

An-Najah National University

Faculty of Engineering and Information Technology

Computer Engineering Department.



All in one game robot

Prepared by:

Danah Aker and Ataa Shaqour

Supervised by:

Dr. Emad Natsheh and Suleiman Abu Kharmeh

Presented in partial fulfillment of the requirements for Bachelor degree in Computer Engineering.

Acknowledgment:

We would like to express our very great appreciation to our supervisors; Dr. Emad Natsheh and Dr. Suleiman Abu Kharmeh for their time, efforts and for their great advice throughout working on this project.

We would also like to thank all the professors and teaching assistants in the Computer Engineering Department for all the cumulative knowledge we have gained from them over the years, without which we would not have completed this project.

Disclaimer Statement:

This report was written by Danah Aker and Ataa Shaqour, students at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Table of Contents:

Abstract	6
Chapter 1: Introduction.....	7
1.1 Statement of the problem.....	7
1.2 Objectives	7
1.3 Scope of the work	7
1.4 Significance of the work.....	8
1.5 Organization of the report.....	9
Chapter 2: Constrains, Standards/ Codes and Earlier coursework.....	10
2.1 Constraints	10
2.2 Standards and codes.....	10
2.3 Earlier coursework.....	11
Chapter 3: Literature Review.....	12
Chapter 4: Methodology.....	14
Chapter 5: Results and Analysis.....	44
Chapter 6: Conclusion and Recommendations.....	44
References.....	45

List of Figures:

4.1 ESP32-Wroom.....	10
4.2 DC-Gear Motor.....	11
4.3 H-Bridge.....	11
4.4 Ultrasonic (HC-SR04).....	12
4.5 Servo Motor.....	12
4.6 IR Sensor.....	13
4.7 Lithium Batteries.....	13
4.8 Front Ultrasonic Sensor.....	14
4.9 Right Ultrasonic.....	14
4.10 Left Ultrasonic.....	15
4.11 Line Right IR Sensor.....	15
4.12 Line Left IR Sensor.....	15
4.13 Front Right IR Sensor.....	16
4.14 Front Left IR Sensor.....	16
4.15 Servo Motor.....	17
4.16 H-Bridge.....	17
4.17 DC-To-DC Convertor.....	18
4.18 Infrared Reciever Module.....	18
4.19 Line Following Flowchart	19
4.20 Obstacle Avoiding Flowchart.....	20
4.21 Follow Me Flowchart.....	21

4.22 Sumo Mode Main Flowchart.....	22
4.23 Sumo Mode Backoff Algo Flowchart.....	24
4.24 Sumo Mode Search Algo Flowchart.....	25
4.25 Sumo Mode Attack Algo Flowchart.....	26
4.26 Remote Control.....	27
4.27 App main screen.....	28
4.28 Car Robot 1.....	29
4.29 Car Robot 2.....	30
4.30 Car Robot 3.....	30

Abstract

In the world of gaming and entertainment, technology has played a crucial role in providing exciting and engaging experiences for people of all ages. With this in mind, we have decided to create an all-in-one robot game that combines multiple functionalities and brings a new level of excitement to the world of game toys.

The main features this game provides are Line Following mode that enables the robot to track a predefined path, Obstacle Avoiding mode that makes use of ultrasonic sensors to detect and avoid any obstacle in its path, Sumo mode that transforms the robot into a sumo wrestler that pushes other particles out of sumo circle autonomously, and finally companion following that allows the robot to follow designated object.

The robot's combination of sensors, actuators, and microcontrollers allows it to perform a collection of tasks and respond to voice and remote commands. Furthermore, the ability to control the robot's moods through a mobile application provides convenience and ease of use, making the experience even more enjoyable.

Chapter 1: Introduction

1.1 Statement of the problem:

The traditional game toy market lacks a comprehensive and versatile solution that combines multiple functionalities in a single device. Existing game toys often focus on a specific function or game, limiting the overall gaming experience and engagement for users. This creates a gap in the market for an all-in-one robot that integrates various features, such as Following an object, Line Following, Obstacle Avoiding, IR Remote Control, Voice Control, Mobile Application, and Sumo capabilities, to provide a diverse and immersive gaming experience.

1.2 Objectives

The objective of this project is to design, develop, and implement an all-in-one robot game that addresses the limitations of existing games. The goal is to provide users with a diverse and engaging gaming experience while offering user-friendly control options as it can be monitored and controlled either through a mobile application or a remote control.

1.3 Scope of the work

The scope of this project is to design and develop an all-in-one robot game toy with multiple functionalities. The design phase will include the selection of appropriate sensors, actuators, and microcontrollers to support the desired functionalities. The focus is on delivering a versatile and engaging game toy through an iterative design process.

1.4 Significance of the work:

The significance of this work lies in its contribution to the toy industry by offering a comprehensive, engaging, and user-friendly all-in-one robot. Instead of having different games each has one specific feature, our all in one robot provides one playground that combines a set of features together.

1.5 Organization of the report

This report will include a walking through all the process that was done to build this project, starting from the constraints, the standards and technologies that were used, then it describes the full process and features in the project, it then discusses the results and ends with a conclusion which summarizes the whole work.

Chapter 2: Constrains, Standards and Earlier work

2.1 Constraints

- IR sensors that we used are sensitive to sunlight, and that affects the behavior of the robot, until we move to a shadowed place.
- We faced some problems with the bad quality of the car robot body which is the only choice we have in the market.

2.2 Standards:

Arduino IDE is used to program the ESP32 WROOM-32 which we used to communicate with the application over Wi-Fi. For the application development MIT App Inventor was used, a cloud-based online tool for creating applications for Android.

2.3 Earlier work

In order to design and build the system in the most professional way, we highly depended on the following courses that we took during computer engineering program:

1. Microcontrollers course: Gave a background on how to deal with microcontrollers and how to control the hardware parts through them. And the lab Brought theoretical knowledge into practice in dealing with I/O,, motors, etc..
2. Programming courses: Mainly C++ and OOP for writing the codes and algorithms.
3. Electronic circuits courses: These courses highly contributed in dealing with different modules and finding the right power sources and connection for each of them.
4. Microcontrollers lab: it brought theoretical knowledge into practice in dealing with I/O, motors, etc..

