Faculty of Information Technology

IN4300 Embedded System

Level 4

Smart Navigating Robot

Group Name: TechWizards

Faculty of Information Technology

University of Moratuwa

2025

Faculty of Information Technology

IN4300 Embedded System

Level 4

Smart Navigating Robot

Group Name: TechWizards

Group Members

Index Number	Name
204116V	Liyanage V.P.S
204119H	Madhuwanthi W.A.I
204121G	Mahathanthri M.H.H
204148U	Pallikkonda D.C.D

Faculty of Information Technology

University of Moratuwa

2025

Table of Contents

Table of Figures

1. Introduction

The Smart Robot is an autonomous robotic vehicle designed to be controlled and monitored through a web-based interface. Powered by a NodeMCU ESP32 (38-pin) microcontroller, the robot integrates various components such as motor controllers, IR speed sensors, and a user-friendly web page for seamless interaction. This project aims to provide a flexible and efficient robotic platform that can be remotely controlled and programmed to follow user-defined paths with minimal human intervention.

The robot is equipped with real-time motion feedback using IR speed sensors, enabling it to make precise adjustments during movement. Users can interact with the robot through a web page, where they can draw custom paths on a grid, start the vehicle, or clear the path. The web interface communicates with the ESP32 microcontroller via the microcontroller server API, ensuring real-time control and monitoring.

To ensure stable power distribution, a buck converter regulates voltage levels, preventing potential fluctuations that may affect motor performance. The robot's modular design allows for easy integration of additional sensors or modules, making it highly adaptable for various applications.

2. Aim & Objectives

2.1 Aim

- To design and develop an autonomous smart car capable of following a user-defined path with real-time command execution via a web-based interface

2.2 Objectives

- Implement a robust motor control system using an ESP32 microcontroller.
- Develop a web-based interface for drawing and controlling movement paths.
- Integrate IR speed sensors for real-time feedback and motion correction.
- Utilize web server communication for real-time interaction.
- Explore future improvements such as obstacle detection and dynamic speed adjustments.

3. Project Components

3.1 Hardware Components

Components	**Purpose**
ESP32 Microcontroller	Processes commands and controls the system.
L298N Motor Driver	Drives the motors based on ESP32 signals.
Buck Converter	Provides stable 5V power for sensitive components.
IR Speed Sensor	Measure wheel rotation for calculate the distance
DC Geared Motors	Provide movement.
11.1V Lipo Battery	Powers the entire system.

