EC299 - MINOR PROJECT

COURSE PROJECT REPORT ON

WIFI CONTROL CAR USING ESP8266 MICROCONTROLLER

COURSE INSTRUCTOR : MR. M. SRINIVAS

SUBMITTED BY :

DIVYA-623102

PALLAVI-623106

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

ANDHRA PRADESH

April 2024

TABLE OF CONTENTS

LIST OF FIGURES

1.ACKNOWLEDGEMENT

The WiFi Controlled Car project has been an invaluable learning experience for our team, providing insights into remote vehicle control and wireless communication.
We would like to to acknowledge the priceless cooperation of our professor MR.M.SRINIVAS who provided us a unique opportunity with a lot of time and effort for his support and encouragement in making hardware model as well as this paper work. Special thanks to the department and lab instructors of ECE for understanding and help during the fact finding mission.

2. ABSTRACT

The WiFi Controlled Car project represents an innovative exploration into the realm of remote vehicle control using WiFi technology. This aims to design and implement a robust system that allows users to wirelessly command and a car through a web interface or a dedicated mobile application .The project integrates essential hardware components , including motors, wheels, and an ESP8266 microcontroller, to enable seamless communication and control over a local WiFi network.

Throughout the development process, the project delves into various communication protocols, motor control algorithms, and user interface designs to ensure efficient and intuitive operation. By leveraging WiFi connectivity, the system offers real-time control over the vehicle, empowering users to navigate with precision and ease. Moreover, the project explores the implications of WiFi- controlled systems in diverse applications, including robotics, automation, and Internet of Things (IoT).

3. INTRODUCTION

The WiFi Controlled Car project aims to explore the possibilities of remote vehicle control using WiFi technology. In today's interconnected world, wireless communication has become increasingly prevalent, offering new opportunities for innovative applications. This project leverages WiFi connectivity to create a system that allows users to control a car remotely through a web interface or a mobile application.

The evolution of wireless communication plays a vital role in the industry 4.0 campaign. Wireless systems are employed in different dimensions like security, manpower, machine learning, database management systems and so on. Here, wireless controlled RCC car is a result of networking patriarchy. Wi-Fi technology is an abundant technology for communication, data transfer as well as data management. But, none of us utilize complete profit out of it rather than us. By using these concepts, an RC car is built using Wi-Fi i.e., ESP8266 NodeMCU in our research work. We will see about the model and working of the car.

