Project Report
for
"Self-Driving RC Car"

Jacob Buchanan	Team Leader
Abdulaziz Alghafli	Member
Zhengkun Ye	Member

Electrical Engineering and Computer Science Department
University of Toledo

Submitted to:

Course Instructor
Dr. M. Niamat
Electrical Engineering and Computer Science, University of Toledo

Project Advisor
Dr. A. Javaid
Electrical Engineering and Computer Science, University of Toledo

Submitted as part of requirements for the course
EECS 4020 Senior Design Project II

EXECUTIVE SUMMARY

This project proposes a design strategy and prototype of a self driving model car that will operate with minimal human intervention. The primary intention of this project is to create a vehicle that would reduce the number of accidents related to distracted driving. This is being achieved by adding several autonomous advancements and features to the car. Accidents caused by unintended lane departure, distracted lane changing, forward collisions, rear collisions, merging ramp collisions and driving negligence may be avoided through the implementation of network of various sensors and a signal processing system. To mitigate lane departure related accidents a car-to-lane-marker positioning system will be implemented, essentially this system functions by monitoring the vehicle's location with respect to a lane boundary marker and provides appropriate feedback/control signals to the steering system. To address forward, rear and merging ramp collisions a proximity sensor system will be implemented; this system functions by monitoring the distance between the vehicle and another object in four areas, front, rear, left and right sides. If an object comes to close in proximity to the vehicle the processing unit reacts by slowing down, speeding up, stopping or changing lanes. To address a portion of driving negligence and distracted driving a video system is being implemented that scans the roadway for posted traffic regulations. This system functions by processing the posted information and setting system parameters or providing a control signal; i.e., reading a speed limit sign and maintaining the posted highway speed, reading a stop sign and stopping when appropriate, reading a lane ending sign and merging into an appropriate lane. The project goal is to deliver a solution capable of reducing the number of accidents caused by distracted driving.

ACKNOWLEDGEMENTS

We would like to take this opportunity to express our gratitude to our faculty advisor Dr. Ahmad Javaid for their guidance throughout this course of this project. We would also like to express our thanks to our professor Dr. Mohammad Niamat for encouraging a high level of professionalism and for encouraging our team to constantly improve upon our performance, ultimately preparing us for our future endeavors.

The design team consists of five members focused on the design, development and prototyping of a Self Driving RC Car. The team members are; Jacob Buchanan an electrical engineering associate, Abdulaziz Alghafli an electrical engineering associate, and Zhengkun Ye a computer science associate. Jacob is a talented professional seeking continuous skill enrichment, his studies focus on electrical circuit design and analysis along with programming logic control systems. Abdulaziz is a talented professional who seeks challenges and takes great pride in his work, his studies focus on electrical circuit design. Zhengkun is an eager and talented professional who focuses on computer programming and computer vision area. The design team acts as a uniform body under the guidance and vision of the team leader, Jacob.

TABLE OF CONTENTS

		Page
1.0	**List of Figures**	**5**
2.0	**Introduction**	**7**
3.0	**Problem Statement**	**8**
3.1	Need	8
3.2	Objective	9
3.3	Background and Technology Survey	10
3.4	Marketing Requirements	15
3.5	Objective Tree	18
4.0	**Requirements Specification**	**19**
4.1	Engineering Requirements	19
5.0	**Design**	**22**
5.1	Functional Decomposition Diagrams	22
5.2	UML Diagrams	26
5.3	Mechanical Design Elements	38
5.4	Electrical Design Elements	42
6.0	**Design Verification and Testing**	**48**
6.1	Unit Testing	48
6.2	Integration Testing	56
6.3	Acceptance Testing	60
6.4	Requirements Verification	61
7.0	**Performance Analysis**	**63**
8.0	**Standards Compliance**	**64**
9.0	**Ethical and Societal Considerations**	**65**
10.0	**Constraints**	**66**
11.0	**Impact of Design as an Engineering Solution**	**67**
12.0	**Conclusion**	**67**
13.0	**Recommendations**	**68**
14.0	**References**	**69**
15.0	**Appendices**	**70**

1.0 - LIST OF FIGURES

Figure 1: Objective Tree
Figure 2: Level 0 Functional Decomposition Diagram
Figure 3: Level 1 Functional Decomposition Diagram
Figure 4: Level 2 Functional Decomposition Diagram
Figure 5: Activity Diagram
Figure 6: Communication Diagram
Figure 7: Component Diagram
Figure 8: Composite Structure Diagram
Figure 9: Deployment Diagram
Figure 10: Interaction Overview Diagram
Figure 11: State Machine Diagram
Figure 12: Object Diagram
Figure 13: Profile Diagram
Figure 14: Sequence Diagram
Figure 15: Timing Diagram
Figure 16: Use Case Diagram
Figure 17: Vehicle Diagram (Side View)
Figure 18: Vehicle Diagram (Front View)
Figure 19: Vehicle Layout Diagram
Figure 20: Test Bed Diagram
Figure 21: Motor Amplifier Circuit
Figure 22: Component Wiring Diagram
Figure 23: Component Wiring Diagram
Figure 24: System Connection Diagram
Figure 25: Raspberry Pi Connection Diagram
Figure 26: Arduino Connection Diagram

Table 1: 5V Battery Pack Test
Table 2: 9V Battery Test
Table 3: Arduino Microcontroller Test
Table 4: Infrared Sensor Test
Table 5: Raspberry Pi Test
Table 6: Arducam Camera Test
Table 7: Servo Test
Table 8: Motor Test

Table 9: Arduino and Raspberry Pi Test
Table 10: Camera and Servo Test
Table 11: Infrared Sensor and Motor Test
Table 12: Camera and Motor Test
Table 13: System Test
Table 14: Requirements Test

2.0 - INTRODUCTION

Have you ever been involved in an automotive accident caused by distracted driving or other conditions beyond your control? With the vast number of highway drivers and the adoption of mobile computing, the rate of automotive accidents has been steadily increasing. Having this knowledge, what can be done to slow this increase or begin a decrease in accidents related to distracted driving. A potential solution is to create a self driving or operator assistance systems for a vehicle. In this study, a model RC car will be retrofitted with components to automate several critical aspects of driving.

In order to understand this report, readers should have a technical background in electrical engineering and computer science or computer engineering and systems engineering. In particular, a background in systems integrations, computer programming and electrical circuit design. Many of the topics covered in this report are of a highly technical aspect, therefore unfamiliar or inexperienced readers may find difficulty fully understanding the scope of the report.

This report is intended to document the work that has been completed on this project. The report also provides a guided explanation of the execution of the project and the overall development strategy being used to complete the project. In summary, this report is a detailed overview of the course of the project. This report does not include the development specifications and engineering requirements needed to develop a full-size vehicle that behaves as the model vehicle is designed to operate.

3.0 - PROBLEM STATEMENT

3.1 NEED

A self driving model car is being proposed to address the needs of this project, this will reduce the initial cost of development and prototyping. Additional research and development will be necessary to extend the design to a full-size motor vehicle. The systems that are developed and integrated into the model vehicle need to be of high accuracy, high resilience and high availability to minimize the potential of system failures resulting in an accident.

One function of the design includes the ability to maintain uniform tracking pattern while traveling within a roadway lane in efforts to prevent unintended lane departures and reduce driver fatigue through steering corrections. The design will fully automate the lane guidance system, requiring no driver intervention, allowing the vehicle to navigate roadways independently. This system greatly reduces the risk of distracted driving related accidents whose primary cause was an unintended lane departure.

Another function of the design shall incorporate the ability to detect stationary and potentially moving objects around the perimeter of the vehicle. The design forces the vehicle to stop if a potential forward or rear collision is detected. In addition to forward and rear collision detection, side collision detection is being implemented to supplement the lane departure prevention system by returning to the original lane during a lane change if a potential collision is detected. This system reduces the risk of distracted driving related accidents whose causes include but are not limited to; failure to stop, failure to yield and abrupt stopping.

The final function of the design provides a image processor and instantaneous speed monitor. The design allows for posted traffic regulations, traffic signs for example, to be processed and an action taken in response to the processed image. The instantaneous speed sensor supplements this system by monitoring the vehicle speed in order to maintain the posted speed limit. This system reduces the risk of distracted driving related accidents who causes include but are not limited to; driver negligence.

3.2 OBJECTIVE

Implementation of Driving Feedback Systems and Autonomous Control Systems, or driving aides, in vehicle designs may reduce the number of motor vehicle accidents. Driving aides may include; unintended lane departure warning and prevention systems, intended lane-change collision prevention, forward and rear collision warning and mitigation, and additional sensor driven monitoring and supervisory systems. As an example, an environment-aware navigation system may be able to provide an early warning when approaching a stop sign or stop the vehicle if no action is taken by the driver. A car that is able to give real time feedback to the driver or even be able to control the car by itself will help with the problem.

For the solution we would use a RC car to keep the project size reasonable, which allows for a safer and lower cost project. However, the project could be applied to cars on a larger scale. One of the main goals is that the car stay in between two lines as it drives. This is where the hardware would come into place. We would need to use sensors in order to realize this goal. We will observe the car in various situations which demonstrate whether or not the response

behavior is satisfactory. Experiments may include stopping or maneuvering if the car senses an obstacle in close proximity, reacting to posted traffic signs, or uniform tracking within a lane. A challenge we could place it in would be to test it in multiple weather conditions and have it adjust accordingly to keep effectiveness.

This project is an effective response to the annual increase in motor vehicle accidents caused by distracted driving. The proposal identifies and attempts to mitigate several behaviors attributed to accidents, yet it is applied to a small scale model more practical for this course. In addition, this project is appropriate for the disciplines of both electrical and computer engineering which encompasses all project participants. Again, this project proposes several solutions intended to address several of the behaviors associated with many vehicular accidents.

3.3 BACKGROUND AND TECHNOLOGY SURVEY

The vehicle must follow the path of a road. One way to create a path to follow is to put a single line down the middle of the road. Then the car will follow this line through the use of a line sensor in the front center of the vehicle. This line sensor would be an infrared sensor which detects light and dark at a close range. “The IR emitter emits a constant IR beam. The white surface reflects most of the beam while the black surface absorbs most of the beam. This reflected beam is picked up by the IR detector and its conduction increases and hence a voltage variation in the output pin (0v for absence of IR rays and 5v for maximum intensity). This is given to the comparator to compare with a reference signal generated by the potentiometer. That is how the line is sensed by the sensors. Back to the post, you have to place the bot over the line and see what it reads for various position of the line. Make sure that the line always detected at

least by one sensor if is within the range of the three sensors. This is something you should take utmost care. The maximum distance between the sensors should not exceed the width of the line" reproduced from [1]. Two of these sensors would offer the greatest utility. The advantages are protection and a decrease in the percentage of accidents caused by lane departure. The main disadvantage would be that all the roads would need a line painted down the middle which would cost time and money. "Driver may get distracted with (Human-Technology Interaction while Driving, Passionate Conversations in the Car, Makeup/Hair in Car), or driving under the Influence. So the line sensor will help them to stay in the line" reproduced from [1]. The estimated cost for the sensor itself is $18.

The other option is using two sensors with two road lines. These would be placed on either side of the car by the currently existing lane markers on the roads. There would be a method through which the CPU could calculate the distance between the lines and make adjustments to the vehicle so that it remains within those lines. This method would also have the same advantage of aiding in preventing accidents caused by lane departure. The other advantage is that the lines on the road would not need to be altered. The disadvantage is that the vehicle would need more processing power which would be allocated to this function. Also, two sensors must be purchased rather than one which increases the cost of the project.

Another important feature of the product is traffic sign recognition. There are two main hardware systems which could be used for application. The first method would be an object detector hooked up to a camera. When an object is detected, sign or not, a picture is taken. The picture is then processed by software which checks for traffic signs in the image. This method would be inexpensive but less effective. The sign would have to be directly in the line of the

detector. There would also be a low number of frames to process while searching the images for signs. If obstruction or glare were to occur on the sign, the program might not be able to detect any sign from the low image count. The other hardware system would be a live video camera mounted to the vehicle [5]. This camera would record video and a program would process it with the aim of recognize traffic signs. This method requires greater processing power and a higher cost but would yield better detection results as there would be a higher frame count and less chance that obstruction or glare could completely hide a sign.