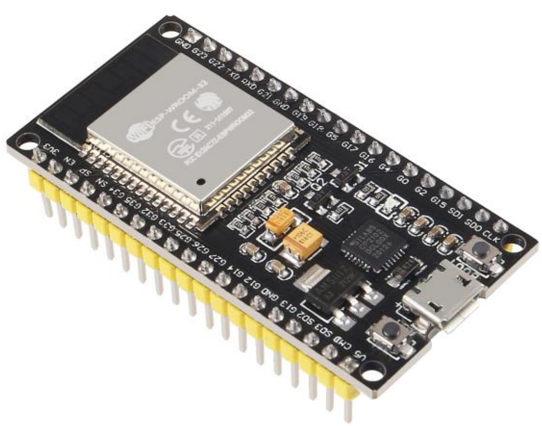

Figure 1 NodeMCU wroom-32 ESP32 (38 pins) board

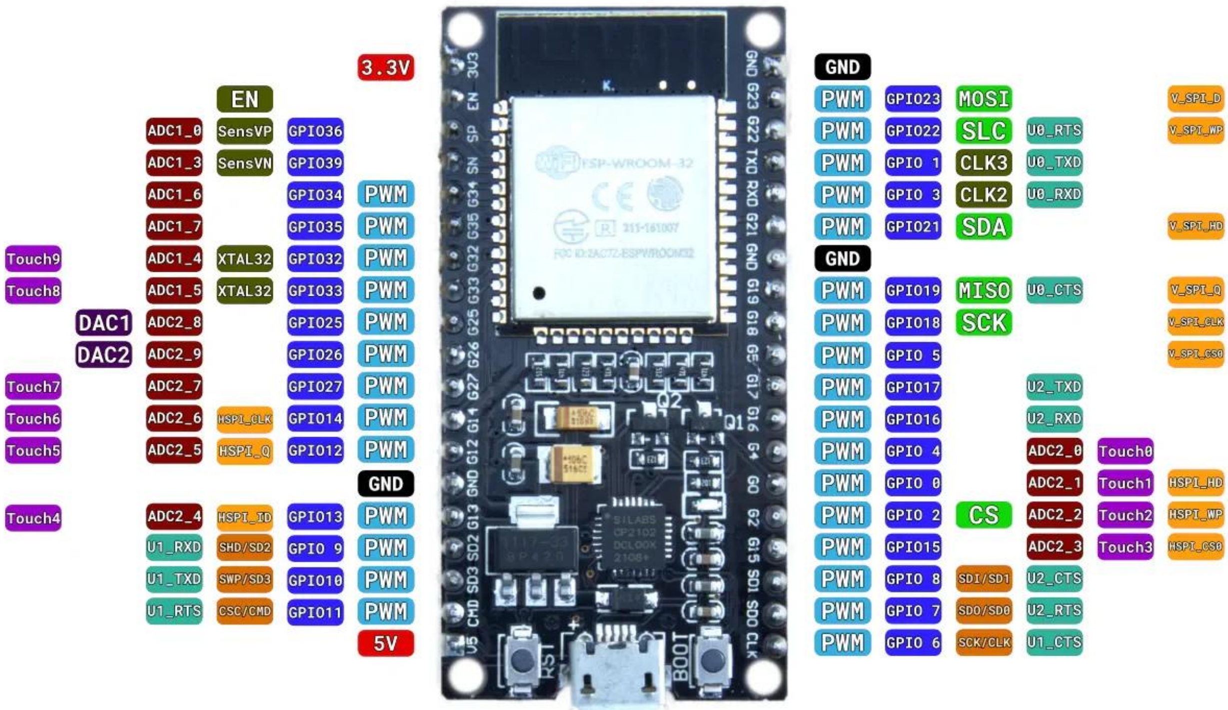

Figure 2 NodeMCU wroom-32 pin diagram

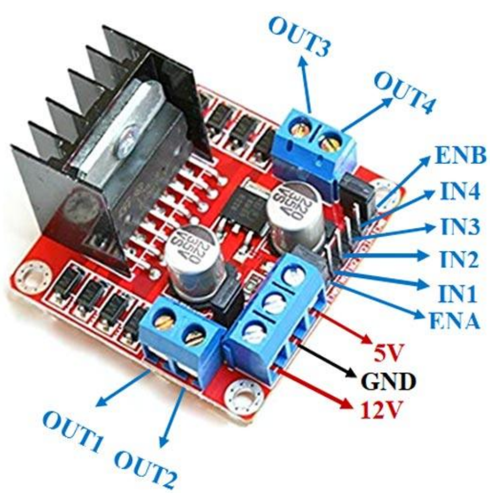

Figure 3 L298N Motor Driver

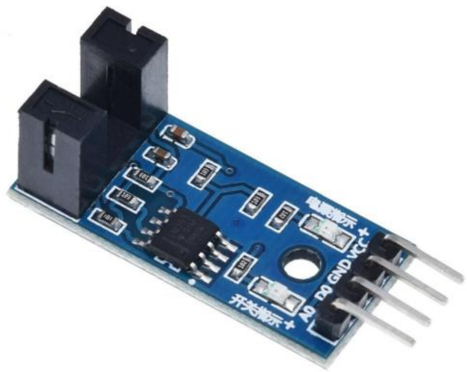

Figure 4 IR Speed Sensor Module (Analog & Digital) Mouse Encoder U-Type Slot Type (MD0369)

3.2 Software Components

Components	**Purpose**
Path-drawing web Interface	Allows the user to draw a path on a grid-based system.
Low Level Motor Controller	Handling motors to go forward, stop, turn right, and turn left. this contains low-level commands for motor controlling
WiFi communicator	Facilitates communication between the web interface and the ESP32
Speed sensor-based Functions	Software logic that uses data from the IR speed sensors to adjust motor movements.
Path String Decorder and Motor Controlling	decode the path string and control the motors according to the path string.

