Gizmo: The Next Frontier in Autonomous Robotics - Building on Falcon's Legacy

Milestone 3

BIA Capstone Project

MGMT – 6134 – 23F

Prepared For:

Professor Mark Bueno

Professor Mona Abou Taka

Prepared By: **PC51 Technologies**

Wilberto Mejia - 1138669 (S32)

Apple Dela Cruz - 1080609 (S31)

Jhay-Ar Aguilar - 1105965 (S32)

Dean Paul Martin - 1091458 (S32)

Anjeline Sayoc - 1104192 (S32)

Roben Woo - 1080811 (S31)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. PROJECT SUMMARY

1.1 INTRODUCTION

Team Digi Destined initiated the project *Going Into Automation: Building an Arduino-based Intelligent Robot To Avoid Obstacles*. The project focused on developing an autonomous four-wheeled robot using the ESP32 wireless module and the Arduino integrated development environment which enables the robot car to make decisions through perception algorithms as part of the BIA Capstone Project (Summer 2023). The scope of the project was to create Falcon, a four-wheel car, with ESP 32 controller that has the following capabilities: line tracking and obstacle detection and avoidance. Due to time constraints, Team Digi Destined only completed the Falcon with line tracking and obstacle detection abilities.

The project opens the door to a new area of knowledge. Integrated Systems and Micro Controllers show the path to AI that can further lead to wide opportunities to the society in fields such as, health, security, and education. The opportunity of getting the experience related to robotics inspired the current team, PC51 Technologies, to improve the robot functionalities and meet current needs in society while Fanshawe provides the steps into this wide and important field. Moving forward, PC51 Technologies aims to expand Falcon's capabilities to include obstacle detection and avoidance, while also introducing a new feature for light tracing. These enhancements will introduce **Gizmo**.

PC51 Technologies embarks on a journey where Falcon's legacy merges seamlessly with Gizmo's potential, creating a bridge between the past and the future. This project is illuminated by the promise of enhanced autonomy - offering solutions to challenges that society may not fully comprehend at this time. The transition from Falcon to Gizmo symbolizes PC51 Technologies' commitment to pushing the boundaries of what this intelligent robot can achieve. This project will also showcase the team's collective effort to keep pace with the rapid advancements in the field and to contribute to the cutting-edge innovations that shape society.

Gizmo, with its newfound capabilities, stands as a testament to the team's dedication to progress. This exploration in the field of robotics represents a significant stride forward. Fanshawe, as an institution, proudly supports this project as it continues to take bold steps into a wide and important field that will benefit the society.

1.2 PROJECT MILESTONE SUMMARY

High-Level Milestone Timeline				
Milestone	**Description**	**Start Date**	**Status**	**Completion Date**
1	Inception	11 September 2023	Completed	27 September 2023
2	Analysis of Deliverables	28 September 2023	Completed	18 October 2023
3	Design of Deliverables	19 October 2023	Completed	8 November 2023
4	Construction, Results, and Discussion of Deliverables		Not Started	TBA
Final Report	Final Report and Evaluations		Not Started	TBA

TABLE 1: PROJECT MILESTONE SUMMARY

Summary Project Status	
Project Start Date	September 11, 2023
Estimated Project End Date	December 8, 2023
Impacted Process	
Potential Financial Impact	

TABLE 2: PROJECT STATUS SUMMARY

Milestone Event Table			
Milestone	**Status**	**Due Date**	**Expected Completion Date**
Milestone 1	Completed	September 27, 2023	September 27, 2023
Milestone 2	Completed	October 18, 2023	October 18, 2023
Milestone 3	Completed	November 8, 2023	November 8, 2023
Milestone 4	To Be Done	November 29, 2023	
Final Milestone	To Be Done	December 8, 2023	

TABLE 3: PROJECT MILESTONE SCHEDULE

2. SYSTEM ARCHITECTURE

2.1 HARDWARE COMPONENTS

- **Robot Car Body (ESP32 Car):** This is the physical platform that integrates all the hardware and tool components and allows the car to move and use all its functions.
- **ESP32-WROVER**: The ESP32-WROVER microcontroller module serves as the main brain (CPU) of the car, handling movements, function controls, and internal and external communication.
- **Motors**: These motors control the movement of the car wheels and their direction.
- **Servos**: The servo is used for controlling specific actions like steering or movement of sensors like the ultrasonic sensors.
- **LED Lights (WS2812):** These LEDs can be used for visual indicators and serve aesthetic effects.
- **LED Matrix**: The LED matrix can display text, light patterns, or sensor data.
- **Battery**: Stores electricity to power the entire car system.
- **Camera:** The camera captures images or video for various applications, such as computer vision.
- **IR Receiver:** The IR receiver component receives infrared signals from a remote control or other IR s, allowing the car to receive commands or data via external infrared communication.
- **Sensors**:
  - **Ultrasonic Wave Sensors**: Used for obstacle detection and distance measurement around the car robot.
  - **Photoresistors**: Light sensors that can be used for following and detecting light.
  - **Line Tracking Sensors (PCF8574**): Used for tracking line routes on the ground or following a specific path.

2.2 SOFTWARE COMPONENTS:

- **Arduino IDE**: This is the development environment for programming the ESP32 Car and controlling the hardware components' functions and behavior.
- **Sketches**: Custom software developed using Arduino IDE to control the motors, servo, and LEDs, and interact with sensors.

2.3 COMMUNICATION MODULES:

- **IR (Infrared**): Used for communication between the car and a remote control or another device via infrared.

- **Wi-Fi**: Allows the car to connect to a local Wi-Fi network for remote control and data exchange.

- **USB:** Provides a versatile communication interface for programming, data transfer, debugging, and user interaction between the robot and external devices such as computers or microcontrollers.

2.4 USER INTERFACE:

- **Freenove Application**: A mobile or desktop application developed for controlling and monitoring the car. It can include features like driving controls, sensor data visualization, and more.
- **Executable File (Main.exe):** For desktop application (Windows).
- **Python Script (main.py):** For desktop application (cross-platform).
- **Freenove Mobile Application:** For mobile application.

2.5 SYSTEM ARCHITECTURE DESCRIPTION

1. The ESP32-WROVER serves as the central controller, running the custom firmware developed in Arduino IDE.
2. The sketches control the motors, servo, and LEDs, and communicate with the various sensors to gather data.
3. Sensor data (from ultrasonic sensors, photoresistors, and line tracking sensors) is processed within the ESP32 and can be used for decision-making and control.
4. The ESP32-WROVER communicates with the Freenove Application using Wi-Fi, allowing users to control the car remotely and receive real-time sensor data.
5. The Freenove Application can be installed on a mobile device or a computer (Windows, macOS, or Linux). It offers a user-friendly interface for driving the car, visualizing sensor data, and sending commands.
6. In addition to Wi-Fi communication, the car can also communicate using IR with a remote control or another IR-enabled device for basic control.
7. The LED lights (WS2812) and LED matrix can display emotions, sensor data, or visual effects as needed.

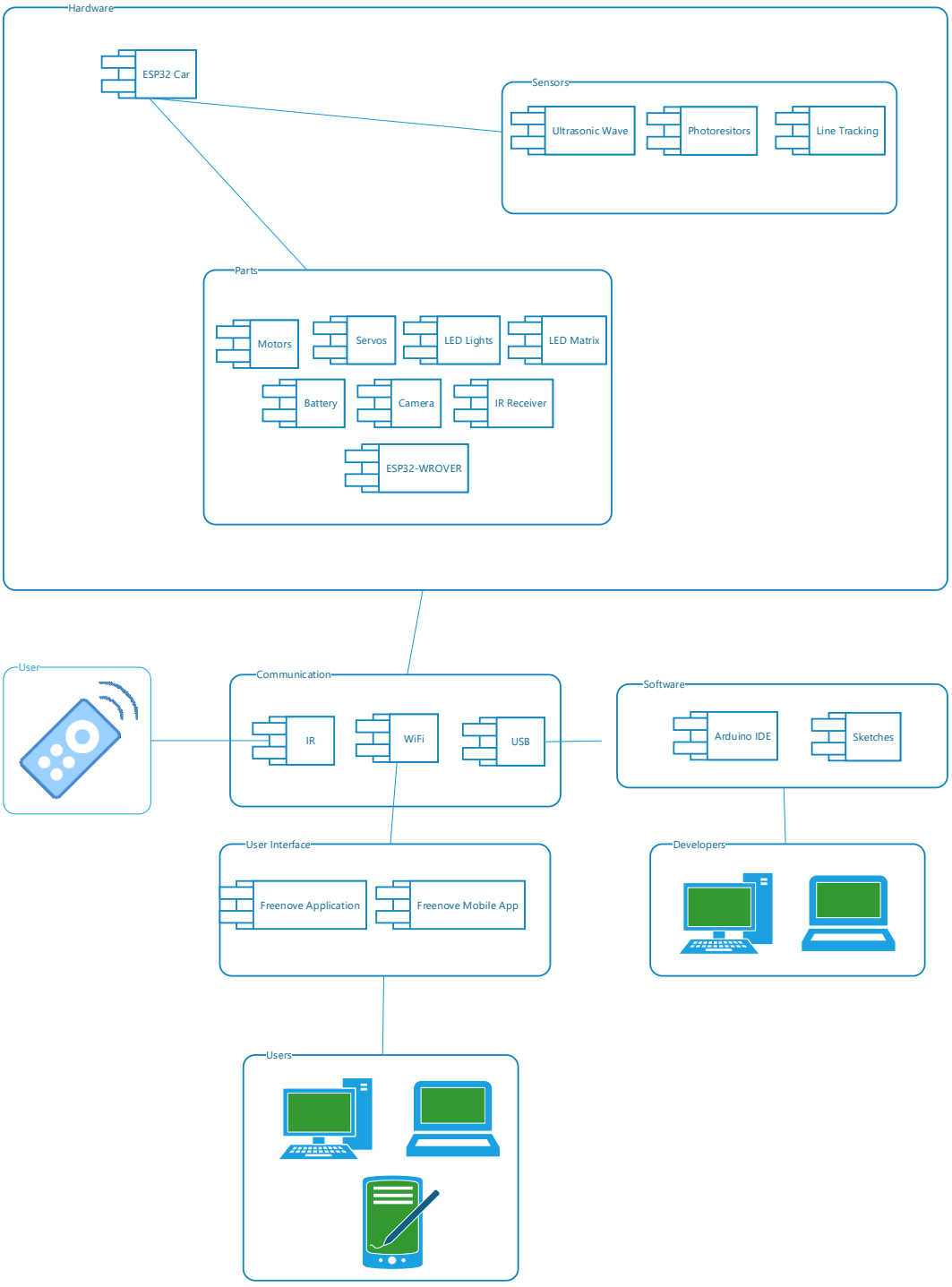


FIGURE 1: SYSTEM ARCHITECTURE

3. PROTOTYPE

User Story ID	As a <type of user>	I want <to perform some task>	so that I can <achieve some goal>
1	Gizmo Operator	Gizmo to autonomously detect and avoid obstacles in its path	Gizmo can continue its route following tasks efficiently and without human intervention.
2	Gizmo Operator	Gizmo to be capable of detecting and tracking a moving light source.	Gizmo can actively follow the light source by adjusting its movements, including moving forward, backward, and turning sideways.
3	Gizmo Operator	To see continuous improvements in the existing features of Gizmo.	I can make the most of the robot's capabilities and enjoy a better user experience.