Chapter 3: Literature Review

This project was built and developed with various features including Follow Me, Line Following, Obstacle Avoiding and Sumo mode, with two control modes: mobile application and remote control.

Similar projects exist in the literature, offering similar specifications; however, they typically focus on providing each feature separately. In contrast, this project takes a unique approach by integrating various features and functionalities into one system.

Furthermore, while previous projects typically offer a single method of controlling parameters, this project stands out by implementing two different control mechanisms. By offering these dual control options, users have the flexibility to interact with the all-in-one robot according to their preferences and convenience.

The aggregation of multiple features into a single system is a key aspect of this project's significance. By combining these functionalities, this project offers a complete, integrated approach.

In summary, the literature reveals that while similar projects exist, this all-in-one robot project sets itself apart by integrating multiple features and providing dual control options. The project's contribution lies in its ability to create a comprehensive and innovative system that caters to various user preferences, offering a seamless and immersive all-in-one robot experience.

Chapter 4: Methodology

In this chapter, we are going to talk about the design of the system, the components that were used to accomplish it, and discuss the development process and the outcome at the end of that process afterward.

4.1 Hardware components

4.1.1 ESP32 WROOM

The ESP32 WROOM-32 is a popular and widely used microcontroller module based on the ESP32 system-on-chip (SoC). It features a powerful 32-bit dual-core processor, integrated Wi-Fi and Bluetooth capabilities, and a wide range of input/output pins, making it suitable for a variety of IoT (Internet of Things) applications. The ESP32 WROOM-32 module provides a convenient and compact form factor for embedding the ESP32 chip into electronic projects, offering capabilities for wireless communication, data processing, and sensor integration.

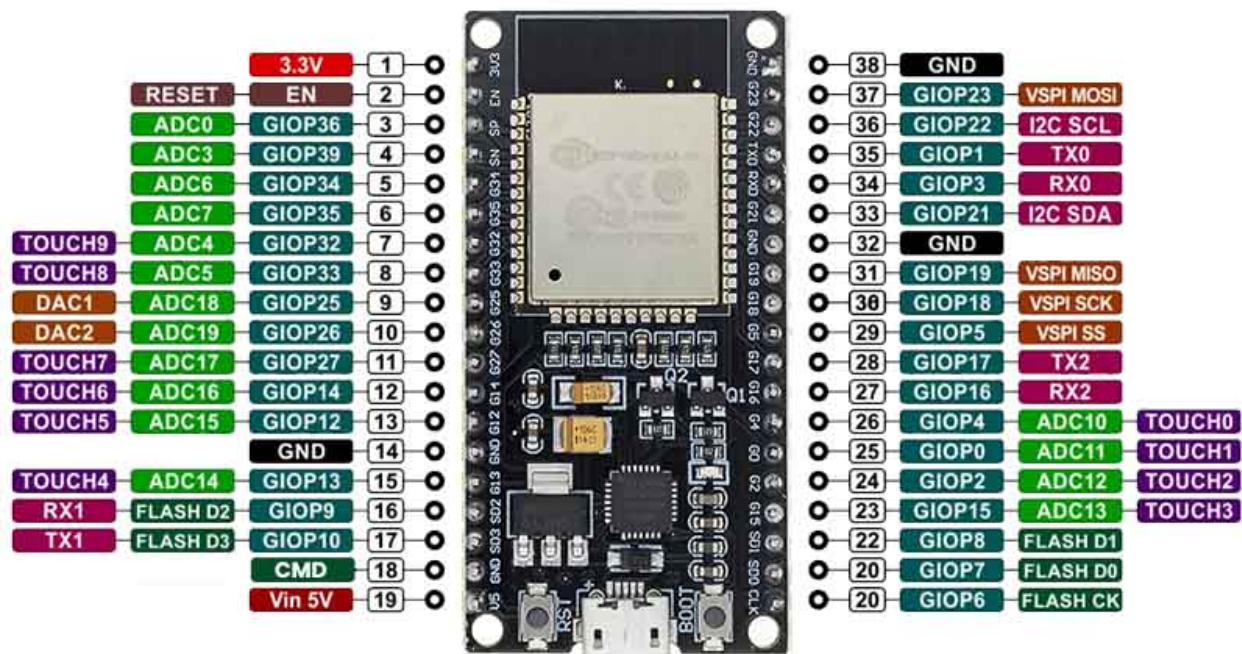


Fig 4.1 ESP32-Wroom

4.1.2 DC gear motor

DC motors are the most common type of motor to move wheels of cars. DC motors normally have just two leads, one positive and one negative. If the two leads were connected directly to a battery, the motor would rotate. If the leads were switched, the motor would rotate in the opposite direction. We needed 4 DC motors to control the 4 wheels of the car robot.



Fig. 4.2 DC-Gear Motor

4.1.3 H bridge

Was used to control the direction of the spin of DC motor, without changing the way the leads are connected. An H bridge is an electronic circuit that can drive the motor in both directions. H-bridges are used in many different applications, one of the most common being to control motors in robots. It is called an H-bridge because it uses four transistors connected in such a way that the schematic diagram looks like an "H".



Fig. 4.3 H-Bridge

4.1.4 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. We used 3 of it to have a wider view for the robot, we put one to the right, one to the left and the third in the front. Three of them were used in these features: follow me, sumo and obstacle avoiding modes.



Fig. 4.4 Ultrasonic (HC-SR04)

4.1.5 Servo Motor

A servo motor is a self-contained electrical device that rotates parts of a machine with high efficiency and with great precision. It serves us in the obstacle avoidance feature. We used it to move the ultrasonic in 180 degree, to help the ultrasonic look for an obstacle.



Fig. 4.5 Servo Motor

4.1.6 IR Sensor

(Infrared Sensor) is an electronic instrument used to detect any kind of Obstacles / objects. We utilized 4 of them, two were needed in the line follower feature and another two for both, the follow me and the sumo modes.

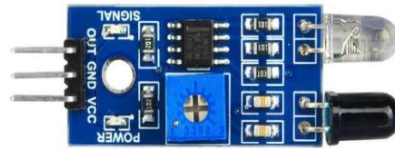


Fig. 4.6 IR Sensor

4.1.7 Lithium Batteries

We used a rechargeable battery with 4.2V.



