functional code. Mathias and Liam spent a large amount of time debugging this code and fixing major problems to make a working program. Usually, these problems were solved by creating the feature in its own standalone program to ensure that it would work first then after looking to integrate into the main code. For most of the semester Liam was writing a lot of the original code leaving Mathias in charge of most of the integration, when more issues arose Mathias and Liam would collaborate to ensure the problem was resolved.

On the hardware front we also had many issues, some with ordering and retrieval of parts and others with the limitations of the hardware that we did receive. The camera and lenses that we used for this project were likely not the correct choice because the lack of auto focus on the lenses was a difficult challenge for us to overcome and the camera did not quite have a high enough resolution to use at the range that we had planned on using it at. This led to some changes being made to ensure our project still functioned at least partially as intended.

As a group the main changes that we would make would be to pick a microcontroller that is more powerful and a camera/lens combination that creates clearer images to make this device more effective. Overall, we have concluded that the code that was created operates exactly as we expected. However, it may be beneficial to create some sort of calibration process that allows the values of the filter to be changed automatically on startup. We had to make many edits to the filter, and it was discovered that different lighting conditions changed the effectiveness of the device.

Resources outside of Capstone

With this project being in a vehicle and with a large amount of code being written there were multiple resources that were utilized outside of the classroom. Firstly, Mark Fong gave us access to an old computer vision project that he worked on, this allowed us to see how the GRIP filter was integrated with code to work properly. Some reverse engineering on behalf of Liam and Mathias allowed us to get this working in a very rough capacity early in the semester. On top of this we used many online resources to get ideas and inspiration on writing code for our device. Due to the very niche concept of our project none of this code worked right off the bat and all of it required many changes but often the code allowed us to debug rapidly.

In addition to this, the largest resource outside of capstone was the operation and use of Mathias's vehicle. The OBD2 reader and code worked with this car so when field testing was being completed, we had to ride in Mathias's car. Some other work could be completed in the lab but anything to do with vehicle speed and real-world testing was completed in this car.