TABLE 4: USER STORIES

See Appendix 1. * Prototypes for User Stories 2 and 3 will be available on Milestone 3.

4. API FUNCTIONS USED

1. **IR (Infrared):**
    - Infrared communication involves transmitting and receiving data using infrared light. It's similar to how remote controls for TVs work.
    - While not explicitly referred to as an API, it involves communication protocols for encoding and decoding data using infrared signals.
    - This communication may require specific libraries or protocols to define how data is modulated onto the IR signal and how the receiver should decode and interpret this data.
2. **Wi-Fi:**
    - Wi-Fi allows the car to connect to a local wireless network, enabling remote control and data exchange.
    - Wi-Fi communication involves well-defined protocols and standards (e.g., TCP/IP) for network communication.
    - To interact with Wi-Fi networks, the software may use Wi-Fi libraries or APIs that provide functions for connecting to networks, transmitting data, and managing network configurations.
3. **USB:**
    - USB (Universal Serial Bus) is a common hardware interface for connecting devices to computers or other microcontrollers.
    - Although not explicitly mentioned as an API, using USB typically involves USB communication protocols and potentially software libraries.

- Libraries or APIs can manage USB connections, allowing data transfer, programming the microcontroller, and debugging.

4. **Freenove Application (Desktop and Mobile):**
    - The Freenove Application provides a user interface for controlling and monitoring the car.
    - APIs for handling user inputs might include libraries for GUI development, allowing the creation of buttons, sliders, and other interactive elements.
    - Sensor data visualization would involve using charting or visualization libraries to display data graphically.
    - Communication with the ESP32-WROVER over Wi-Fi may involve using networking libraries or protocols to send and receive commands and data between the application and the car.

5. **Executable File (Main.exe) and Python Script (main.py):**
    - Main.exe and main.py are components of the desktop application.
    - Communication with the ESP32-WROVER over Wi-Fi or other interfaces may require network communication libraries or APIs.
    - Python may use libraries like **socket** for network communication.

6. **Arduino IDE:**
    - While not explicitly an API, the Arduino IDE serves as the development environment for programming the ESP32 Car.
    - The IDE provides a set of libraries and APIs for working with hardware components, such as sensors, motors, servos, and LEDs.
    - Developers can use these libraries to control the hardware and define the behavior of the car.

5. TEST CASES

See Appendix 2.

6. DEPLOYMENT DIAGRAM

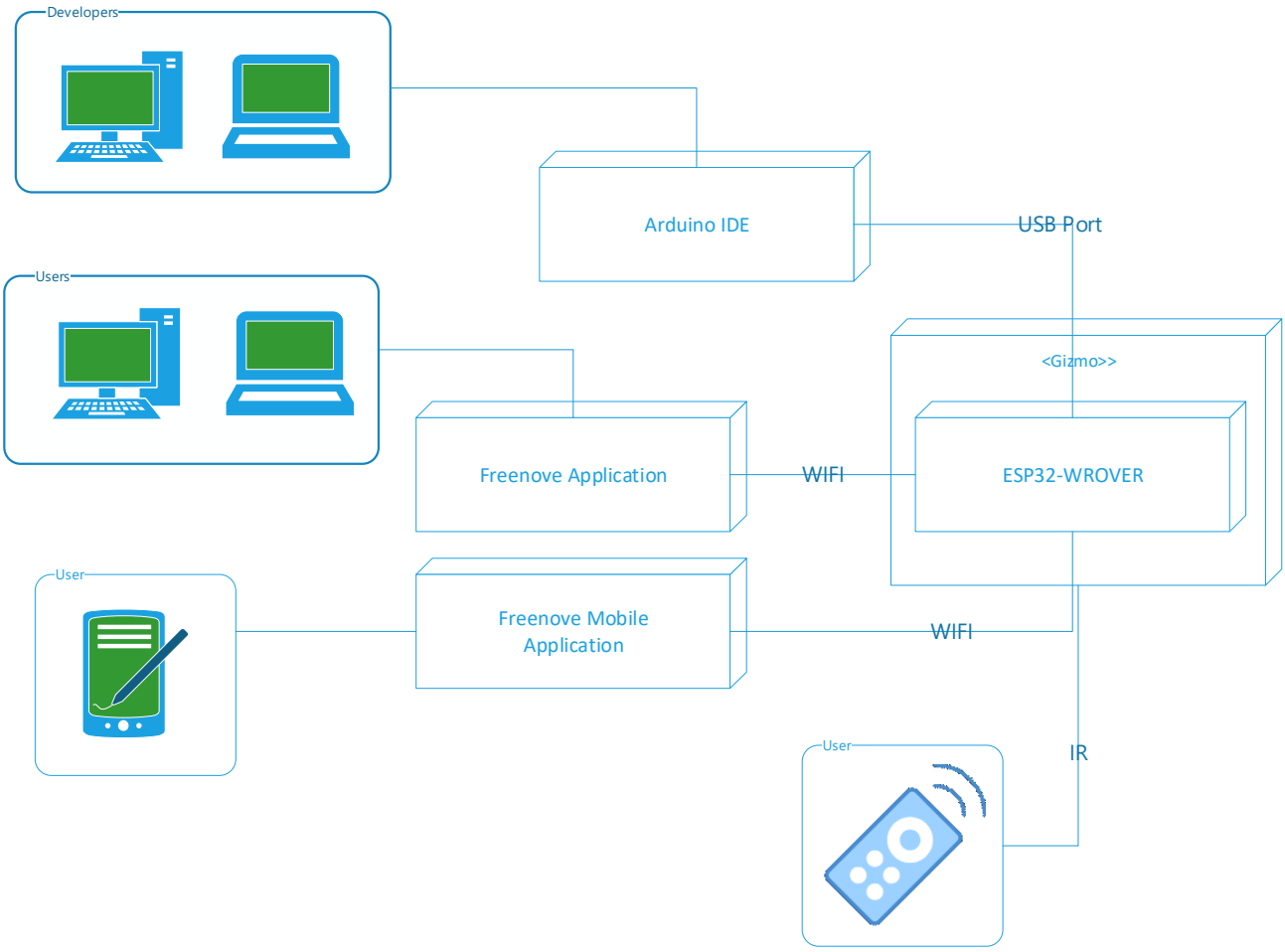


FIGURE 2: DEPLOYMENT DIAGRAM

- Gizmo ESP32 Wrover is the physical hardware where the sketch is deployed.
- The developer's computer or laptop can communicate with Gizmo through the USB port using the Arduino IDE for sketch deployment.
- The Freenove Application can be installed on a computer/laptop to control Gizmo via Wi-Fi.
- The Freenove Mobile Application can be installed on a mobile device to control Gizmo via Wi-Fi.
- An IR Remote Control device can also be used to operate Gizmo.
- Gizmo's hardware communicates with the computer/laptop via USB for sketch deployment.
- Gizmo's hardware communicates with the computer/laptop and mobile devices over Wi-Fi for remote control and operating Gizmo functions.

7. PRODUCT / SPRINT BACKLOG

ID	As a…	I want to be able to…	So that…	Priority	Status
BID001	User	Download and install Arduino IDE	I can create, edit, and save C++ codes	Sprint 1 / Milestone 2	DONE
BID002	User	Connect Gizmo to the computer using a USB cable	I can upload C++ codes for Gizmo's functionalities	Sprint 1 / Milestone 2	DONE
BID003	User	Turn Gizmo's power to ON	I can use and control Gizmo's behavior and movements	Sprint 1 / Milestone 2	DONE
BID004	User	Use Gizmo's ultrasonic sensors (circle shapes)	I can use it to detect circle-shaped obstacles in front of Gizmo	Sprint 1 / Milestone 2	DONE
BID005	User	Utilize Gizmo's ultrasonic sensors to emit waves	I can calculate the distance of the obstacle in front of Gizmo	Sprint 1 / Milestone 2	DONE
BID006	User	Combine Gizmo's direction and movement with obstacle detection calculation	Gizmo can move around the circle-shaped obstacle while detecting it ahead	Sprint 1 / Milestone 2	DONE
BID007	User	Use Gizmo's reflective optical sensors and line tracking sensors	I can use it to have Gizmo detect black-line routes on the ground for direction	Sprint 1 / Milestone 2	DONE
BID008	User	Utilize Gizmo's reflective optical sensors and line tracking sensors to emit infrared lights	I can use it to have Gizmo detect and compute the black-line route to move on	Sprint 1 / Milestone 2	DONE
BID009	User	Combine Gizmo's direction and movement with line track computation	Gizmo can move and traverse the black-line routes	Sprint 1 / Milestone 2	DONE
BID010	User	Integrate Gizmo's obstacle avoidance and line tracking capabilities (circle shapes)	I can use it to have Gizmo traverse a black-line route and avoid circle-shaped obstacles, temporarily	Sprint 2 / Milestone 3	DONE