Fig. 4.7 Lithium Batteries

4.2 system design

The main control device that is used in this project is ESP32-Wroom. We chose this device because it has 32-pins that we needed the most of. Also, it has an integrated wifi and bluetooth.

The following figures show the ports that connected with the ESP32-Wroom.



Fig. 4.8 Front Ultrasonic Sensor

We used this ultrasonic sensor to measure the distance from the Front side.



Fig. 4.9 Right Ultrasonic Sensor

We used this ultrasonic sensor to measure the distance from the right side.



Fig. 4.10 Left Ultrasonic Sensor

We used this ultrasonic sensor to measure the distance from the Left side.

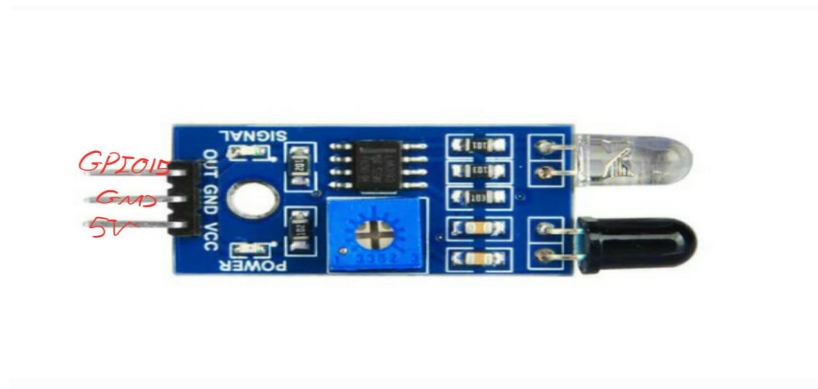


Fig. 4.11 Line Right IR Sensor

We used this IR sensor to detect the black line from the Left side.



Fig. 4.12 Line Left IR Sensor

We used this IR sensor to detect the black line from the right side.

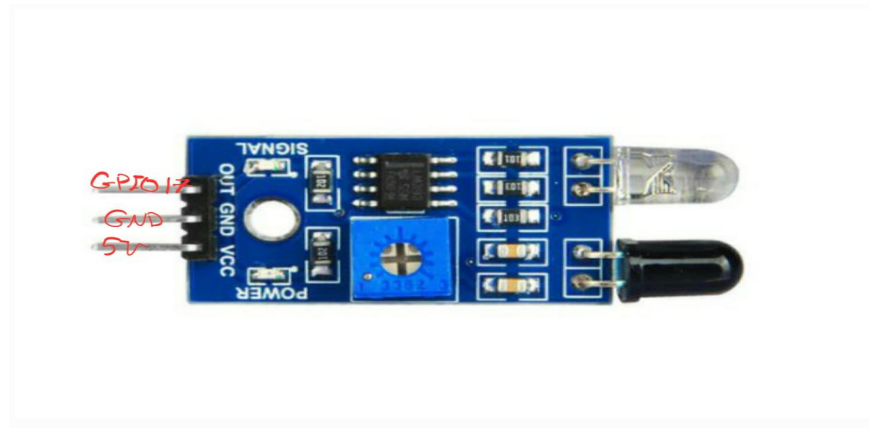


Fig. 4.13 Front Right IR Sensor

We used this IR sensor to detect the object from the right side.



Fig. 4.14 Front Left IR Sensor

We used this IR sensor to detect the object from the left side.



Fig. 4.15 Servo Motor

We used this servo motor to rotate the front ultrasonic.

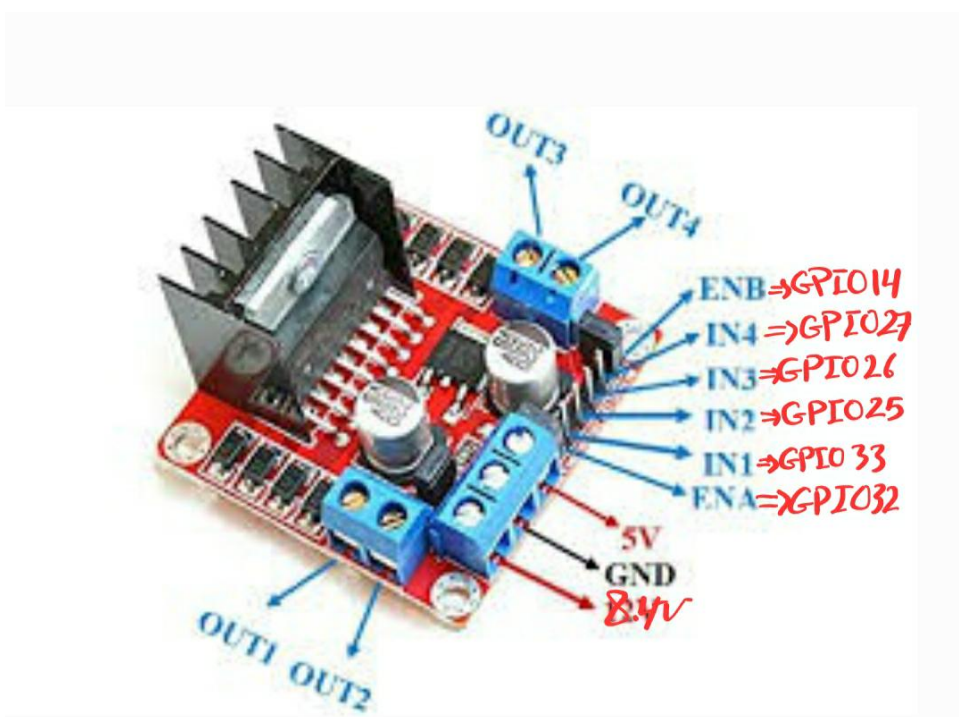


Fig. 4.16 H-Bridge

We used this H-bridge to control the direction of the spin of the DC motor.



Fig. 4.17 DC-To-DC Convertor

We used this convertor module to convert the input voltage from 8.4V to 5V. To provide the sensors and the ESP32-Wroom with the suitable voltage.