Appendix A

Figure A1. Team Gantt Chart

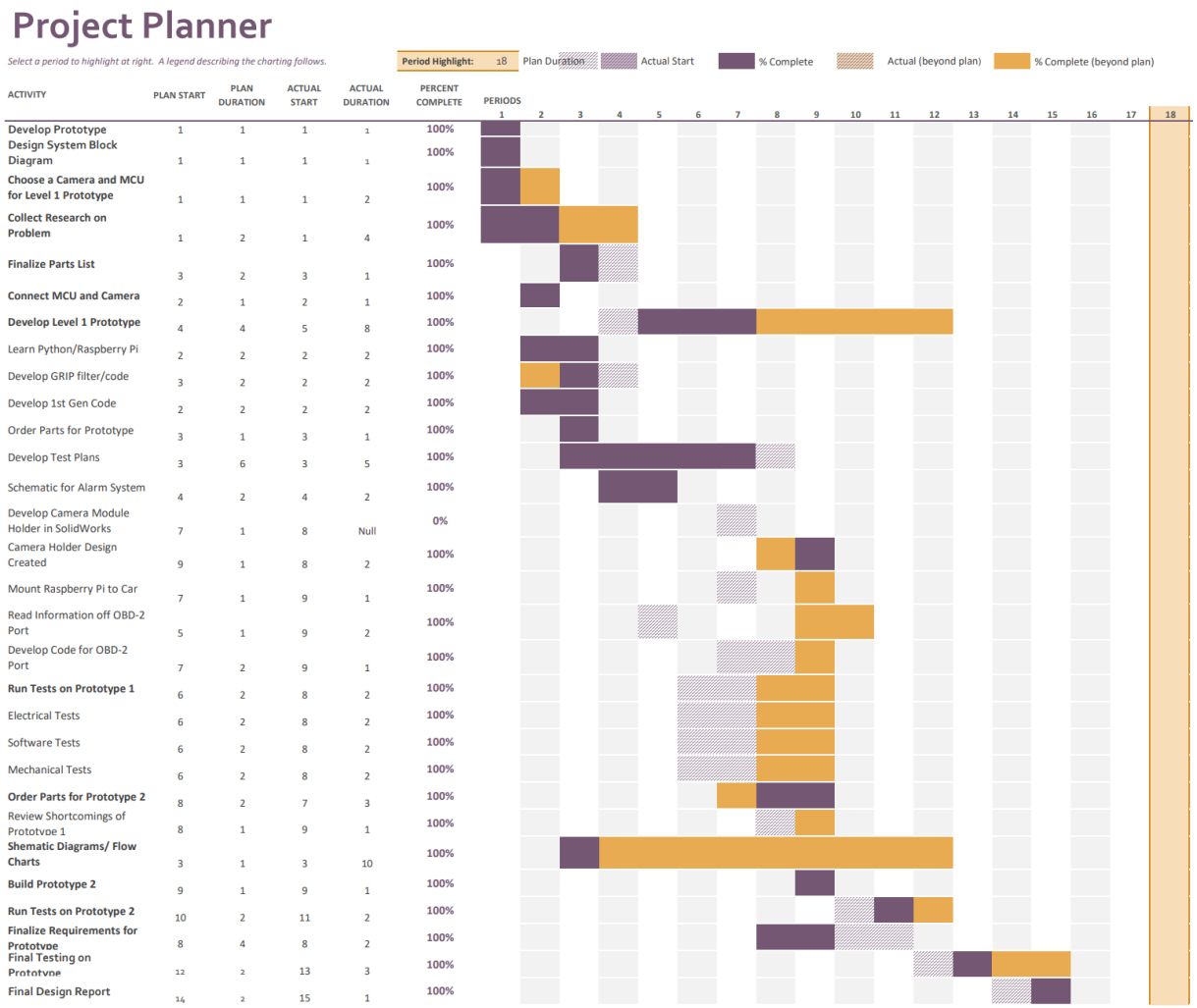


Figure A2. Team Schedule

Week #	Member Name	Task Description	hrs	Task Finished?
1	All Members	Create Time Sheet	1	Yes
1	All Members	Create Gantt Chart	1	Yes
1	All Members	Create Team Charter	1	Yes
1	All Members	Create Block Diagram	1	Yes
2	All Members	Define Test Conditions and plan	2	Yes
2	All Members	Each member creates and presents prototype ideas	2 (each)	Yes
2	All Members	Team Meeting-Present Prototypes, 3905-To-Do List	2	Yes
2	Megan	Research how to connect Raspberry Pi to computer, and python	1	Yes
2	Ben	Research Camera and MCU for prototype	4	Yes
3	Liam	Working on reading information from webcam to computer	4	Yes
3	Liam, Ben, Megan	Team Meeting-discuss progress, finalizing parts to order, and connect webcam to software	2	Yes
4	Liam	Develop first generation code for reading red light	12	Yes
5	Ben/Megan	Have the physical prototype wired together (MCU, Camera)	10	Yes
5	Megan	PCB Master Class- Learning skills to needed for alert system in car	3	Yes
5	All Members	Team Meeting Discussing Progress and Writing Test Plans for Prototype 1	3	Yes
5	Liam, Mathias	Develop a method of audio/visual output	10	Yes
5	All Members	Have a working red light reading prototype integrated	5	Yes
6	Liam	Read information from OBD2 port	8	Yes
6	All Members	Test first prototype with actual pictures taken of red lights	2	Yes
7	Ben	Set up HQ camera with lens	2	Yes
8	Mathias	Integrate Pi Camera with Grip image processing	4	Yes
12	Liam	Develop code to automatically power on system	3	No
13	All Members	Perform more tests on the system	4	Yes
15	All Members	Complete final paper + presentation	20	No