			moving out of the path, and then go back to the route		
BID011	User	Create a black-line route on a mat	I can use it to test Gizmo's obstacle avoidance with line tracking capabilities while in movement	Sprint 2 / Milestone 3	DONE
BID012	User	Use Gizmo's photoresistors	I can use it to have Gizmo detect the movement and direction of the light	Sprint 3 / Milestone 4	NEXT
BID013	User	Utilize Gizmo's photoresistors to get ADC values from the detection of light	I can use it to have Gizmo determine and follow the direction of light	Sprint 3 / Milestone 4	NEXT
BID014	User	Use Gizmo's ultrasonic sensors (rectangle shapes)	I can use it to detect rectangle-shaped obstacles in front of Gizmo	Sprint 3 / Milestone 4	NEXT
BID015	User	Integrate Gizmo's obstacle avoidance and line tracking capabilities (circle and rectangle shapes)	I can use it to have Gizmo traverse a black-line route and avoid circle and rectangle shaped obstacles, temporarily moving out of the path, and then go back to the route	Sprint 3 / Milestone 4	NEXT
BID016	User	Display Gizmo's eyes rotating via LED lights	I can add and see Gizmo's eye functionality to rotate	Sprint 3 / Milestone 4	NEXT
BID017	User	Display Gizmo's eyes to blink via LED lights	I can add and see Gizmo's eye functionality to blink	Sprint 3 / Milestone 4	NEXT
BID018	User	Display Gizmo's eyes to smile via LED lights	I can add and see Gizmo's eye functionality to smile	Sprint 3 / Milestone 4	NEXT
BID019	User	Display Gizmo's eyes to cry via LED lights	I can add and see Gizmo's eye functionality to cry	Sprint 3 / Milestone 4	NEXT
BID020	User	Install Freenove's mobile application via App Store (for iOS phones)	I can setup Gizmo to be controlled using Freenove's mobile application	Sprint 3 / Milestone 4	NEXT

BID021	User	Install Freenove's mobile application via Google Play Store (for Android phones)	I can setup Gizmo to be controlled using Freenove's mobile application	Sprint 3 / Milestone 4	NEXT
BID022	User	Install Freenove's desktop PC application	I can setup Gizmo to be controlled using Freenove's desktop PC application	Sprint 3 / Milestone 4	NEXT
BID023	User	Set Gizmo's Wi-Fi connection password	Connect to Gizmo via Wi-Fi Access Point	Sprint 3 / Milestone 4	NEXT
BID024	User	Set Gizmo's hotspot connection password	Connect to Gizmo via its Hotspot Network using ESP32	Sprint 3 / Milestone 4	NEXT
BID025	User	Select Gizmo's network in the list of available connection networks in my mobile phone	I can establish network connection to access Gizmo	Sprint 3 / Milestone 4	NEXT
BID026	User	Select Gizmo's network in the list of available connection networks in my desktop PC	I can establish network connection to access Gizmo	Sprint 3 / Milestone 4	NEXT
BID027	User	Open and use Freenove's mobile application	Select 4WD Car for ESP32 which is Gizmo's device type	Sprint 3 / Milestone 4	NEXT
BID028	User	Open and use Freenove's desktop PC application	Select 4WD Car for ESP32 which is Gizmo's device type	Sprint 3 / Milestone 4	NEXT
BID029	User	Enter Gizmo's IP address in the mobile application	Set Gizmo's IP address to connect to	Sprint 3 / Milestone 4	NEXT
BID030	User	Enter Gizmo's IP address in the desktop PC application	Set Gizmo's IP address to connect to	Sprint 3 / Milestone 4	NEXT
BID031	User	Press the connect button in the mobile application	I can connect the mobile application to Gizmo's network	Sprint 3 / Milestone 4	NEXT

BID032	User	Press the connect button in the desktop PC application	I can connect the desktop PC application to Gizmo's network	Sprint 3 / Milestone 4	NEXT
BID033	User	Use Freenove's mobile application to use the control car running feature	I can control Gizmo's direction and movement using mobile application	Sprint 3 / Milestone 4	NEXT
BID034	User	Use Freenove's mobile application to use the control camera feature	I can control Gizmo's camera angle and position using mobile application	Sprint 3 / Milestone 4	NEXT
BID035	User	Use Freenove's mobile application to display Gizmo's LED emotion feature	I can show Gizmo's LED emotions using mobile application	Sprint 3 / Milestone 4	NEXT
BID036	User	Use Freenove's mobile application to display Gizmo's RGB LED lights feature	I can show Gizmo's RBG LED emotions using mobile application	Sprint 3 / Milestone 4	NEXT
BID037	User	Use Freenove's mobile application to produce Gizmo's buzzer feature	I can show Gizmo's buzzer sound using mobile application	Sprint 3 / Milestone 4	NEXT
BID038	User	Use Freenove's desktop PC application to click the Turn Left button	Gizmo can turn left	Sprint 3 / Milestone 4	NEXT
BID039	User	Use Freenove's desktop PC application to click the Turn Right button	Gizmo can turn right	Sprint 3 / Milestone 4	NEXT
BID040	User	Use Freenove's desktop PC application to click the Forward button	Gizmo can move forward	Sprint 3 / Milestone 4	NEXT
BID041	User	Use Freenove's desktop PC application to click the Backward button	Gizmo can move backward	Sprint 3 / Milestone 4	NEXT

BID042	User	Use Freenove's desktop PC application to click the Up button	Gizmo's servo/camera angle turn to face up	Sprint 3 / Milestone 4	NEXT
BID043	User	Use Freenove's desktop PC application to click the Down button	Gizmo's servo/camera angle turn to face down	Sprint 3 / Milestone 4	NEXT
BID044	User	Use Freenove's desktop PC application to click the Left button	Gizmo's servo/camera angle turn to face left	Sprint 3 / Milestone 4	NEXT
BID045	User	Use Freenove's desktop PC application to click the Right button	Gizmo's servo/camera angle turn to face right	Sprint 3 / Milestone 4	NEXT
BID046	User	Use Freenove's desktop PC application to set Gizmo's RGB color	I can set and show Gizmo's RGB LED lights	Sprint 3 / Milestone 4	NEXT
BID047	User	Use Freenove's desktop PC application to set Gizmo's RGB color mode	I can set and show Gizmo's RGB LED lights according to different modes	Sprint 3 / Milestone 4	NEXT
BID048	User	Use Freenove's desktop PC application to display Gizmo's battery level	I can see and check Gizmo's battery level	Sprint 3 / Milestone 4	NEXT
BID049	User	Put batteries in Remote Control	I can use the remote control to navigate Gizmo's movements via Infrared	Sprint 3 / Milestone 4	NEXT
BID050	User	Click the power button on the Remote Control	I can use the remote control to use Gizmo's movements via Infrared	Sprint 3 / Milestone 4	NEXT
BID051	User	Click the plus (+) button on the Remote Control	I can use the remote control to use Gizmo's move forward via Infrared	Sprint 3 / Milestone 4	NEXT
BID052	User	Click the rewind (<<) button on the Remote Control	I can use the remote control to Gizmo's turn left via Infrared	Sprint 3 / Milestone 4	NEXT

BID053	User	Click the forward (>>) button on the Remote Control	I can use the remote control to Gizmo's turn right via Infrared	Sprint 3 / Milestone 4	NEXT
BID054	User	Click the minus (-) button on the Remote Control	I can use the remote control to Gizmo's to move back via Infrared	Sprint 3 / Milestone 4	NEXT
BID055	User	Click the play (>) button on the Remote Control	I can use the remote control to Gizmo's to stop via Infrared	Sprint 3 / Milestone 4	NEXT
BID056	User	Click the 0 button on the Remote Control	I can use the remote control to Gizmo's control servo 1 turn left via Infrared	Sprint 3 / Milestone 4	NEXT
BID057	User	Click the 1 button on the Remote Control	I can use the remote control to Gizmo's control servo 1 turn right via Infrared	Sprint 3 / Milestone 4	NEXT
BID058	User	Click the 2 button on the Remote Control	I can use the remote control to Gizmo's to turn on displaying random emoticons via Infrared	Sprint 3 / Milestone 4	NEXT
BID059	User	Click the 3 button on the Remote Control	I can use the remote control to Gizmo's control servo 2 turn right via Infrared	Sprint 3 / Milestone 4	NEXT
BID060	User	Click the 4 button on the Remote Control	I can use the remote control to Gizmo's control servo 1 turn 90-degree angle via Infrared	Sprint 3 / Milestone 4	NEXT
BID061	User	Click the 5 button on the Remote Control	I can use the remote control to Gizmo's to turn off displaying random emoticons via Infrared	Sprint 3 / Milestone 4	NEXT
BID062	User	Click the 6 button on the Remote Control	I can use the remote control to Gizmo's control servo 2 turn 90-degree angle via Infrared	Sprint 3 / Milestone 4	NEXT
BID063	User	Click the 7 button on the Remote Control	I can use the remote control to turn on Gizmo's random display of WS2812 via Infrared	Sprint 3 / Milestone 4	NEXT

BID064	User	Click the 8 button on the Remote Control	I can use the remote control to turn off Gizmo's random display of WS2812 via Infrared	Sprint 3 / Milestone 4	NEXT
BID065	User	Click the C button on the Remote Control	I can use the remote control to Gizmo's control servo 2 turn left via Infrared	Sprint 3 / Milestone 4	NEXT
BID066	User	Click the test button on the Remote Control	I can use the remote control to Gizmo's buzzer sound via Infrared	Sprint 3 / Milestone 4	NEXT

8. GANTT CHART

See Appendix 3.

9. REFERENCES

Freenove (n.d.). *Tutorial.pdf*. https://github.com/Freenove/Freenove_4WD_Car_Kit_for_ESP32/Blob/master/Tutorial.pdf

Freenove Videos (n.d.). *Freenove Videos*. https://www.youtube.com/@Freenove/videos

Arduino (2023). *Arduino IDE*. https://www.arduino.cc/

APPENDIX 1 – PROTOTYPE

Line Tracking with Obstacle Avoidance (obstacle circle)

if it is on track

Sensor scan 60 90 120 degrees

If there is an Obstacle head stop

Sensor scan

If it is too near move back

If it is too far move forward

Then Stop at a certain distance.