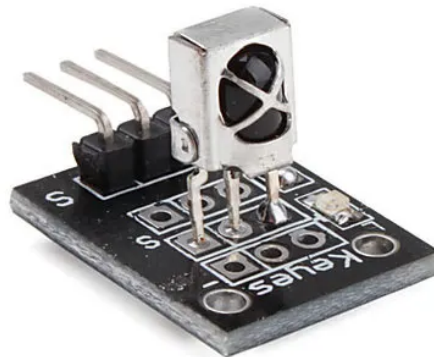


Fig. 4.18 Infrared Receiver Module

It sends information from an infrared remote control to another device by receiving and decoding signals.

4.3 process of work

In this section we will demonstrate the process and the algorithm of each feature of the all in one game robot.

- Line following mode:

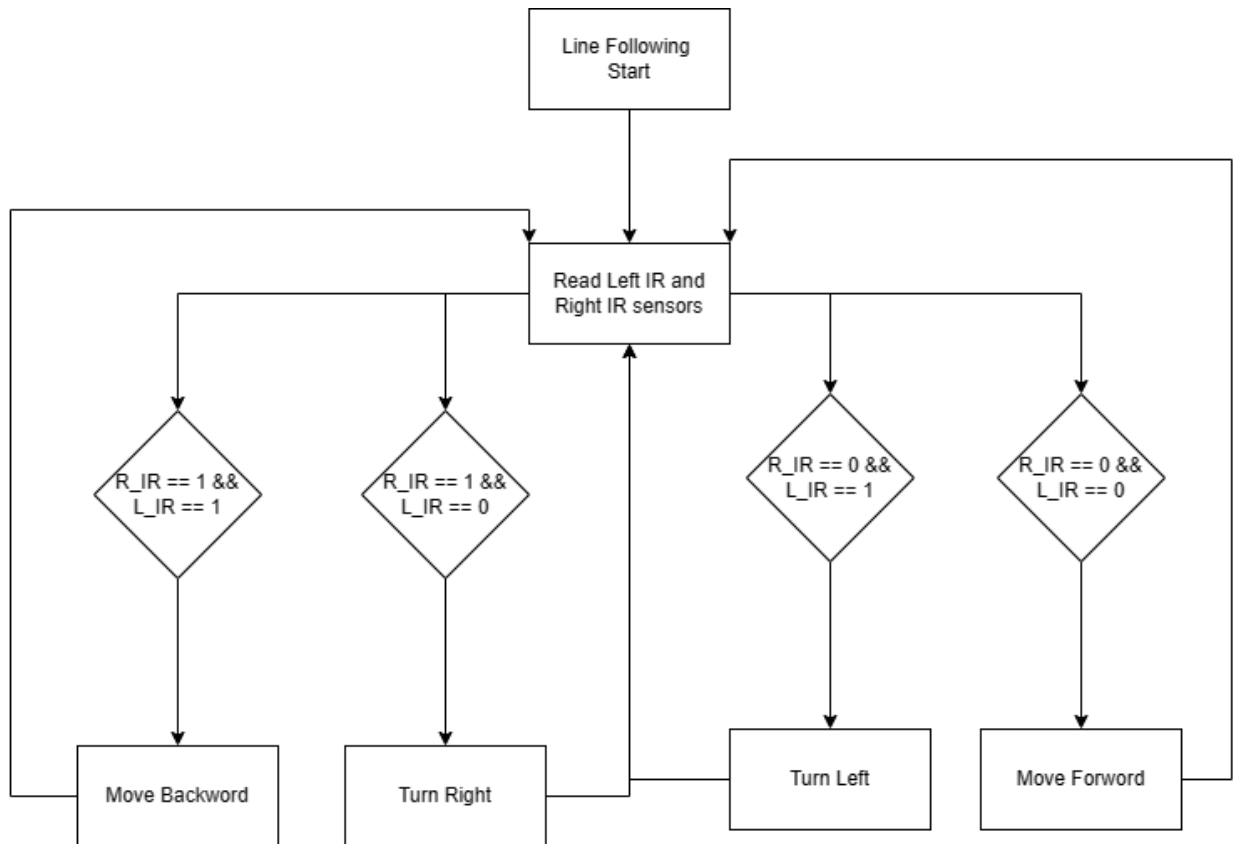


Fig. 4.19 Line Following Flowchart

As appears in the diagram, the line following feature of the robot relies on two main infrared sensors (left and right) to guide its movement. When both the left and right sensors detect the line (outputting a value of zero), the robot continues moving forward along the line. If one of the sensors detects the line (one) while the other does not (zero), the robot adjusts its direction accordingly, veering either left or right to realign itself with the line.

- Obstacle avoiding:

