SYSC4805: Final Report

Remote Controlled Car with Autonomous Abilities

Group name: Bitter Sweet
Group members

Michael Pruss 101008219
Michael Skalecki 100969837
Keyan Cassis 101011524
Zubaer Ahmed 101001925

Table of Contents

1.0 Project Charter

1.1 Overall Objective

Advanced Driver Assistance Systems (ADAS) provide assistance to drivers by detecting and avoiding obstacles on the road.[1] The goal of this project is to design and implement an ADAS system on a car that will avoid obstacles and prevent the car from moving out of specified boundaries. The vehicle will be remote controlled through a user while ADAS will run in the background to assist in avoiding collisions.

1.2 Overall Deliverables

The project includes several deliverables that are expected to be functional and complete by the project deadline. Many deliverables represent the ADAS functionality the vehicle will support.

1. The vehicle shall move in any direction and follow a path defined by black line limits on both sides of the path without crossing the black line limits.
2. The vehicle shall autonomously avoid obstacles while staying within the path.
3. The ADAS system shall override user remote control when the vehicle is about to collide with an obstacle or when the vehicle is about to pass over the black line limits.
4. The vehicle shall signal the user through an LED when the battery powering the vehicle has low power remaining.

The vehicle will also support remote controls from the user. The deliverables for vehicle remote controls are:

5. The user shall be able to control the vehicle through a wireless remote control.
6. The user shall be able to control the vehicle in four directions; forward, backwards, right, and left.
7. The user shall be able to turn on and shutdown the vehicle’s motors.
8. The user shall be able to turn on and off the ADAS systems.

1.3 Success Criteria

The success criteria of the Remote Controlled Car with Autonomous Abilities is twofold; one, in its capacity to meet all requirements listed in the Overall Deliverables section, and two, in its robustness and ability to pass all the software and hardware tests. These tests will be divided into two; the module tests and the unit tests. The module tests will consist of testing the functionality of the car’s individual modules as a whole, whereas the unit tests are for testing individual components in themselves. The unit tests will be carried out on each of the hardware peripherals on the car, which are listed in the Requirements section, as well as on their software drivers. **Table 1.3.1** below shows the planned tests for the project:

Table 1.3.1: Success Tests for the Remote Controlled Car

Component	**Test Description**	**Test Steps**	**Expected Outcome**
LED	Hardware unit test for the car's headlights LEDs	● Connect the LED in series with an appropriate resistor to a power source ● Disconnect the LED from the circuit	The LED should turn on when connected to a power source and turn back off when disconnected.
Push Buttons	Hardware unit test for the car's push buttons	● Wire the push button circuit ● Connect the push button to an arduino pin and run the code ● Press the push button	The Arduino serial monitor shows a value change when the push buttons are pressed.
H-Bridge controlled motors	Hardware unit test for each of the car's motors	● Wire the h-bridge controlled motors circuit ● Connect the motors to Arduino pins ● Run the code that activates the motors by applying a voltage on the selected pins	The car's motors spin at the desired speeds and in the right direction when a voltage is placed on the according arduino nano pins
Sonar Sensors	Hardware unit test for each sonar sensor	● Wire the sonar sensor circuit ● Connect the sonar sensor to an Arduino pin ● Run the sonar sensor driver code ● Move the sensor close to and away from different objects	The Arduino serial monitor accurately displays changing values from the sonar sensor as the sensor is placed in front of different objects at varying distances.
Infrared Sensors	Hardware unit test for the infrared sensors	● Wire the infrared sensor circuit ● Connect the infrared sensor to an Arduino pin ● Run the infrared sensor driver code ● Move the sensor over and away from black objects such as tape	The Arduino serial monitor accurately displays changing values from the infrared sensor as the sensor is placed in front of different colors.
Car	Module test for the object detection and avoidance system of the car	● Wire the sonar sensors, the piezo, and the motors on the car ● Move the car towards obstacles	The vehicle beeps when approaching objects in its course and avoids them by turning around them.
Car	Module test to ensure the car does not go beyond the obstacle course limits	● Wire the infrared sensors and the motors to the car ● Place the car on course to pass the black delimiting tape ● Let the car move forward	The vehicle does not go past the black limit lines. When approaching them, it turns inwards back to the course
Car	Module test for the vehicle's wireless controller	● Wire the motors, and the wireless controller receiver to the car ● Press the buttons on the controller to speed up, turn, and reverse the car	The vehicle's speed and direction change according to user input from the wireless controller.
Car	Module test for the car's ADAS system	● Wire the infrared sensors, sonar sensors, piezo, motors, and wireless controller receiver to the car	The ADAS systems overrides user remote control input when the vehicle is about to collide with an

		● Using the wireless controller, steer the car towards obstacles or towards the course limits	obstacle or when the vehicle is about to pass over the black line barriers. The vehicle then avoids the collision or turns back into the course.

2.0 Scope

2.1 Requirements

This project consists of many hardware components. The base for the vehicle will be the provided 3D printed car with motor assembly. This car will be operated by an Arduino Uno microcontroller with four 1.5V AA batteries. Components necessary for our system are:

- Push button for turning on and off the motors
- 2 infrared sensors to detect the black curb lines
- H-bridge controller for controlling the motors
- 3 sonar sensors for detecting obstacles
- LED for engine light
- Remote RF controller transmitter with 4 directions
- RF receiver with 4 directional inputs into Arduino

The vehicle will be manually driven by a user in the forward, reverse, right, and left directions. As the vehicle is driving, ADAS will monitor the infrared sensors as well as the sonar sensors. If the infrared sensors detect the black lines then the vehicle will automatically be diverted back onto the road, and user input will be overridden. If the sonar sensors detect obstacles along the path of the road in the form of walls, vehicles, and other obstructions, then ADAS will take over control of the vehicle and will avoid collisions with the obstructions.

In addition to the movement of the vehicle, the user will have the ability to manually, shut off and turn on the motors (engine killswitch), turn on and off ADAS.

2.2 Work Breakdown Structure

The work for the Remote Controlled Car with Autonomous Abilities project has been divided into four parts; Implementation, Progress Revision, Testing, and Closeout. **Figure 2.2.1** below shows the full work breakdown structure diagram.