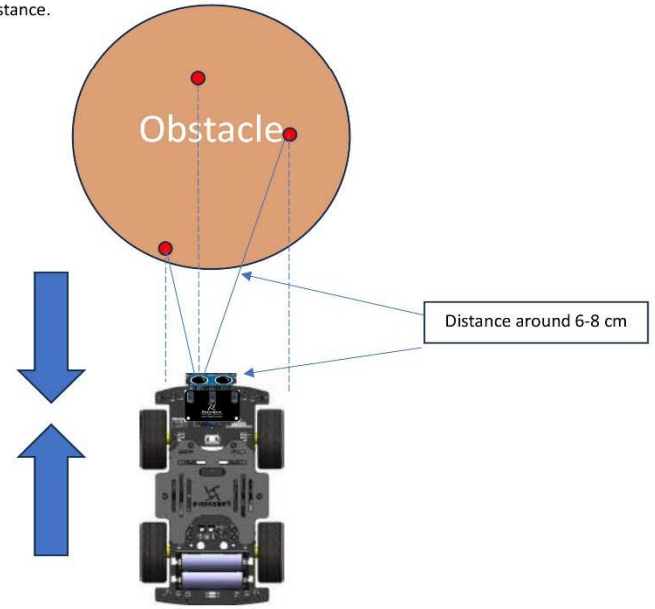


Illustration 1 (Obstacle Avoidance)

The sensor scans the object again.

Scans 60 90 120 degrees

Determine if the object is closer to the left or right.

If the object is on the left scanner turns 176 degrees, and then turns left at a certain distance.

If the object is on the right scanner turns 0 degrees, and then turns right at a certain distance.

then stops.

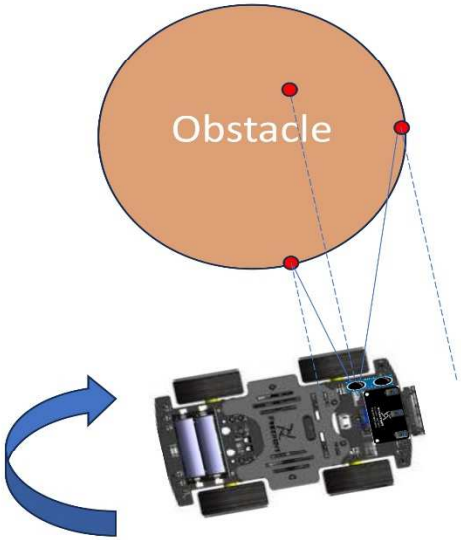


Illustration 2 (Obstacle Avoidance)

Before moving forward

Gizmo moves forward between 18-20 distances.

If distance is less than 18 Gizmo goes far

If distance is greater than 20 Gizmo goes near

Then loops to move forward between 18 to 20 distances.

Moves forward at a certain distance then stops.

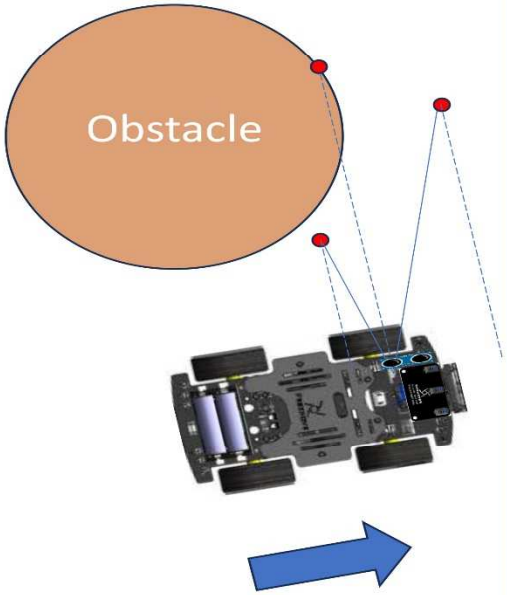
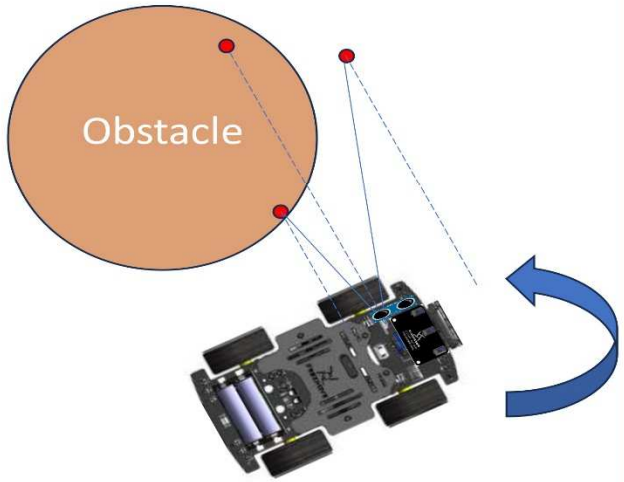


Illustration 3 (Obstacle Avoidance)

If the previous turn was left, Gizmo will turn to the right at a certain distance and then stop.

If the previous turn was right, Gizmo will turn to the left at a certain distance and then stop.



Repeat move forward at a certain distance then stop then turn at a certain distance then stop.

And stop when line track is located while moving and turning.

Illustration 4 (Obstacle Avoidance)

from obstacle avoidance, Gizmo detects 111 then stops.

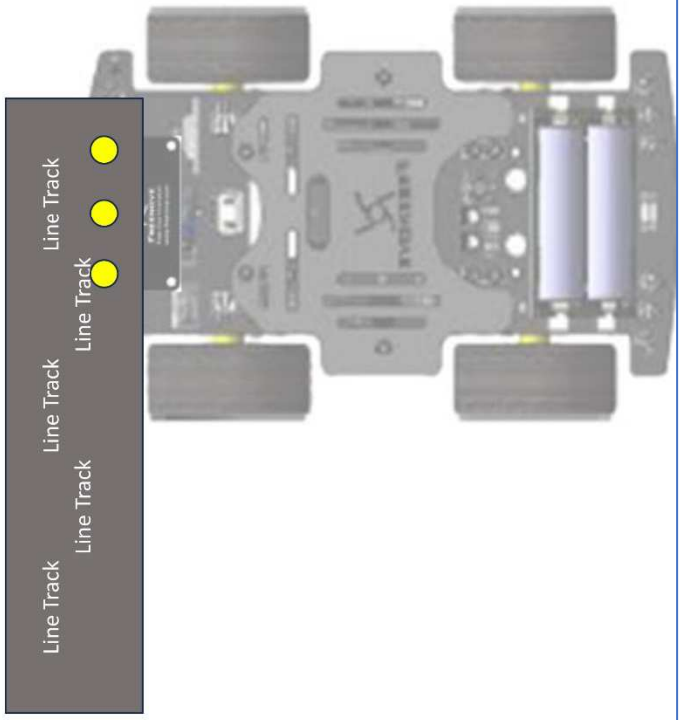


Illustration 1 (Return to Track)

Gizmo move forwards until 000.

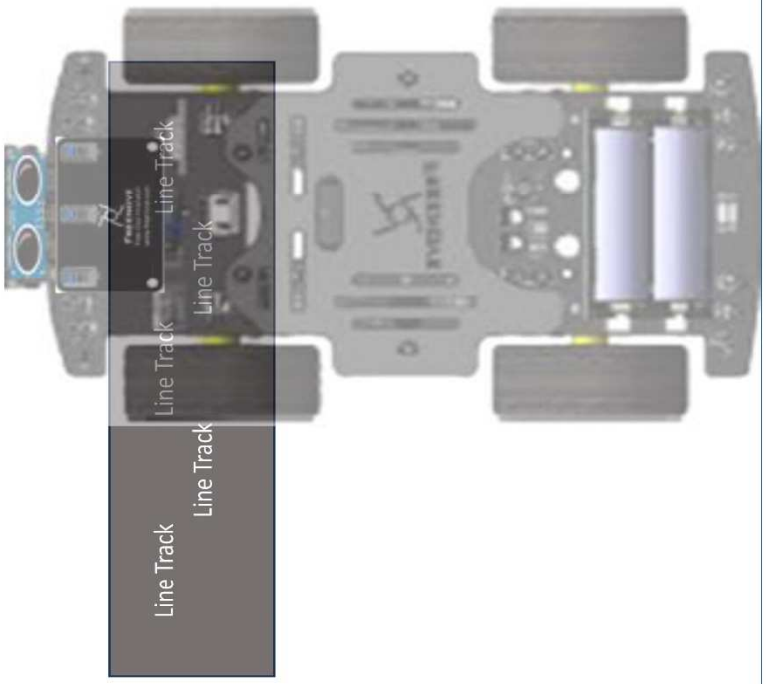


Illustration 2 (Return to Track)

Gizmo turns far from obstacle until 111

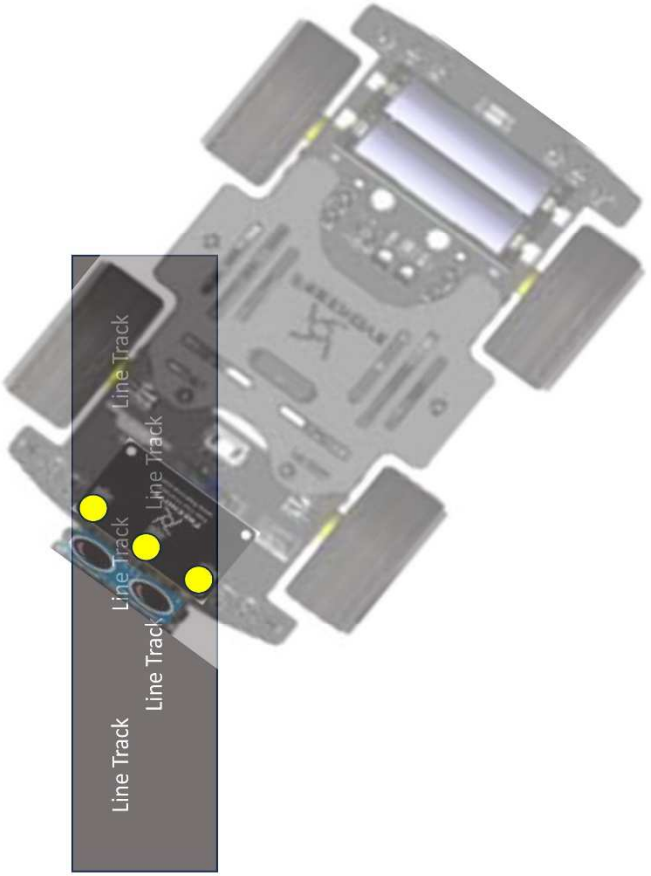


Illustration 3 (Return to Track)

Move forward until 110 or 011.

