International Journal of
Advanced Control and
System Engineering

Volume 2, Issue 1, 2024
January–June
DOI (Journal): 10.37591/IJACSE

https://journals.stmjournals.com/ijacse

Review IJACSE

Smart Lane Following and Mobile Controlled Robotic Car

Umesh Pinjarkar[1],*, Hitesh Raktade[2], Saurabh Popalghat[2], Vishakha Polas[2], Aayush Kadam[2]

Abstract

One major development in autonomous vehicle technology is the emergence of mobile-controlled robotic cars and smart lane-following vehicles. To improve user control and vehicle autonomy, these systems integrate mobile technology, computer vision, and machine learning. This review work highlights the potential of mobile-controlled and smart lane-following robotic automobiles to transform transportation and related sectors by examining their design, functioning, applications, and obstacles. Autonomous driving relies heavily on smart lane-following technology, which allows cars to maintain their lane boundaries without the need for driver interaction. Furthermore, the incorporation of mobile control facilitates remote interaction and direction of these vehicles by users. This assessment covers the future directions, existing applications, and technological underpinnings of robotic cars that can follow a smart lane and be directed by a mobile device.

Keywords: smart lane, mobile technology, computer vision, machine learning, robotic automobiles

INTRODUCTION

The potential of autonomous vehicles to increase transportation efficiency, safety, and convenience has attracted a lot of interest in recent years. In the contemporary landscape of technological advancement, robotics stands out as a field offering innovative solutions across various domains. Among these solutions, the development of autonomous systems, such as line following robots, holds significant promise for addressing complex challenges and enhancing productivity.

A line following robot, designed to navigate predefined paths autonomously, represents a compelling solution for applications in industrial automation, logistics optimization, and educational exploration. Its ability to track lines or markings on the ground using sensors enables precise navigation in dynamic environments where manual control or fixed infrastructure may be impractical. Motivated by the limitations of traditional navigation methods in industries and the educational value of robotics projects, the conception of line following robots aims to streamline material flow in industries and provide hands-on learning opportunities for students. These robots serve as tangible embodiments of interdisciplinary principles, bridging concepts from engineering, computer science, and robotics to create tangible solutions with real-world impact. This report delves into the intricacies of designing a line following robot, exploring its components, algorithms, and methodologies. Robotic automobiles with mobile controls and smart lane-following are major advancements in autonomous vehicle technology.

***Author for Correspondence**
Umesh Pinjarkar
E-mail: umesh.pinjarkar@aiml.sce.edu.in

[1]Assistant Professor, Department of Electrical Engineering, Saraswati College of Engineering, Pune, Maharashtra, India
[2]Student, Department of Electrical Engineering, Saraswati College of Engineering, Pune, Maharashtra, India

Received Date: April 29, 2024
Accepted Date: June 29, 2024
Published Date: July 30, 2024

Citation: Umesh Pinjarkar, Hitesh Raktade, Saurabh Popalghat, Vishakha Polas, Aayush Kadam. Smart Lane Following and Mobile Controlled Robotic Car. International Journal of Advanced Control and System Engineering. 2024; 2(1): 36–47p.

These systems improve transportation's efficiency, convenience, and safety by utilising mobile technologies, computer vision, and machine learning. Even though there are still obstacles to overcome, continued progress should increase their potential and increase the range of uses. Smart robotic automobiles are expected to play a major part in transportation and other areas in the future as research and development continue.

LITERATURE SURVEY

Shivaraju *et al*. presented a Wi-Fi-controlled robot car for home automation, allowing users to control it via an Android app or SMS commands. Tasks such as home security and object transportation are facilitated. Future upgrades aim to integrate a spy camera and utilize solar power for sustainability [1]. Talele *et al*. developed a smartphone-controlled robot car with an ESP8266 at its core, allowing for various tasks with minimal technology. It incorporates a front-mounted camera for live viewing on smartphones and the capability for Wi-Fi-based control and video streaming. Challenges in surveillance system development include ensuring dynamic functionality and mobility, especially for large areas. The Internet of Things (IoT) plays a crucial role, enabling seamless communication between the car and mobile devices. This technology finds applications in tasks like animal rescue and zoo surveillance, offering accessibility to otherwise restricted areas. The project utilizes an Arduino Uno motor driver and an ESP32 camera module for efficient communication and control [2]. Dadoria proposed using a microcontroller-based line-following robot equipped with IR sensors to deliver medicines within hospitals efficiently. A proximity sensor detects obstacles and alarms accordingly. This system streamlines material supply, reducing manual work and ensuring timely medication delivery, enhancing patient care [3]. The system architecture of an autonomous car and its real-time adaptive vision system for following roads is described by Kuan *et al*. The vehicle is a robotically controlled 10-ton armoured personnel carrier [4].