3.3 Circuit Diagram & Wiring

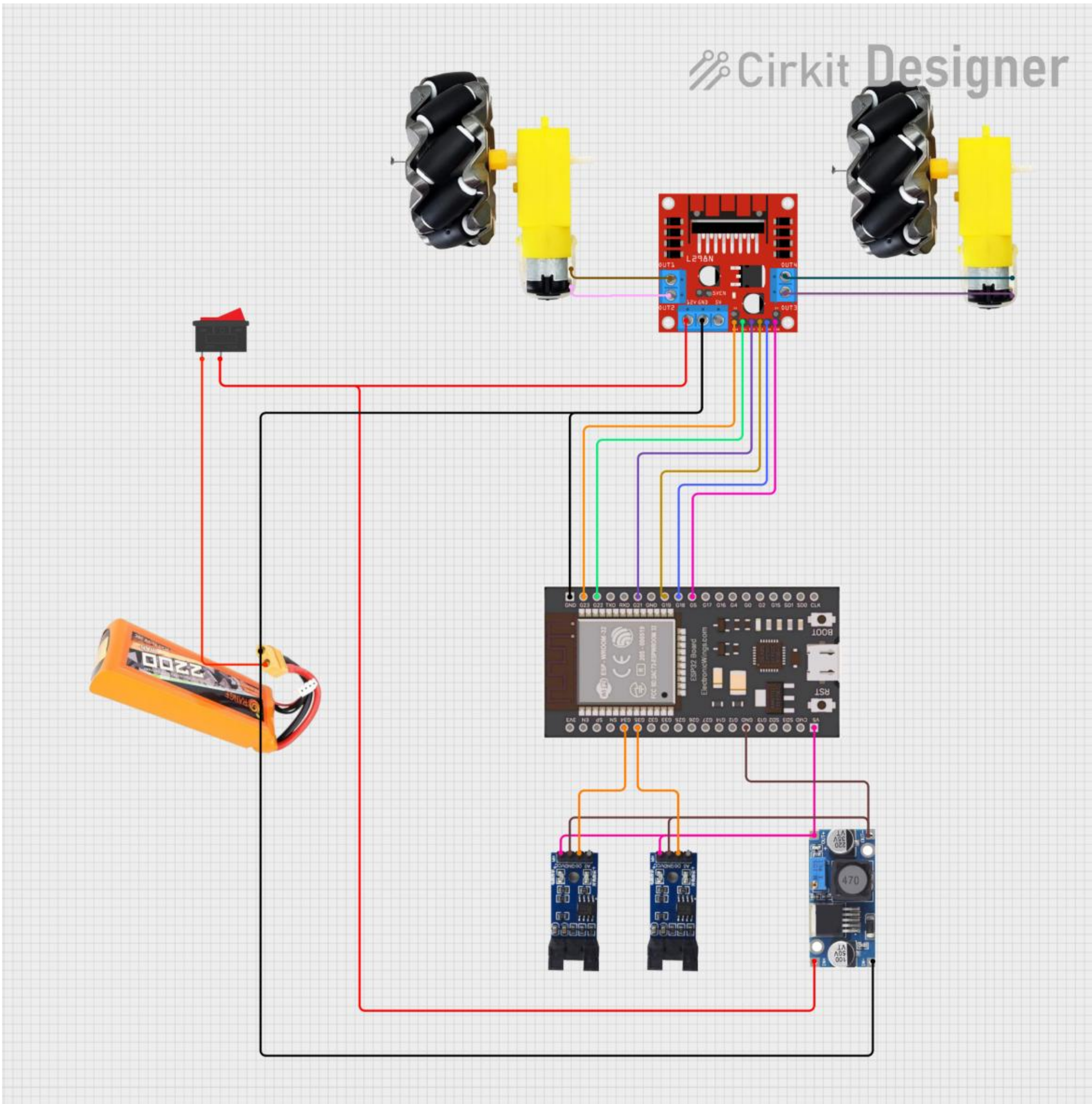

Figure 5 Smart Navigating Robot circuit diagram

4. System Architecture

4.1 Block Diagram of the system

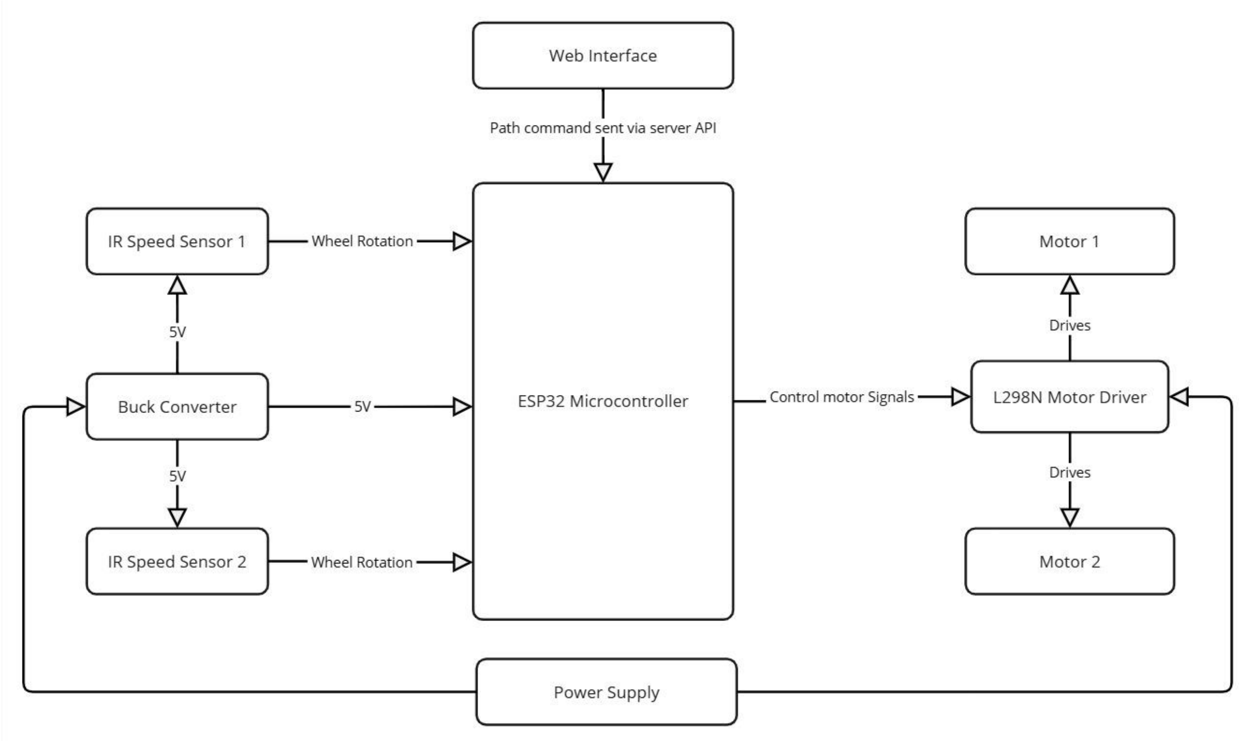

Figure 6 Smart Navigating Robot block diagram

Explanation of the Block Diagram:

1. NodeMCU ESP32 Microcontroller
    - The main processing unit that controls the system.
    - Receives commands from the web interface and feedback from speed sensors.
    - Sends control signals to the motor driver.
2. L298N Motor Driver
    - Controls the speed and direction of the DC geared motors.
    - Takes signals from the ESP32 and drives the motors using the 11.1V battery.
3. Buck Converter
    - Converts the 11.1V from the LiPo battery to 5V.
    - Powers the ESP32 and IR speed sensors.
4. IR Speed Sensors
    - Measure wheel rotation to provide feedback on speed and distance.

- Send real-time data to the ESP32 for adjustments.

5. DC Geared Motors
    - Provide mechanical movement to the system.
    - Controlled by the L298N motor driver.
6. 11.1V LiPo Battery
    - Supplies power to the entire system.
    - Provides 11.1V to the motor driver and buck converter.