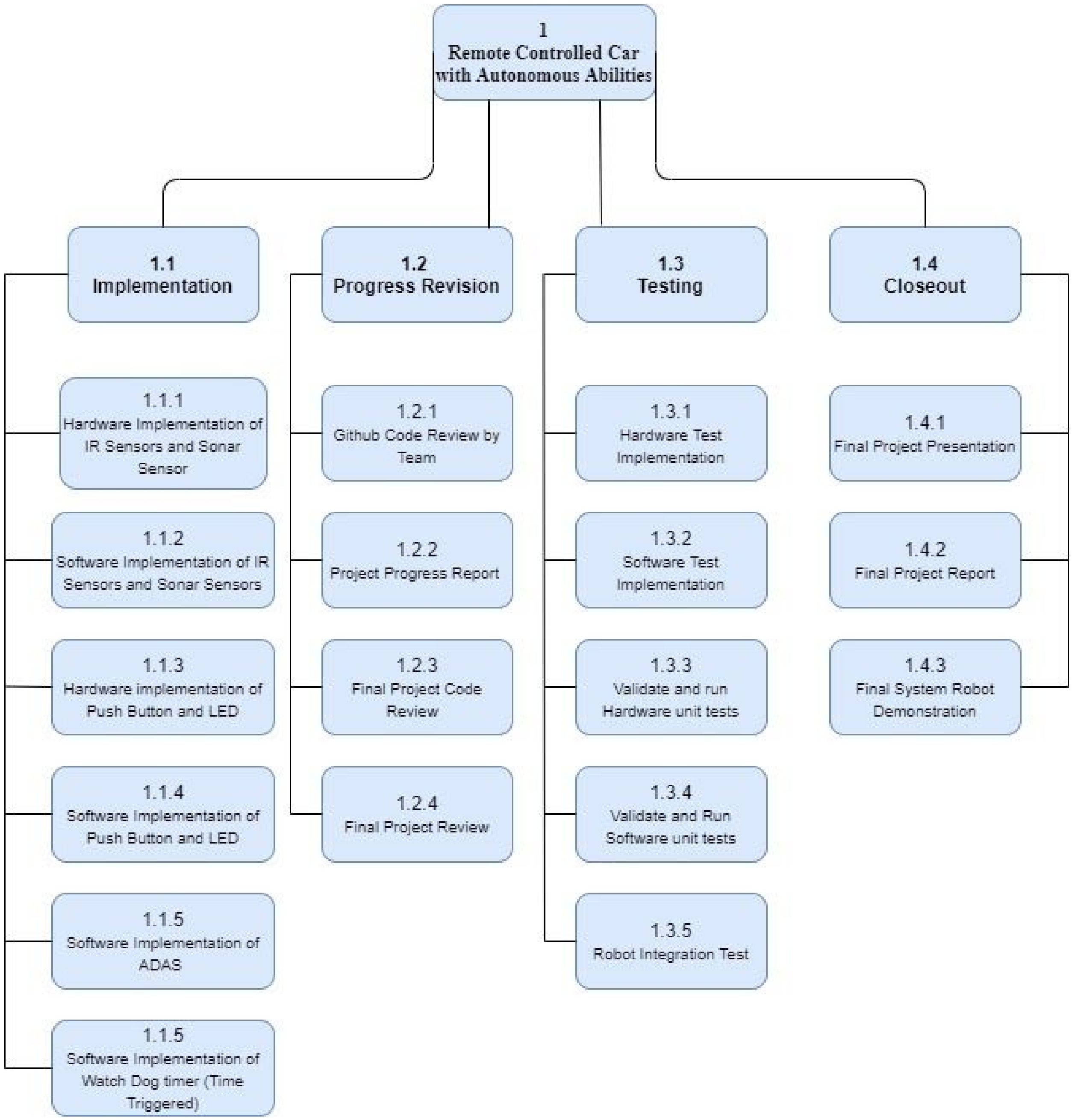


Figure 2.2.1: Project Work Breakdown Structure Diagram

3.0 Schedule

3.1 Schedule Network Diagram

The activities required in the Remote Controlled Car with Autonomous Abilities project can be classified as software/hardware and documentation. **Figure 3.1.1** below shows a graphical representation of the logical dependencies, among the project schedule activities.

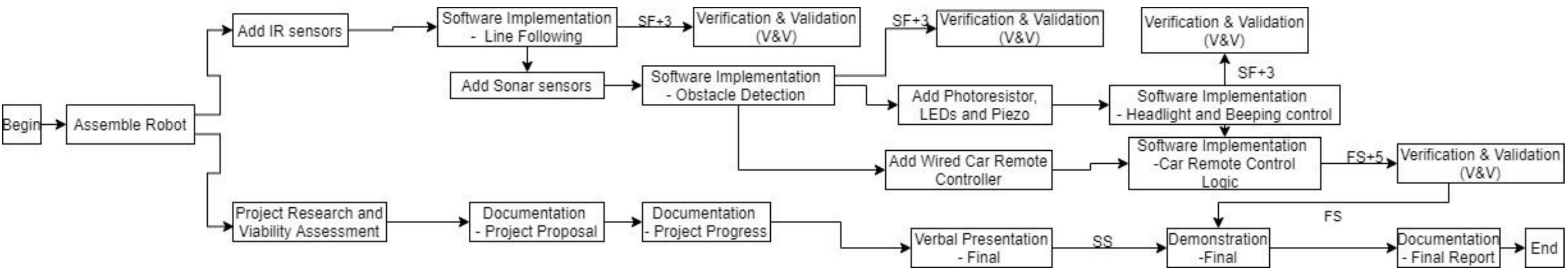


Figure 3.1.1: Project Schedule Network Diagram

3.2 Gantt chart

The workflow for the group has been broken down into Opportunity assessment, Implementation, and testing. Each task has been defined and assigned to a team member. **Figure 3.2.1** below represents a Gantt chart that outlines the workflow for the project.