4.APPARATUS

4.1 ESP8266 MICROCONTROLLER

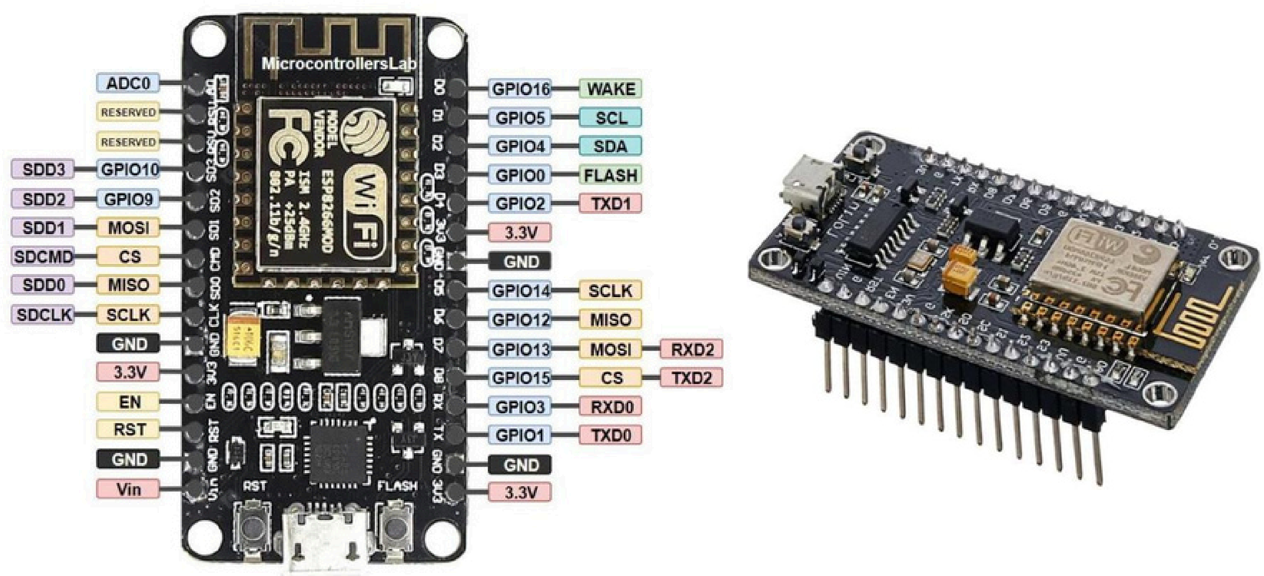


Figure 1 ESP8266

Microcontroller

The ESP8266 microcontroller, commonly used in WiFi-controlled car projects, offers: 32-bit Tensilica Xtensa LX106 processor running upto 80 MHz for handling WiFi and motor control tasks, 64 KB instruction RAM, 96 KB data RAM, and external flash memory up to 4 MB for program storage. Built-in WiFi connectivity for remote control over local networks or the internet, GPIO pins for interfacing with sensors, motor drivers, and LEDs, Analog inputs for sensor readings and voltage measurements. Communication interfaces including UART, SPI and I2C. Low power consumption, suitable for battery-powered applications. Programmable using Arduino IDE with support for C/C++ languages.

4.2 L298N MOTOR DRIVER

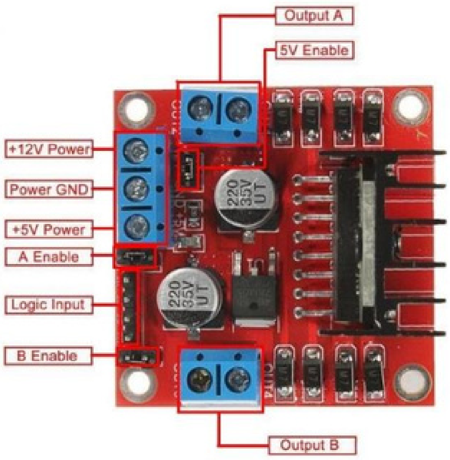


Figure 2 L298N Motor driver

The L298N is a popular dual H-bridge motor driver IC used for controlling DC motors. It's widely used in robotics and other applications where precise motor control is needed. The L298N can control motors with voltages ranging from 5V to 35V. L298N motor driver module is required

4.3 JUMPER WIRES

Figure 3 Jumper wires

Jumper wires are electrical connectors used to create temporary connections between components in electronics projects. They come in different types, colours, and lengths, and are commonly used with breadboards for prototyping circuits.

4.4 BO MOTOR AND BO WHEELS

Figure 4 BO Motors and Wheels

These are commonly used in robotics and small vehicle applications due to their high torque and compact size. "Bo" stands for "Brushed DC motors with Output shafts." They typically have two terminals for power supply and are controlled using PWM (Pulse Width Modulation) signals. Here, we used the Bo Motor And The RPM Of the Motor Is 150rpm.

4.5 7.4V BATTERY

These types of cars are often controlled through a smartphone app or a remote control unit that communicates with the car via Wi-Fi signals. The 12-volt battery would provide the necessary power to drive the car's motors and operate any other electronic components.