Image recognition software is an important part of the product. The program needs to run on somewhat low processing power, yet still be able to function with the highest accuracy. As the Arduino is unable to process video, the Raspberry Pi would be a better option so long as a portable power supply is used.

The system could be trained to handle recognition during inclement weather conditions. A number of algorithms can be used to recognize real-world images, namely scale invariant feature transform (SIFT), histogram of oriented gradients (HOG), and principal component analysis (PCA). Each of these is able to recognize traffic signs in real time using a video camera. HOG is considered more accurate when compared to SIFT and can achieve an 80-90% average detection rate [4]. HOG and PCA can be realized using OpenCV, a computer vision and machine learning library which is compatible with Raspberry Pi.

The implementation of object detection and avoidance is an important feature in the design of this product. The primary goal of this component is to detect a stationary or moving object and reroute or stop to avoid a potential collision. This goal is accomplished by monitoring the distance between the vehicle and an object in front of or behind the vehicle and appropriately

reacting if the distance is continually decreasing or reaches a specified minimum. There are two potential implementations for this system: infrared-based and sonar-based distance measuring.

The first implementation utilizes an infrared sensor which operates by emitting infrared light and calculating the time required for the light to return to a photo sensor after reflecting off an object. Infrared sensors are capable of accurately measuring a distance of 10 to 80 centimeters which translates to a range of three inches to just over two and a half feet. The variable voltage output from the sensor is appealing as it allows a simpler integration with computational hardware. The advantages of this sensor include a low cost design, compact construction, and simple output. Although this sensor is appealing, it has several caveats. The sensor optics are affected by debris and liquids, washing of the sensor may deteriorate the housing, direct sunlight impairs the accuracy and precision of the sensor, and intense color variations of the reference object impair sensor readings. Sensor information obtained from [2].

The second implementation utilizes a sonar sensor which operates by emitting sound waves and calculating the time required for it to return to the sensors after reflecting off an object. Sonar sensors are capable of accurately measuring a distance of six inches to just over twenty-one feet. The variable voltage output from the sensor is appealing, as it allows a simpler integration with computational hardware. The advantages of this sensor include further detection range, capability to detect small objects, and compact sensor design. Also, fewer sensors may be used to cover an area and light does not affect the operation of the sensor. The disadvantages of this sensor include higher cost and a six-inch minimum distance measurement (resulting in closer objects being reported at a distance of six inches). Additionally, the measurement range of six

inches to twenty inches can be impaired by acoustic phase effects and the sensor precision and accuracy may be affected by debris and liquid. Sensor information obtained from [3].

Speed detection is an essential requirement of a successful RC self-driving car model. One easy way to achieve this is by using a speed-sensing radar gun. It can be used as a handheld radar for speed detection or it can be mounted to a vehicle for moving use. For handheld use, the radar can be powered by the vehicle or it can operate from internal rechargeable batteries enclosed in the radar´s removable handle. A speed-sensing radar gun is small and lightweight, and one can be found easily. The primary disadvantages are that the speed-sensing radar gun has a high cost and that most are only able to detect speed in the range of 15 to 200 miles per hour. Due to the low operating speeds of the RC car, a speed-sensing radar gun is a poor choice for speed monitoring. A technology similar to radar speed detection may be used where a laser is utilized in place of radar. This solution suffers from high cost and the size of the equipment is unrealistic for effective implementation in the RC car.

A third solution may be a speed detection camera system that uses image processing techniques on video streams. A digital camera mounted on the car and QuickBird sensors can be used to detect the speed of the moving car [7]. It is similar to a micro GPS tracking chip. An object-based automated method of vehicle extraction from images was also proposed. The speed of a RC vehicle could be detected from digital images and mini GPS system with high accuracy [6]. The GPS system could help ensure the location and trajectory of the car model. The GPS tracking device needed is a real time live mini GPS tracker which can easily connect to smartphones. The micro GPS tracking system including chip and sensors may cost from $10 to $100.

3.4 MARKETING REQUIREMENTS

The design proposed must conform to the following marketing requirements. The required functions that the system is intended to fulfill are specified with a great deal of care to ensure the accurate and complete construction of the system. Marketing requirements are related to at least one engineering requirement to realize the application in the design and the construction of the final product.

(1) The system must provide functionality in efforts to reduce the number of collisions related to distracted driving. Various sensory zones shall be strategically located around the perimeter of the vehicle to ensure intended operation of the collision detection system. Each of these sensors should be able to individually report a potential collision to the central processing unit, instilling the appropriate behavior. The sensory system should be providing real time object proximity information to the central processing unit to ensure that mitigation efforts are selected accurately and are effective. A maximum distance should be established to avoid false reporting of objects that leads to erroneous unwanted behavior.

(2) The system should be scalable, allowing it to be effective for more than one implementation. The functionality should be able to be targeted toward multiple platforms or scale-factors of a car-like model. The components of the system should be cost effective and functionally effective and efficient, while maintaining a reasonable range of scalability or have scaled functionality-equivalent counterparts. Design

strategies should be geared toward constructing components that are modular and therefore compatible in several different implementations.

(3) The system must use GPS system to receive the navigation from a wirelessly transmitted signal. The GPS signal will be constantly received by the vehicle. The GPS system should better be capable of differential GPS utilizing WAAS or RTCM, which can provide accuracy up to meters. Processing the GPS data is composed of two parts: receiver and parser. Receiver is an interrupt service routine that stores data into a certain variable. The GPS receiver was utilized for localization. The RC car must be able to determine its location in the global coordinate system from GPS signal. The GPS chip used within the system must conform to the specifications listed in the engineering requirements.

(4) The system must be able to detect the speed of vehicle within a small margin of error. The system must navigate to and from a target destination using GPS while following the speed limit we set. The system requires a variable speed of operation in order to navigate to a location while passing various environmental obstacles. The input part of the GPS system will provide data about the current position of the car and feed the data to the processing unit. After processing the data, the processor will communicate with the GPS chip via serial communication to receive environmental data such as latitude, longitude, altitude, velocity, and heading.

(5) The system must be able to read in data from a live video camera mounted on the vehicle. The camera must be of a high resolution and framerate. The incoming data must then be processed by an on-board computer which will search the images for a number of

different traffic signs. After identifying a sign, the computer will then instruct the car to react accordingly. In order to ensure a proper reaction to a sign, the image processor must recognize the same traffic sign across several frames. Additionally, the system will improve itself over time, gaining more accuracy in varying weather conditions.

(6) The system must perform its work safely in an environment with humans, animals and other physical obstacles that may be on the roads. The system must have the ability to recognize physical impediments within a reasonable range to avoid a collision. Direct monitoring of the physical environment is required to safely stop the system in case of unplanned confrontation or hindrances such as people or animals. After recognizing an obstacle, the system must stop motion at acceptable rate according to the engineering requirements.

(7) The system must utilize sensors which control the car's wheels in order to stay between the two lane markers on the road. To follow the road, the system must be capable of slowing the traveling speed through curves so that the sensors can keep the car driving in the lines. The vehicle must also be able to recalculate its route due to the obstacles that may be in its path.

3.5 OBJECTIVE TREE

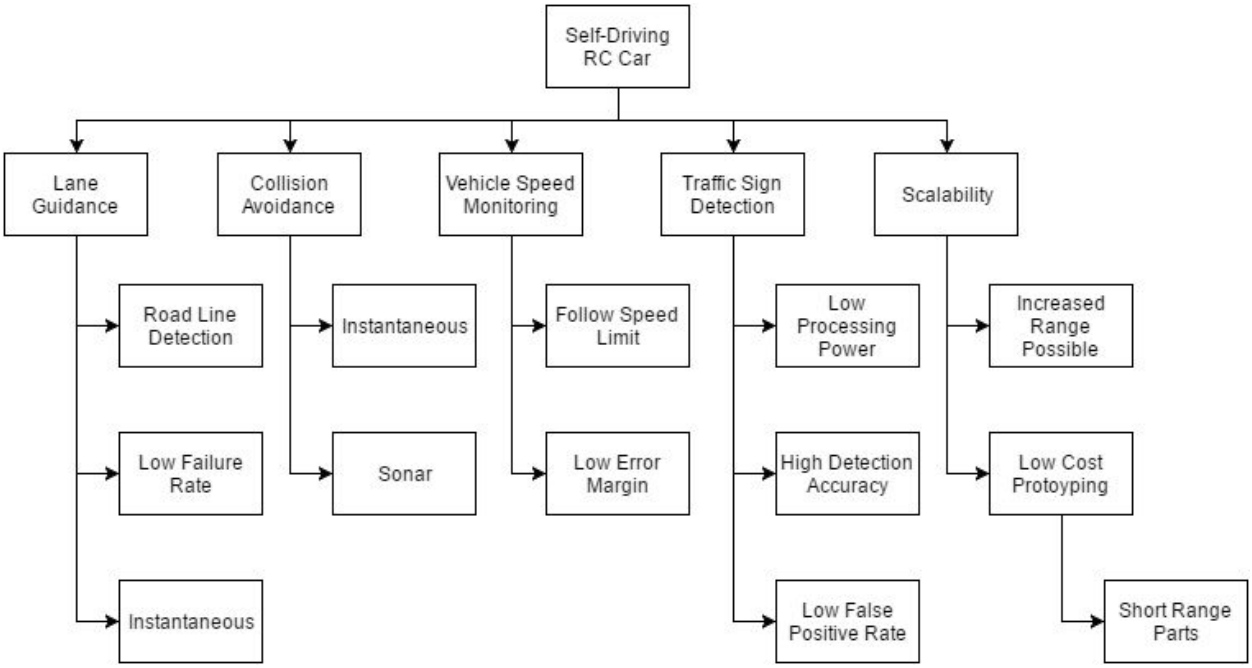

Figure 1: Objective Tree

4.0 - REQUIREMENTS SPECIFICATION

4.1 ENGINEERING REQUIREMENTS

1. The collision detection system must operate at a sampling rate of at least two samples per second.
2. The collision detection system must be able to stop the car with a minimum vehicle-to-object distance of six inches if a collision cannot be maneuvered around.
3. The maximum loss of effectiveness in a scalability application should not exceed 10%.
4. The product cost must not exceed $250.
5. The system must operate at a maximum speed of approximately five miles per hour.
6. The system must have a GPS system which includes a GPS chip, antenna, and software driver.
7. The camera must operate at a minimum 640x480 resolution and 15 frames per second.
8. Traffic signs must be accurately detected and identified at least 90% of the time.
9. The false positive detection rate should be less than one percent.
10. The system must be able to safely stay in between the lanes of a road at all times.

Engineering Requirement 1

The system is required to monitor for potential collisions and send signals to the central processing unit for countermeasure calculation. Due to the high variance in the surrounding environment, a reasonably high sampling rate is required to ensure the system is providing accurate and complete information.

Engineering Requirement 2

The system is required to stop the vehicle if a collision is unavoidable by other countermeasures. This last resort safety effort is in place to reduce the risk of a collision leading to bodily injuring from an accident.

Engineering Requirement 3

Scalability must maintain effectiveness. Failure to observe the effects of dynamically enlarging or reducing a system may lead to system inefficiencies resulting in higher costs from design modifications. Designing the system to be modular initially allows the extra cost of the system to be spread across multiple platforms rather than incurring cost on individual projects.

Engineering Requirement 4

Various electrical, mechanical, and computational components will need to be purchased to accomplish this design and construct a prototype. The combined cost of all necessary components must not exceed $250.

Engineering Requirement 5

The system is required to run in a predetermined path. The original vehicle usually comes equipped with a standard DC motor. This type of motor is designed to propel the car at speeds up to 25 mph. For this application, the desired speed is less than 5 mph and the stock motor could not reliably produce enough torque in this range.

Engineering Requirement 6

The system is required to have a GPS-based electronic navigation system. The antenna of the GPS system connects directly to the chip and receives location data wirelessly from the GPS network.

Engineering Requirement 7

In order to identify a traffic sign within an image, a decent resolution and framerate are needed. The larger both of these parameters are, the more likely a sign can be recognized within a set of images.