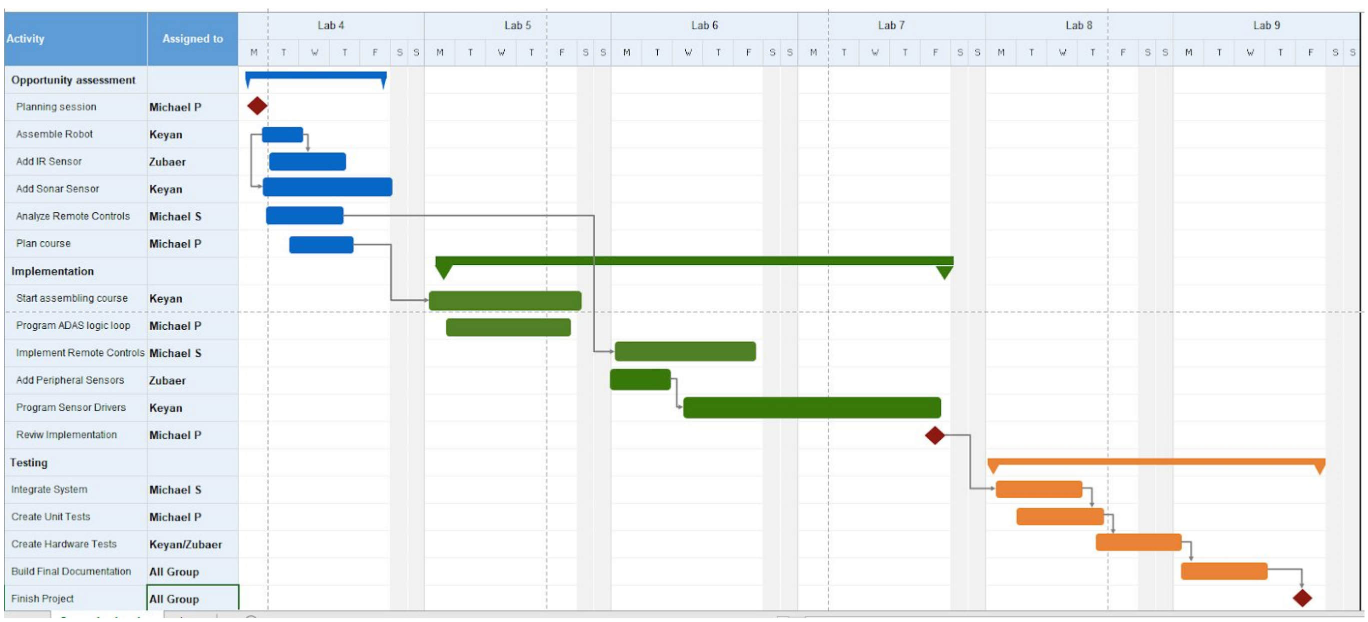


Figure 3.2.1: Group Gantt chart for the project.

4.0 Human Resources

4.1 Responsibility Assignment Matrix

Each member of the Remote Controlled Car with Autonomous Abilities project has been assigned responsibilities. **Table 4.1.1** showcases the responsibility breakdown for the project, where R is responsible, A is approver, C is consult+ed, and I is informed.

Table 4.1.1: Responsibility Assignment Matrix for Group

Task	Michael S.	Michael P.	Keyan	Zubaer
Robot Assembly	C	A	R	A
IR sensor Implementation	A	A	C	R
Sonar Sensor Implementation	C	A	R	A
Remote Control Implementation	R	C	A	A
Design testing course	A	R	A	C
Course embly	A	C	R	A
Manual driving operation	R	C	A	A
ADAS driving operation	C	R	A	A
Peripheral sensors	C	C	C	R
Review implementation	I	R	I	I
Integrate System	R	C	C	C
Develop unit tests	C	R	C	C
Develop hardware tests	I	I	R	R
Write final documentation	R	R	R	R
Complete project	R	R	R	R

R = Responsible, A = Approver, C = Consulted, I = Informed

5.0 Progress Report Deliverables

5.1 Individual Work on Github

Table 5.1.1 below showcases the features each team member worked and committed onto the Github repository.

Table 5.1.1: Individual Work Done on Github

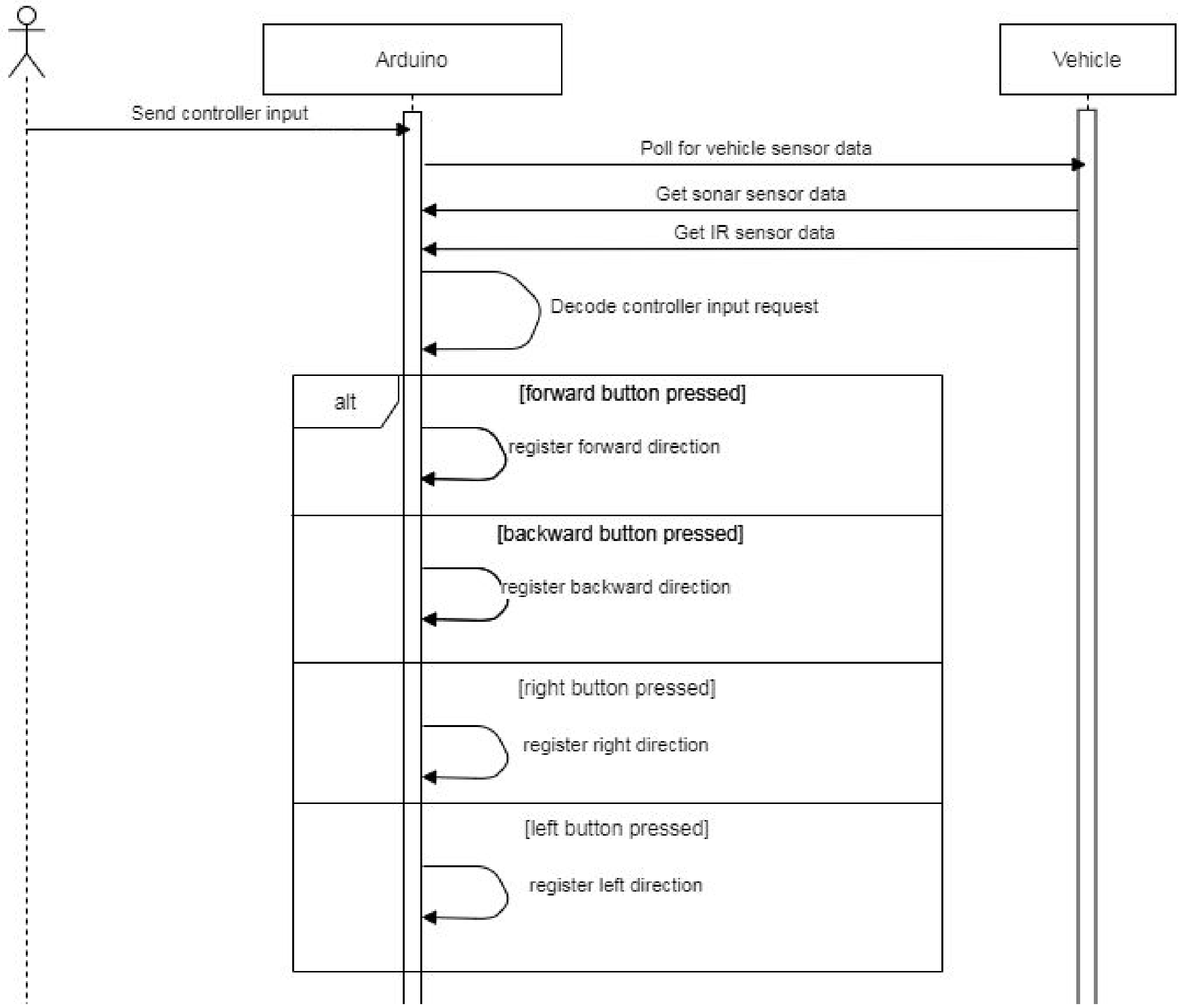
Group Member	Feature Worked On
Michael Pruss	● WatchDog Timer ● Added SONAR sensors
Michael Skalecki	● Added remote controls for vehicle ● Added vehicle movement
Keyan Cassis	● Added car on/off switch ● Added ADAS functionality
Zubaer Ahmed	● Added IR sensor ● Added logic to follow black line

5.2 Time Triggered Implementation

The design for the Remote Control Car with Autonomous Abilities will be timed triggered. To achieve safe and reliable autonomous abilities, several sensors must be polled periodically to learn the surrounding environment. Using a polling approach and reading each sensor value periodically with a fast refresh rate will result in faster and sooner obstacle detection. The sensors are polled every 20ms and are constantly polled in the main loop.