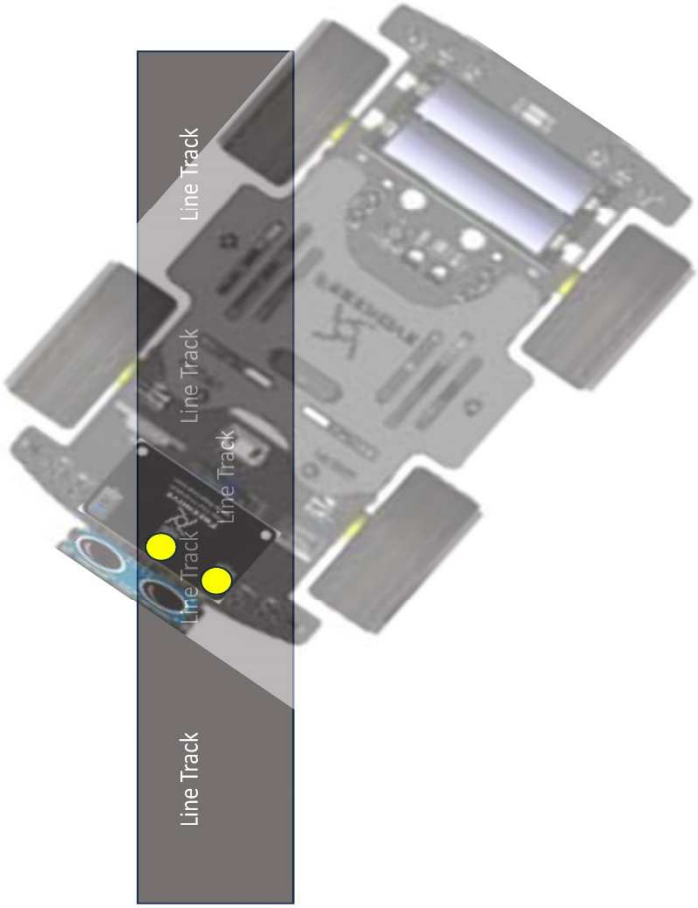
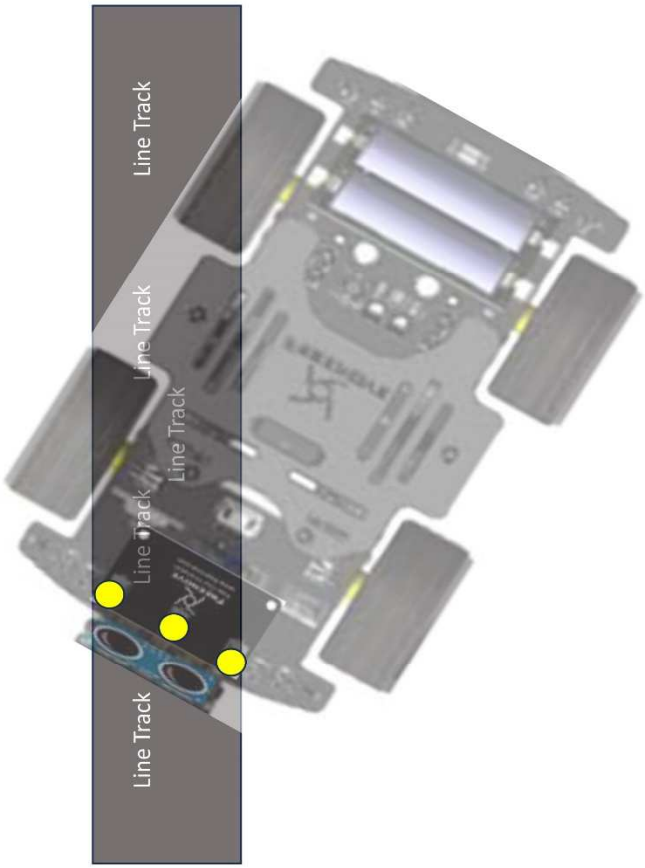


Illustration 4 (Return to Track)

Turn far from obstacle until 111.



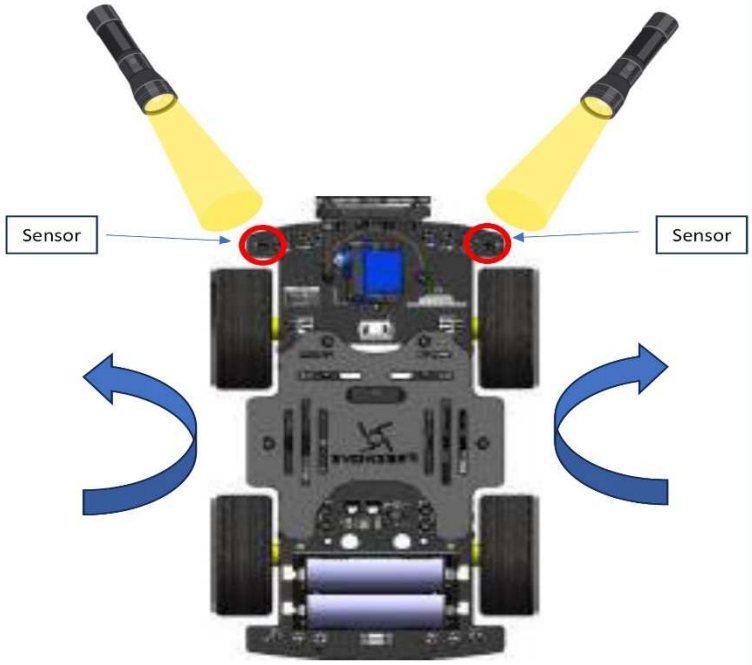
Repeat Gizmo turns far from obstacle until 111 and move forward until 110 or 011, until it is parallel to the line.

Same as Illustration 3 (Return to Track)

Light Tracing

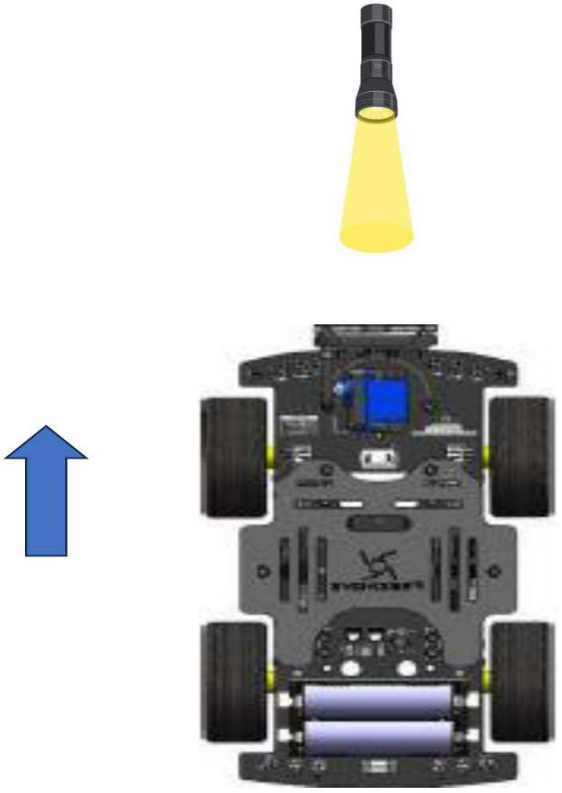
If the light is on the right sensor Gizmo rotates right

If the light is on the left sensor Gizmo rotates left



If the light is on the center of two sensor Gizmo will move forward

If there is no light Gizmo will stop



APPENDIX 2 – TEST CASES

User Story 1: Gizmo to autonomously detect and avoid obstacles in its path.

Test Case ID	Step No.	Operator Action / Input Specifications	Expected Results	Assumption / Operator Input	Status Pass / Fail	QC Comments / Actual Results
Pre-conditions: **1. Arduino IDE, USB-SERIAL CH340 (COMx), and necessary libraries installed in Gizmo Operator workstation.** **2. Gizmo battery is charged and installed in the battery compartment.** **3. Necessary sensors are integrated with Gizmo's car shield and ESP32-WROVER.**						
US1-001	**Loading of Obstacle Avoidance while in route sketch**					
	1	Connect your computer and Gizmo's ESP32 with a USB cable.	ESP2 has communication with the computer.	N/A	**Pass**	
	2	Open "07.1_Line_Tracking_with_Obstacle_Avoidance" folder in "Freenove_4WD_Car_Kit_for_ESP32\Sketches", double-click "07.1_Line_Tracking_with_Obstacle_Avoidance.ino".	Correct sketch selected.	Sketch is free of code errors	**Pass**	
	3	Select development board. Click Tools on the Menu bar, move your mouse to Board: "Arduino Uno", select ESP32 Arduino and then select ESP32 Wrover Module.	Correct development board selected.	N/A	**Pass**	
	4	Select serial port. Click Tools on the Menu bar, move your mouse to Port and select COMx on your computer. The value of COMx varies in different computers, but it won't affect the download function of ESP32, as long as you select the correct one.	Correct serial port selected.	N/A	**Pass**	
	5	Click "Upload Using Programmer" and the program will be downloaded to Gizmo's ESP32.	Sketch successfully downloaded in Gizmo's ESP32.	N/A	**Pass**	Ensure that these libraries are added: Freenove_VK16K33_Lib_For_ESP32.zip, PCF8574.zip
	6	A message "Done Uploading" and the console will have the message "Leaving..., Hard resetting via RTS pin..."	Correct console output with no warnings or failures.	N/A	**Pass**	
	8	Unplug the USB cable from Gizmo.		N/A	**Pass**	