FIGURE 5 3.7V Batteries(2)

4.6 MOBILE APP

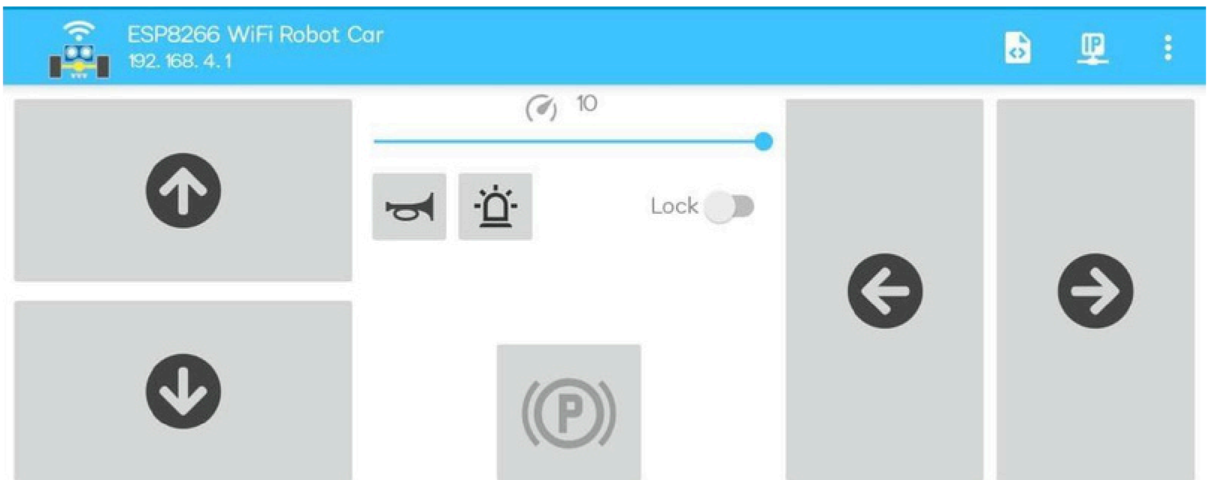


FIGURE 6 MOBILE APP

5.CIRCUIT CONNECTIONS

- ➢ Connect the GPIO pins of the ESP8266 module to the input pins of the motor driver to control the motors' direction and speed. (D5, D6, D8, D7, D4, D3 to ENA, ENB, pin1, pin2, pin3, pin4 respectively)
- ➢ Connect the output pins of the motor driver to the DC motors.
- ➢ Batteries connected in series.
- ➢ Power supply connections from batteries to L298N and from L298N to Wi-Fi module
- ➢ Connect the ground (GND) of all components together.
- ➢ Once you have the hardware set up, program the ESP8266 to connect to Wi-Fi network and listen for commands.

6.CIRCUIT DIAGRAM

Here is the circuit diagram for this Wi-Fi controller car using the ESP8266 Project. We will control the two dc motors via L298N Motor Driver IC. I used the L298N because the it is a high-power motor driver capable of running 5V to 35V DC at a maximum of 25W.

The Connection Of ESP8266 And L298N Motor Driver.

- ENA GPIO14(D5)
- ENA GPIO12(D6)
- IN_1 GPIO15(D8)
- IN_2 GPIO13(D7)
- IN_3 GPIO2(D4)
- IN_4 GPIO0(D3)