COMPONENTS REQUIRED

The proposed system utilizes the following components [1]:

1. *NodeMCU:* Serves as the microcontroller unit, providing the computational power and connectivity necessary for controlling the robot's operations.
2. *IR sensors:* These sensors are employed for line detection, enabling the robot to follow predefined paths accurately.
3. *L298 motor driver:* Facilitates the control of DC motors, allowing the robot to move along the tracked path with precision and agility.
4. *Geared DC motors:* Responsible for propelling the robot forward or backward, as directed by the microcontroller and motor driver.

FOUNDATIONS OF TECHNOLOGY

- *Computer vision:* The foundation of intelligent lane-following systems is computer vision. In order to identify lane markers on the road, cameras and image processing algorithms are used. Important methods consist of:
    - *Edge detection:* The edges of lane markings are identified by algorithms such as Canny edge detection.
    - *Hough transform:* This method finds the image's straight lines, which are utilised to indicate lane borders.
    - *Region of interest (ROI):* To speed up processing and increase accuracy, focus on particular regions of the image, such as the lower half where lanes are anticipated.
- Robotic learning: The ability of robotic automobiles to comprehend challenging driving settings is improved by machine learning. Methods like:
    - *Convolutional neural networks (CNNs):* CNNs aid in the identification of lanes, cars, and people. They are used for object detection and image categorization [5].
    - By teaching the vehicle to make judgements based on input from its surroundings, reinforcement learning helps the vehicle become more adept at following lanes over time.

- Robotic automobiles can communicate with mobile control systems through cell-phones or tablets. Important technologies consist of:
  - *Bluetooth and Wi-Fi:* Enable wireless connectivity between the automobile and the mobile device.
  - *Mobile applications:* Offer user interfaces for navigating, watching live video feeds, and managing vehicle motions. Real-time tracking and navigation are made possible with GPS integration.

Node MCU ESP8266

It is the device's central component. It offers the IoT platform. It is a Wi-Fi module with firmware for the esp8266 inside of it. This microcontroller is connected to every other sensor. It receives the measured values from them and uploads them all to the cloud for analysis. This board is developed by ESP8266 Opensource Community. XTOS is the name of its operating system. The ESP8266(LX106) is the CPU [6]. This microcontroller is connected to every other sensor as shown in Figure 1 [3]. It features a 4 MB storage capacity and 128 kb of built-in RAM.

IR Sensors (Infrared Sensor)

An IR sensor detects infrared radiation emitted or reflected by objects in their vicinity. It converts these signals into electrical impulses for further processing. Commonly used for proximity sensing, motion detection, and temperature measurement, IR sensors find applications in various fields, including consumer electronics, security systems, and industrial automation, due to their reliability and versatility as shown in Figure 2.

L298 Motor Driver

The L298 motor driver is a popular integrated circuit (IC) used to control the speed and direction of DC motors as shown in Figure 3. It provides dual H-bridge configurations, allowing bidirectional control of two motors or a single stepper motor. With built-in protection features such as thermal shutdown and overcurrent protection, the L298 offers robust motor control for a wide range of applications. It is commonly utilized in robotics, automation, and motor control projects due to its simplicity, reliability, and versatility in driving DC motors efficiently.

Geared DC Motors

Geared DC motors are electric motors equipped with gears that reduce the motor's speed while increasing its torque output as shown in Figure 4. These motors combine the benefits of DC motors, such as simple control and high efficiency, with the advantages of gearing, which include increased torque and improved precision. Geared DC motors find widespread use in applications requiring precise speed control, high torque, and low-speed operation, such as robotics, automation, and consumer electronics.