4.2 System Workflow

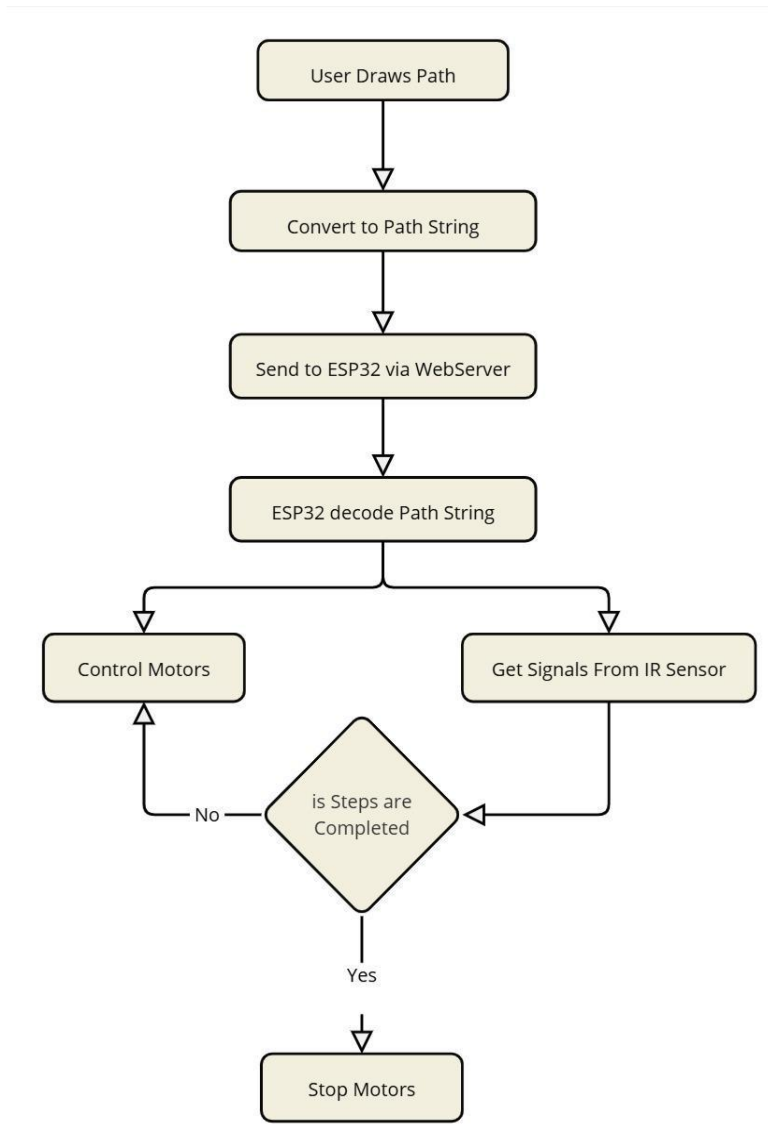

Figure 7 Smart Navigating Robot workflow

5. Hardware and Software Implementation

The hardware implementation involved assembling and connecting components such as the ESP32, L298N motor driver, IR speed sensors, and DC geared motors. The software implementation included developing a web interface, ESP32 firmware, and a server API for real-time communication. The integration of hardware and software enabled the robot to follow user-defined paths accurately.

5.1 Hardware Implementation

5.1.1 Component Assembly

ESP32 Microcontroller: The central processing unit was mounted on the robot chassis. It is connected to the L298N motor driver, IR speed sensors, and power supply using the following pins:

- IR Speed Sensors:
  - Left Sensor: Connected to GPIO 34 (#define SENSOR_LEFT 34).
  - Right Sensor: Connected to GPIO 35 (#define SENSOR_RIGHT 35).
- L298N Motor Driver:
  - Left Motor:
    - PWM: GPIO 5 (#define MOTOR_LEFT_PWM 5).
    - IN1: GPIO 18 (#define MOTOR_LEFT_IN1 18).
    - IN2: GPIO 19 (#define MOTOR_LEFT_IN2 19).
  - Right Motor:
    - PWM: GPIO 23 (#define MOTOR_RIGHT_PWM 23).
    - IN1: GPIO 21 (#define MOTOR_RIGHT_IN1 21).
    - IN2: GPIO 22 (#define MOTOR_RIGHT_IN2 22)

L298N Motor Driver: Connected to the ESP32 for motor control and to the DC geared motors for movement. The motor driver was powered directly by the 11.1V LiPo battery.

IR Speed Sensors: mounted near the wheels to measure rotation. The sensors were connected to the ESP32's digital input pins (GPIO 34 and 35) for real-time feedback.

Buck Converter: Used to step down the 11.1V battery voltage to 5V for powering the ESP32 and IR sensors.

DC Geared Motors: Attached to the wheels and connected to the L298N motor driver for movement.