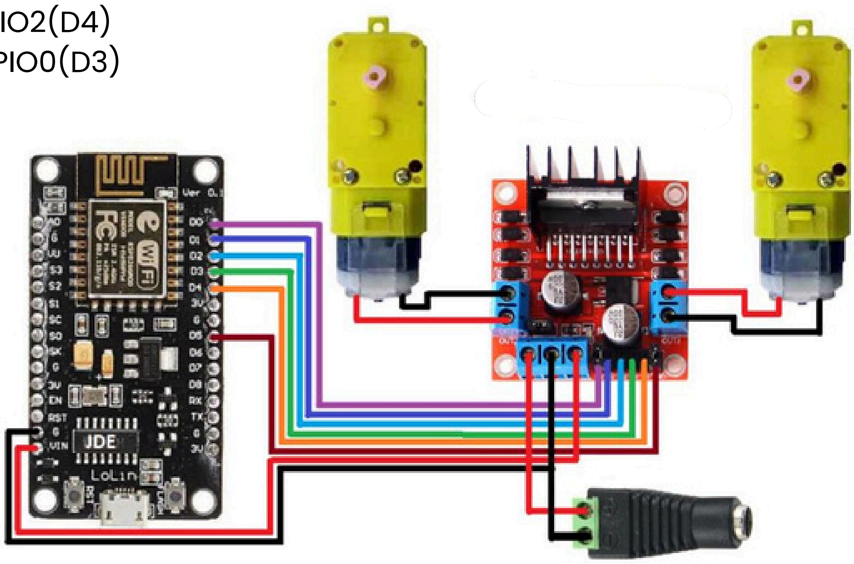


FIGURE 7 CIRCUIT DIAGRAM

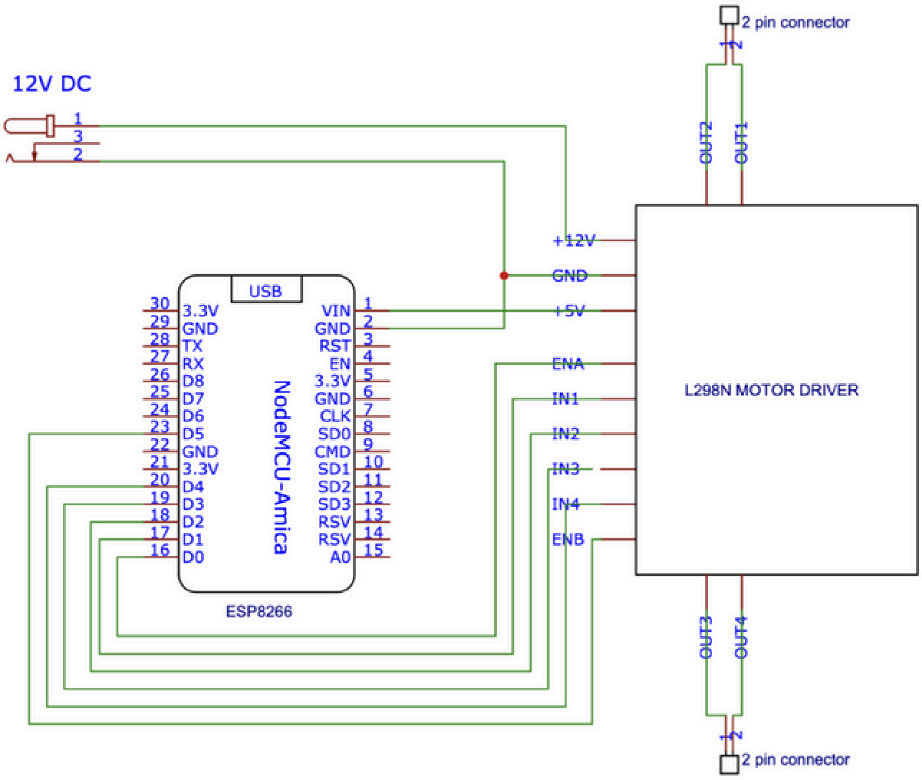


Figure 8 PCB DESIGN

7.CODE

FIGURE 9 ESP8266 CODE

```
// Motor Driver pins
#define ENA D5  // GPIO14
#define IN1 D8  // GPIO15
#define IN2 D7  // GPIO13
#define IN3 D4  // GPIO2
#define IN4 D3  // GPIO0
#define ENB D6  // GPIO12

char command = 0;


#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char*ssid="NodeMCU Car"
WiFi.softAP(ssid)

void setup() {
  // Set motor pins as output
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
```

FIGURE 9.1 CODE (PART 1)

```
  // Set speed (0-1023)
  analogWrite(ENA, 800);
  analogWrite(ENB, 800);

  // Start serial communication (if app uses Serial)
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {
    command = Serial.read();

    switch (command) {
      case 'F': // Forward
        forward();
        break;
      case 'B': // Backward
        backward();
        break;
      case 'L': // Left
        left();
        break;
      case 'R': // Right
        right();
        break;
      case 'S': // Stop
```

FIGURE 9.2 CODE(PART 2)

```
      case 'S': // Stop
        stop();
        break;
    }
  }
}

// Movement functions
void forward() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void backward() {
```

FIGURE 9.3 CODE(PART 3)

```
void backward() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void left() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void right() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void stop() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
```

FIGURE 9.4 CODE(PART 4)

Code Overview

- Motor driver pins are connected to GPIOs on ESP8266.(as explained in above sections)
- ESP8266 receives character commands from the mobile app via WiFi (like 'F', 'B', 'L', 'R', 'S').
- Each command activates specific GPIOs to control motor direction.
- analogWrite() is used for motor speed control.
- digitalWrite() gives High and low for GPIO pins.