The watchdog for the system will be event triggered and will fire an event when the system is in an irrecoverable state.

5.3 Sequence Diagram



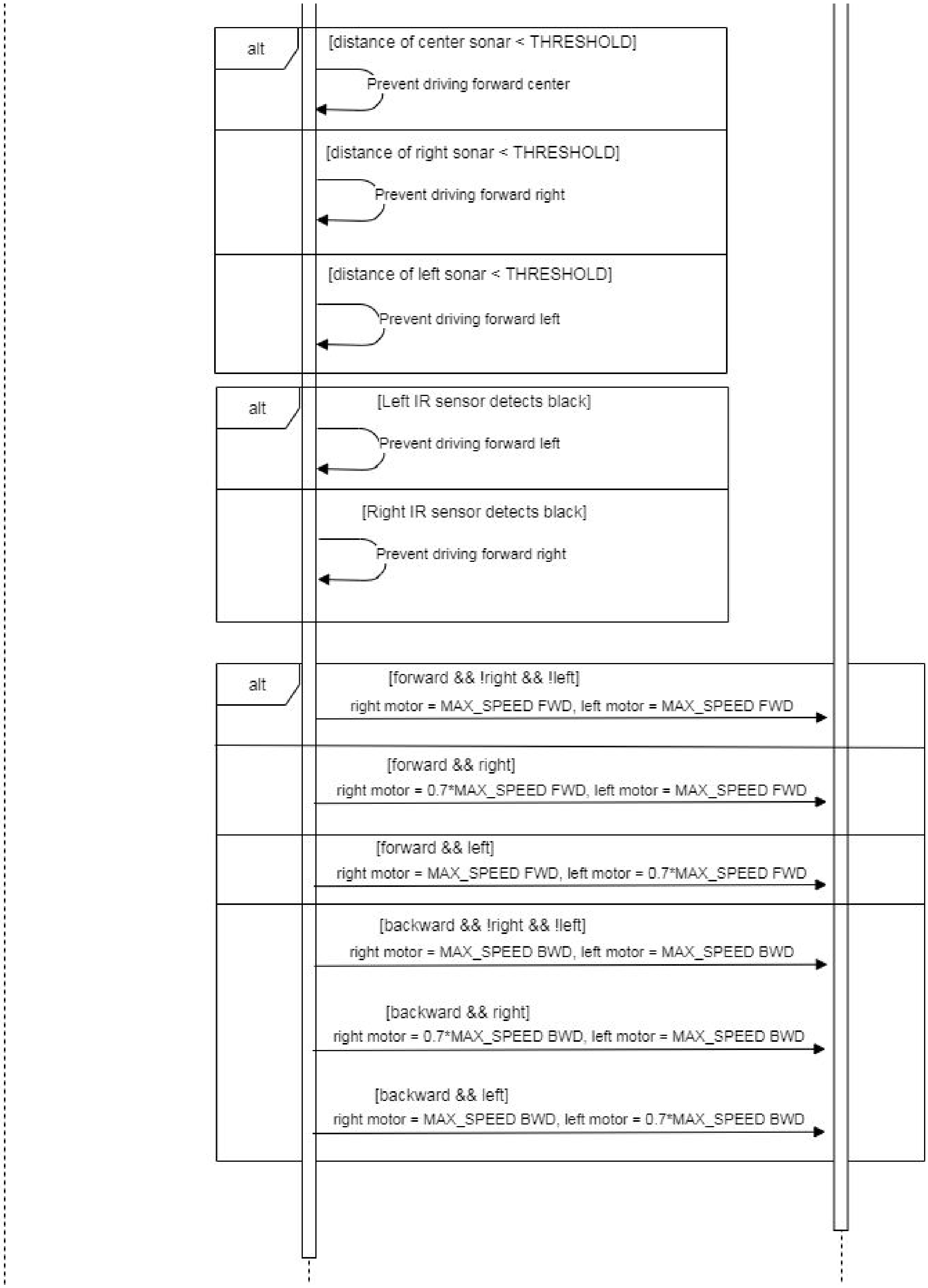


Figure 5.3.1: Sequence diagram for vehicle movement

Referring to Figure 5.3.1, the vehicle is primarily controlled by the user who provides input for the direction in which they want to travel. The polling for the controller input as well as the sensor data is time triggered and the entire functionality is looped. The Arduino will decode the inputs and will retrieve sensor data from the vehicle. The sensor data will be parsed and used for validation of the road environment. For the sonar data, the right, center, and left sonars will be tested against threshold values to determine if the vehicle can travel safely in the requested directions. If the distances are less than the thresholds, then the vehicle will be prevented from travelling in the respective directions. For the IR sensor, the left and right sensors will check to see if they are over the black road border. If they are, then the vehicle will be prevented from travelling in the respective directions.