Engineering Requirement 8

The system should be able to detect and correctly recognize a sign the majority of the time. This is necessary so that the vehicle can obey signs and safely drive alongside others. In order to ensure this, the detection rate must be as high as possible.

Engineering Requirement 9

A high detection rate is meaningless if traffic signs are not present. Not only must the system be able to accurately identify signs that are present, it must also prevent itself from seeing signs that are not actually there. A false positive is a danger to the vehicle and to others, so it must be minimized.

Engineering Requirement 10

Each sensor in the lane monitoring system is independently monitored and provides feedback to the central processing unit to control the vehicle steering. Light-dependent resistors are used to detect contrast variances on a surface. Information gathered from the sensors is analyzed and a physical position is calculated. This position is used to keep the vehicle tracking uniformly between the lane markers.

5.0 - DESIGN

5.1 FUNCTIONAL DECOMPOSITION DIAGRAMS

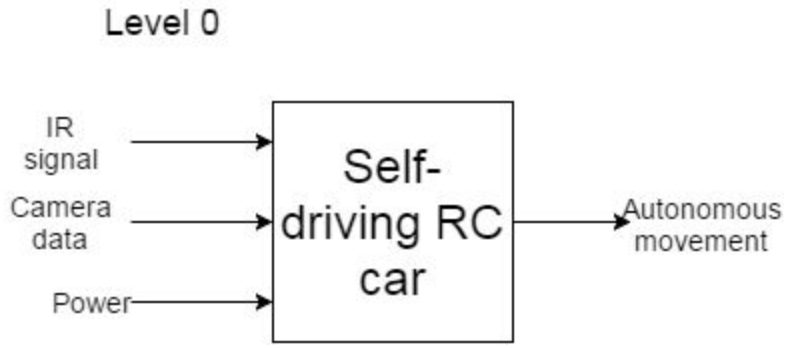

Figure 2: Level 0 Functional Decomposition Diagram

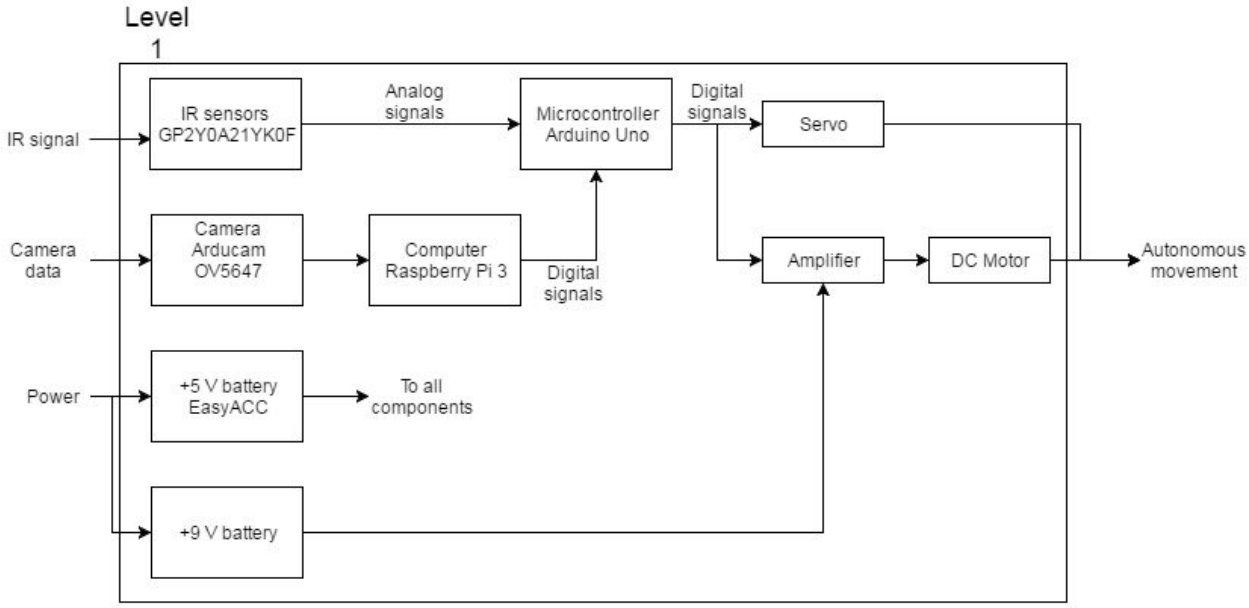

Figure 3: Level 1 Functional Decomposition Diagram

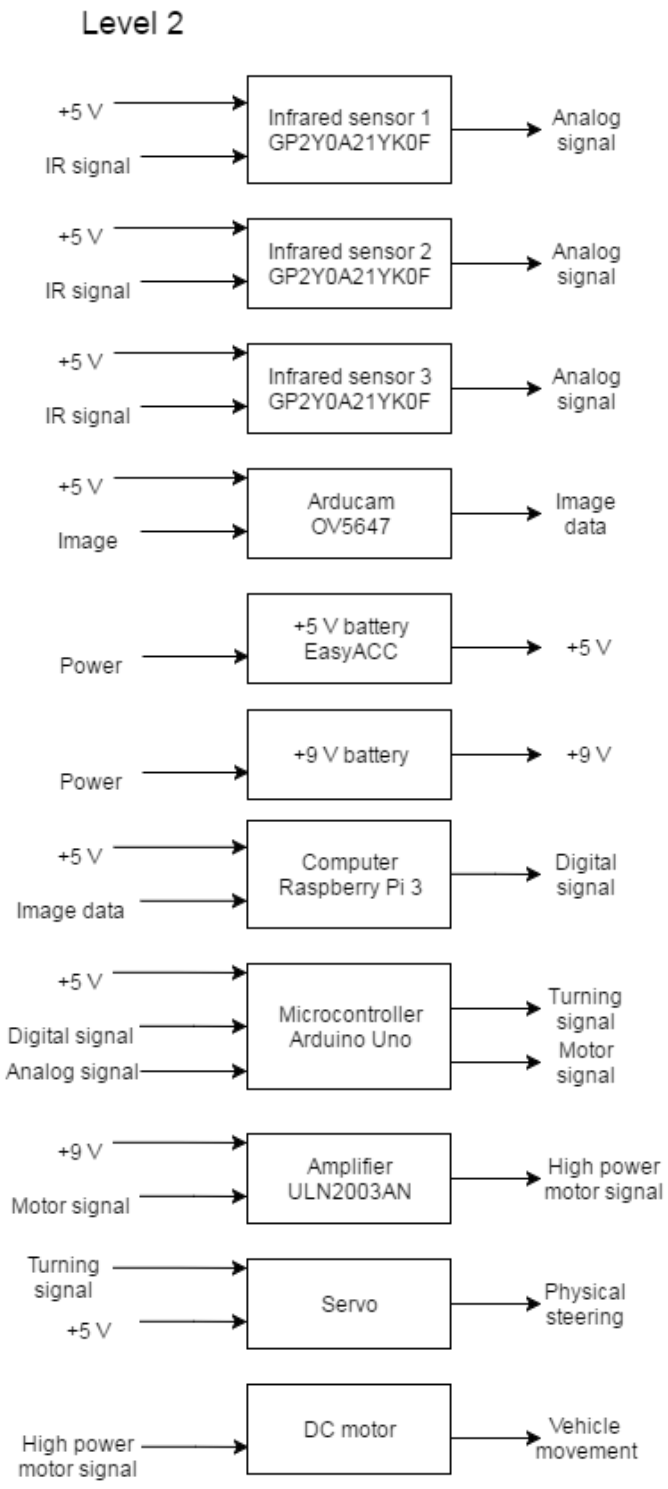

Figure 4: Level 2 Functional Decomposition Diagram

Module	Arduino
Input	Analog sensor signals and digital control signals
Output	Motor and servo control signals
Function	Microcontroller; gather inputs to decide on an output action

Table 1 Arduino Module

Module	Raspberry Pi
Input	Camera image data
Output	Digital control signals
Function	Process images and send actions to Arduino

Table 2 Raspberry Pi

Module	Battery
Input	Power via charging
Output	Power to components
Functionality	Provides power to the vehicle

Table 3 Battery Module

Module	Transistor Amplifier
Input	Motor control signal
Output	High current motor control signal
Functionality	Amplify the control signal to the motor

Table 4 Amplifier Module

Module	Servo
Input	Digital control signal
Output	Rotation angle
Functionality	Change front tire angle to turn vehicle

Table 5 Servo Module

Module	DC Motor
Input	Motor control signal
Output	Forward propulsion
Functionality	Physically moves the vehicle

Table 6 Motor Module

5.2 UML DIAGRAMS

This activity diagram indicates the process which the vehicle/system undertakes. Once activated, the sensors will begin taking in information. This information will be processed accordingly depending on the sensor from which it came. The data is gathered in one place and a decision is made. The actions form a chain which gets the vehicle from one location to another.

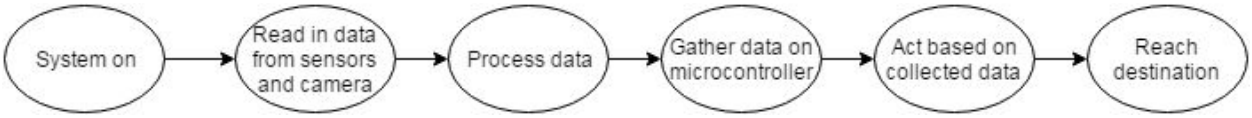

Figure 5: Activity Diagram

The communication diagram shows the flow of information within the system. The data is created by the sensors and camera. It is passed to the computer and microcontroller where it is then converted to physical outputs by the servo and motor.

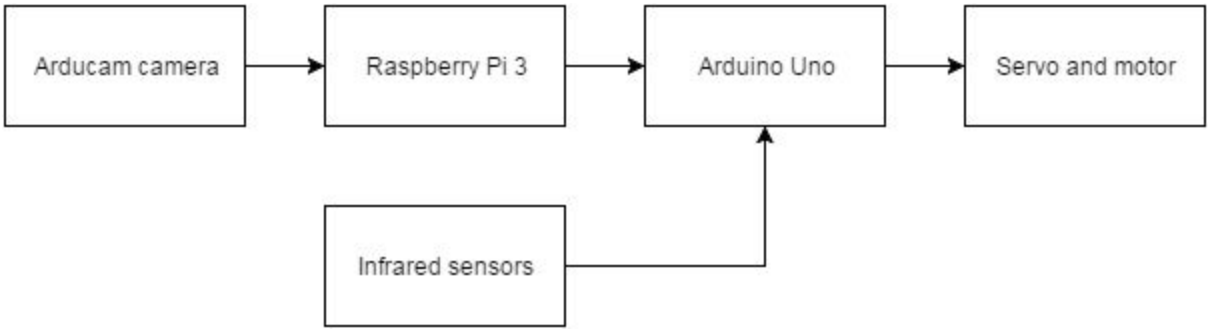

Figure 6: Communication Diagram

The component diagram shows the physical integration of the components within the system. The information/signal/data that each component sends or receives can also be read from this diagram.