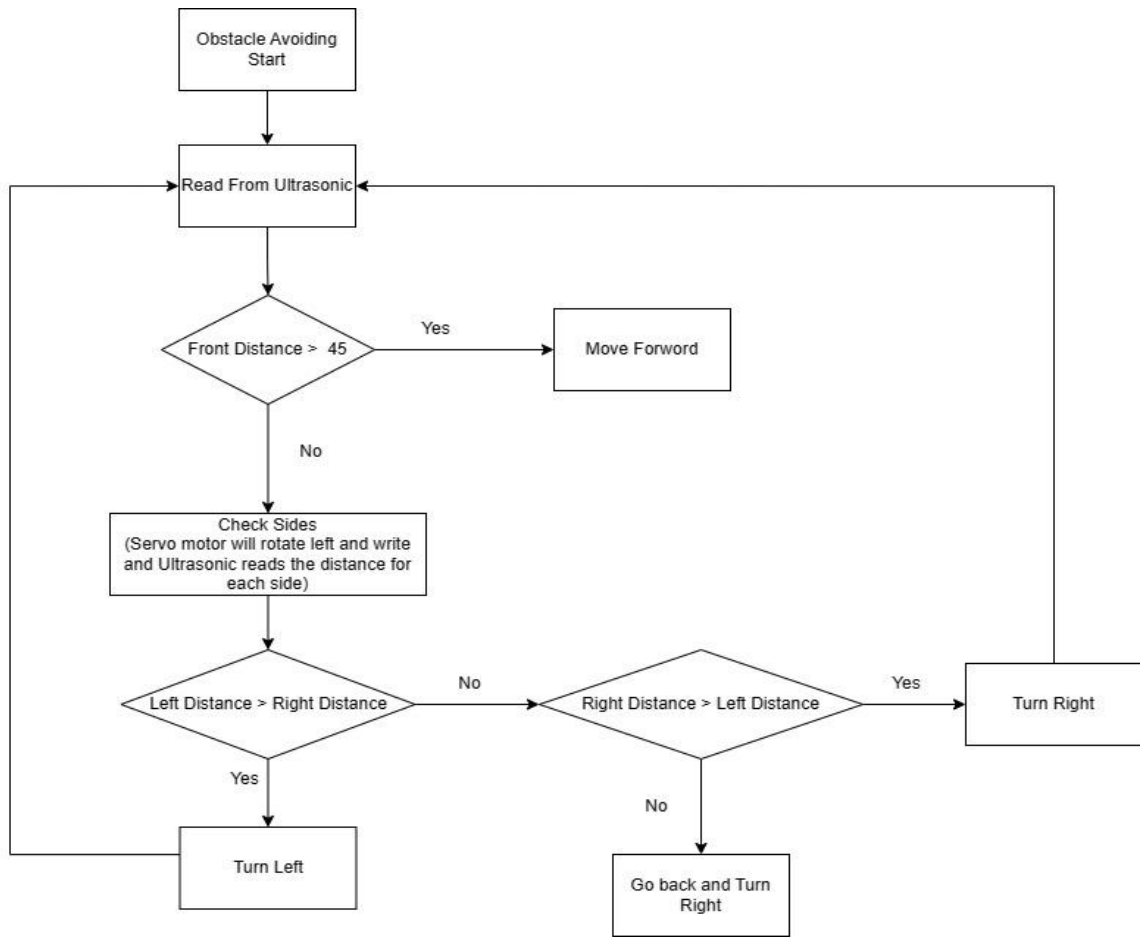


Fig. 4.20 Obstacle Avoiding Flowchart

In obstacle avoiding mode, the robot uses an ultrasonic sensor to measure the distance in front of it. Depending on the measured distance, the robot decides its movement. If the distance in front is above 45 cm, it means there is no obstacle, so the robot moves forward while continuously checking the distance. However, if the distance is less than 45 cm, the robot takes another action. It first turns right and measures the distance, then turns left and measures the distance again. Based on these measurements, the robot determines which direction to turn. If both the right and left distances are small, it means there is an obstacle blocking the path entirely. In that case, the robot goes back and then turns right to find an alternate path. This way, the robot can effectively avoid obstacles and navigate its environment without colliding with any objects.

- Following me mode:

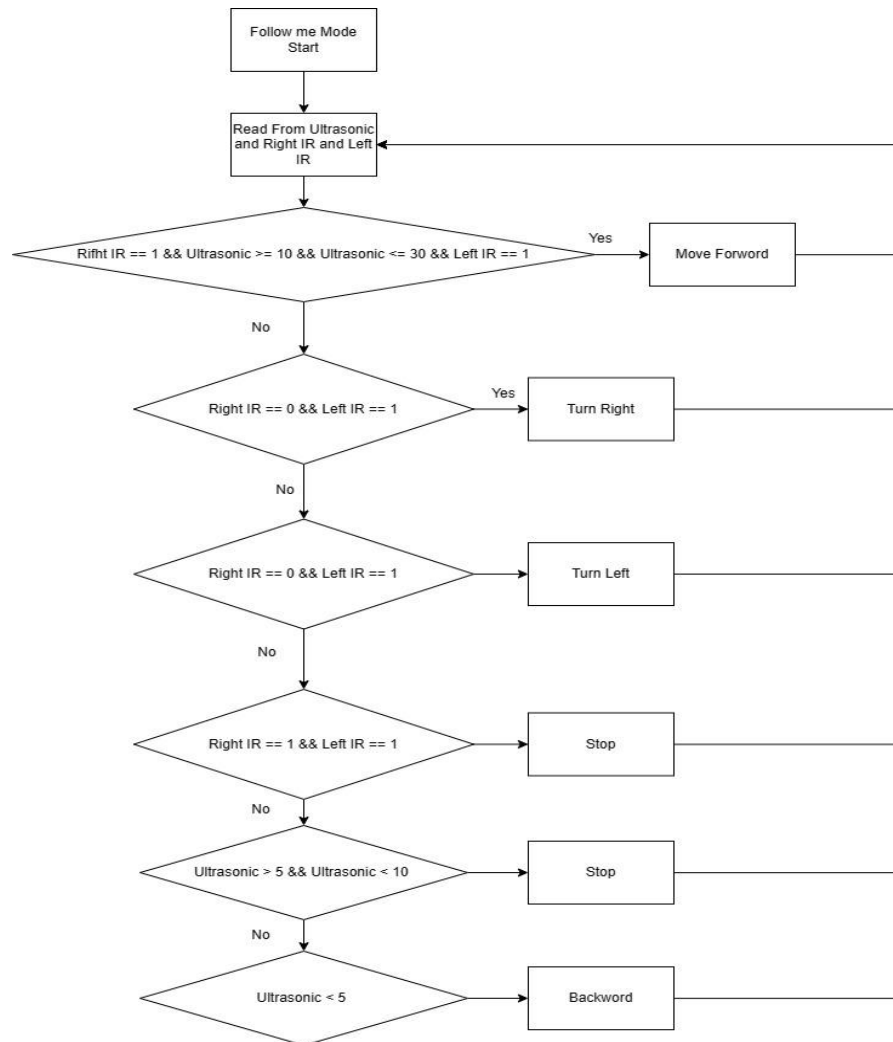


Fig. 4.21 Follow Me Flowchart

In the "follow me" mode, we make use of three sensors: the ultrasonic sensor to measure the distance in the front, as well as the left and right IR sensors. So here's how it works: first, the robot checks the readings from the right and left sensors. If there's no object detected by either sensor, and the ultrasonic sensor measures an object at a distance greater than 10 cm but less than 30 cm, then the robot starts moving forward. Now, if there's an object detected on either the right or left side, the robot changes its behavior to follow that object. However, if both the right and left sensors detect an object at the same time, or if the ultrasonic sensor measures an object at a distance between 5 cm and 10 cm, the robot stops. Lastly, if the ultrasonic sensor measures a

distance less than 5 cm, indicating a very close object, the robot reverses and moves backward to create a safe distance.