5.4 Circuit Diagram

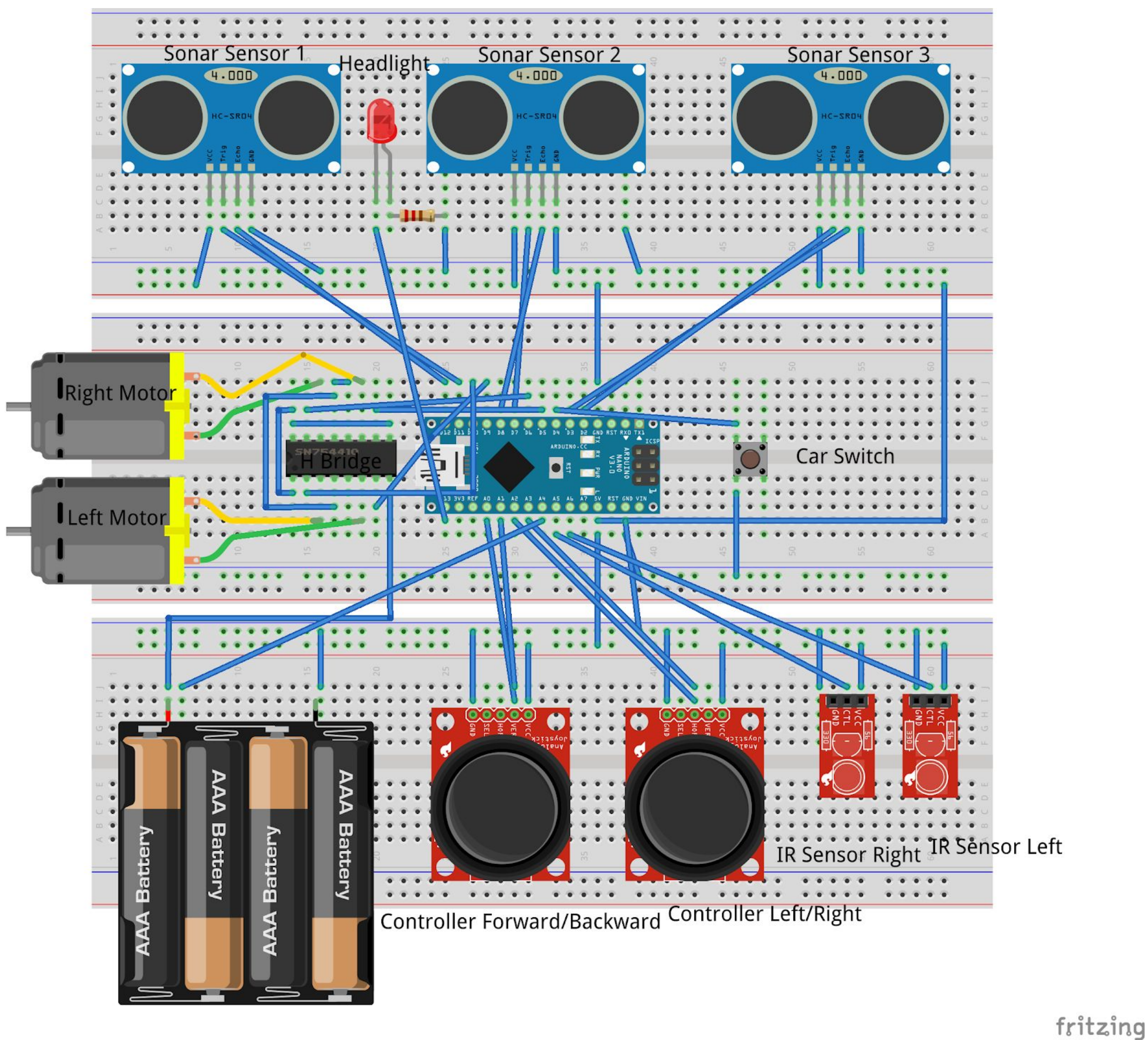


Figure 5.4.1: System Circuit Diagram of robot

Figure 5.4.1 above gives a Circuit diagram of the robot that has been proposed for this project. The circuit diagram has been created through Fritzing. The sensors that are seen in the circuit diagram include the two IR sensors (left and right), the two motors (left and right) and the three sonar sensors (left, right and center). Additionally the circuit diagram also shows the wiring required for the LED and the push button which will be the car switch.

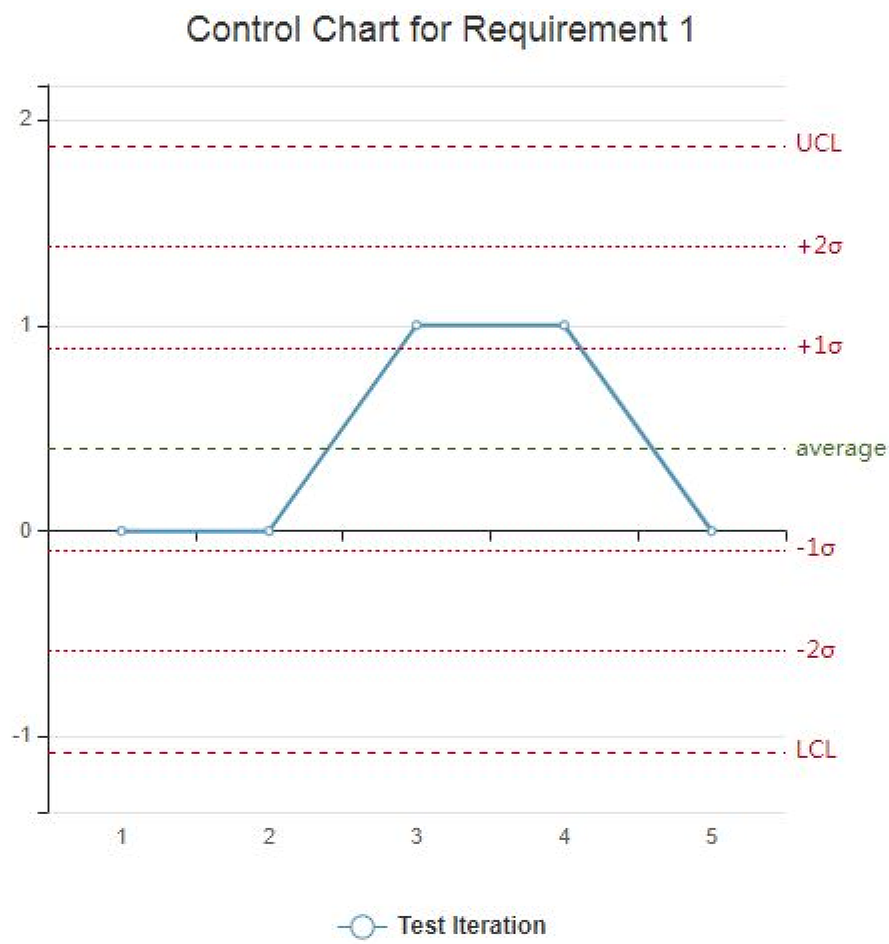
5.5 Watchdog Description

The watchdog timer is enabled in the setup procedure of the robot by default. The timer indicates when the program has stalled by printing a watchdog timer message that it has expired. Following a watchdog timer message the program is reset. The watchdog timer is enabled through a timer interrupt using the TCCR registers. When the timer compares interrupt fires by comparing the maximum time set and current time executing an instruction, the watchdog resets the code. Through this functionality the robot is able to properly implement the watchdog timer in the system.