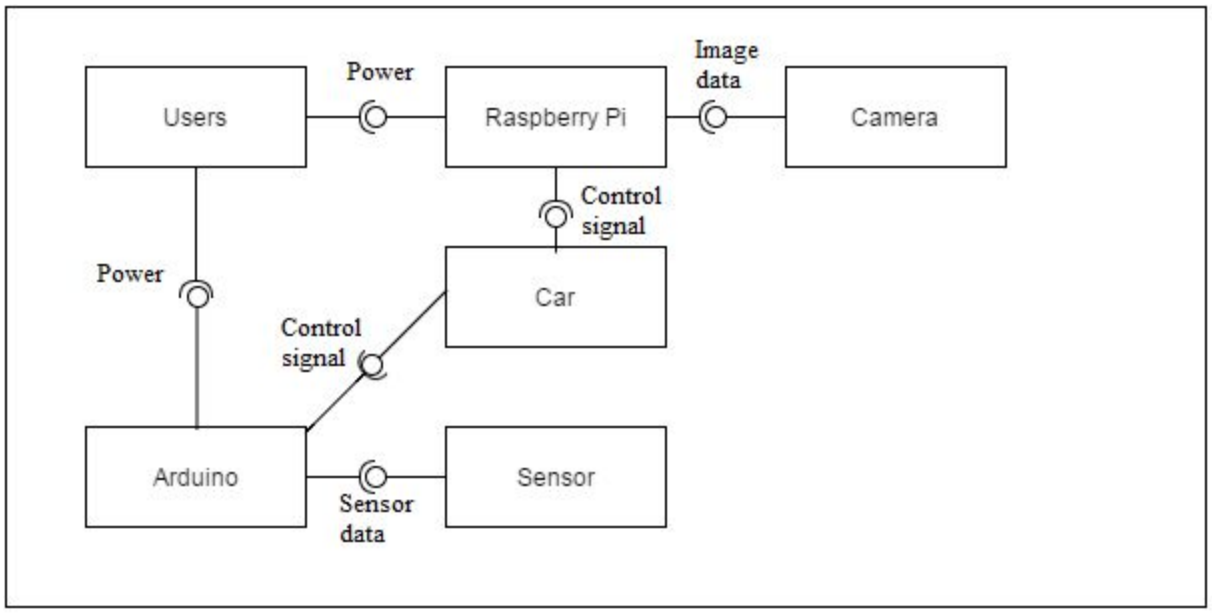

Figure 7: Component Diagram

The composite structure diagram shows the signals send between different main components and also the user. The user must be present to start the process. When started, the Raspberry Pi sends data to the Arduino which in turn controls components which move the vehicle (and the user).

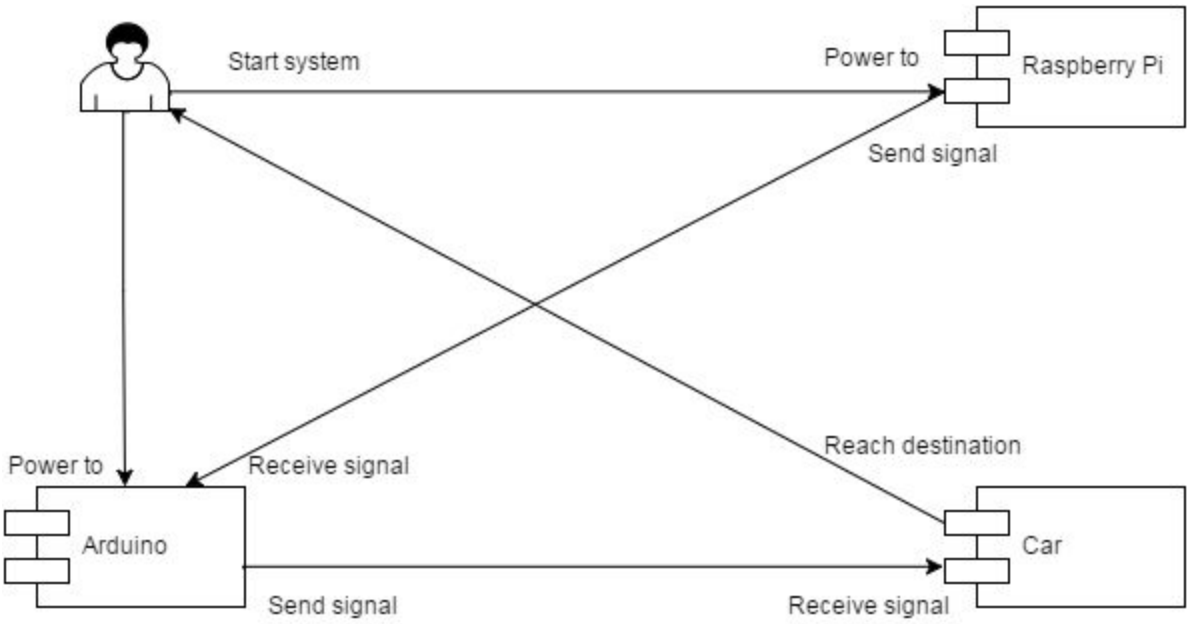

Figure 8: Composite Structure Diagram

The deployments diagram shows the main pieces of the system. The user interacts with the system to start it. The Raspberry Pi sends information to the Arduino. The Arduino processes the available data and moves the car accordingly.

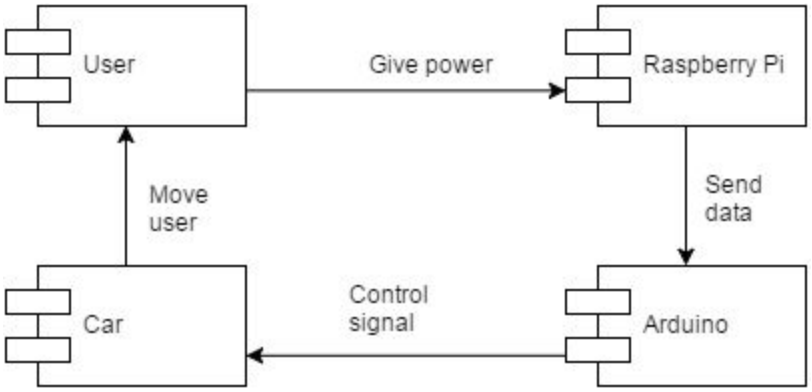

Figure 9: Deployment Diagram

The interaction overview diagram shows the process which is executed by the system. The end goal is the move the user to a destination.

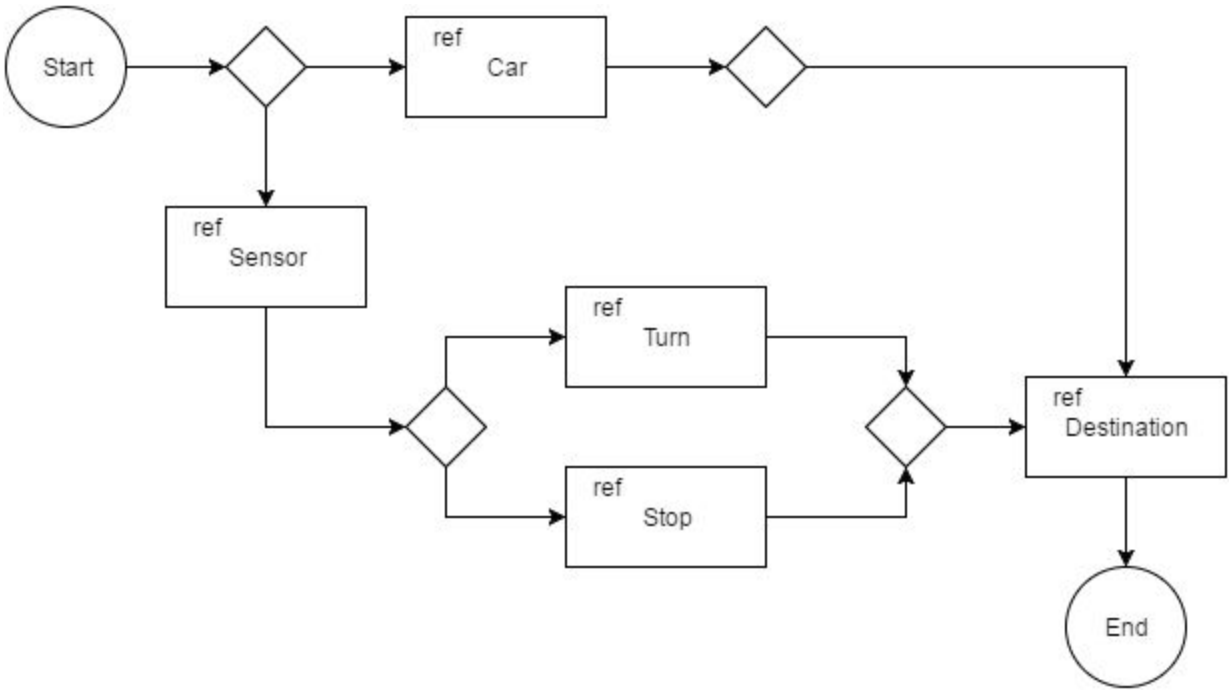

Figure 10: Interaction Overview Diagram

The state machine diagram shows the states which the system occupies as it executes. The transition of these states can be seen and are vital to the operation of the system.

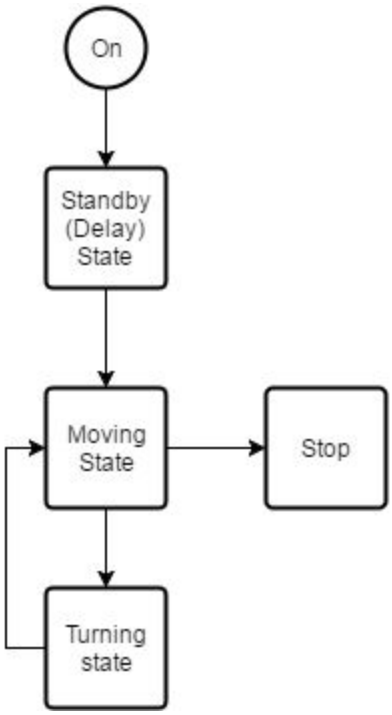

Figure 11: State Machine Diagram

The object diagram shows the distribution of objects in the programs and how they relate to the components of the system.

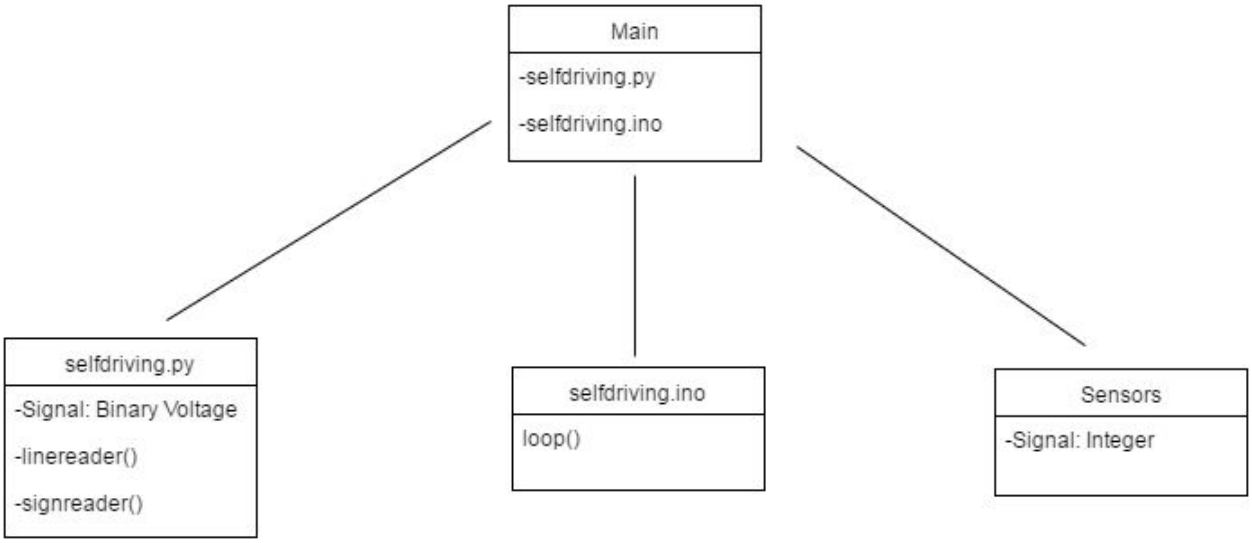

Figure 12: Object Diagram

The profile diagram shows the organization of the subprograms. The programs are distributed across the Raspberry Pi and the Arduino.

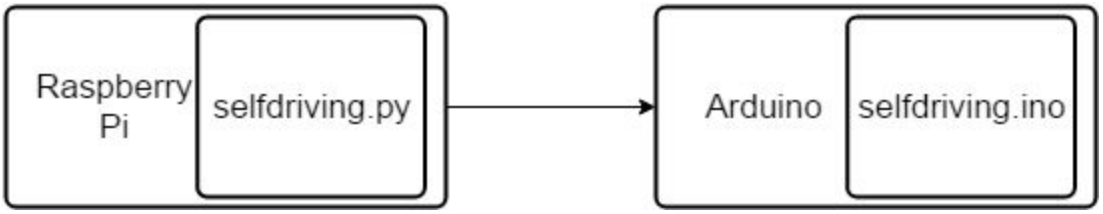

Figure 13: Profile Diagram

The sequence diagram shows how each component exchanges data. These exchanges allow the system to function properly.