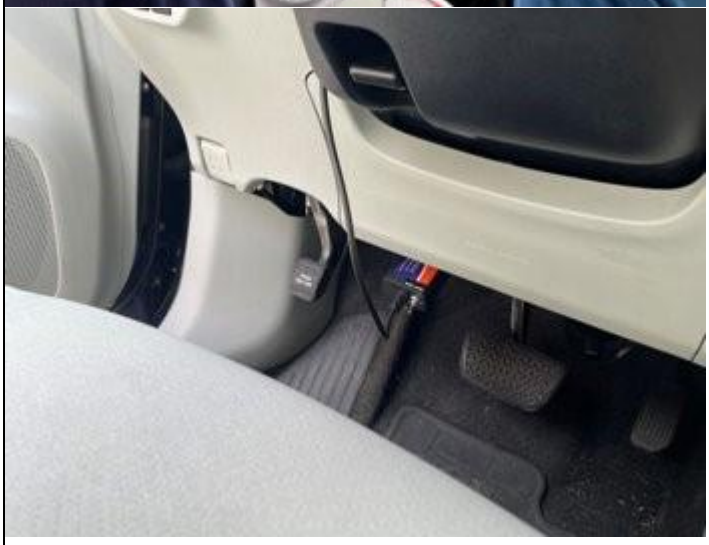
Appendix B

Test Sheet 1 Basic Functionality Test

1.1 Basic Functionality Test	
Method of Verification: Inspection	
Type of Test: First Article	
Date(s)/ Total Times: **3/9/23 12:00 PM-2:00 PM**	**Location: Dreese Capstone Lab**
Test Equipment: Speaker, Raspberry Pi, Camera and Lens	
Test Engineer: Ben Moskos **Witness: Megan Murphy**	
Criteria for Success: • **Passes – All systems operate correctly. The camera and lens display a clear image, the speaker produces a sound, and the Raspberry Pi powers and interfaces with the camera.** • **Fails – One or more components do not function properly like described in the passing criteria.**	
Test Procedure: 1. Power the Raspberry Pi with the micro-USB. 2. Attach the telescopic lens to the Raspberry Pi High Quality camera. 3. Open the Raspberry Pi terminal and enter raspistill –t 0 to view an image through the camera. 4. Plug the speaker into the USB-A port of Raspberry Pi and play the downloaded beep sound.	

Test Results:

Test	Powered On?	Did Software Boot?	Did the image display?	Did the speaker produce a sound?
Raspberry Pi	Yes	Yes		
Camera and Lens	Yes		Yes	
Speaker	Yes			Yes



Action Items: **Passed test**	
Test Engineer: Ben Moskos	Witness: Megan Murphy

Test Sheet 2 Brake Test

1.2 Brake Test	
Method of Verification: Inspection	
Type of Test: First Article	
Date(s)/ Total Times: 3/9/23 12:00 PM-2:00 PM	**Location: Dreese Capstone Lab**
Test Equipment: Speaker, Raspberry Pi, Camera and Lens	

Test Engineer: Ben Moskos
Witness: Megan Murphy

Criteria for Success:

- **Passes – System stays secure and does not become a hazard to the driver**
- **Fails – System moves from holder**

Test Procedure:

1. Start system up
2. Start driving and reach 15 mph
3. Come to a stop with the deceleration of approaching a stop sign
4. Once stopped record if the system stayed in place
5. Repeat steps 2-4 for 35 mph and 25 mph
6. For the 25 mph test, the braking should be an aburt stop like the driver forgot to brake for a stop sign

Test Results:

Test	System remains rigidly secure?	System has become loose?	System has failed?	Speed of test?
Low Speed Braking	X			15 mph
Standard Braking	X			35 mph
Slamming on Brakes	X			25 mph

Action Items:
Passed test

Test Engineer: Ben Moskos	Witness:Megan Murphy

Test Sheet 3 Speed Bump Test

1.3 Speed Bump Test

Method of Verification:

Inspection	
Type of Test: Human Acceptance and Endurance	
Date(s)/ Total Times: 4/9/23 12:00-1:00 PM	**Location: St. John Parking Lot**
Test Equipment: Car, System, System Fastenings	
Test Engineer: Ben Moskos **Witness: Megan Murphy**	

Criteria for Success:

- **Passes – System stays secure while the car contacts speedbump or pothole**
- **Fails – System becomes unsecure and begins to move freely within the car after contact with speedbump or pothole**

Test Procedure:

1. Begin by placing all systems in their desired locations. Make sure all Velcro, suctions, and connections are made securely.
2. Ensure that the system is operating correctly based on the system component's locations.
3. Begin by testing standard driving conditions. This would include turning and driving on a finished surface
4. Next, in a controlled environment like an empty parking lot containing speed bumps, test hitting a speed bump at speeds under 10 mph.
5. Assume for higher speeds that speedbumps would be a very rare occurrence at high speeds.

Test Results:

Test	System remains rigidly secure?	System has become loose?	System has failed?	Speed of test?
Standard Driving	X			25 mph
Low Speed Speedbump	X			10 mph

Action Items: Passed

Test Engineer: Ben Moskos	Witness: Megan Murphy

Test Sheet 4 Power Supply Test

2.1 Power Supply Test
Method of Verification: Demonstration
Type of Test:

First Article and Acceptance

Date(s)/ Total Times: 3/28/23 12:30-2:00 PM	**Location: Dreese Capstone Lab**

Test Equipment: OBD-2, Red Light Camera Module, Speaker

Test Engineer: Megan Murphy
Witness: Liam Kearns

Criteria for Success:

- **Passes – All system components are powered on**
- **Fails – System fails to power all components**

Test Procedure:

1. Connect the Red Light Camera module and OBD-2 port to the car.
2. Check each component of the system to make sure the device is powered on.
3. Record if any error warning lights or errors appear on the devices