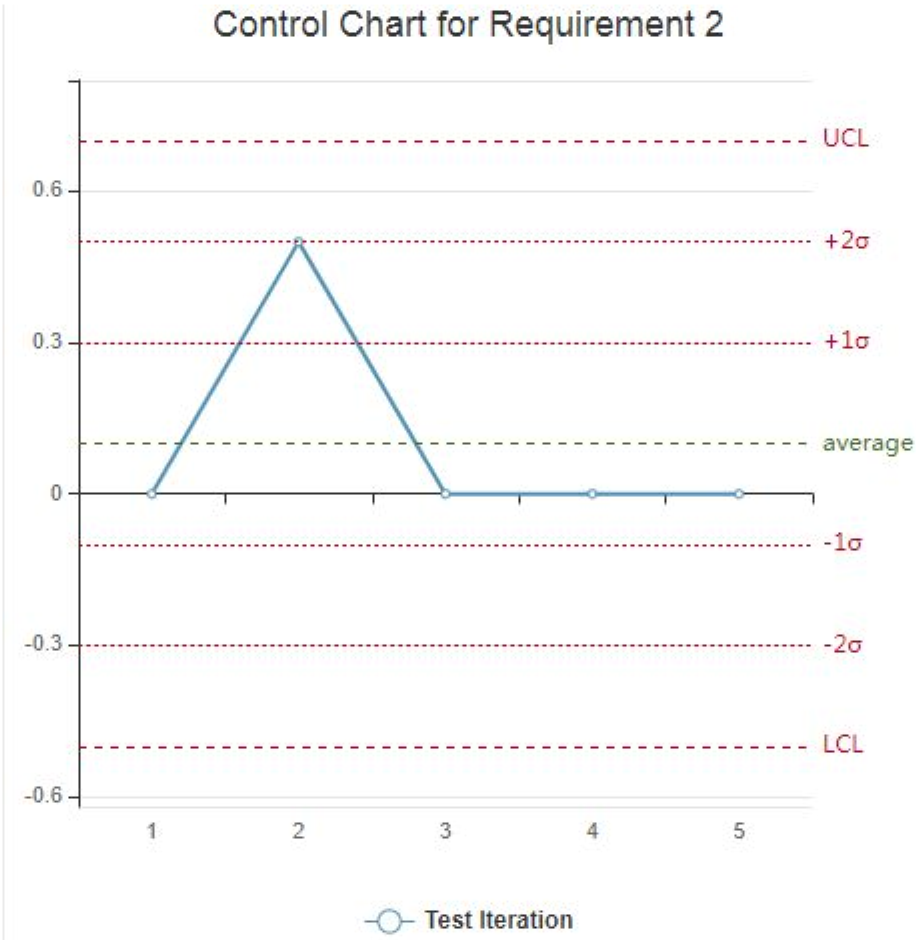
6.0 Final Report Deliverables

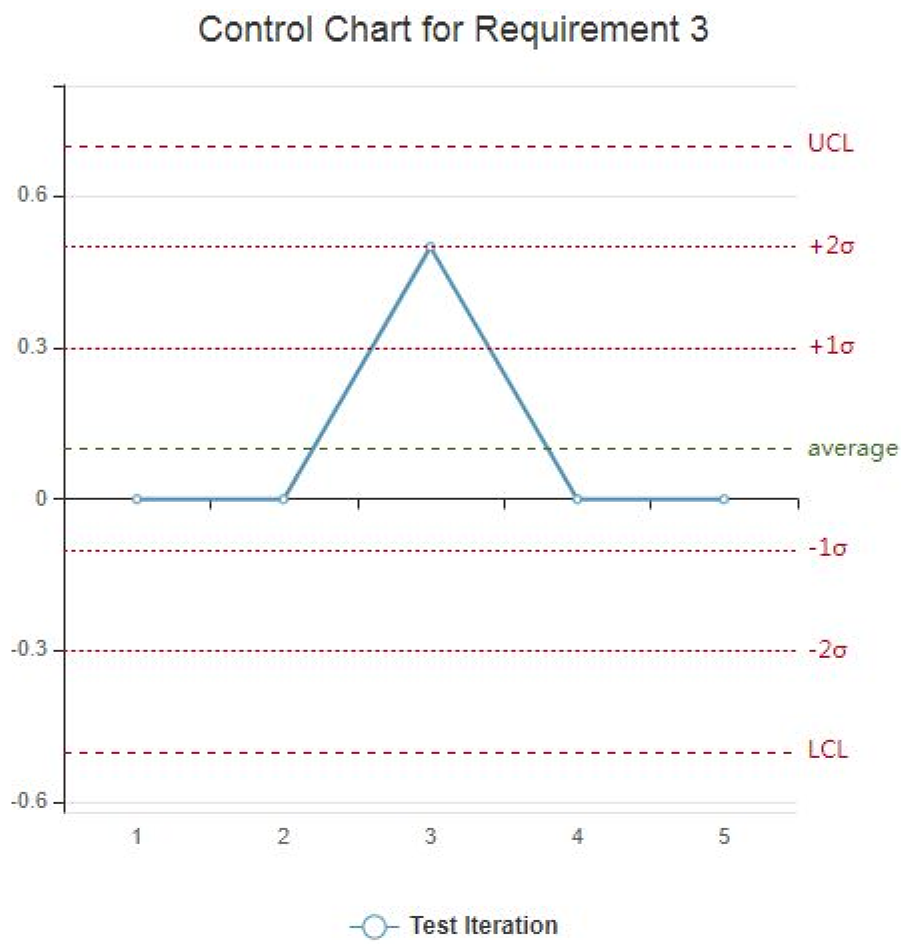
6.1 Requirements Control Charts

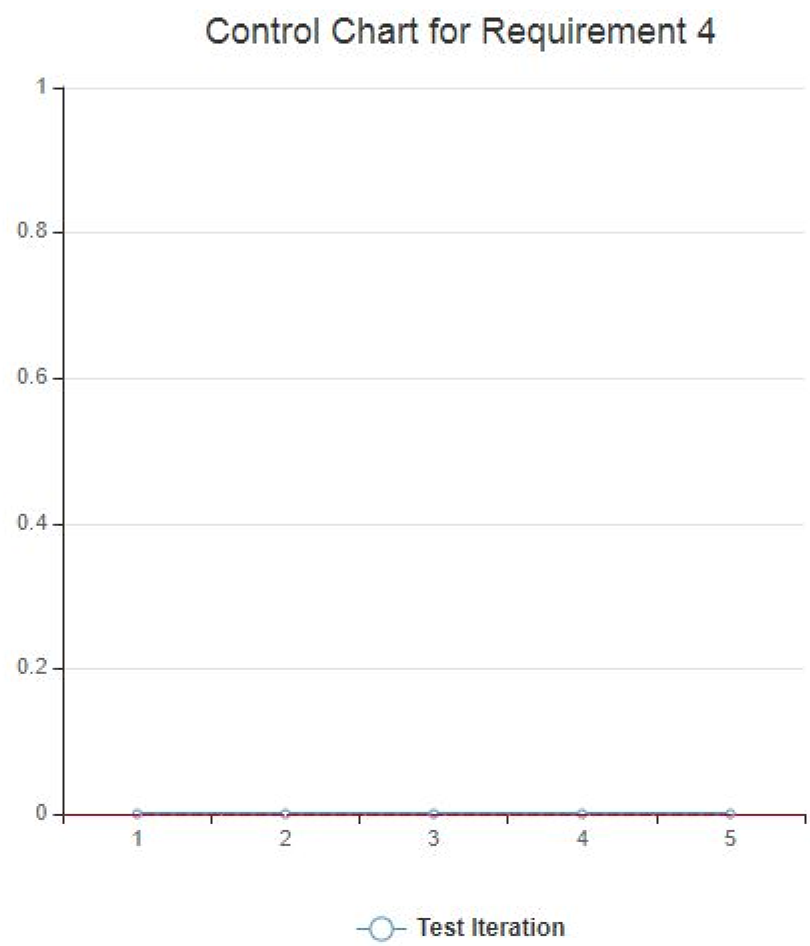
Control charts were used to test and evaluate the vehicle's ability to meet the project requirements listed in section 1.2. Each deliverable was tested five times and each test iteration was assigned a value between 0, 0.5, and 1. A value of 0 means that the test passed without error, 0.5 means that the test passed however there is room for improvement in the vehicle's behaviour, and 1 means that the test failed, i.e. the vehicle did not meet the requirement.