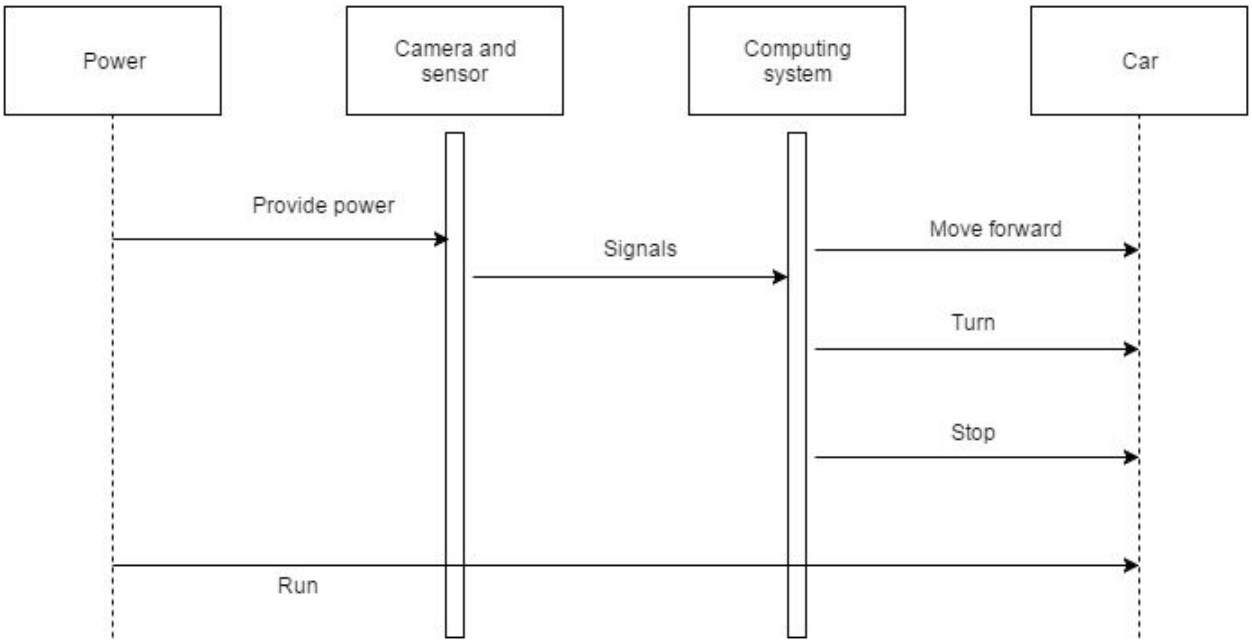

Figure 14: Sequence Diagram

The timing diagram shows the times which certain processes take. The sample rate can also be seen.

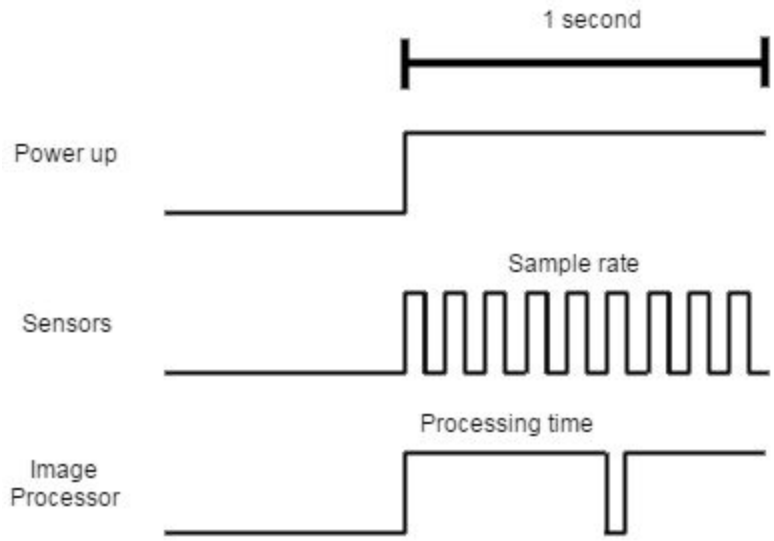

Figure 15: Timing Diagram

The use case diagram shows how an outsider, the user, will interact with the system.

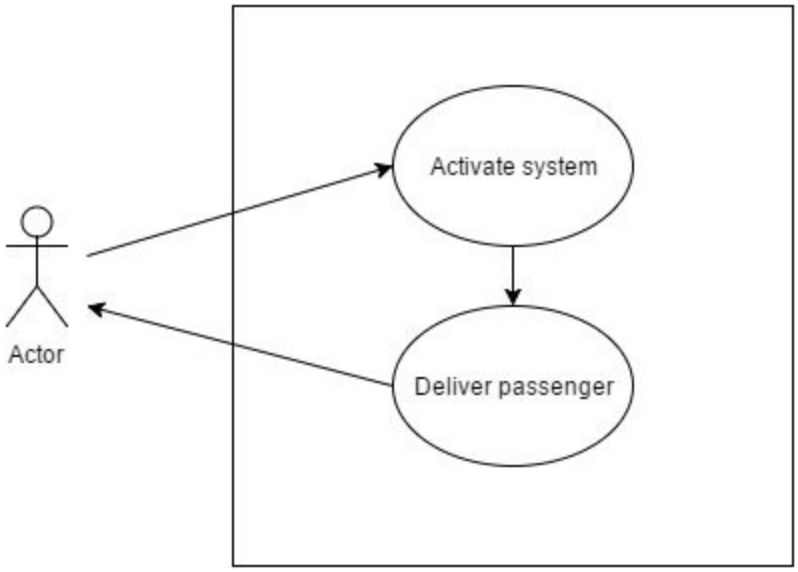

Figure 16: Use Case Diagram

5.3 MECHANICAL DESIGN DIAGRAMS

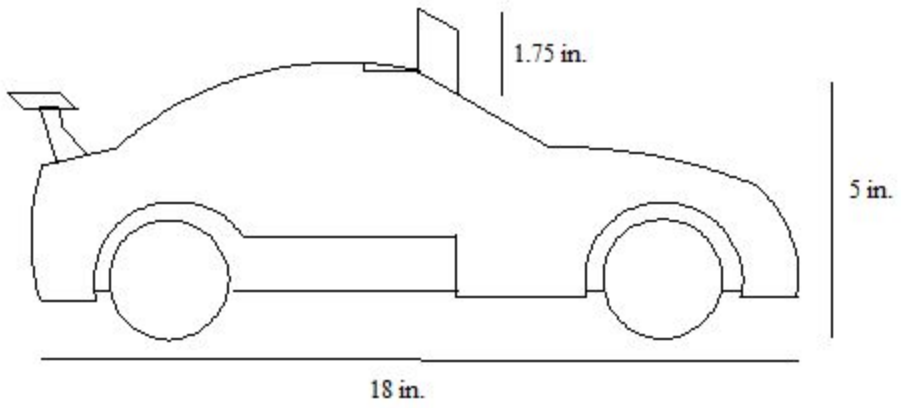

Figure 17: Vehicle Diagram (Side View)

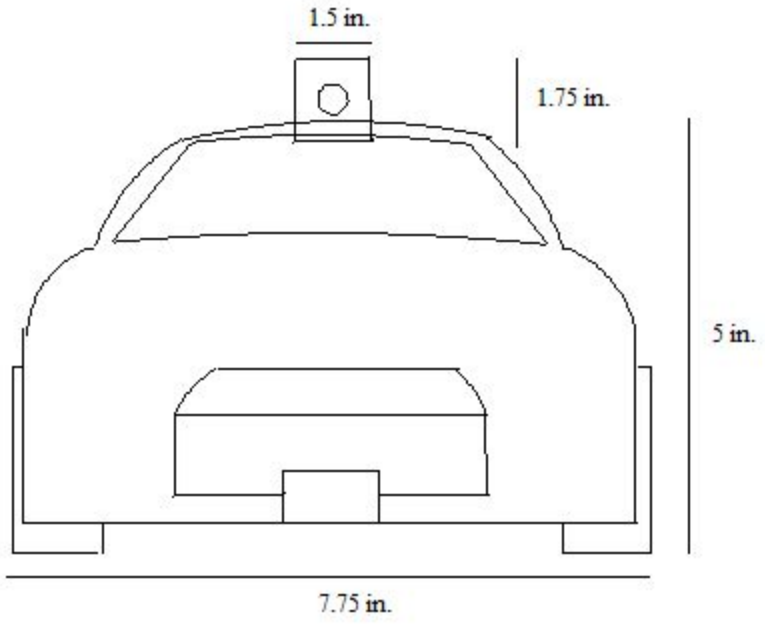

Figure 18: Vehicle Diagram (Front View)