- Sumo mode:

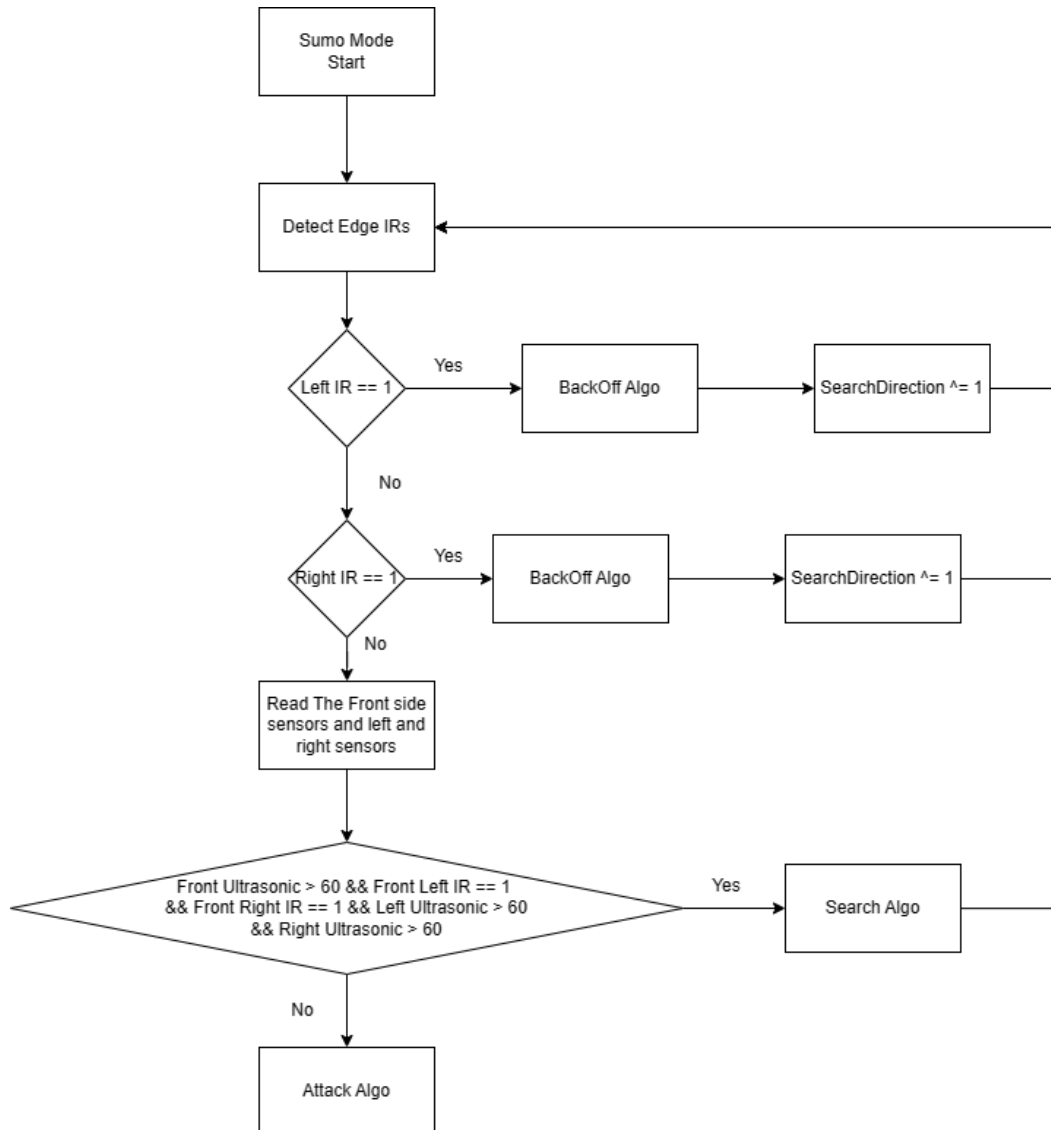


Fig. 4.22 Sumo Mode Main Flowchart

In the "Sumo" mode, we make use of seven sensors: the front ultrasonic sensor to measure the distance in the front, as well as the left and right Ultrasonic sensors. In addition, we used the left and right line IR sensors to detect the black line and the front left and front right IR sensors to detect the opponent object . So here's how it works: first, the robot checks if it reached the black line (ring) or not. If it reaches the black line it will activate the Backoff algorithm (as we will explain in fig. 4.23) and then it will

toggle the searchDirection variable to change the direction of the car to reach different angles. Now if the right and left line IR sensors didn't detect the black line. It will read the measured distance from the front, right and left ultrasonic. If the measured distance for the ultrasonics are greater than 60cm and the front left and front right IR sensors didn't detect an object in front of them. Then it will activate the search algorithm (as we will explain in fig. 4.24) to search for an object inside the ring. On the other hand, it will activate the Attack (It will be explained in fig. 4.25) mode and push the object outside the ring.