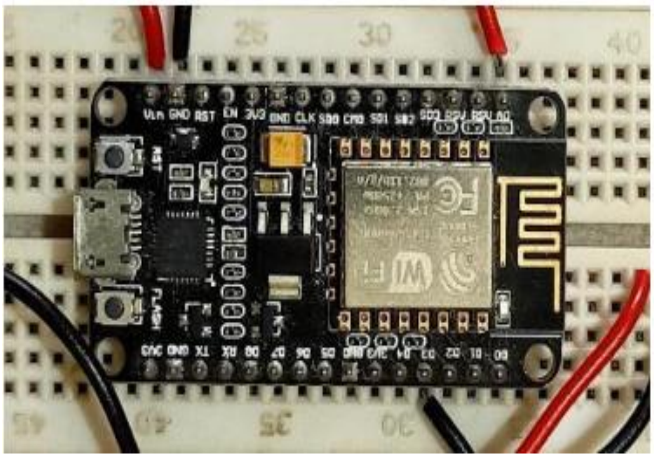


Figure 1. Node MCU ESP8266.

Figure 2. IR sensor.

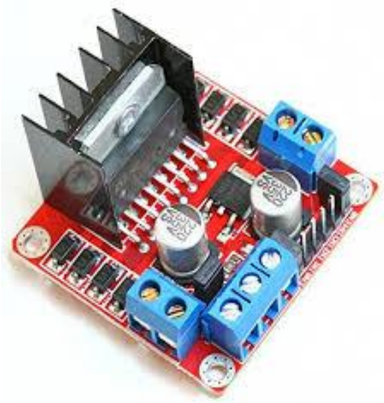


Figure 3. Motor driver L298.

Figure 4. Geared DC motors.

PROBLEM STATEMENT

To address the challenges in modern industrial and healthcare industry, there is a need for a reliable and autonomous system capable of navigating predefined paths with precision and efficiency. In modern industrial and healthcare departments, the efficient management and distribution of resources pose significant challenges. Manual handling of tasks such as material transport and medication delivery often leads to inefficiencies, delays, and errors. Furthermore, the increasing demand for personalized and timely patient care necessitates innovative solutions to streamline processes and enhance service quality. In this context, the development of our line following robot presents a compelling opportunity to address these challenges [7]. The primary problem statement revolves around the need for a reliable and autonomous system capable of navigating predefined paths with precision and efficiency. Specifically, there is a demand for a solution that can automate the delivery of materials, such as medication, within healthcare facilities, ensuring timely access and minimizing reliance on manual intervention.

PROPOSED SYSTEM

Algorithm

- *Initialization:*
  - Set up pins and variables.
  - Connect to Blynk server.
- *Setup function:*
  - Initialize components and Blynk.
- *Blynk virtual pin handlers:*
  - Update mode and speed variables based on Blynk inputs.
- *Main loop:*
  - Run Blynk tasks.
- *Control car based on mode:*
  - In mobile control mode (0), respond to Blynk inputs for direction and speed.
  - In line follower mode (1), adjust car movement based on IR sensor inputs [8–10].
- *Motor control functions:*
  - Control car movement: left, right, forward, stop.
  - Blynk inputs.

The flowchart of the proposed system is shown in Figure 5.

The block diagram of the proposed system is shown in Figure 6. The circuit diagram of the system and its connections with the microcontroller to the sensor are shown in Figure 7 [7].