	9	Turn ON the power switch.	Gizmo successfully powered on.	N/A	**Pass**	
US1-002	**Line Tracking using predefined path**					
	1	Steps 1 to 9 of US1-001 successfully completed.		Use 04.2_Track_Car.ino	**Pass**	
	2	Scenario 1: With a straight-line path, Gizmo will travel from point A to point B and stop at the end of the track.	Gizmo will not derail from the path and stops at the end of the track.	N/A	**Pass**	*Videos can be found in GDrive (US1-002-Scenario1)
	3	Scenario 2: With curve track, Gizmo will loop indefinitely.	Gizmo will not derail from the path.	N/A	**Pass**	*Video can be found in GDrive (US1-002-Scenario2)
US1-003	**Obstacle Detection**					
	1	Steps 1 to 9 of US1-001 successfully completed.		N/A	**Pass**	
	2	While Gizmo is moving, it identifies an obstacle within the predefined distance using its front ultrasonic sensor.	Gizmo will stop and continue to scan the obstacle.	The obstacle has a circle shape or rounded.	**Pass**	*Video can be found in GDrive(US1-003-Step2)
US1-004	**Obstacle Detection and Avoidance**					
	1	Steps 1 to 9 of US1-001 successfully completed.		N/A	**Pass**	
	2	While Gizmo is moving, it identifies an obstacle within the predefined distance using its front ultrasonic sensor.	Gizmo will stop and continue to scan the obstacle.	Obstacle has a circle shape or is rounded.	**Pass**	*Video can be found in GDrive (US1-003-Step2)
	3	Gizmo will adjust its navigation to create distance from the obstacle.	Gizmo will move away if too near the obstacle or move closer if too far from the obstacle.		**Pass**	*Videos can be found in GDrive (US1-004-Step3)
	4	Gizmo will continue to scan the obstacle as it navigates within the perimeter of the object.	Gizmo will successfully navigate around the obstacle to avoid it.		**Pass**	*Video can be found in GDrive (US1-004-Step4)

US1-005	**Obstacle detection and avoidance within the predefined path**					
	1	Steps 1 to 9 of US1-001 successfully completed.		N/A	**Pass**	
	2	While Gizmo is moving at a predefined path (straight path), it identifies an obstacle within the predefined distance using its front ultrasonic sensor.	Gizmo will stop and continue to scan the obstacle.	Obstacle has a circle shape or is rounded.	**Pass**	*Video can be found in GDrive (US1-003-Step2)
	3	Gizmo will adjust its distance with the obstacle.	Gizmo will move away if too near the obstacle or move closer if too far from the obstacle.	There is only one obstacle within the predefined path.	**Pass**	*Videos can be found in GDrive (US1-004-Step3)
	4	Gizmo will continue to scan the obstacle as it navigates within the perimeter of the object.	Gizmo will avoid obstacle as it navigate around it.		**Pass**	*Video can be found in GDrive (US1-004-Step4)
	5	Gizmo will scan and attempts to return to the predefined path.	Gizmo will return to the predefined path.			
	6	Repeat steps 2 to 5 with the following scenarios: Curve path with 1 obstacle Straight line path with 2 or more obstacle Curve path with 2 or more obstacle	Gizmo will avoid 1 or more obstacles and will return to the predefined path.			

User Story 2: Gizmo to be capable of detecting and tracking a moving light source

Test Case ID	Step No.	Operator Action / Input Specifications	Expected Results	Assumption / Operator Input	Status Pass / Fail	QC Comments / Actual Results
Pre-conditions: **1. Arduino IDE, USB-SERIAL CH340 (COMx), and necessary libraries installed in Gizmo Operator workstation.** **2. Gizmo battery is charged and installed in the battery compartment.** **3. Necessary sensors are integrated with Gizmo's car shield and ESP32-WROVER.**						
US2-001	**Loading of Light Tracing sketch**					
	1	Connect your computer and Gizmo's ESP32 with a USB cable.	ESP2 has communication with the computer.	N/A		
	2	Open "03.3_Photosensitive_Car" folder in "Freenove_4WD_Car_Kit_for_ESP32\Sketches", double-click "03.3_Photosensitive_Car.ino".	Correct sketch selected.	Sketch is free of code errors		
	3	Select development board. Click Tolos on the Menu bar, move your mouse to Board: "Arduino Uno", select ESP32 Arduino and then select ESP32 Wrover Module.	Correct development board selected.	N/A		
	4	Select serial port. Click Tools on the Menu bar, move your mouse to Port and select COMx on your computer. The value of COMx varies in different computers, but it won't affect the download function of ESP32, as long as you select the correct one.	Correct serial port selected.	N/A		
	5	Click "Upload Using Programmer" and the program will be downloaded to Gizmo's ESP32.	Sketch successfully downloaded in Gizmo's ESP32.	N/A		
	6	A message "Done Uploading" and the console will have the message "Leaving..., Hard resetting via RTS pin..."	Correct console output with no warnings or failures.	N/A		
	7	Unplug the USB cable from Gizmo.		N/A		
	8	Turn ON the power switch.	Gizmo successfully powered on.	N/A		
US2-002	Light Tracing					
	1	Steps 1 to 9 of US2-001 successfully completed.		N/A		

	2	While Gizmo is moving, it identifies a light source at its right side within the predefined distance using its photoresistors.	Gizmo will turn right.			
	3	While Gizmo is moving, it identifies a light source at its left side within the predefined distance using its photoresistors.	Gizmo will turn left.	N/A		
	4	While Gizmo is moving, it identifies a light source in front of the car within the predefined distance using its photoresistors.	Gizmo will move straight.	N/A		
	5	While Gizmo is moving, after turning or moving after the light source is identified, turn-off the light source.	Gizmo will stop.	N/A		
US2-003	**Light tracing with obstacle detection and avoidance within the predefined path (WISHLIST)**					
	1	Steps of US1-005 successfully completed.		N/A		
	2	Gizmo to get off from the predefined path	Gizmo will enable light tracing capability.	N/A		
	3	Use a light source to guide Gizmo back to track	Gizmo will enable line tracking capability.			

User Story 3: To see continuous improvements in the existing features of Gizmo

Test Case ID	Step No.	Operator Action / Input Specifications	Expected Results	Assumption / Operator Input	Status Pass / Fail	QC Comments / Actual Results
Pre-conditions: **1. Arduino IDE, USB-SERIAL CH340 (COMx), and necessary libraries installed in Gizmo Operator workstation.** **2. Gizmo battery is charged and installed in the battery compartment.** **3. Necessary sensors are integrated with Gizmo's car shield and ESP32-WROVER.**						
US3-001	**Loading of sketch**					
	1	Connect your computer and Gizmo's ESP32 with a USB cable.	ESP2 has communication with the computer.	N/A		
	2	Open folder in "Freenove_4WD_Car_Kit_for_ESP32\Sketches", double-click sketch (*.ino) file.	Correct sketch selected.	Sketch is free of code errors		
	3	Select development board. Click Tools on the Menu bar, move your mouse to Board: "Arduino Uno", select ESP32 Arduino and then select ESP32 Wrover Module.	Correct development board selected.	N/A		
	4	Select serial port. Click Tools on the Menu bar, move your mouse to Port and select COMx on your computer. The value of COMx varies in different computers, but it won't affect the download function of ESP32, as long as you select the correct one.	Correct serial port selected.	N/A		
	5	Click "Upload Using Programmer" and the program will be downloaded to Gizmo's ESP32.	Sketch successfully downloaded in Gizmo's ESP32.	N/A		
	6	A message "Done Uploading" and the console will have the message "Leaving..., Hard resetting via RTS pin..."	Correct console output with no warnings or failures.	N/A		
	7	Unplug the USB cable from Gizmo.		N/A		
	8	Turn ON the power switch.	Gizmo successfully powered on.	N/A		