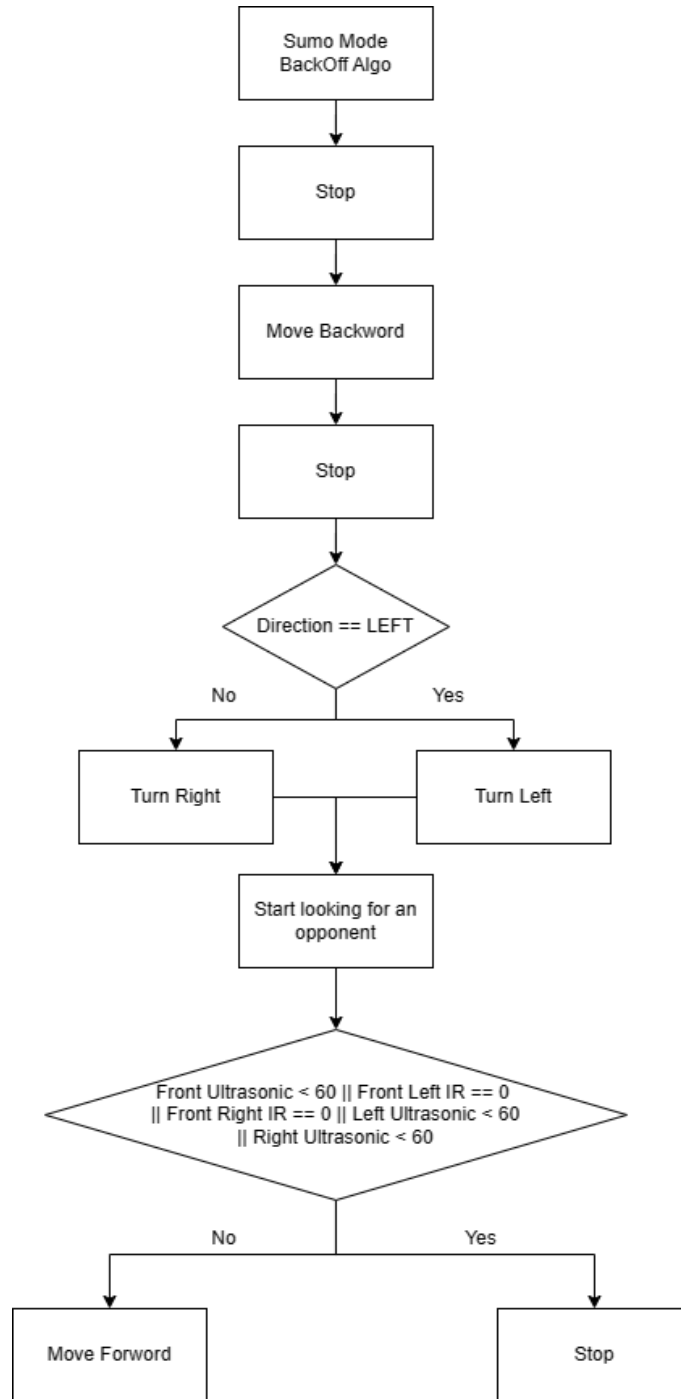


Fig. 4.23 Sumo Mode Backoff Algo Flowchart

When the left and right line IR sensors detect the black line. The car will stop and then it will move backward, and then it will stop. And then see if the direction is equal to left, the car robot will turn to left. If it doesn't, the car robot will turn to the right. After that it starts looking for an opponent, if one of the sensors detect an object, The car robot will stop. If it didn't, the car robot will move forward.

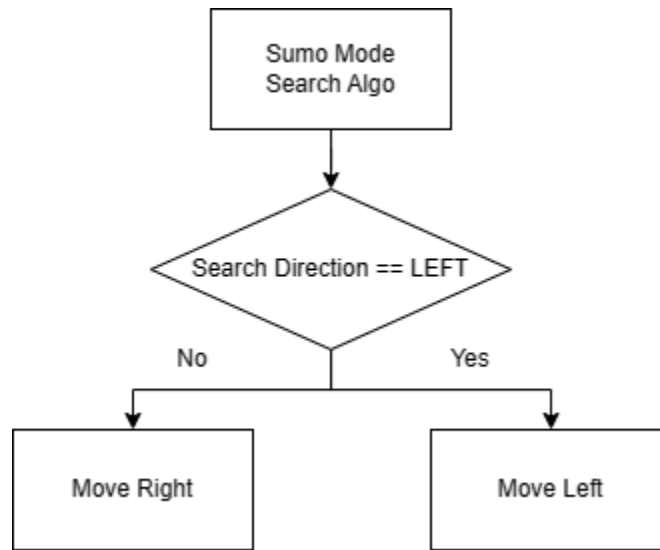


Fig. 4.24 Sumo Mode Search Algo Flowchart

In the search algorithm, if the search direction equals to left. The car robot will move to the left. If it doesn't, the car robot will move to the right.

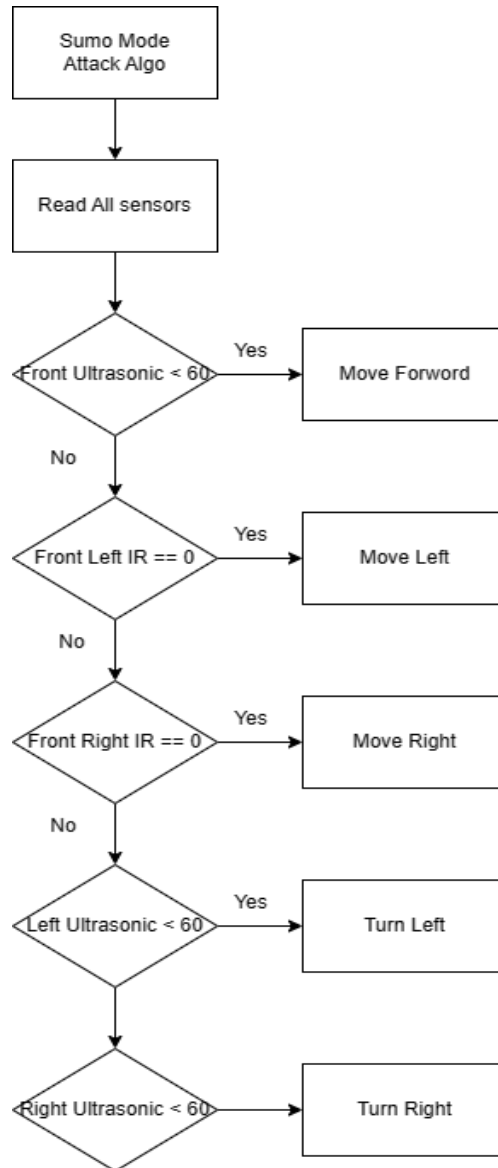


Fig. 4.25 Sumo Mode Attack Algo Flowchart

In the Attack algorithm, The car robot will get the measurements from all the sensors. If the distance we got from the front ultrasonic is less than 60cm. Then, The car robot will move forward.

If it didn't, then if the front left IR sensor detected an object. Then, the car robot will move to the left. If it didn't, Then if the front right IR sensor detected an object. Then, the car robot will move to the right. If it didn't. Then, if the distance we got from the left ultrasonic is less than 60. Then, the car robot will turn to the left. If it didn't. Then, if the distance we got from the right ultrasonic is less than 60. Then, the car robot will turn to the right.

- Manual controlling mode:

In this mode, we can control the movement of the robot using an IR remote control. This feature provides an intuitive and interactive method for users to navigate the robot in different directions and different modes, we can control its movement to go right, left, forward or backward, or select the mode to follow from the modes we previously mentioned.