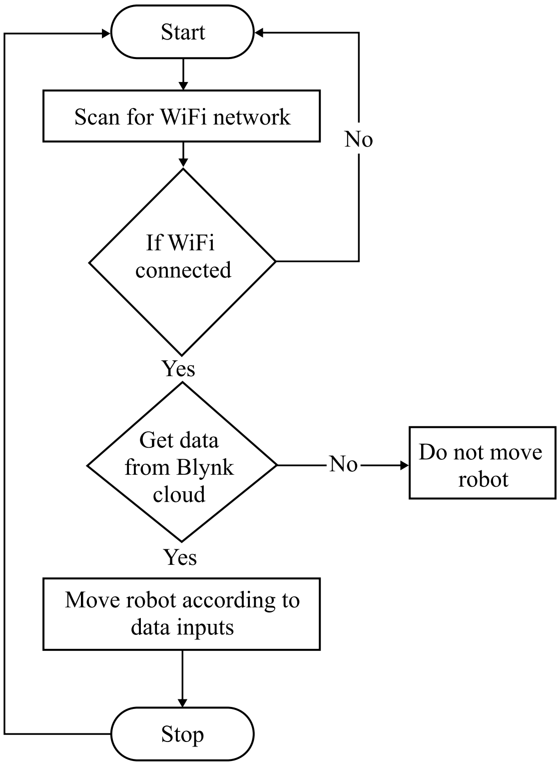


Figure 5. Flow chart of the working algorithm.

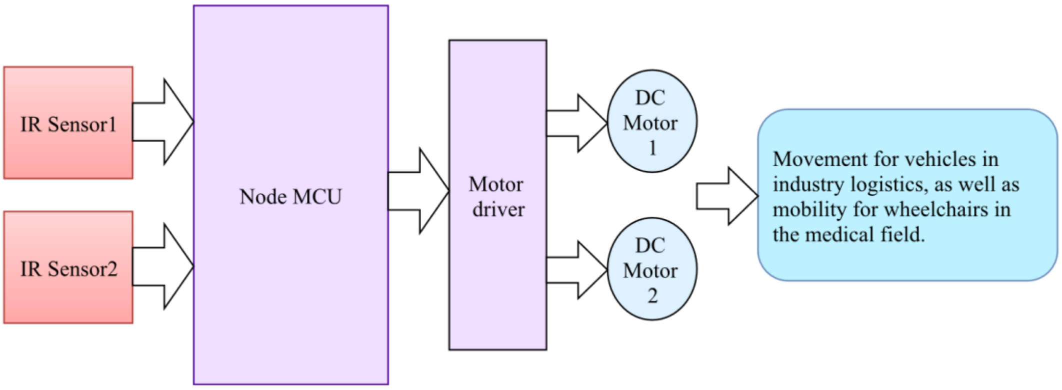


Figure 6. Block diagram of smart robotic car.

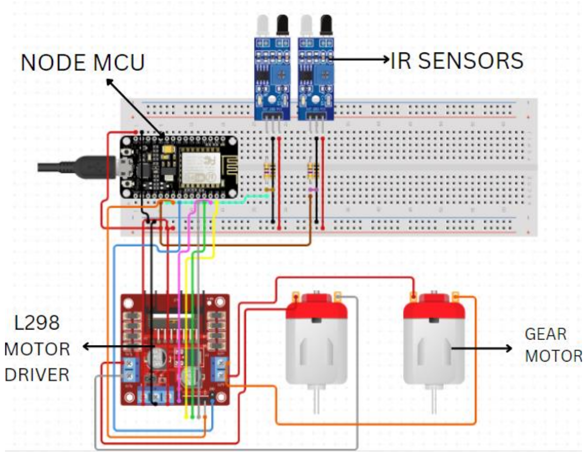


Figure 7. Circuit diagram of smart robotic car.

PROPOSED WORK

Here is a proposed work based on the project:

1. *Hardware setup:*
   i. List the required hardware components: ESP8266 microcontroller, two IR sensors, L298 Motor driver, Toggle switch, Gear DC Motor, Jumper Wires.
   ii. Provide a schematic diagram illustrating how the components are connected.
2. *Software implementation:*
   i. *Initialization:*
      - Set up pins and variables.
      - Connect to Blynk server.
   ii. *Setup function:*
      - Initialize components and Blynk.
   iii. *Blynk virtual pin handlers:*
      - Update mode and speed variables based on Blynk inputs.
   iv. *Main loop:*
      - Run Blynk tasks.
      - Control car based on mode:
        - ○ In mobile control mode (0), respond to Blynk inputs for direction and speed [11, 12].
        - ○ In line follower mode (1), adjust car movement based on IR sensor inputs.
   v. *Motor control functions:*
      - Control car movement: left, right, forward, stop.
      - Blynk inputs.