11.1V LiPo Battery: Provided power to the entire system. It was connected to the buck converter and motor driver.

5.1.2 Wiring and Connections

- The ESP32 was connected to the L298N motor driver using GPIO pins for control signals (e.g., IN1, IN2, IN3, IN4).
- The IR speed sensors were connected to the ESP32's analog input pins (GPIO 34 and 35) for feedback.
- The buck converter was wired to the battery and provided 5V output to the ESP32 and sensors.
- The motors were connected to the L298N motor driver's output terminals.

5.1.3 Chassis and Mechanical Setup

- The robot chassis was designed to accommodate all components securely.
- The motors were mounted on the chassis, and the wheels were attached to the motor shafts.
- The IR sensors were positioned close to the wheels to accurately measure rotation.

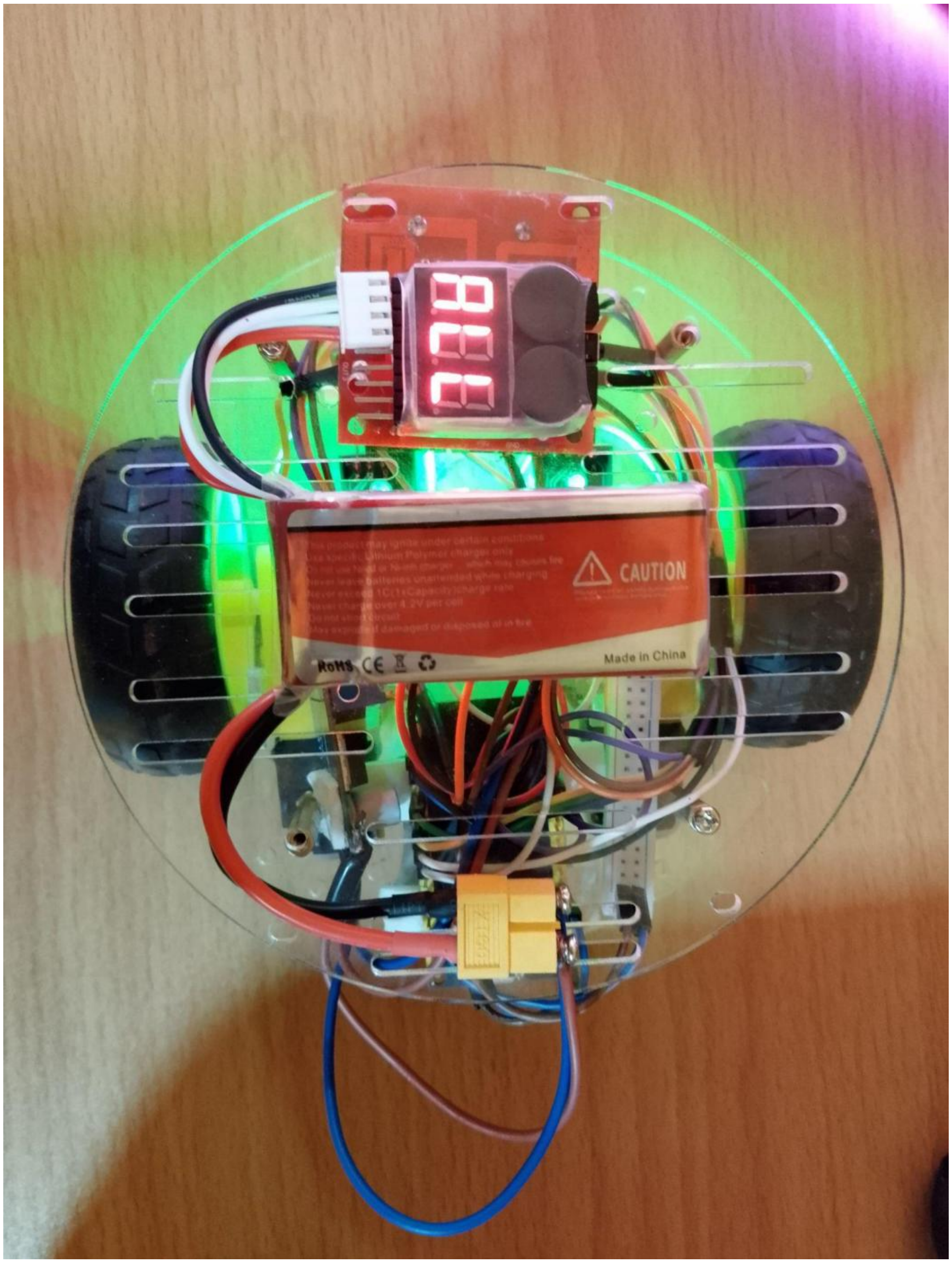

Figure 8 Smart Navigating Robot Image 01

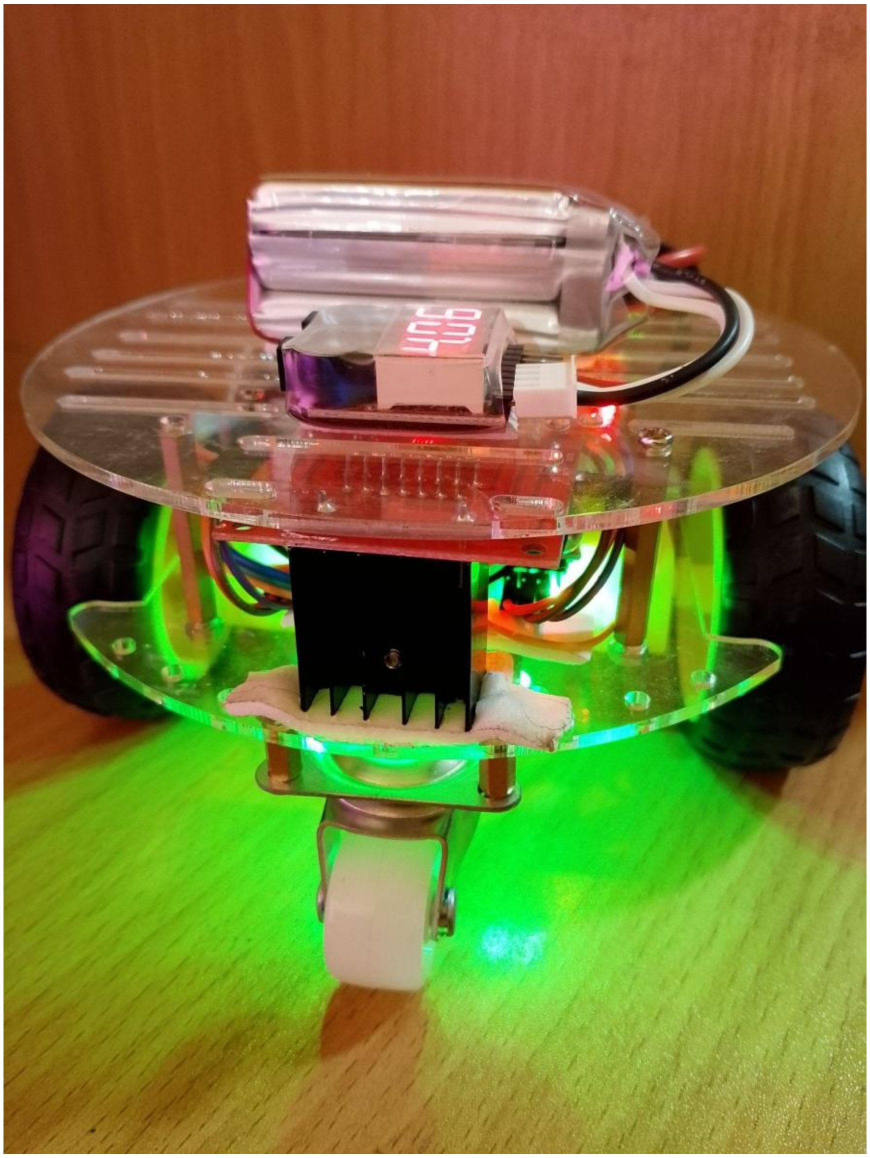

Figure 9 Smart Navigating Robot Image 02

5.2. Software Implementation

5.2.1 Web Interface Development

- A web-based interface was created using NextJS and TypeScript.
- The interface allowed users to draw paths on a grid and send commands to the ESP32 via a server API.

- WebServer communication was implemented for real-time interaction between the web interface and the ESP32.