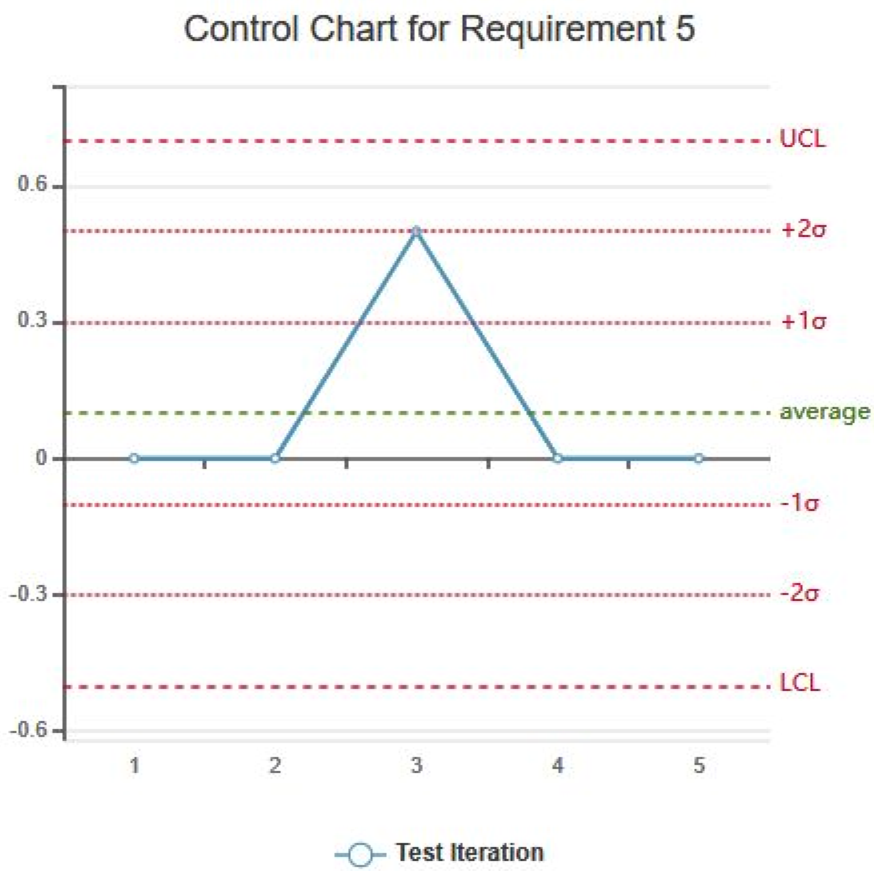
Note that two of the test iterations failed for requirement 1. This is due to the vehicle's left infrared sensor malfunctioning during development. The other successful cases occurred when the vehicle's ability to remain within the black line limits were tested on the right side.


Control Chart for Requirement 2
UCL
+2σ
+1σ
average
-1σ
-2σ
LCL
0.6
0.3
0
-0.3
-0.6
1
2
3
4
5
Test Iteration

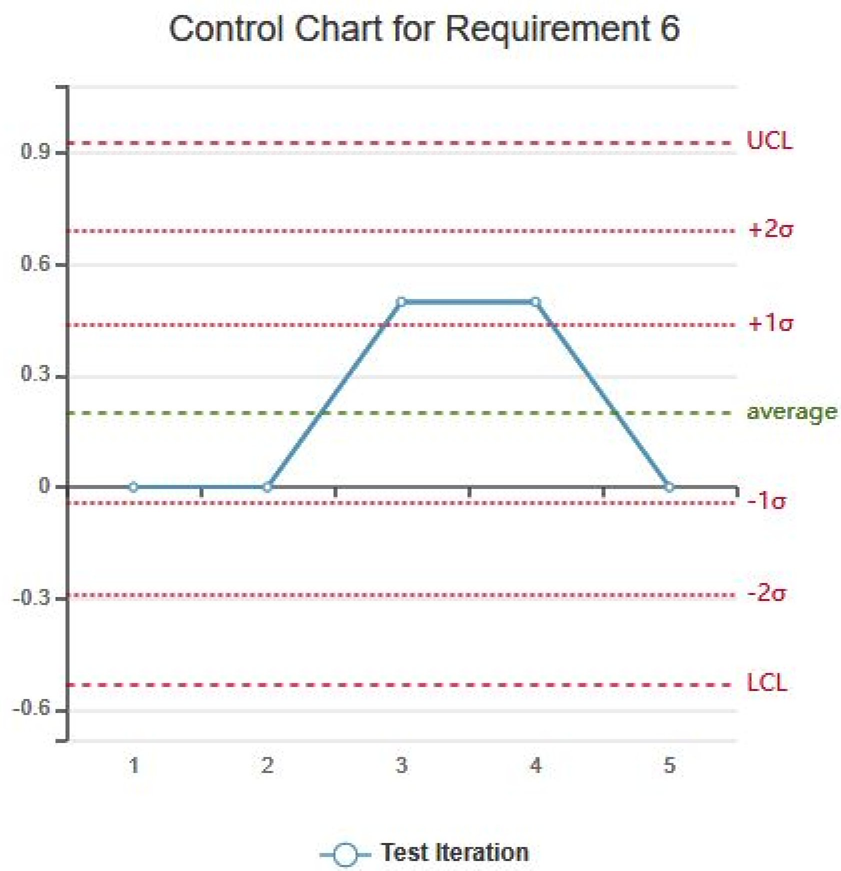

Control Chart for Requirement 3
0.6
0.3
0
-0.3
-0.6
1
2
3
4
5
UCL
+2σ
+1σ
average
-1σ
-2σ
LCL
Test Iteration