CODE

```
9#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#define ENA D0
#define IN1 D1
#define IN2 D2
#define IN3 D3
#define IN4 D4
#define ENB D5
#define S1 D6 // Assuming S1 pin of NodeMCU is connected to IR sensor 1
#define S2 D7 // Assuming S2 pin of NodeMCU is connected to IR sensor 2

int Speed=255; // Default speed
int mode=0; // 0 for mobile control mode, 1 for line follower mode

char auth[]="uxpEmykaQSPyWEjspp8wD-B9Z6JP7yhE"; //Enter your Blynk auth token
char ssid[]="Redmi 8A Dual"; //Enter your WIFI name
char pass[]="123456789"; //Enter your WIFI password
#define BLYNK_TEMPLATE_ID "TMPL3U5F20PZY"
#define BLYNK_TEMPLATE_NAME "ultimatec3"

void setup() {
Serial.begin(9600);
Serial.println("Setup started...");

pinMode(ENA, OUTPUT);
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
```

```
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
pinMode(ENB, OUTPUT);
pinMode(S1, INPUT);
pinMode(S2, INPUT);

Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

  // Attach a function to be called whenever the virtual pin changes
Blynk.syncVirtual(V4); // Sync initial state

Serial.println("Setup completed.");
}

BLYNK_WRITE(V4) {
  mode=param.asInt(); // Update mode based on value received from Blynk
  if (mode==0) {// Mobile control mode
Serial.println("Switched to Mobile control mode.");
  }
}
BLYNK_WRITE(V3) {
  if (mode==0) {// Mobile control mode
    Speed=param.asInt(); // Update the speed based on the value received from the slider