Test Results:
X means yes, and blank is no or N/a

Component	Powered on	Errors/Warnings
Raspberry Pi	X	
Camera Module	X	
OBD-2	X	
Speaker	X	

Action Items: Passed Test

Test Engineer: Megan Murphy	Witness: Liam Kearns

2.2 Operation Duration Test

Method of Verification:
Analysis

Type of Test:
Acceptance and Endurance

Date(s)/ Total Times: 3/30/23	**Location: St. John Parking Lot and Streets of Columbus**

Test Equipment: Car, Red Light Camera Module, Timer, Thermometer

Test Engineer: Megan Murphy
Witness: Ben Moskos

Criteria for Success:

- **Passes – The system operates for 1 hour without becoming a heat hazard**
- **Fails – The system stops working or overheats**

Test Procedure:

1. Connect the Red Light Camera Module to the car. Have a passenger and driver in the car.
2. The passenger is responsible for monitoring the temperatures of the components at the different time intervals.
3. Power on all devices and have the passenger make sure none of the devices are hot to the touch.
4. The driver starts the car and start driving. When the car starts driving, the passenger will start the timer.
5. Continue driving and record if the temperature of a device changes(normal, warm, and hot) of the components every 15 minutes for 60 minutes.

Test Results:
Normal is normal device temperature
Warm is elevation is temperature since turning the device on
Hot is the device had a noticeable difference in temperature

	Temperature (degrees F)			
Time(minutes)	Raspberry PI	Camera Lens	Camera Module	OBD-2 Port
0	Normal	Normal	Normal	Normal
15	Normal	Normal	Normal	Normal
30	Normal	Normal	Normal	Normal
45	Warm	Normal	Warm	Normal
60	Warm	Normal	Warm	Normal

Action Items: **Passed Test. In the future, repeat test with temperatures.**

Test Engineer: Megan Murphy	Witness: Ben Moskos

Test Sheet 6 Alert System Test

2.3 Alert System Test

Method of Verification:
Demonstration

Type of Test:
First Article and Acceptance

Date(s)/ Total Times: **3/28/23** **12:30-2:00 PM**	**Location: Dreese Capstone Lab**

Test Equipment: OBD-2, Red Light Camera Module, Speaker, Decibel Reading Device

Test Engineer: Megan Murphy
Witness: Liam Kearns

Criteria for Success:

- **Passes – The speaker plays sound above 70 dB with error ±10**
- **Fails – Sound is below 60 or above 80 dB**

Test Procedure:

1. Power on the camera system.
2. Have the system set to a default alert that will play a sound without the feedback system of a red light.
3. Record the decibels of the sound for 10 trials.

Test Results:

Trials	1	2	3	4	5	6	7	8	9	10
Decibel	68.1	72.5	69.6	66.9	64.0	68.4	70.6	71.1	63.9	60.9

Mean decibel is 67.63 dB

 Screenshot of decibel reader	
Action Items: **Passed Test**	
Test Engineer: Megan Murphy	Witness: Liam Kearns

Test Sheet 7 Time of Day Testing

3.1 Time of Day Testing	
Method of Verification: Test	
Type of Test: Acceptance and Environmental	
Date(s)/ Total Times: 4/4/23	**Location: St. Johns arena parking lot/traffic light**
Test Equipment: Full Detection system, Car, low traffic area with traffic light	
Test Engineer: Liam Kearns **Witness: Mathias Dawit**	
Criteria for Success: • **Passes – Device can detect traffic light in no light, low light, and full light situations in less than 1.5 seconds** • **Fails – Device fails to detect traffic light in all situations or detects slower than 1.5 seconds**	
Test Procedure: 1. Set-up device in vehicle outside of St. Johns arena 2. Ensure all operations are functioning as anticipated 3. Drive towards the traffic light as it is red 4. Note whether the alert sounds as the vehicle approaches the light 5. Repeat steps 3 and 4 for the appropriate number of times for the test 6. Repeat steps 3-5 at the other 2 lighting conditions	

Test Results: (Time in Seconds)

Lighting conditions	Detects correctly?	Detects with additional noise?	Doesn't detect?
No Light (Night)	X		
Low Light (Morning/Evening)	X (0.3, 0.2, 0.6)		
Full Light (Noon)		X (1.1,0.7,1.7)	

Test Engineers will approach the light multiple times at the three listed lighting conditions and mark how many times each outcome takes place.

Action Items:
When testing in full lighting the camera would sometimes see other red items and send false alarms, this is what is meant by additional noise.