8.WORKING

The working prototype of Wi-Fi controlled NodeMCU begins with connecting motors with wheels, a bit of soldering work which includes motors to be connected to the sockets of driver. Before connecting the motor with the provided L298N motor driver, we have a most significant principle to follow, i.e., we should run the code of WiFi module before getting through the circuit.

Then we should the check the mobility of the Wi-Fi controlled car's forward, backward, leftward or rightward conditions. By means mobile application, we can connect the IP address through which the car is being controlled. In case of any discrepancies, we should check whether the power supply is in a proper way.

We should not overrate the power supply more than 12v. And ESP8266 can't tolerate more than 5V. Power supply from batteries(>5V~7.4V) given to 12V pin of L298N module and voltage regulation happens there to 5V and that is given to esp through it's Vin pin. That's how this prototype works. The RCC car is controlled by the NodeMCU car application which contains an IP address based connection system. So by using our IP address of the Wi-Fi module, we can control our car. The screenshot of the app and the circuit diagram is given above.

WORKING PRINCIPLE:

- The ESP8266 NodeMCU is programmed to act as a WIFI server.
- When powered on, the ESP8266 creates its own hotspot and we connect our mobile phone to it via WiFi.
- An app is hosted on the ESP8266, which contains control buttons (Forward, Backward, Left, Right).
- A user connects to the ESP8266 using a smartphone via WiFi and opens the app developed for this project.
- When a control button is pressed:

  1.This app sends a command (like "forward", "left", etc.) to t

  2.The ESP8266 receives the command and processes it.

  3.Based on the command, the ESP8266 activates the corresponding GPIO pins connected to L298N.

  4. The L298N then controls the DC motors (their direction and speed) according to those signals.
- As a result, the car moves in the desired direction.

8.1 SPECIFICATIONS

8.1.1 MICRONTROLLER - ESP8266
8.1.2 MOTOR DRIVER - L298N 8.1.3
BATTERY – 7.4V

9.ADVANTAGES

- **Wireless Communication:** Wi-Fi control eliminates the need for physical tethering between the controller device and the car, providing greater freedom of movement and flexibility during operation.
- **Longer Range:** Compared to other wireless communication technologies like Bluetooth, Wi-Fi typically offers a longer range, allowing the controller device to maintain connectivity with the car over greater distances.
- **Multi-Device Compatibilty:** Wi-Fi control enables multiple controller devices to interact with the car simultaneously, making it suitable for group activities or multiplayer games.
- **Remote Access:** With Wi-Fi connectivity, users can control the car from anywhere within the range of the Wi-Fi network, offering oppurtunnities for remote operation and monitoring.

10.DISADVANTAGES

- Limited operational range due to WiFi signal range constraints.
- Dependence on WiFi infrastructure, making it unsuitable for remote areas.
- Potential for signal interference from other devices or networks.
- Increased power consumption, impacting battery life.
- Complexity of setup and configuration, requiring technical expertise.
- Security concerns related to WiFi network vulnerabilities.
- Car can change it’s direction based on the orientation of the first wheel and wind movement.

11.APPLICATIONS

- **Remote surveillance**: WiFi-controlled cars equipped with cameras can be used for remote surveillance in areas where human access is difficult or unsafe, such as disaster zones, construction sites, or hazardous environments

- **Security Patrols:** WiFi-controlled cars can pantrol and monitor premises for security purposes, providing real-time video surveillance and alerts for suspicious activities or intrusions.