Serial.print("Speed updated to: ");
Serial.println(Speed);
  }
}

BLYNK_WRITE(V0) {
  if (mode==0) {// Mobile control mode
    int forwardState=param.asInt();
    if (forwardState==HIGH) {
carForward();
Serial.println("Car moving forward.");
    } else {
carStop();
Serial.println("Car stopped.");
    }
  }
}

BLYNK_WRITE(V1) {
  if (mode==0) {// Mobile control mode
    int leftState=param.asInt();
    if (leftState==HIGH) {
carLeft();
Serial.println("Car turning left.");
    } else {
carStop();
Serial.println("Car stopped.");
    }
  }
```

```
}

BLYNK_WRITE(V2) {
  if (mode==0) {// Mobile control mode
    int rightState=param.asInt();
    if (rightState==HIGH) {
carRight();
Serial.println("Car turning right.");
    } else {
carStop();
Serial.println("Car stopped.");
    }
  }
}

void loop() {
Blynk.run();

  if (mode==1) {// Line follower mode
    if (digitalRead(S1)==HIGH &&digitalRead(S2)==LOW) {
carLeft();
Serial.println("Line follower mode: Car turning left.");
    } else if (digitalRead(S1)==LOW &&digitalRead(S2)==HIGH) {
carRight();
Serial.println("Line follower mode: Car turning right.");
    } else {
carForward();
Serial.println("Line follower mode: Car moving forward.");
    }
  }
}

void carLeft() {
analogWrite(ENA, Speed);
analogWrite(ENB, Speed);
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
}

void carRight() {
analogWrite(ENA, Speed);
analogWrite(ENB, Speed);
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
}

void carBackward() {
analogWrite(ENA, Speed);