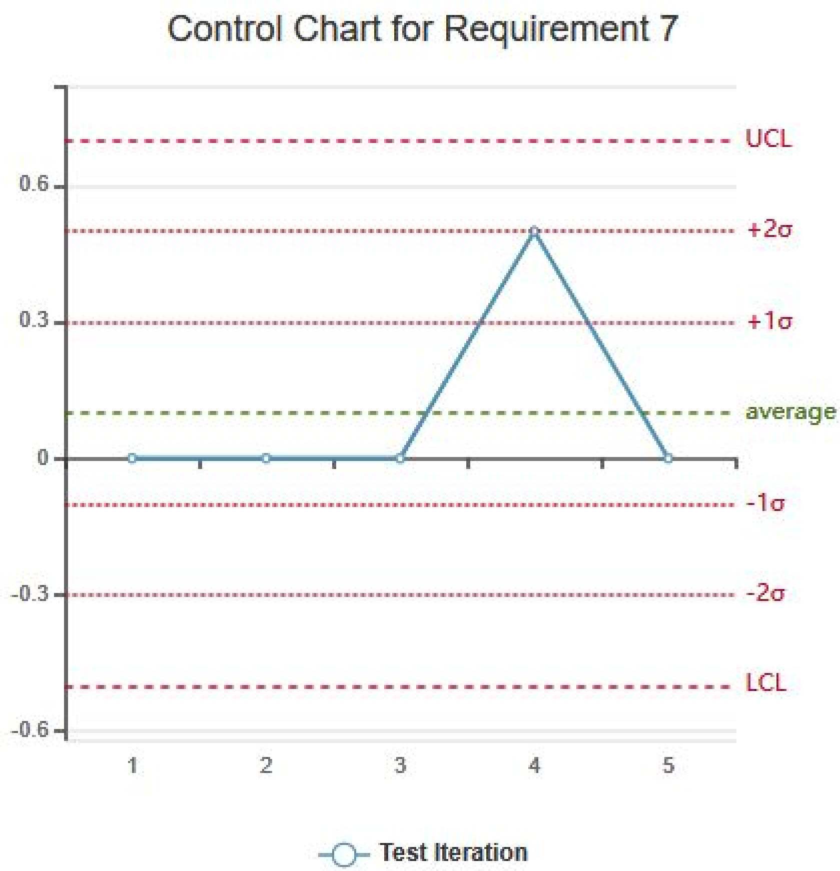
Note that in the tests for requirement 4, the vehicle passed each test iteration and therefore there is no upper control limit to be defined.



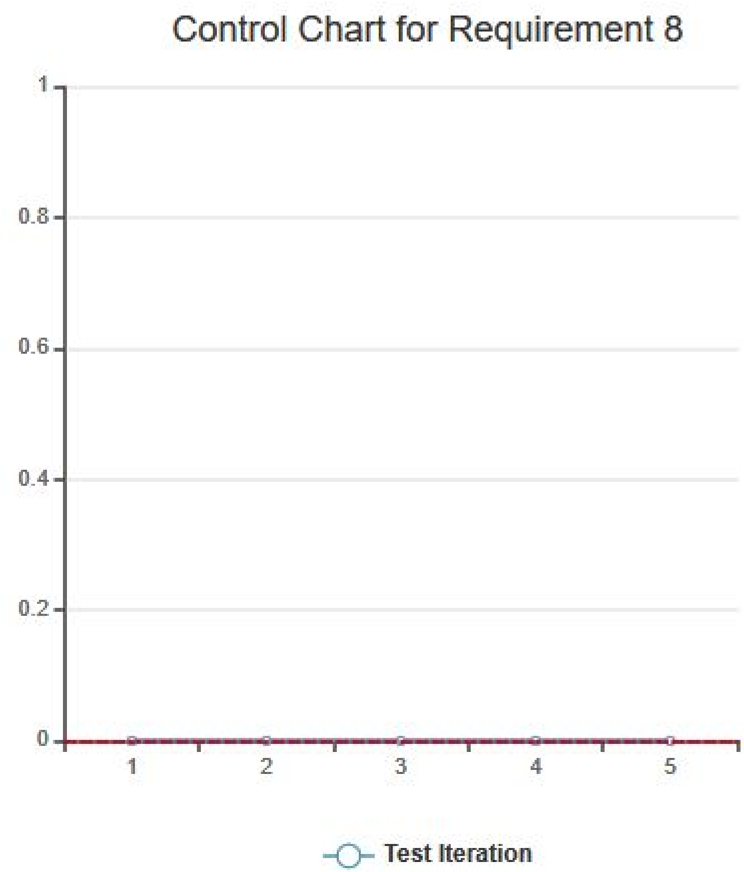
The average determined from the test iterations for Requirement 5 was 0.1 which is deemed an appropriate pass.



During test iterations 3 and 4 the remote control functionality for direction control was assessed, specifically this was left and backwards. During these test iterations they were deemed as a pass, the 0.5 score is based upon a longer response time during the input of the direction control.



During the test iteration in Requirement 7, remote control ability to turn the motors on and off was assessed.



Note that in the tests for requirement 8, the vehicle passed each test iteration and therefore there is no upper control limit to be defined.

6.2 Individual Contribution

The following table details individual contributions on the project and for the final report.

Table 5.1.1: Individual Contributions to Project and Report

Group Member	Project Contribution	Report Contribution
Michael Pruss	• WatchDog Timer • Added SONAR sensors • Worked on testing course	1.1, 1.2, 3.2, 5.1, 6.2
Michael Skalecki	• Added remote controls for vehicle • Added vehicle movement • Worked on testing course	2.1, 4.1, 5.3

	• Tested vehicle	
Keyan Cassis	• Added car on/off switch • Added ADAS functionality • Worked on testing course	1.3, 3.1, 5.4, 6.1
Zubaer Ahmed	• Added IR sensor • Added logic to follow black line • Worked on testing course	2.2, 5.2, 6.1

7.0 References

[1] L. Li, D. Wen, N. Zheng and L. Shen, "Cognitive Cars: A New Frontier for ADAS Research," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 395-407, March 2012.T