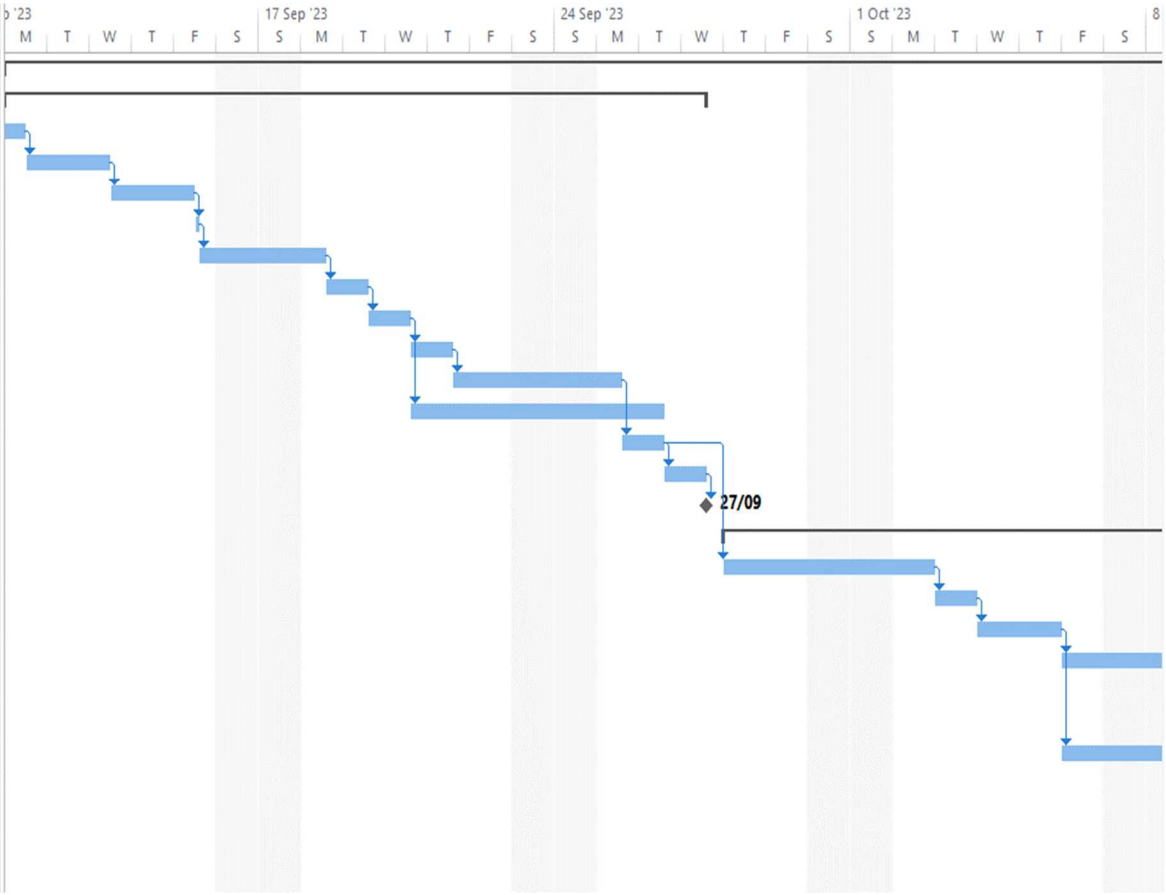
US3-002	Gizmo functionalities					
	1	Steps 1 to 9 of US3-001 successfully completed.		Correct sketch uploaded to perform the succeeding steps		
	2	Setup different surfaces and allow Gizmo scan obstacles. - Using puzzle mat - Using carpeted floor Using smooth floor or surface	Gizmo can scan obstacles and avoid them regardless of surfaces.	N/A		
	3	Run Gizmo for Obstacle avoidance and line tracking for 30 minutes	Gizmo will continue detecting obstacle and able to track line without failure.	Battery used is fully charged.		
	4	LED displays	Gizmo displays emotions or navigation visuals.	N/A		
	5	Multi-functional Infrared Car	Gizmo to perform functions using the IR remote. See Figure XXX for Remote Functions			
	6	WiFi Car	Gizmo to perform functions using the Freenove Application via computer or mobile Device.	Successfully comfigured WiFi car see Freenove Tutorial **Chapter 7 WiFi Car**		
	7	Perform Power Off	Power off Gizmo using power button.	N/A		

ICON	KEY Value	Function	ICON	KEY Value	Function
+	FF02FD	Move forward	TEST	FF22DD	Control the buzzer
⏮	FFE01F	Turn left	2	FF18E7	Random emoticons
⏭	FF906F	Turn light	5	FF38C7	Turn off emoticons
−	FF9867	Move back	7	FF42BD	Random display of WS2812
▶	FFA857	Stop the car	8	FF4AB5	Turn off WS2812 display
0	FF6897	Control servo 1 turn left	C	FFB04F	Control servo 2 turn left
1	FF30CF	Control servo 1 turn right	3	FF7A85	Control servo 2 turn right
4	FF10EF	Control servo 1 turn to 90°	6	FF5AA5	Control servo 2 turn to 90°

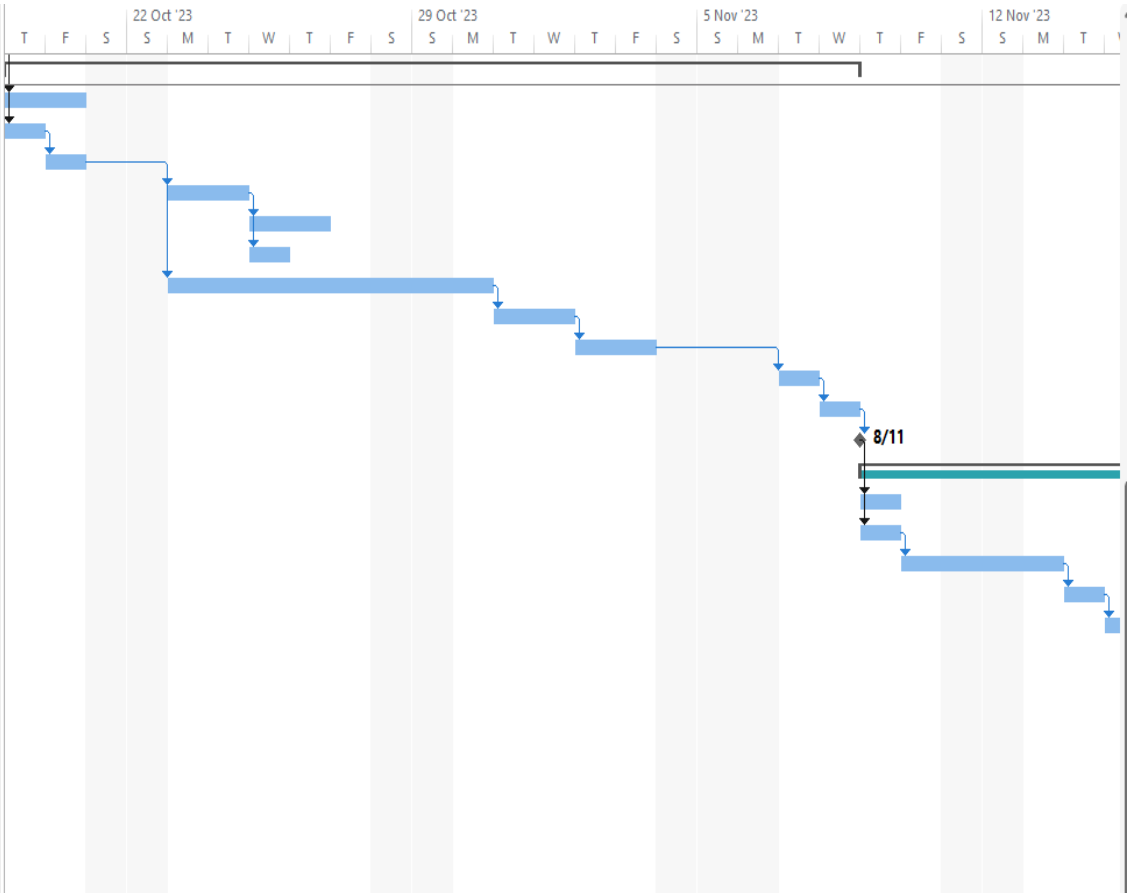
FIGURE 3: IR REMOTE FUNCTONS

APPENDIX 3 – GANTT CHART

Task Mode	Task Name	Duration	Start	Finish	Predecess
	Capstone Project	**102 days**	**Mon 11/09/2**	**Wed 31/01/2**	
	Inception	**12,75 days**	**Mon 11/09/2**	**Wed 27/09/2**	
	Team formation	4 hrs	Mon 11/09/2	Mon 11/09/2	
	Project selection	2 days	Mon 11/09/2	Wed 13/09/2	3
	Sponsor approval	2 days	Wed 13/09/2	Fri 15/09/23	4
	Assigment of roles	2 hrs	Fri 15/09/23	Fri 15/09/23	5
	Logo creation	1 day	Fri 15/09/23	Mon 18/09/2	6
	Project charter	1 day	Mon 18/09/2	Tue 19/09/23	7
	Project scope	1 day	Tue 19/09/23	Wed 20/09/2	8
	Gantt chart	1 day	Wed 20/09/2	Thu 21/09/23	9
	Requirement gathering	2 days	Thu 21/09/23	Mon 25/09/2	10
	Hardware analysis	4 days	Wed 20/09/2	Tue 26/09/23	9
	Relevance	1 day	Mon 25/09/2	Tue 26/09/23	11
	Sponsor Sign-Off	1 day	Tue 26/09/23	Wed 27/09/2	13
	Milestone 1: Inception	0 days	Wed 27/09/2	Wed 27/09/2	14
	Sprint 1 - Goal 1	**15 days**	**Thu 28/09/23**	**Wed 18/10/2**	
	Web Search	3 days	Thu 28/09/23	Mon 2/10/23	13
	Use case Diagrams - Draft	1 day	Tue 3/10/23	Tue 3/10/23	17
	Class Diagrams - Draft	2 days	Wed 4/10/23	Thu 5/10/23	18
	Prototype - Draft	3 days	Fri 6/10/23	Tue 10/10/23	19
	System Architecture - Draft	3 days	Wed 11/10/2	Fri 13/10/23	20
	Test Case	1 day	Wed 11/10/2	Wed 11/10/2	20
	Development - Goal 1	5 days	Fri 6/10/23	Thu 12/10/23	19
	Unit Testing - Goal 1	2 days	Fri 13/10/23	Mon 16/10/2	23
	Sponsor Demo	1 day	Tue 17/10/23	Tue 17/10/23	24
	Deployment - Goal 1	1 day	Wed 18/10/2	Wed 18/10/2	25
	Milestone 2: Sprint 1 - Goal 1	0 days	Wed 18/10/2	Wed 18/10/2	26



Task Mode	Task Name	Duration	Start	Finish	Predecess
	Sprint 2 - Goal 2	**14 days**	**Thu 19/10/2:**	**Wed 8/11/2:**	
	System Architecture	2 days	Thu 19/10/2:	Fri 20/10/23	27
	Use Case Diagrams	1 day	Thu 19/10/2:	Thu 19/10/2:	27
	Class Diagrams	1 day	Fri 20/10/23	Fri 20/10/23	30
	Prototype	2 days	Mon 23/10/2	Tue 24/10/2:	31
	System Architecture	2 days	Wed 25/10/2	Thu 26/10/2:	32
	Test Case	1 day	Wed 25/10/2	Wed 25/10/2	32
	Development - Goal 2	6 days	Mon 23/10/2	Mon 30/10/2	31
	Unit Testing - Goal 2	2 days	Tue 31/10/2:	Wed 1/11/2:	35
	Integration Testing	2 days	Thu 2/11/23	Fri 3/11/23	36
	Sponsor Demo	1 day	Tue 7/11/23	Tue 7/11/23	37
	Deployment - Goal 2	1 day	Wed 8/11/2:	Wed 8/11/2:	38
	Milestone 3: Sprint 2 - Goal 2	0 days	Wed 8/11/2:	Wed 8/11/2:	39
	Sprint 3 - Improvements	**60 days**	**Thu 9/11/23**	**Wed 31/01/2**	
	Use Case Diagram	1 day	Thu 9/11/23	Thu 9/11/23	40
	Class Diagram	1 day	Thu 9/11/23	Thu 9/11/23	40
	Prototype	2 days	Fri 10/11/23	Mon 13/11/2	43
	Test Case	1 day	Tue 14/11/2:	Tue 14/11/2:	44
	Development	5 days	Wed 15/11/2	Tue 21/11/2:	45
	Unit Testing	2 days	Wed 22/11/2	Thu 23/11/2:	46
	Integration Testing	1 day	Fri 24/11/23	Fri 24/11/23	47
	Sponsor Demo	1 day	Mon 27/11/2	Mon 27/11/2	48
	Deployment	1 day	Tue 28/11/2:	Tue 28/11/2:	49
	Recommendations	1 day	Tue 28/11/2:	Tue 28/11/2:	49
	Summary	1 day	Wed 29/11/2	Wed 29/11/2	51
	References	1 day	Wed 29/11/2	Wed 29/11/2	51
	Milestone 4: Sprint 3 - Improvements	0 days	Wed 29/11/2	Wed 29/11/2	53