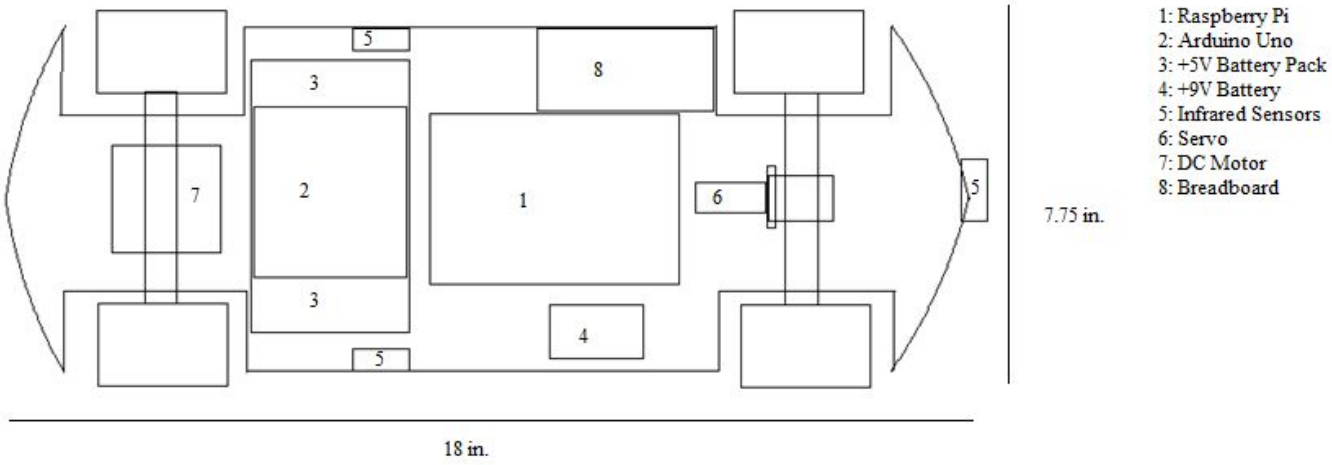

Figure 19: Vehicle Layout Diagram

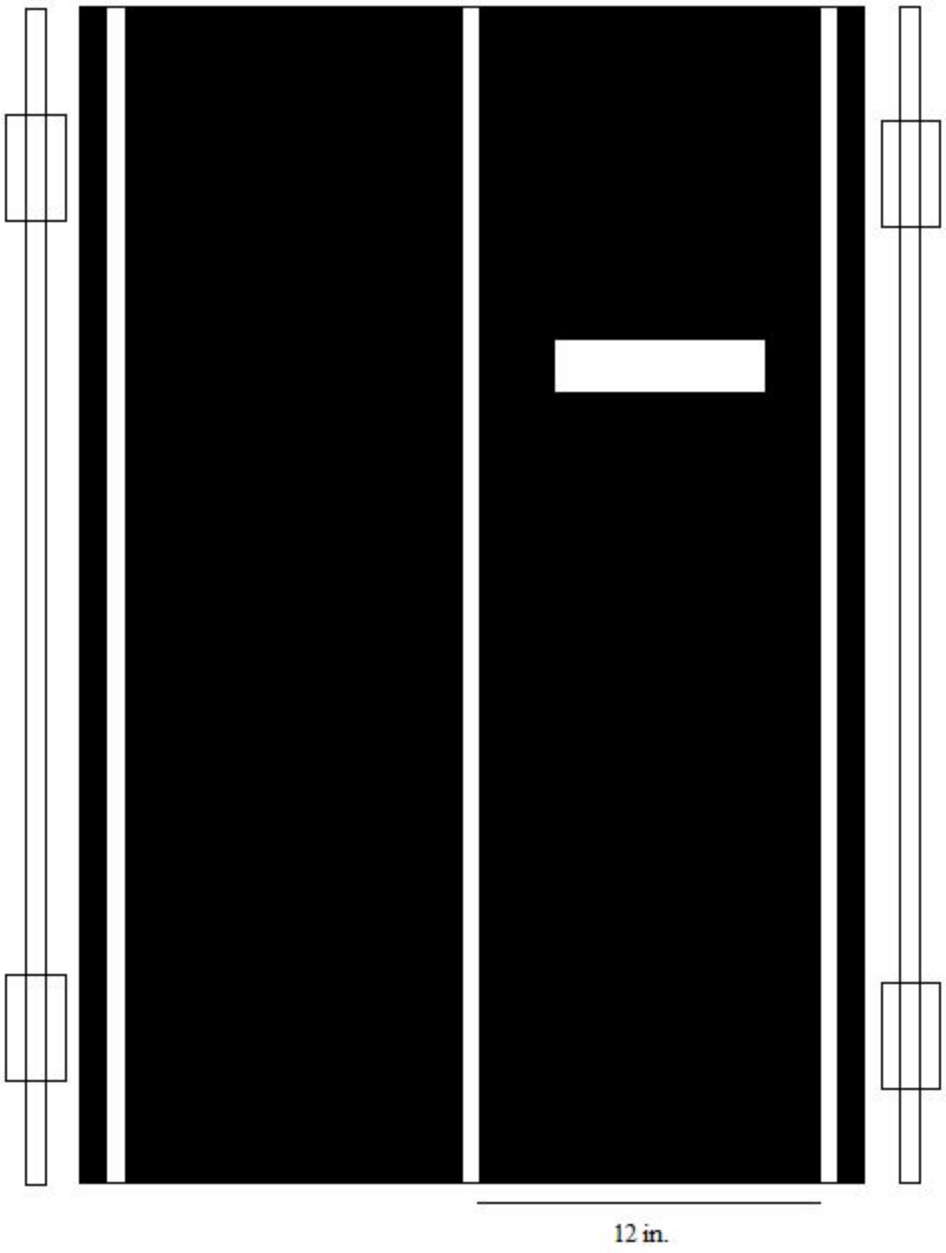

Figure 20: Test Bed Diagram

5.4 ELECTRICAL DESIGN DIAGRAMS

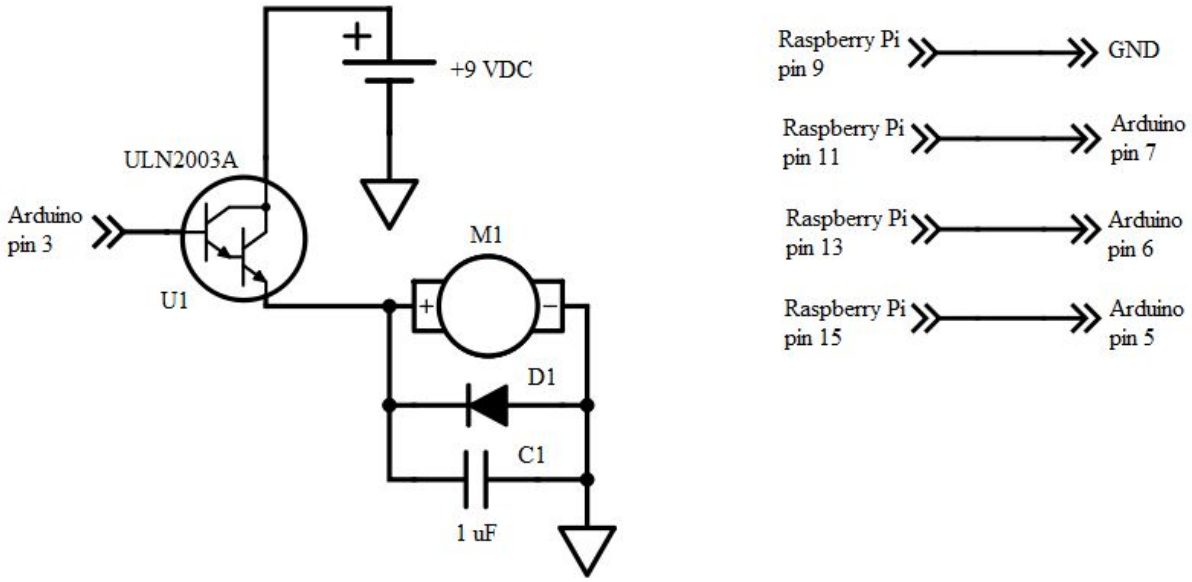

Figure 21: Motor Amplifier Circuit

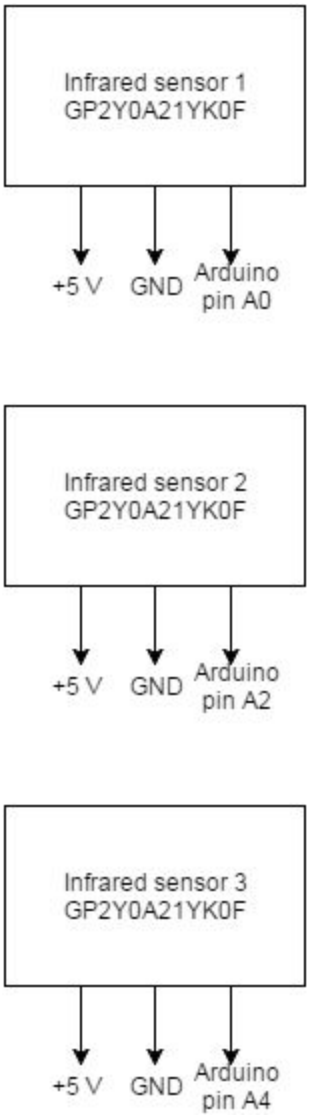

Figure 22: Component Wiring Diagram

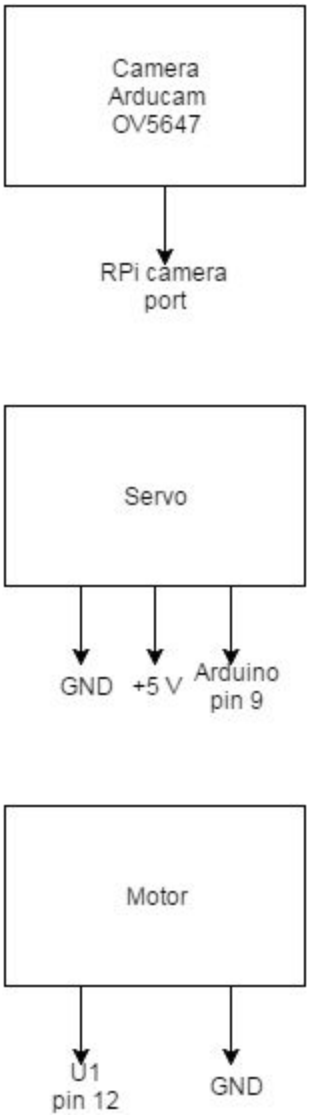

Figure 23: Component Wiring Diagram

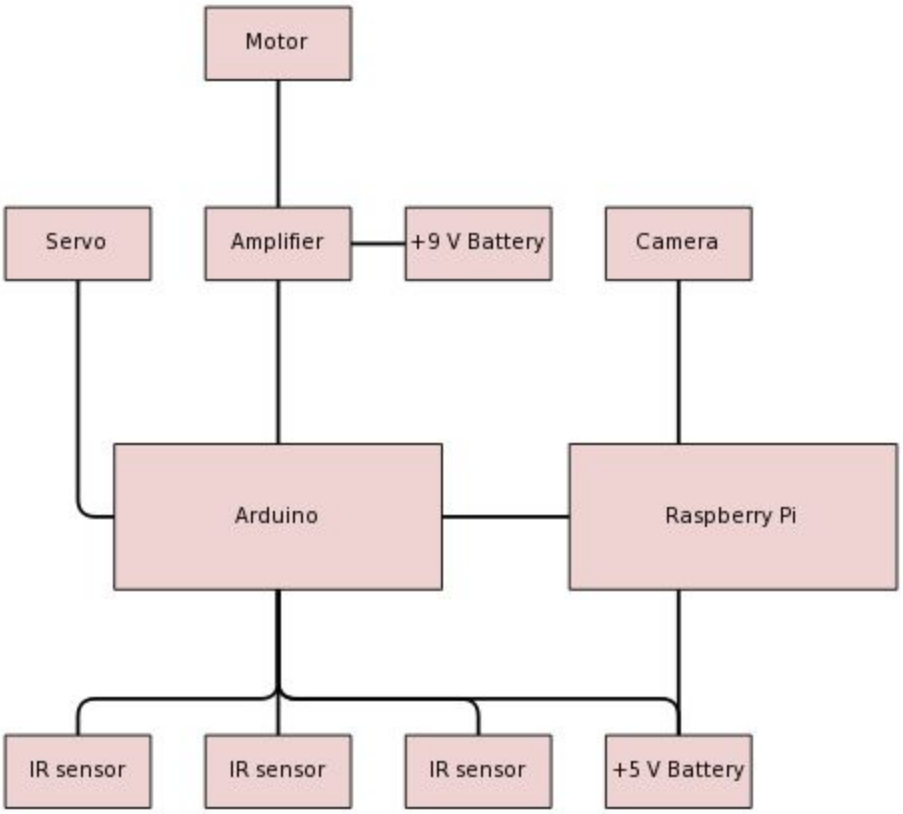

Figure 24: System Connection Diagram

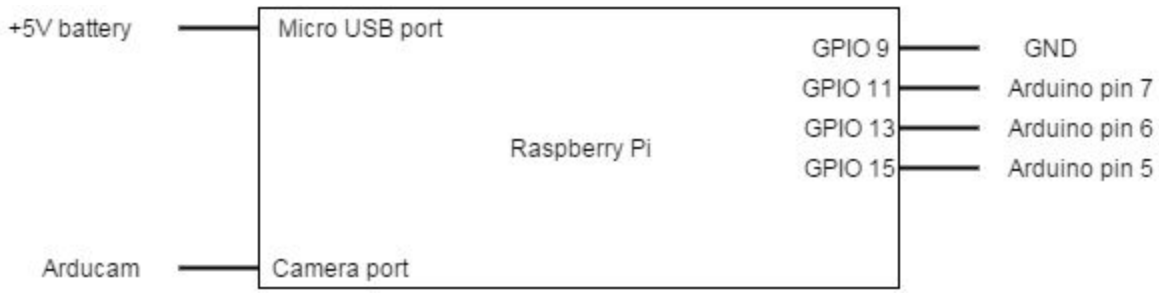

Figure 25: Raspberry Pi Connection Diagram

Figure 26: Arduino Connection Diagram

6.0 - DESIGN VERIFICATION AND TESTING

6.1 UNIT TESTING

Test Name: 5V Battery Pack Test				**Test Number:** T01	
Test Description: Test the battery pack to ensure the proper output voltage. This battery pack is used to power the Arduino, Raspberry Pi, and the sensors. It must have an output of +5 volts and a lifetime of several hours.					
Test Information					
Setup: Connect a USB to open wire cable					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Ensure all else is disconnected	X			
2	Press the button to check the level and verify that is has charge	X			
3	Read the voltage across the two wire terminals	X			
4	Verify that the voltage is +5 V (+/- 0.5 V)	X			

Table 1: 5V Battery Pack Test

Test Name: 9V Battery Test				**Test Number:** T02	
Test Description: Test the battery to ensure the proper output voltage. This battery is used to drive the motor amplifier. It provides a higher voltage and more current to run the motor.					
Test Information					
Setup: None					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Disconnect the 9V battery	X			
2	Read the voltage across the terminals	X			
3	Verify that the voltage is +9 V (+/- 1 V)	X			