- **Exploration and Mapping**: These cars can explore and map indoor or outdoor environments, collecting data and creating digital maps for navigation, inventory management, or archaeological surveys.

- **Education and Research**: WiFi-controlled cars serve as educational tools for teaching robotics, programming, and wireless communication concepts in schools, colleges, and research institutions.

- **Entertainment and Gaming**: WiFi-controlled cars offer interactive gaming experiences, allowing users to compete in races, obstacle courses, or challenges remotely via mobile apps or web interfaces.

- **Remote Inspection**: In industrial settings, WiFi-controlled cars can conduct remote inspections of machinery, pipelines, or infrastructure, enabling maintenance and troubleshooting without physical access.

- **Environmental Monitoring**: Equipped with sensors, WiFi-controlled cars can monitor environmental parameters such as air quality, temperature, and humidity in urban or natural settings.

- **Search and Rescue Operations**: These cars can assist in search and rescue operations by exploring rugged terrain, locating missing persons, or delivering supplies to remote locations.

12.CHALLENGES FACED

- **Switch Issue:** Although the power supply from the batteries was stable and all components were connected properly, the prototype initially didn't work due to a faulty switch. This was a tricky issue that could easily be overlooked.
- To fix this, we bypassed the switch by giving direct wire connections to the battery terminals.
- To control the power manually, we connected the female pin jumper wire from the battery and the male pin jumper wire from the 12V pin of the L298N motor driver. We joined them to power ON, and separated them to power OFF.

- **Battery Drain:** The batteries drained faster than expected during testing, affecting the prototype's performance. We replaced them with fresh batteries. (Note: the used batteries can also be recharged.)

- **Wire Burn Due to High Voltage:** At one point, the wires overheated and got damaged due to high voltage. We identified the affected wires and replaced them with new ones.

- **APK Installation Issue:** The intended mobile application for controlling the car via WiFi couldn't be downloaded. As a workaround, we found and used an alternate app from the Play Store to successfully send commands to the car.

- **Incorrect Motor Connections:** The output connections from the motor driver to the motors were initially reversed. As a result, the car moved backward when 'forward' was pressed, and left when 'right' was pressed (and vice versa). We corrected this by swapping the motor wires to their proper polarity.