APPENDIX 4 – REQUIREMENTS TRACEABILITY MATRIX

REQT ID	BUSINESS REQUIREMENT	CATEGORY	PRIORITY	FUNCTIONAL / NON-FUNCTIONAL SPECIFICATION REFERENCE	ACCEPTANCE CRITERIA
	USID-1 Autonomously detects and avoids obstacles in its path				
1		Functional	High	OD1	The ultrasonic head is successfully installed in Gizmo.
2		Functional	High	OD2	The ultrasonic head scans at different angles while detecting obstacles.
3		Functional	High	CA1	Gizmo can avoid the obstacle and continue navigating the pre-defined path.
4		Functional	High	CA2	Gizmo slows down, stops or steers away from obstacles.
5		Functional	High	UWE1	Gizmo successfully transmits sound waves from its transmitter
6		Functional	High	UWE2	Gizmo successfully receives sound waves in its receiver.
7		Functional	Medium	TIM1	Gizmo accurately gets pingTime using pulseIn method.
8		Functional	Medium	TIM2	Gizmo accurately gets pingTime using pulseIn method.
9		Functional	Medium	DC1	Gizmo accurately calculate the distance using the formula distance = velocity * time / 2.
10		Functional	Medium	DC2	Gizmo accurately calculate the distance using the formula distance = velocity * time / 2.

11		Non-Functional	Low	ER1	Gizmo to function effectively on a variety of surfaces.
12		Non-Functional	Low	ER2	Gizmo has stable and reliable performance regardless of the surface type.
13		Non-Functional	Low	PE1	Gizmo is capable of performing obstacle avoidance and line tracking without failure for more than 30 minutes of continuous operation.
14		Non-Functional	Low	PE2	Gizmo is capable of performing obstacle avoidance and line tracking without failure for more than 30 minutes of continuous operation.
15		Non-Functional	Low	RA1	Gizmo is capable of performing obstacle avoidance and line tracking without failure for more than 30 minutes of continuous operation.
16		Non-Functional	Low	RA2	Gizmo can continue to operate even there is an error in the code and will be able to be stopped using power off button.
17		Non-Functional	Low	RA3	Gizmo can be stopped using power off button.
18		Non-Functional	Low	UUS2	Gizmo will utilize LED matrix to display its actions.
19		Non-Functional	Low	UUS3	Gizmo can be powered off using the power button.
	USID-2 detecting and tracking a moving light source				
1		Functional	High	LD1	Gizmo will be able to detect light using its photoresistors.
2		Functional	High	LD2	Gizmo will be able to detect light using its photoresistors and evaluate ADC values.

3		Functional	High	LTB1	Gizmo will be able to detect light and be able to steer toward the light source.
4		Functional	High	LTB2	Gizmo will be able to detect light and be able to steer toward the light source.
5		Functional	Medium	ER3	Gizmo will be able to detect varying light source.
6		Functional	Medium	ER4	Gizmo will be able to detect varying light source and be able to steer toward the light source.
7		Non-Functional	Low	RA1	Gizmo is capable of light tracing without failure for more than 30 minutes of continuous operation.
8		Non-Functional	Low	RA2	Gizmo can continue to operate even there is an error in the code and will be able to be stopped using power off button.
9		Non-Functional	Low	RA3	Gizmo can be stopped using power off button.
10		Non-Functional	Low	UUS3	Gizmo can be powered off using the power button.
	USID-3 Improvements in the existing features of Gizmo				
		Non-Functional	Low	LLC1	Gizmo and its documentation shall adhere to the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.
		Non-Functional	Low	LLC2	Copy of license shall be included in the capstone documentation.
		Non-Functional	Low	LLC3	Gizmo will be used solely for the BIA Capstone Project for Fall 2023.

		Non-Functional	Low	S1	Gizmo can accommodate integration of other features like LED lights, buzzer, and alike.
		Non-Functional	Low	S2	Gizmo can integrate to the code features like LED lights, buzzer, and alike.
		Non-Functional	Low	US1	Freenove application is user-friendly, easy to install and configure.

Category	Code	Requirement Group	Description
FUNCTIONAL	OD	**Obstacle Detection**	
	OD1		The robot shall employ an ultrasonic ranging module to detect obstacles within a specified range around the car during moving in route.
	OD2		The ultrasound system gathers data from multiple directions, evaluates this data independently in each direction, and subsequently manages the car's actions to steer clear of obstacles while moving in its route.
FUNCTIONAL	CA	**Collision Avoidance**	
	CA1		The obstacle avoidance system shall actively control the car's movement to prevent collisions with detected obstacles.
	CA2		When an obstacle is detected, the system shall initiate one or more of the following actions to avoid collision: o Slow down or stop the car. o Steer the car away from the obstacle
FUNCTIONAL	UWE	**Ultrasonic Wave Emission**	

Category	Code	Requirement Group	Description
	UWE1		The ultrasonic ranging module shall be equipped to emit ultrasonic waves in a controlled manner.
	UWE2		The emitted ultrasonic waves shall propagate through the environment and interact with obstacles, causing them to reflect the waves back towards the module.
FUNCTIONAL	TIM	**Time Interval Measurement**	
	TIM1		The system shall precisely measure the time interval between the transmission of ultrasonic waves and the reception of their echoes.
	TIM2		The time difference, measured in microseconds or milliseconds, shall be a reliable indicator of the total travel time of the ultrasonic waves from transmission to reception
FUNCTIONAL	DC	**Distance Calculation**	
	DC1		The module shall utilize the measured time interval to calculate the distance to encountered obstacles based on the speed of sound in the environment.
	DC2		The distance calculation shall provide accurate and real-time information regarding the proximity of obstacles to the car.
FUNCTIONAL	LD	**Light Detection**	
	LD1		The car shall be equipped with two photoresistors, strategically placed at the front of the vehicle to detect variations in light intensity.
	LD2		The system shall utilize the Analog to Digital Converter (ADC) values obtained from the photoresistors to accurately measure the light intensity.
FUNCTIONAL	LTB	**Light Tracing Behavior**	
	LTB1		The car shall be programmed to respond to the detected light source by autonomously steering towards it.
	LTB2		The degree of steering shall be proportional to the difference in ADC values between the two photoresistors, ensuring precise alignment with the light source.
NON-FUNCTIONAL	ER	**Environmental Requirements**	
	ER1		The robot car shall be designed and calibrated to function effectively on a variety of surfaces, including but not limited to carpets, smooth floors, and rough outdoor terrains.
	ER2		The system shall adapt its driving parameters and behavior to ensure stable and reliable performance regardless of the surface type.
	ER3		The system shall demonstrate robust performance in varying lighting conditions, including low-light environments and areas with intense light sources.

Category	Code	Requirement Group	Description
	ER4		The smart car's light tracking behavior shall remain accurate and responsive, adjusting its steering in accordance with changes in light intensity, without significant deviations or errors caused by fluctuations in lighting conditions.
NON-FUNCTIONAL	LLC	**Legal and Licensing Compliance**	
	LLC1		All files, materials, and instructional guides utilized in the development and documentation of this capstone project, including those related to the Freenove 4WD smart car, shall adhere to the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.
	LLC2		A copy of this license shall be prominently displayed in the project's documentation and provided with any derivative works.
	LLC3		The project team shall ensure that any resources, software, or materials utilized are not employed for commercial purposes and are used in strict compliance with the licensing terms and conditions specified.
NON-FUNCTIONAL	PE	**Performance Efficiency**	
	PE1		The system shall aim to maximize its operational time on a single battery charge under typical usage conditions.
	PE2		The system shall strive to provide responsive obstacle detection and avoidance capabilities to ensure the smart car can efficiently respond in dynamic environments.
NON-FUNCTIONAL	RA	**Reliability and Availability**	
	RA1		The smart car shall be designed to operate continuously without any system failures for the duration of its battery capacity, ensuring reliable performance throughout its operational cycle.
	RA2		The system shall include built-in fault detection mechanisms to promptly identify and recover from common errors or sensor malfunctions that may occur during the battery's operational capacity.
	RA3		In case of a critical system failure within the battery's operational capacity, there should be a straightforward and rapid system restart procedure that allows the smart car to resume normal operation.
NON-FUNCTIONAL	S	**Scalability**	
	S1		The system architecture shall be designed to allow for easy integration of additional sensors or modules to enhance the smart car's capabilities.
	S2		The software shall be modular and scalable to accommodate future upgrades and improvements without requiring significant code rewrites.
NON-FUNCTIONAL	UUS	**Usability and User Experience**	

Category	Code	Requirement Group	Description
	UUS1		The user interface for controlling the smart car shall be intuitive and user-friendly, ensuring that operators with varying levels of technical expertise can easily interact with the system
	UUS2		The system shall provide clear and informative feedback to the user, including obstacle detection alerts and status updates, through both visual and auditory cues.
	UUS3		The smart car shall be designed with a physical emergency stop button that is easily accessible to the operator, allowing for immediate manual intervention in case of unexpected behavior or emergencies.