```

```
analogWrite(ENB, Speed);
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
}

void carForward() {
analogWrite(ENA, Speed);
analogWrite(ENB, Speed);
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
}

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

PROJECT OUTPUT

Figure 8 presents the proposed robotic car. The line follower mode is shown in Figure 9. Figure 10 represents the application interface by which we control the car in mobile control mode. Figure 11(a) shows representation how to assign virtual pins. Figure 11(b) shows Website Dashboard Interface Figure 12 shows Device Information Interface. Figure 13 shows utilization of e-yantra resource development centre. Figure 14 shows Participation in Avishkar Competition.

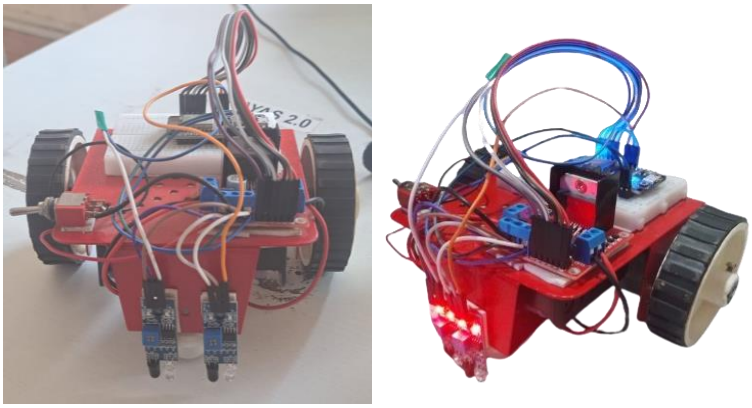


Figure 8. Describe hardware of robotic car.

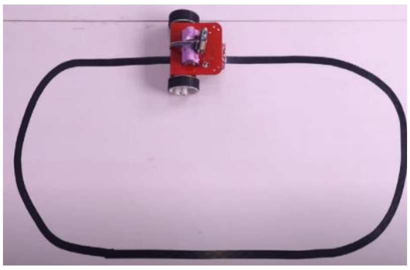


Figure 9. Describe how the robotic car follow line.

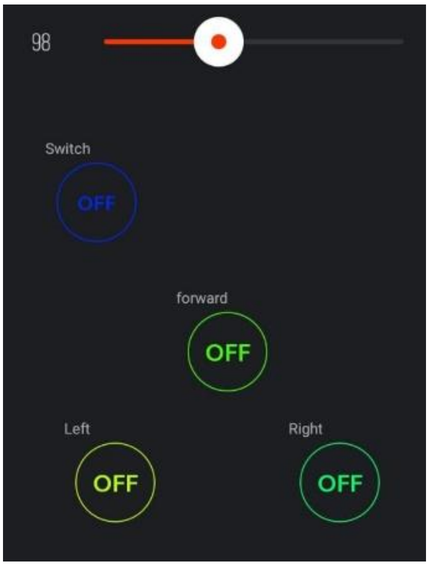
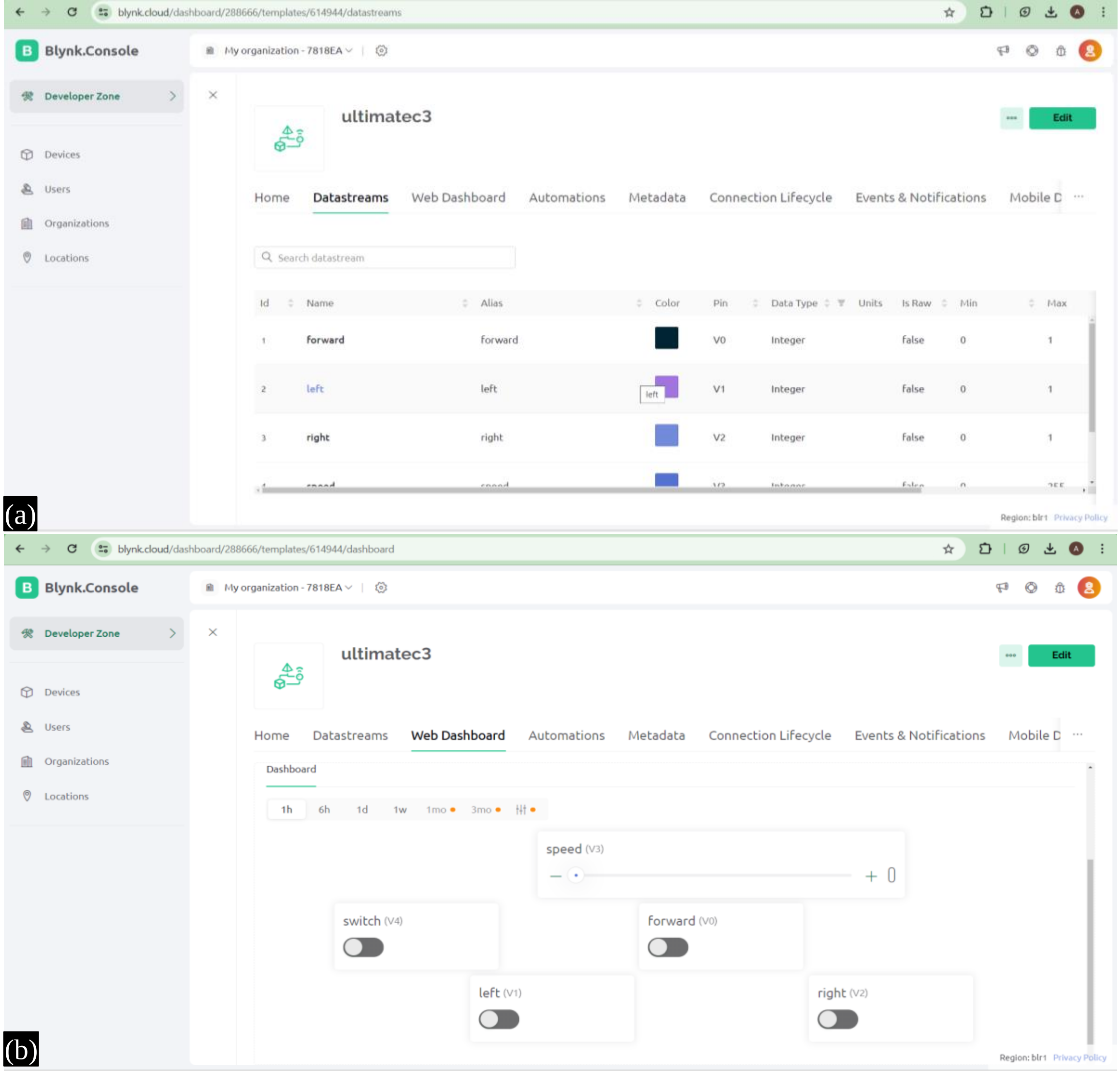
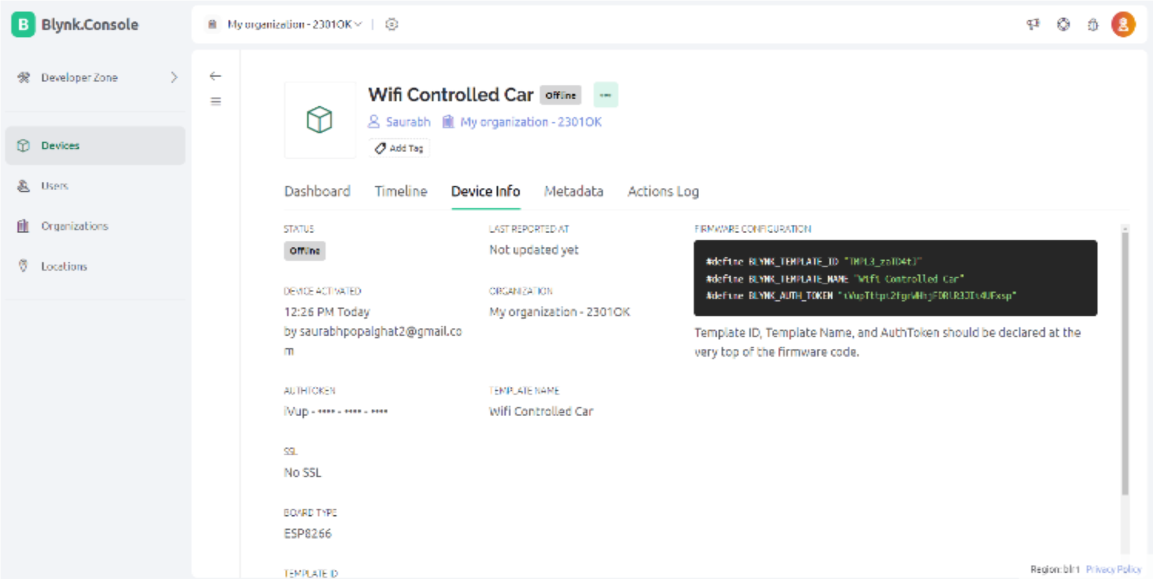


Figure 10. Application interface.



Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min	Max
1	forward	forward	■	V0	Integer		false	0	1
2	left	left	■	V1	Integer		false	0	1
3	right	right	■	V2	Integer		false	0	1

Figure 11. (a) Website DataStream Interface, (b) Web dashboard.



Wifi Controlled Car Offline

Saurabh My organization - 2301OK

STATUS
Offline

LAST REPORTED AT
Not updated yet

FIRMWARE CONFIGURATION