Table 2: 9V Battery Test

Test Name: Arduino Microcontroller Test				**Test Number:** T03	
Test Description: Test the Arduino microcontroller. This microcontroller is used to process sensor data and control the outputs which in turn control the vehicle.					
Test Information					
Setup: Get a USB B cable					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Verify the correct model of the Arduino (Arduino Uno)	X			
2	Connect a USB B cable to the Arduino and a power source	X			
3	The power LED should light up on the Arduino	X			

Table 3: Arduino Microcontroller Test

Test Name: Infrared Sensor Test				**Test Number:** T04	
Test Description: Test the infrared sensors. These sensors are distance measuring devices. There are a total of three which all must be tested individually to ensure proper operation					
Test Information					
Setup: Complete T03					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect the sensor wires according to the schematics	X			
2	Verify that the analogread test code is uploaded to Arduino	X			
3	Open the serial window	X			
4	Verify that the numbers change depending on the distance away	X			
5	Repeat for all infrared sensors	X			

Table 4: Infrared Sensor Test

The Infrared Sensor Test verifies that the infrared sensors and Arduino comply with Engineering Requirement #1: *The collision detection system must operate at a sampling rate of at least two samples per second.*

Test Name: Raspberry Pi Test				**Test Number:** T05	
Test Description: Test the Raspberry Pi. This device is the on board computing unit used for image processing pictures from the camera.					
Test Information					
Setup: Get a micro USB cable					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Verify the correct model of the Raspberry Pi (Raspberry Pi 3)	X			
2	Connect a micro USB cable to the Raspberry Pi and a power source	X			
3	The power LED should light up on the Raspberry Pi	X			

Table 5: Raspberry Pi Test

Test Name: Arducam Camera Test				**Test Number:** T06	
Test Description: Test the Arducam camera. This camera is used to capture realtime images or video of the road ahead of the vehicle. It connects directly to the Raspberry Pi					
Test Information					
Setup: Complete T05. Get a monitor, mouse, and keyboard					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect the camera to the Raspberry Pi	X			
2	Connect an HDMI display, mouse, and keyboard to the Raspberry Pi	X			
3	Run the camera test code	X			
4	Verify that the camera is taking pictures	X			

Table 6: Arducam Camera Test

The Arducam Camera Test verifies that the camera and Raspberry Pi complies with Engineering Requirement 7: *The camera must operate at a minimum 640x480 resolution and 15 frames per second.*

Test Name: Servo Test				**Test Number:** T07	
Test Description: Test the servo. The servo is a small motor which has a controllable, set angle. This is used in the vehicle for directional steering.					
Test Information					
Setup: Complete T03					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect the servo wiring according to the schematic	X			
2	Upload the servo test code to the Arduino	X			
3	Power on the unit	X			
4	Verify that the servo angle changes	X			

Table 7: Servo Test

Test Name: Motor Test				**Test Number:** T08	
Test Description: Test the motor. The motor is used to physically move the car using an electrical control signal. The motor is a small 5V DC motor.					
Test Information					
Setup: Complete T03					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect the motor wiring according to the schematic	X			
2	Upload the motor test code to the Arduino	X			
3	Power on the unit	X			
4	Verify that the motor spins	X			

Table 8: Motor Test

6.2 INTEGRATION TESTING

Test Name: Arduino and Raspberry Pi Test					**Test Number:** T09
Test Description: Test the communication between the Raspberry Pi and Arduino boards. This verifies that the two devices are able to send and receive one another's data.					
Test Information					
Setup: Complete T03, T05					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect the Raspberry Pi GPIO and Arduino together according to the schematic	X			
2	Upload selfdriving.ino to the Arduino	X			
3	Power on the units	X			
4	Run selfdriving.py on the Raspberry Pi	X			
5	Check the GPIO signal voltages	X			

Table 9: Arduino and Raspberry Pi Test

Test Name: Camera and Servo Test				**Test Number:** T10	
Test Description: Test the camera and servo integration. The camera will take in image data and the data is processed in the Raspberry Pi. A signal is sent to the Arduino which then turns the servo.					
Test Information					
Setup: Complete T09					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect the Raspberry Pi GPIO and Arduino together according to the schematic	X			
2	Upload selfdriving.ino to the Arduino	X			
3	Power on the units	X			
4	Run selfdriving.py on the Raspberry Pi	X			
5	Move the vehicle side to side over the test bed	X			
6	The servo should turn left when right of center and right when left of center	X			

Table 10: Camera and Servo Test

The Camera and Servo Test verifies that the camera, servo, Arduino, and Raspberry Pi comply with Engineering Requirement #10: *The system must be able to safely stay in between the lanes of a road at all times.*

Test Name: Infrared Sensor and Motor Test				**Test Number:** T11	
Test Description: Test the connection between the infrared sensor and motor. When an object in front of the sensor reaches some certain distance, the sensor is triggered and the motor will stop spinning.					
Test Information					
Setup: Complete T04 and T08					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect the Raspberry Pi GPIO and Arduino together according to the schematic	X			
2	Upload selfdriving.ino to the Arduino	X			
3	Power on the units	X			
4	Run selfdriving.py on the Raspberry Pi	X			
5	Verify that the motor is spinning	X			
6	Place an object in front of the vehicle	X			
7	The motor should stop spinning	X			

Table 11: Infrared Sensor and Motor Test

The Infrared Sensor and Motor test verifies that the infrared sensor, Arduino, and motor comply with Engineering Requirement #2: *The collision detection system must be able to stop the car with a minimum vehicle-to-object distance of six inches if a collision cannot be maneuvered around.*

Test Name: Camera and Motor Test				**Test Number:** T12	
Test Description: Test the camera and motor integration. This is also the stop sign detection test. When a red octagon is seen by the camera, the car will stop.					
Test Information					
Setup: Complete T04 and T06					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect the Raspberry Pi GPIO and Arduino together according to the schematic	X			
2	Upload selfdriving.ino to the Arduino	X			
3	Power on the units	X			
4	Run selfdriving.py on the Raspberry Pi	X			
5	Hold a red octagon in the camera's field of vision	X			
6	The motor should stop	X			

Table 12: Camera and Motor Test

6.3 ACCEPTANCE TESTING

Test Name: System Test					**Test Number:** T13
Test Description: Test the system under ideal conditions. A test bed is required in this procedure so one must be built. Refer to Section 5.3: Mechanical Design Diagrams for a test bed layout.					
Test Information					
Setup: Complete all previous tests, construct a test bed					
Name of Tester:		**Date:**			**Time:**
#	**Procedure**	**Pass**	**Fail**	**N/A**	**Comments**
1	Verify that selfdriving.ino is uploaded to the Arduino	X			
2	Ensure that all connections are made	X			
3	Place the vehicle on a test bed	X			
4	Power on the units	X			
5	Run selfdriving.py on the Raspberry Pi	X			
6	The vehicle should begin moving	X			
7	The vehicle should stay in between the lines	X			
8	The vehicle should stop before colliding with any obstacles	X			
9	The vehicle should stop when facing a stop sign	X			

Table 13: System Test

The System Test verifies that the system complies with Engineering Requirement #5: *The system must operate at a maximum speed of approximately five miles per hour.*

The System Test verifies that the system complies with Engineering Requirement #8: *Traffic signs must be accurately detected and identified at least 90% of the time.*

The System Test verifies that the system complies with Engineering Requirement #9: *The false positive detection rate should be less than one percent.*

6.4 REQUIREMENTS VERIFICATION

Engineering Requirements	Test Verification
The collision detection system must operate at a sampling rate of at least two samples per second.	Show that objects are detected at a minimum of two times per second. Test: Infrared Sensor Test Test #: T04
The collision detection system must be able to stop the car with a minimum vehicle-to-object distance of six inches if a collision cannot be maneuvered around.	Show that the vehicle stops to a distance of six inches Test: Infrared Sensor and Motor Test Test #: T11
The maximum loss of effectiveness in a scalability application should not exceed 10%.	The system should be able to be scaled up effectively Test: N/A Test #: N/A
The product cost must not exceed $250.	Show that cost of materials is less than or equal to $250 Test: N/A Test #: N/A
The system must operate at a maximum speed of approximately five miles per hour.	Show that the vehicle moves at a speed of 5 mph maximum Test: N/A Test #: N/A
The system must have a GPS system which includes a GPS chip, antenna, and software driver.	Show that a GPS chip is integrated and can help with guidance Test: N/A Test #: N/A
The camera must operate at a minimum 640x480 resolution and 15 frames per second.	Show that the camera resolution is at least 640x480

	Test: Camera Test Test #: T06
Traffic signs must be accurately detected and identified at least 90% of the time.	Show that, when presented with a stop sign, the vehicle will detect it 90% of the time Test: Camera and Motor Test Test #: T12
The false positive detection rate should be less than one percent.	Show that false positive sign detections are minimized to under 1% Test: Camera and Motor Test Test #: T12
The system must be able to safely stay in between the lanes of a road at all times.	Show that the car stays in between the lines of the test bed Test: Camera and Servo Test Test #: T10

Table 14: Requirements Test

There are a couple engineering requirements which were not tested for during these procedures:

- Engineering Requirement #3 (*The maximum loss of effectiveness in a scalability application should not exceed 10%)* was not exactly a testable requirement along with the rest. It is one which must be proven via comparative calculations.
- Engineering Requirement #4 (*The product cost must not exceed $250*) is satisfied by the system. It is not tested for but the system does comply.
- Engineering Requirement #6 (*The system must have a GPS system which includes a GPS chip, antenna, and software driver*) is not satisfied by the system. The GPS chip integration was scrapped before even making it into the vehicle. With the vehicle operating over such a small area, GPS resolution would be too low to provide meaningful data to the car unless the track was made much larger and with more routes.
- Engineering Requirement #9 (The false positive detection rate should be less than one percent) was not tested for. We simply did not have a large enough sample size to determine this precisely. Given more time, this would have been thoroughly tested.

7.0 - PERFORMANCE ANALYSIS

Performance analysis was done via multiple tests. The tests contained within section 6.0: Design Verification and Testing were used for this. Unit testing analyzed the individual components while integration testing analyzed how they worked together. Acceptance testing shows that the vehicle functions under certain set conditions. Requirements verification shows that the project meets most of the design requirements which were selected before.

Each of the individual components used for the vehicle were shown to work correctly and efficiently. The applied tests were able to determine that each component was not only working, but that the component was the best to achieve its end goal. Tests were performed to show their integration as well. The components formed their own systems and those systems were integrated with one another. The tests show that these systems function with one another very well.

During the course of the design and testing phases, several changes were made to the vehicle. Initially, the intention was to have the vehicle change lanes when confronted with an obstacle as long as the other lane is available. While testing this, the car was able to detect and avoid obstacles, but it would sometimes hit the obstacles while turning. As a result, we determined that the safest action for the car to take in such a situation is to slow to a stop, This allows the vehicle to still avoid collisions while alse being safer than before.

8.0 - STANDARDS COMPLIANCE

The original RC car was build to follow industry standards for wireless communication. There was a remote control transmitting device and a receiver on the car. Due to the nature of this prototype, these were both removed in their entirety. The ultrasonic distance sensors may have needed to meet certain compliance standards, but they were taken out in favor of infrared sensor. The GPS chip was also scrapped during the design process. Because of this, the vehicle does not emit or attempt to receive any wireless communication signals.

According to the Arduino manufacturer's website, "All boards are labelled with the FCC and CE logo, as they meet the electromagnetic compatibility standards set in their respective jurisdictions. Arduino/Genuino products meet the essential requirements of EU Directive 2001/95/CE General directive on products safety and Directive 93/68/CE" [8]. Additionally, "This device complies with Part 15 of the FCC Rules" [8].

The Raspberry Pi is not intended to be used in commercial applications. This product is a tech demonstration and not a commercially available product. From the manufacturer's website, "This evaluation board/kit is intended for use for ENGINEERING DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY and is not considered by BeagleBoard.org to be a finished end-product fit for general consumer use" [9].

9.0 - ETHICAL AND SOCIETAL CONSIDERATIONS

As automation and computerization continue to advance and replace more and more of our world, cars may not be left out. Cars which contain a number of sensors to assist a driver are already becoming commonplace today. Self-driving cars have been theorized, developed, and even made commercially available in the past couple years. A number of automakers are currently testing these sorts of vehicles with planned releases within the next couple years.