13.PROJECT DEMONSTRATION

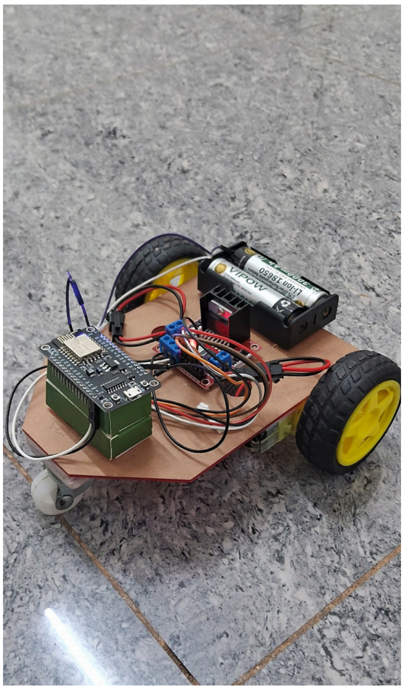


FIGURE 10.1 FRONT VIEW OF CAR PROTOTYPE

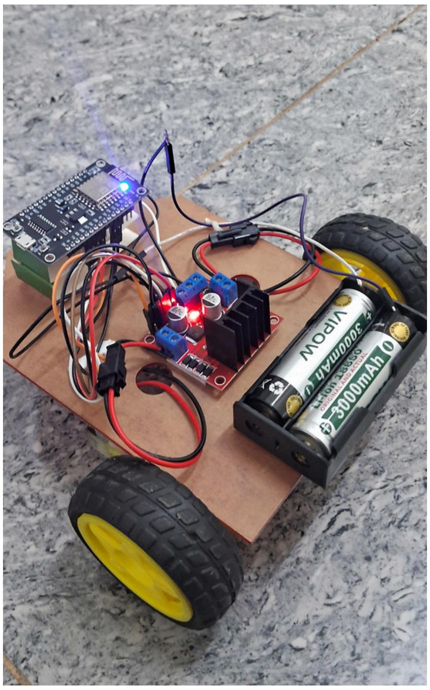


FIGURE 10.2 BACK VIEW OF CAR PROTOTYPE

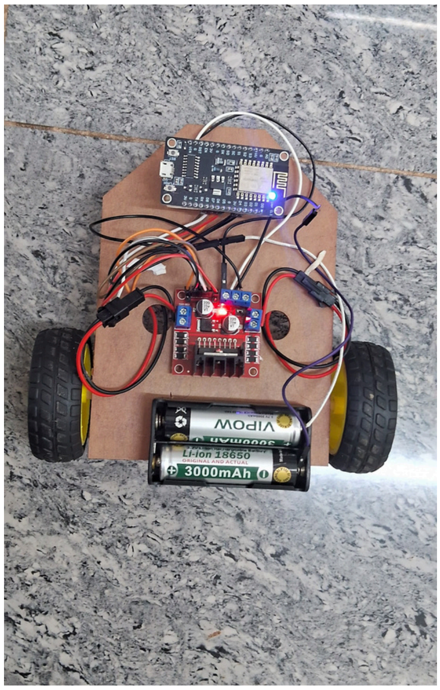
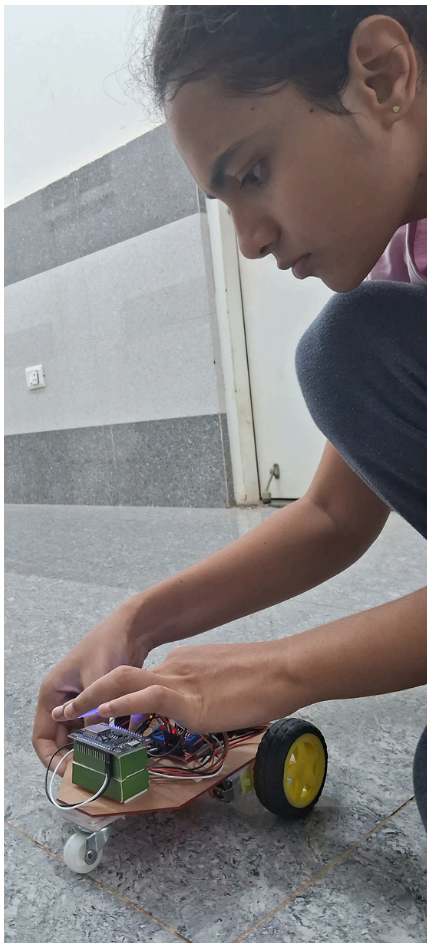


FIGURE 10.3 TOP VIEW OF CAR PROTOTYPE

The working model successfully performs the intended functionality. Scan the QR code below to view the live demonstration.

FIGURE 10.4 QR CODE(TO VIEW LIVE WORKING OF CAR PROTOTYPE)



14.CONCLUSION

This ideology can be used in automation industries, real-time control systems engineering, industrial purposes. The whole prototype is applicable in both PC as well as android mobile with a Wi-Fi technology. A physical living matter can end up with imperfections but, a built-in prototype can't make imperfections (or) errors. This ideology may possess finite drawbacks as similar with IoT.

Throughout the project, we have explored the capabilities of the ESP8266 microcontroller, including its processing power, memory, and communication interfaces, to establish reliable and responsive WiFi communication between the car and remote devices. By interfacing with motor drivers, sensors, and other hardware components, we have enabled precise control and feedback mechanisms, enhancing the functionality and usability of the WiFi-controlled car.

15.REFERENCES

[1] https://justdoelectronics.com/how-to-make-a-wifi-car-using-nodemcu/

[2] https://www.hackster.io/diyprojectslab/make-wi-fi-controller-car-using-esp8266-297e49