5.2.2 ESP32 Firmware

The ESP32 was programmed using the Arduino IDE.
The firmware included:

- Path Parsing: The ESP32 received path commands from the web interface and parsed them into movement instructions.
- Motor Control: The ESP32 sent control signals to the L298N motor driver to control the motors' speed and direction.
- Feedback Loop: The ESP32 used data from the IR speed sensors to adjust motor movements in real-time.

5.2.3 Server API

A server API was developed to facilitate communication between the web interface and the ESP32.
The API handled:

- Sending path commands from the web interface to the ESP32.
- Receiving status updates (e.g., sensor data) from the ESP32.

5.2.4 Feedback Control Algorithm

- A feedback control algorithm was implemented to ensure accurate path tracking.
- The algorithm used data from the IR speed sensors to:
- Calculate the distance traveled.
- Adjust motor speed and direction to correct deviations from the path.

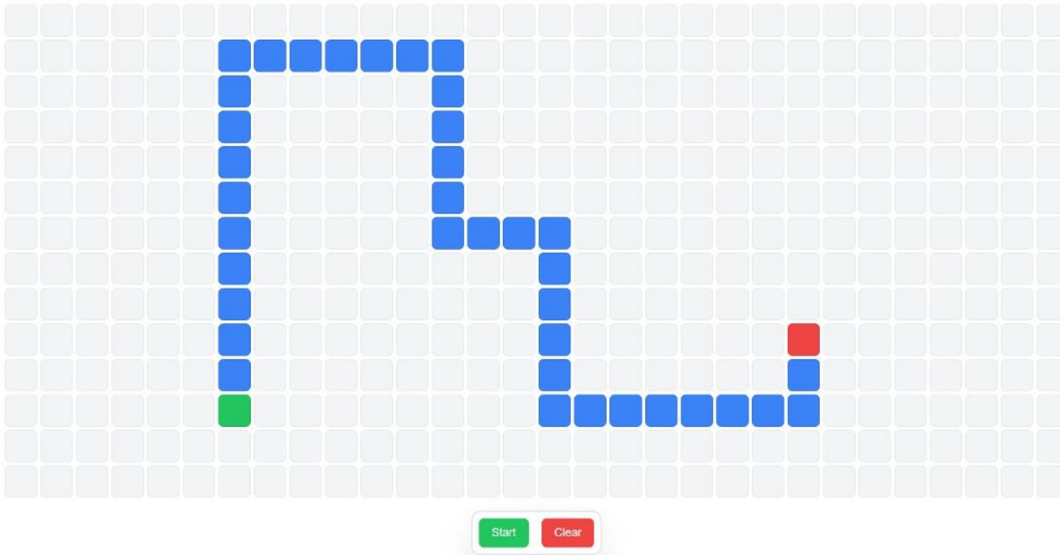

Figure 10 Smart Navigating Robot Web Interface

6. Cost & Expenditure

Component	**Quantity**	**Unit Cost (Rs.) .00**	**Total Cost (Rs)**
ESP32 Microcontroller	1	1270	1270
L298N Motor Driver	1	460	460
Buck Converter	1	200	200
IR Speed Sensors	2	150	300
11.1V LiPo Battery	1	2700	2700
LiPo Battery USB Charger	1	730	730
2-Wheel Round Double Deck Smaart car chassis with DC Geared Motors	1	1150	1150
Battery Voltage Tester	1	430	430
Miscellaneous	-		1150
Total	-	-	**8390**

7. Reference

1. https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html
2. https://github.com/espressif/arduino-esp32
3. https://randomnerdtutorials.com/esp32-dc-motor-l298n-motor-driver-control-speed-direction/
4. https://forum.arduino.cc/t/setup-communication-between-desktop-pc-and-esp32-over-wifi/1152476
5. https://docs.sunfounder.com/projects/umsk/en/latest/01_components_basic/07-component_speed.html
6. https://forum.arduino.cc/t/control-12dcv-moter-using-l298n-and-esp-32/952515
7. https://tronic.lk/product/ir-speed-sensor-module-analog-digital-mouse-encoder-u-t
8. https://tronic.lk/product/l298n-motor-driver-module-for-arduino-normal
9. https://tronic.lk/product/2-wheel-round-double-deck-smart-car-chassis-kit