A self-driving car is a hotbed of ethical debates. Car crashes are one of the leading causes of death in developed nations. This is with a person behind the wheel. A reliable computer system, possibly interfaced with other computer systems, has the possibility of making driving far safer for everyone involved. A number of people have suggested that cars should be automated only if they are absolutely proven to be statistically safer than human drivers; if the number of accidents decreases by a considerable amount. If this is the case, then it would be unethical not to explore it.

Another common debate is on how the vehicle will make decisions. A human driver can decide what to do in a situation for him/herself. With a fully automated car, the driver does not have an input. The computer system of the car will choose the best possible action which may or may not be at odds with what would be the human driver's decision.

One more consideration is that a self-driving car is after all still a computer. Computers are susceptible to viruses and hacking. A person may be able to send malicious software to a car which causes it to fail. If vehicles are networked together with one another, then an outsider

made be able to take control of a vehicle remotely without the driver's permission. These issues are ones that must be addressed going forward with the automation of vehicles.

10.0- CONSTRAINTS

The self-driving RC car is a small vehicle. This means that it cannot necessarily function the same as a full-sized vehicle. Everything is small scale and a number of components are not what they would be for a full-sized vehicle. The devices used are merely meant to be a simulation of the real car's components. Theoretically, the system could be scaled up to a regular car.

The camera, although vital, does constrain the system under non-ideal conditions. The lighting of the room in which the vehicle is used can have an effect on the ability of the system to function correctly. Excessively dark or bright environments create problems with the line detection and sign recognition. This means that the vehicle might not operate very well while in imperfect lighting conditions. These can be fixed for a scaled up version, but for this small and cost effective model, there is not much that can be done.

The processing speed is also a constraint. The object detection system can operate at a very high speed. The Arduino reads in these samples extremely quickly and can react quite fast. The Raspberry Pi takes more time doing it processing though. The data output rate is quite small, far smaller than what is wanted for a full-sized vehicle. However, this is not much of an issue for this small-scale prototype.

11.0- IMPACT OF DESIGN AS AN ENGINEERING SOLUTION

The automation of vehicles is something that will very likely happen. Not only that, but it will happen sooner rather than later. Showing that a small-scale model of such a self-driving car can be made easily and cheaply proves that far better systems can be made currently. The fact that a non-negligible number of these cars could be on the road within a few years is enough to start planning for them. Assuming the transition goes smoothly, things may not change much at first. As more of these cars replace human drivers, and as networking systems come into place, there will be a sharp decrease in the number of accidents and a more efficient flow of traffic overall. People will be safer and will probably spend less time on their commutes.

12.0- CONCLUSION

The self-driving RC car is a well-made scale model system of what could be seen on the roads in just a few years. Piece by piece, the system is proven to work. Integrated together these pieces are shown to interact properly. Although there are a few issues with the system, it works as a functional scaled-down model. Given more time and money, there is no reason that the project shouldn't be completely doable. In fact, with more time and money, a far better model could be made. The requirements we had initially set out to fulfill were almost entirely completed.

The implications of such a design must be thought of going forward. Price is a limiting factor of any design. The cheapest isn't always the best. The self-driving RC car we made must

be thought of in this sense as well. It is a great design which meets the needs of the project, but designs can always be improved upon. Looking into the future, a self-driving vehicle prototype is somewhat of a hot commodity. The next couple years will be a race to find the best system to create a full-sized self-driving car. Making small models helps meet this end.

13.0- RECOMMENDATIONS

The design can surely be upgraded. A couple years from now, more powerful and smaller computing devices will be available. These devices will boost the effectiveness of a project like this manyfold. Improved processing speed would help every part of the design.

A slightly larger model would be a great idea. With a larger model, the design gets a step closer to being more realistic. Larger-scale components could be used and more realistic systems could be put in place, systems which completely mimic a real car. However, this larger model would surely increase the cost of the project. With a larger budget, such a thing is doable.

A GPS chip was one of the requirements that was not met for this project. Looking back, this was something that could not reasonably be used for our design. GPS resolution is too large for something on this scale. A slightly larger model would be able to accommodate a GPS chip for navigation.

14.0- REFERENCES

[1] Siddharth. (2013, June 4). Programming a Line Follower Robot [Online]. Available:http://embedjournal.com/programming-line-follower-robot/

[2] SHARP. (2006, December 1). GP2Y0A21YK0F [Online]. Available: http://www.sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf

[3] MaxBotix. (2015, February 20). LV-MaxSonar -EZ Series [Online]. Available: http://maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf

[4] Bram Alefs et al. (2007, June 13-15). Road Sign Detection from Edge Orientation Histograms [Online]. Available: http://www.vision.caltech.edu/VisionWiki/images/4/4b/Alefs07road.pdf

[5] Jack Greenhalgh et al. (2012, December). Real-Time Detection and Recognition of Road Traffic Signs [Online]. Available: http://www.cs.bris.ac.uk/home/greenhal/jg_trans_its_2012.pdf

[6] Wayne. (2015, January.) Navigating with GPS [Online]. Available: https://sites.google.com/site/wayneholder/self-driving-rc-car/navigating-with-gps

[7] Zheng Wang. (2015) Self Driving RC Car [Online]. Available: https://zhengludwig.wordpress.com/projects/self-driving-rc-car/

[8] Arduino. Products Warranty. [Online]. Available: https://www.arduino.cc/en/Main/warranty

[9] Raspberry Pi. Compliance Testing. [Online]. Available: https://www.raspberrypi.org/blog/compliance-testing/

15.0- APPENDICES

Arduino Code:

```
#include <Servo.h>

Servo servoPin;
int motorPin = 3;
int distPinFront = A0;
int distPinRight = A2;
int distPinLeft = A4;
int stopPin = 5;
int moveRight = 6;
int moveLeft = 7;

void setup()
{
  servoPin.attach(9);
  pinMode(motorPin, OUTPUT);
  pinMode(distPinFront, INPUT);
  pinMode(distPinLeft, INPUT);
  pinMode(distPinRight, INPUT);
  pinMode(stopPin, INPUT);
  pinMode(moveRight, INPUT);
  pinMode(moveLeft, INPUT);

  Serial.begin(9600);
}

void loop()
{
  int position;
  int distFront;
  int distRight;
  int distLeft;
  int valStop;
  int valMoveLeft;
  int valMoveRight;

  distFront = analogRead(distPinFront);
  distRight = analogRead(distPinRight);
  distLeft = analogRead(distPinLeft);

  Serial.print(distFront);
  Serial.print(" ");

  valStop = digitalRead(stopPin);
  valMoveRight = digitalRead(moveRight);
```

```
  valMoveLeft = digitalRead(moveLeft);

  analogWrite(motorPin, 255);
  servoPin.write(90);

  Serial.print(distFront);
  Serial.print(" ");

  if (distFront > 300){
    analogWrite(motorPin, 0);
    delay(5000);
  }

  if (valStop == HIGH){
    analogWrite(motorPin, 0);
    delay(5000);
  }

  if (valMoveLeft == HIGH){
    servoPin.write(100);
    delay(300);
  }

  if (valMoveRight == HIGH){
    servoPin.write(80);
    delay(300);
  }

  delay(100);
}
```

Raspberry Pi Code:

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
from PIL import Image
import numpy as np
import RPi.GPIO as GPIO


camera = PiCamera()
camera.resolution = (1008, 480)
camera.framerate = 15
rawCapture = PiRGBArray(camera, size=(1000, 480))
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

```

```
GPIO.setup(11, GPIO.OUT) #move left
GPIO.setup(13, GPIO.OUT) #move right
GPIO.setup(15, GPIO.OUT) #stop

time.sleep(3)

GPIO.output(11, GPIO.LOW) #move left
GPIO.output(13, GPIO.LOW) #move right
GPIO.output(15, GPIO.LOW) #stop

def linereader():
    i = 240
    j = k = 500
    countleft = 0
    countright = 0

    img = cv2.imread('/home/pi/Pictures/car/road.png')
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray,50,150,apertureSize = 3)

    lines = cv2.HoughLines(edges,1,np.pi/180,150)
    for rho,theta in lines[0]:
        a = np.cos(theta)
        b = np.sin(theta)
        x0 = a*rho
        y0 = b*rho
        x1 = int(x0 + 1000*(-b))
        y1 = int(y0 + 1000*(a))
        x2 = int(x0 - 1000*(-b))
        y2 = int(y0 - 1000*(a))
        cv2.line(img,(x1,y1),(x2,y2),(0,0,255),2)

    cv2.imwrite('/home/pi/Pictures/car/houghlines3.jpg',img)
    cv2.imwrite('/home/pi/Pictures/car/edge.jpg', edges)

    imhough = cv2.imread('/home/pi/Pictures/car/houghlines3.jpg')

    lower_red = np.array([0,0,200])
    upper_red = np.array([25,25,255])

    mask = cv2.inRange(imhough, lower_red, upper_red)
    dst = np.zeros(shape=(480,1000))
    red = cv2.bitwise_and(imhough, imhough, dst, mask=mask)

    cv2.imwrite('/home/pi/Pictures/car/red.jpg',red)

    col = Image.open('/home/pi/Pictures/car/red.jpg')
    gray = col.convert('L')

    bw = np.asarray(gray).copy()
```

```
    bw[bw < 50] = 0
    bw[bw >= 50] = 255

    imfile = Image.fromarray(bw)
    imfile.save('/home/pi/Pictures/car/result_bw.jpg')
    bwarray = np.asarray(bw).copy()

    while (bwarray[i][j] == 0):
        countleft += 1
        j -= 1
        if (countleft == 450):
            break
    while (bwarray[i][k] == 0):
        countright += 1
        k += 1
        if (countright == 450):
            break
    diff = countleft - countright
    print(diff)
    if -100 <= diff <= 100:
        print('go straight')
        GPIO.output(11, GPIO.LOW) #move left
        GPIO.output(13, GPIO.LOW) #move right
        GPIO.output(15, GPIO.LOW) #stop
    if diff < -100:
        print('go right')
        GPIO.output(11, GPIO.LOW) #move left
        GPIO.output(13, GPIO.HIGH) #move right
        GPIO.output(15, GPIO.LOW) #stop
    if diff > 100:
        print('go left')
        GPIO.output(11, GPIO.HIGH) #move left
        GPIO.output(13, GPIO.LOW) #move right
            GPIO.output(15, GPIO.LOW) #stop


     def signreader():
         reds = cv2.imread('/home/pi/Pictures/car/road.png')

         boundaries = [
         ([40, 40, 150], [150, 150, 255])
             ]

         for (lower, upper) in boundaries:
             lower = np.array(lower, dtype = "uint8")
             upper = np.array(upper, dtype = "uint8")
             redmask = cv2.inRange(reds, lower, upper)
             output = cv2.bitwise_and(reds, reds, mask = redmask)

             cv2.imwrite('/home/pi/Pictures/car/redresult.png', output)
```

```
117.
118.         kernel = np.ones((7,7),np.float32)/25
119.         dst = cv2.filter2D(output,-1,kernel)
120.
121.         outputgray = cv2.cvtColor(dst, cv2.COLOR_BGR2GRAY)
122.         ret,thresh = cv2.threshold(outputgray,100,225,1)
123.         cv2.imwrite('/home/pi/Pictures/car/thresh.png', thresh)
124.
125.         stopit = cv2.medianBlur(thresh, 1)
126.
127.         contours,h = cv2.findContours(stopit,mode=1,method=2)
128.         for cnt in contours:
129.             approx = cv2.approxPolyDP(cnt,0.01*cv2.arcLength(cnt,True),True)
130.             if len(approx)==8:
131.                 print "Stop sign detected"
132.                 cv2.drawContours(stopit,[cnt],0,255,-1)
133.                 cv2.imwrite('/home/pi/Pictures/car/stopcontour.jpg', stopit)
134.                 GPIO.output(15, GPIO.HIGH) #stop
135.
136.
137. for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
138.     road = frame.array
139.     stopsign = 0
140.
141.     #cv2.imshow('Frame', road)
142.     key = cv2.waitKey(1) & 0xFF
143.
144.     cv2.imwrite('/home/pi/Pictures/car/road.png', road)
145.
146.     linereader()
147.     signreader()
148.
149.     rawCapture.truncate(0)
```