```
#define BLYNK_TEMPLATE_ID "TMPL3_zaID4tJ"
#define BLYNK_TEMPLATE_NAME "Wifi Controlled Car"
#define BLYNK_AUTH_TOKEN "iVupTttpt2fgrWHrjFDRLR3JIt4UFxsp"
```

Template ID, Template Name, and AuthToken should be declared at the very top of the firmware code.

DEVICE ACTIVATED
12:26 PM Today
by saurabhpopalghat2@gmail.com

ORGANIZATION
My organization - 2301OK

AUTHTOKEN
iVup - **** - **** - ****

TEMPLATE NAME
Wifi Controlled Car

SSL
No SSL

BOARD TYPE
ESP8266

TEMPLATE ID

Figure 12. Description of device info.

Figure 13. E-yantra resource development center website interface.

CONCLUSION

In conclusion, our proposed design for a line following robot offers a promising solution to the challenge of autonomous navigation along predefined paths. By leveraging advanced control algorithms and sensor technologies, the robot demonstrates efficient line tracking capabilities, with potential applications in industrial automation, logistics, health care and education. Further development and experimentation are necessary to validate the effectiveness and reliability of the proposed system in real-world scenarios.

Figure 14. Participation in Avishkar competition.

REFERENCES

1. Shivaraju V, Karthik Kumar V. Robot Controlled Car using Wi-Fi Module. International Journal of Scientific Engineering and Technology Research (IJSETR). 2017 Feb; 06(04): 0759–0762.
2. Talele A, Mahajan R, Mahajan T, Kannake H, Khan Z, Late K. Wi-Fi Controlled Car. Int Res J Eng Technol. 2022; 9(12): 1105–1108. [Online]. Available: www.irjet.net
3. Rohit Dadoriya. (2016 Dec 18) Final report of line follower robot. [Online]. Available from https://www.slideshare.net/slideshow/final-report-of-line-follower-robot/70242590
4. Kuan D, Phipps G, Hsueh AC. Autonomous robotic vehicle road following. IEEE Trans Pattern Anal Mach Intell. 1988 Sep; 10(5): 648–58.
5. Fernandes LC, Souza JR, Pessin G, Shinzato PY, Sales D, Mendes C, Prado M, Klaser R, Magalhães AC, Hata A, Pigatto D. CaRINA intelligent robotic car: architectural design and applications. J Syst Architect. 2014 Apr 1; 60(4): 372–92.
6. Ali MA, Mailah M, Jabbar WA, Moiduddin K, Ameen W, Alkhalefah H. Autonomous Road roundabout detection and navigation system for smart vehicles and cities using laser simulator–fuzzy logic algorithms and sensor fusion. Sensors. 2020 Jul 1; 20(13): 3694.
7. Palaniswamy T, Alofi A, Alghamdi FS, Aljuaid NM, Alahmadi R. Automated Mobile Robots–A Survey on Controls. Commun Appl Electron. 2017; 6(9): 22–6.
8. Tzafestas SG. Mobile robot control and navigation: A global overview. J Intell Robot Syst. 2018 Jul; 91(8): 35–58.
9. Martínez-Barberá H, Herrero-Pérez D. Multilayer distributed intelligent control of an autonomous car. Transp Res Part C: Emerg Technol. 2014 Feb 1; 39: 94–112.
10. Hong TH, Rasmussen C, Chang T, Shneier M. Road detection and tracking for autonomous mobile robots. In SPIE Unmanned Ground Vehicle Technology IV. 2002 Jul 17; 4715: 311–319.
11. Waykole S, Shiwakoti N, Stasinopoulos P. Review on lane detection and tracking algorithms of advanced driver assistance system. Sustainability. 2021 Oct 15; 13(20): 11417.
12. Eleftheriou G, Doitsidis L, Zinonos Z, Chatzichristofis SA. A fuzzy rule-based control system for fast line-following robots. In 2020 IEEE 16th International Conference on Distributed Computing in Sensor Systems (DCOSS). 2020 May 25; 388–395.