Test Engineer: Liam Kearns	Witness: Mathias Dawit

Test Sheet 8 Traffic Testing

3.2 Traffic Testing

Method of Verification:
Test

Type of Test:
Acceptance and Environmental

Date(s)/ Total Times: 4/6/23	**Location: St. Johns arena parking lot/traffic light**

Test Equipment: Full Detection system, Car, low traffic area with traffic light

Test Engineer: Liam Kearns
Witness: Mathias Dawit

Criteria for Success:

- **Passes – The device still detects red light despite other vehicles in front in less than 1.5 seconds of entering camera FOV**
- **Fails – The device sends warning slower than 1.5 seconds**

Test Procedure:

1. Power on the system and orient the camera towards the windshield.
2. Begin driving on the road around St. Johns arena, while other vehicles are present.
3. Drive at various distances from vehicles
4. Continue driving until there are 3 examples of red light approach with zero vehicles in front.
5. Repeat steps 3 and 4 with one vehicle in front while approaching red light
6. Repeat steps 3 and 4 with more than one vehicle in front while approaching red light

Test Results: (Time in Seconds)

Lighting conditions	Detects correctly?	Detects with additional noise?	Doesn't detect?
No Vehicles in front	X		
One vehicle in front		X (0.7, 0.4, 0.3)	
Many Vehicles in front		X (0.4, 0.6, 0.9)	

Action Items:

When there were vehicles in front the camera detected the brake lights of these vehicles if they were in the FOV of the camera, the camera's view was changed to limit this however tall vehicles were still detected.

Test Engineer:	Witness:
Liam Kearns	Mathias Dawit

3.3 Speed Beep Testing

Method of Verification:
Test

Type of Test:
Acceptance and Environmental

Date(s)/ Total Times: 4/6/23	**Location: St. Johns arena parking lot/traffic light, 315 Highway**

Test Equipment: Full Detection system, Car, low traffic area with traffic light, high speed area

Test Engineer: Liam Kearns
Witness: Mathias Dawit

Criteria for Success:

- **Passes – The system does not send alerts while below or above the speed threshold set by the team**
- **Fails – The system continues to send alerts despite being outside of the threshold that has been set**

Test Procedure:

1. The code device will be hard coded to alert the system that it is seeing a red light
2. We will ensure the code is set to disable these alerts when outside of the certain speed values
3. Drive within the speed threshold (5-60 mph) and ensure that the alert is being triggered
4. After alerts are going off stop the car to simulate stopping at a traffic light
5. Ensure that the alerts stop sending
6. Repeat steps 3-5 however this time start within the threshold and accelerate out of the threshold (60 mph) to ensure alerts stop sending
7. Repeat steps 3-5 however this time start within the threshold and decelerate out of the threshold (5mph) to ensure alerts stop sending

Test Results: (Last speed before turning off mph)

Speed(mph) Condition	Alerts properly	Doesn't alert
5-60	X	
> 60		X (59.43, 60.12, 61.33)
<		X (1.68, 1.82, 2.02)

Action Items:
The speed listed was the last speed shown on the velocity output before the alerts were cut off.

Test Engineer:	Witness:

Liam Kearns	Mathias Dawit

Appendix C

In this section, we've included a link to the GitHub repository where all the code for this project is stored. It can be downloaded from https://github.com/mathiasdawit/capstone4905repo

Works Cited

[1] "Inside NIOSH: Higher Injury and Death Rates Found in Motor Vehicle Towing Industry," *Centers for Disease Control and Prevention*, 20-Feb-2019. [Online]. Available: https://www.cdc.gov/niosh/research-rounds/resroundsv4n8.html#:~:text=Deaths%3A%20CFOI%20data%20from%202011,all%20U.S.%20private%20industries%20combined. [Accessed: 01-Dec-2022].

[2] "Red Light Running," *IIHS*, Jul-2022. [Online]. Available: https://www.iihs.org/topics/red-light-running. [Accessed: 01-Dec-2022].

[3] "Driver Care - Know Your Stopping Distance." *Automotive Fleet*, www.automotive-fleet.com/driver-care/239402/driver-care-know-your-stopping-distance.

[4] *UCTronics* https://www.uctronics.com/arducam-8-50mm-c-mount-zoom-lens-for-imx477-raspberry-pi-hq-camera-with-c-cs-adapter.html

[5] *Adafruit* https://www.adafruit.com/product/4561

[6] Amazon https://www.amazon.com/dp/B07MQ8GHG3?ref_=cm_sw_r_cp_ud_dp_BAPP8215EG8KDFZ7TMTB