By pressing the appropriate buttons on the remote control, users can send infrared signals to the robot, triggering the corresponding actions in real-time. The robot receives and interprets these signals, translating them into commands to execute the desired movements.



Fig. 4.26 Remote control

The following shows the keys and what it will do when the user press on it:

- + => Move Forward.
- - => Move Backward.
- ► => Turn Right.
- ◀ => Turn Left
- 1 => Line Following Mode.
- 2 => Manual Mode.
- 3 => Obstacle Avoiding Mode.
- 4 => Follow Me Mode.
- 5 => Sumo Mode.

- Voice commands:

The app gets the sound command via google assistant, and it converts it to a text. After getting the text we compare it with the following texts so each text have a specific action:

- "Go Forward" => Move Forward.
- "Go Backward" => Move Backward.
- "Turn Right" => Turn Right.
- "Turn Left" => Turn Left.
- "Stop" => Stop.

After getting the correct text it sends a http request to the ESP32-Wroom via wifi connection and it does that action.

4.4 software application



Fig. 4.27 App main screen

We utilized the MIT App Inventor, an open-source online platform, to develop the mobile application for our game robot. Leveraging the ESPAsyncWebServer library and the built-in Wi-Fi module in the ESP,, we established communication between the mobile application and the robot through HTTP requests.

The mobile application features a single interface, as appears in the accompanying image, enabling comprehensive control over all the provided features.

In the first section of the screen, users can effortlessly command the robot's movement in multiple directions using four buttons or voice commands.

The second section of the screen allows users to select the desired operating mode for the robot. They can choose between line following, obstacle avoidance, sumo, follow me, or manual control to move using the remote control feature.

Additionally, the application incorporates a slider that enables users to adjust the robot's speed according to their preferences

4.5 Final Product

This section will show figures of the final Car Robot.

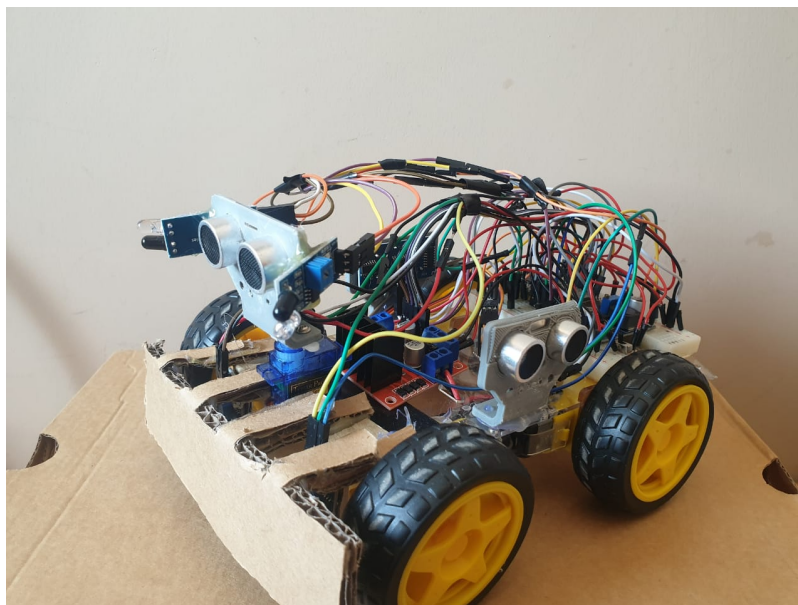


Fig. 4.28 Car Robot 1

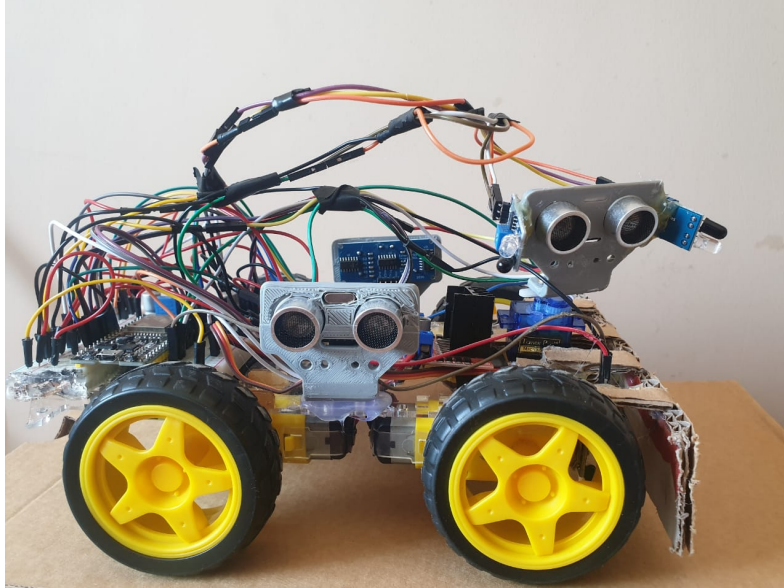


Fig. 4.29 Car Robot 2



Fig. 4.30 Car Robot 3

Chapter 5: Results and discussion

In the end, we successfully built our toy robot with all the desired features, including line following, obstacle avoidance, follow me, and sumo modes. Additionally, we incorporated two methods of control: IR remote control and a mobile application. The integration of Wi-Fi connectivity and voice recognition further enhanced the robot's capabilities. Overall, our robot showcases a comprehensive set of features, enabling versatile control options and providing an interactive and engaging experience for users.

Chapter 6: Conclusion and Recommendations

Conclusion :

In conclusion, our all-in-one toy robot project has been a resounding success. We have successfully designed and developed a robot that incorporates a wide range of features.

Recommendations:

We have some features we would like to add to our application as future plans.

- Gesture Recognition: Implementing gesture recognition technology allows the robot to interpret and respond to user gestures. This feature could enable users to control the robot's actions or trigger specific behaviors through hand movements
- We would like to improve the sumo mode to be able to battle with more than one object
- We would also build a cross platform app that could be used in both android and iphone mobile phones.
- We would improve the material which the car robot made up of.

References:

1. [MIT App Inventor | Explore MIT App Inventor.](#)
2. ESP32 WROOM-32 data sheet. (2022). Retrieved May 26, 2022 from https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf