Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 03 Issue: 12 | Dec-2025
ISSN (O) 3107-6696

ELEVATING WIFI-CONTROLLED CAR USING ESP32

Asst. Prof. Kotresh Korlahalli[1], Praveensingh G Rajput[2], Revansiddayya P Hiremath[3], Rakesh Bellad[4]

[1,2,3,4] Dept. of Electronics & Communication Tontadarya College of Engineering Gadag, Karnataka, India.
[1]korlahalli@gmail.com, [2]praveensinghrajput18@gmail.com, [3]rphiremath06@gmail.com, [4]belladrakesh04@gmail.com

---

Abstract - The key innovation is a web-based interface hosted directly on the ESP32, eliminating the need for dedicated applications or internet connectivity. This design choice enhances user-friendliness and reduces system complexity. The hardware setup involves fundamental components: an ESP32 development board, an L298N motor driver, and standard DC motors, ensuring cost-effectiveness and ease of replication. The ESP32's dual-core ar-chitecture and integrated Wi-Fi module allow it to simultaneously serve web pages and manage real-time motor control. Users can access the intuitive web interface through any Wi-Fi enabled device's browser to send commands to the car. The commands are wirelessly transmit in the ESP32, which then processes them to drive the L298N motor driver. The L298N, in turn, regulates the power supplied to the DC motors, enabling precise control over the robot's movement. Primarily intended for educational pur-poses and rapid prototyping, this project effectively demonstrates the principles of wireless robotic control with minimal hardware and software overhead. The app-free and internet-independent nature of the system makes it an ideal platform for students, hobbyists, and researchers exploring robotics and the Internet of Things. The paper elaborates on the hardware configuration, the software implementation on the ESP32, and the functionality of the user-friendly web control interface. The advantages of this approach, including its simplicity, low cost, and self-contained operation, are also discussed. Index terms such as ESP32, Wi-Fi robot, IoT car, web server control, and remote control categorize the core aspects of this work. This research contributes to making wireless robotic control more accessible and easier to implement.

Keywords - ESP32, WiFi robot, Iot car, Robotic car, Remote control, Web server control

Introduction

The advancement of microcontrollers has enabled the cre-ation of simple yet effective robotic systems for learning and experimentation. One such innovation is the Wi-Fi-controlled robotic car, which provides a platform for exploring wireless communication and basic automation. Utilizing the ESP32 microcontroller, this project focuses on remote vehicle control through Wi-Fi connectivity without relying on any automated input mechanisms. The goal is to build a cost-effective system that emphasizes manual control and responsiveness.

It creates a wireless access point and hosts a web interface that allows users to control the car from any device with a browser. The vehicle is powered by DC motors and uses a motor driver to manage movement in all directions. A key feature of this model is its ability to elevate or lift slightly, allowing it to clear minor obstacles through user command alone.

The car is fully operated through a browser-based interface, offering buttons or commands for forward, reverse, turns, and elevation control. Since no automatic detection or environmental feedback is used, the entire system is designed to respond directly to user input. This makes the project ideal for beginners and those interested in understanding the fundamentals of wireless robotic control without added complexity.

In conclusion, the Wi-Fi-controlled car using ESP32 demonstrates how functional and interactive a simple robotic system can be. Without the use of automated systems or external input devices, it provides a reliable platform for learning core concepts in embedded programming, motor control, and web-based communication. Its minimal hardware requirements and focus on manual operation make it perfect for educational use and early-stage prototyping.

II. LITERATURE REVIEW

Various approaches have been explored for WiFi-controlled robotic systems. Traditional methods often rely on RF modules or Bluetooth, which suffer from limited range and connectivity issues. Recent advancements have shifted focus toward using WiFi-enabled microcontrollers for more reliable and long-distance control.

In [1], a WiFi-based robotic car was developed using the NodeMCU microcontroller. While it offered decent control over a local network, the system lacked advanced features such as feedback mechanisms and had limited GPIO support for expansion.

Another study [2] implemented a robotic car controlled via a custom mobile app, using an ESP8266 module for WiFi communication. Although functional, the use of the ESP8266 limited the processing power and expandability, restricting the system's performance for advanced operations like sensor integration or multi-functional movements.

A more advanced design [3] employed the ESP32 microcontroller for controlling a robotic vehicle over WiFi, taking advantage of its dual-core processor and integrated peripherals. Despite offering improved performance, the design did not address the mechanical flexibility required for diverse terrain navigation.

Our proposed system addresses these limitations by incorporating an elevating chassis mechanism for terrain adaptability, WiFi control using the ESP32, and real-time feedback via an I2C LCD display. This low-cost yet flexible design enhances both mobility and user interaction, making it suitable for educational and prototyping purposes.

III. PROPOSED SYSTEM

The proposed system comprises a robotic car equipped with an ESP32 microcontroller, enabling wireless control and an elevating mechanism for enhanced terrain navigation. The key components include:

ESP32 Microcontroller: Acts as the central control unit, providing WiFi capabilities for wireless communication and control via a smartphone or web interface.

Motor Driver Module (L298N or similar): Controls the DC motors that drive the car's wheels and the elevating mechanism.

DC Gear Motors: Power the car's movement and allow the platform to elevate for better navigation over obstacles or uneven surfaces.

WiFi-Based Control Interface: A smartphone or web-based interface (e.g., using a custom webpage or Blynk app) for real-time wireless control of movement and elevation.

Power Supply (Li-ion Battery): Provides energy to the ESP32 and motors, ensuring sufficient power for both mobility and lifting actions.

Chassis with Elevation Mechanism: A specially designed base that includes an elevating frame or actuator system for lifting the car body as needed.

Limit Switches or Position Sensors (optional): Ensure safe and precise control of the elevating mechanism, preventing overextension.

LED Indicators: Provide visual feedback on system status such as WiFi connection, movement mode, or elevation state.

A. System Architecture

The Elevating WiFi Car is built around the ESP32 microcontroller, which handles both WiFi communication and motor control. A web-based interface lets users send commands over WiFi, which the ESP32 processes to control the L293D motor driver for driving the DC motors. An additional motor manages the elevation mechanism, triggered by specific commands. An I2C LCD displays real-time movement and elevation status.

Manual push buttons provide elevation control override, while PWM signals from the ESP32 adjust motor speed. GPIO pins are manage the direction and control through the motor driver. With its dual-core processing, the ESP32 efficiently handles both wireless communication and real-time motor operations, enabling smooth, responsive navigation and elevation.

B. Locomotion and Elevation Mechanism

The car uses four DC motors controlled through an L298N motor driver to achieve directional movement. For elevation, a geared DC motor connected to a screw jack mechanism lifts the platform vertically. Both movement and elevation are controlled via GPIO pins of the ESP32. PWM signals control the motor speed, while digital pins determine direction. The screw jack provides stable and smooth vertical movement, capable of lifting moderate loads. The motor for elevation is powered separately to ensure consistent torque. Safety limits are programmed to prevent over-extension of the elevating platform. The mechanical design ensures balanced weight distribution to maintain stability during lifting.

C. Web-Based Control Interface

A web server is hosted on the ESP32, allowing users to control the car via a smartphone or computer browser. The interface provides buttons for directional control and elevation, enabling real-time interaction without the need for any external applications. The interface is coded using HTML, CSS, and JavaScript embedded in the ESP32 firmware. It automatically refreshes motor states based on user input, giving immediate feedback. Commands sent through the web interface are parsed by the ESP32 to trigger specific GPIO actions. The system is compatible with any device having a web browser, ensuring

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 03 Issue: 12 | Dec-2025
ISSN (O) 3107-6696

high accessibility. This interface allows multiple users to connect sequentially to test or operate the system.

IV. HARDWARE IMPLIMENTATION

The hardware components are selected for their cost-efficiency and compatibility with the ESP32 microcontroller. The integration of these modules enables wireless control and mechanical elevation of the robotic car.

A. ESP32 Microcontroller

The ESP32 serves as the central controller, managing mo-tor control and wireless communication. Its integrated WiFi capability eliminates the need for external network modules, and its GPIO pins support multiple peripheral connections.

B. L293D Motor Driver Module

The ESP32 sends control signals to the L293D motor driver, which regulates the power supplied to the motors, enabling forward and reverse movement for both the driving and lifting mechanisms

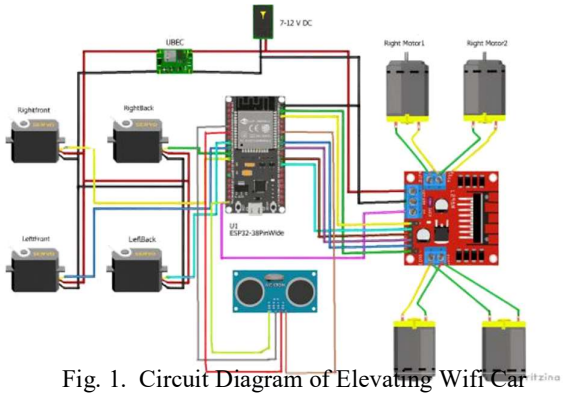

Fig. 1. Circuit Diagram of Elevating Wifi Car

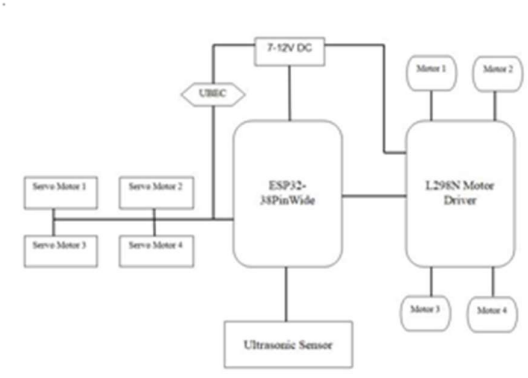

Fig. 2. Block Diagram

C. Dc Motors

Two DC motors are used for forward and reverse motion, and one additional motor is employed for the vertical elevation mechanism. These motors are powered and controlled through the L293D driver.

D. Push Button

Push buttons are included on the car chassis to allow manual control of the elevation system. These provide a local override option, useful during testing or network interruptions.

E. Power Supply Module

A regulated power module delivers stable voltage to the ESP32 and motor driver. It ensures consistent operation, especially when both movement and elevation systems are active simultaneously.

F. WiFi Interface

The ESP32's built-in WiFi module enables wireless com-mand reception without external adapters. It hosts or connects to a local network, it allowing real-time car control through a web-based interface.

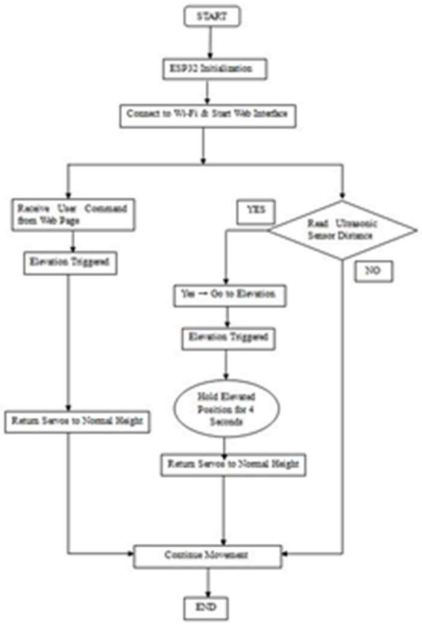

Fig. 3. Flow chart

V. SYSTEM OPERATION

The system Operates in the following sequence:

1) System Initialization

- ESP32 powers on and initializes all hardware components.
- The system connects to a WiFi network or starts its own hotspot.
- Motors and control pins are set to a default idle state.

2) WiFi Control

- The user accesses a web-based interface on a smartphone or computer.
- Direction and elevation commands are sent over WiFi to the ESP32.
- ESP32 processes the commands and triggers the respective motor actions.

3) Motion Control

- ESP32 conveys directional signals to the L293D motor driver.
- Drive motors rotate forward, backward, left, or right as commanded.
- Motor speed is controlled via PWM signals from the ESP32.

4) Elevation Control

- Elevation commands are triggered through the web interface or push buttons.
- ESP32 controls the lifting motor to raise or lower the platform
  - The motor stops automatically after reaching the desired lift duration.

5) Manual Override

- Push buttons provide direct control of the elevation system.
- Useful for testing or when WiFi connection is unavailable.
- Overrides web commands when pressed.

6) System Standby

- After task execution, the system awaits the next command.
- Motors return to idle if no input is received.
- ESP32 maintains a stable connection and responds to incoming WiFi commands.

VI. USER INTERACTION AND FUNCTION PRIORITIZATION

A. Function Prioritization Strategy

When multiple commands are received via WiFi at the same time, the system processes them in the following order:

1) Driving Commands

- Movement commands (forward, backward, left, right) take the highest priority to ensure immediate manual control.

2) Elevation Commands

- These are secondary and will only execute when there is no active movement command to avoid operational conflict.

3) Simultaneous Command Handling

- If both driving and elevation commands are received together, driving commands are processed first.
- Elevation commands are queued until driving stops.

B. Additional User Interaction Features

1) Full App-Based Control:

- All commands are issued via a smartphone app, including directional movement and elevation actions.

2) Command Execution Feedback (Future Scope):

- Future versions may include in-app visual feedback (e.g., indicators or status messages) to show successful command execution.

3) Auto-Reset Between Commands:

- After any command completes, the system returns to an idle state, ready for the next instruction.

4) Software-Based Status Monitoring:

- The user relies on the app interface for all control and monitoring; no physical feedback components are used.

VII. WORKING PROCESS

1) Idle State
   - Car remains inactive until the user connects via WiFi.

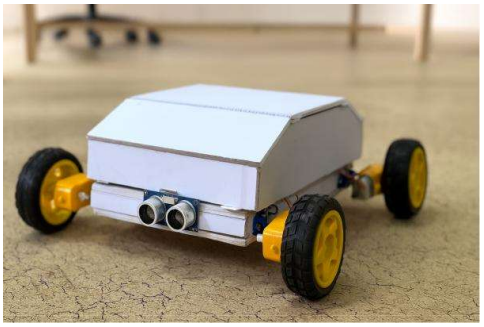

Fig. 4. Side View of Car

- Control interface becomes accessible through smartphone or browser.

2) Command Received

- ESP32 receives motion or elevation instructions via the web interface.

3) Execution Phase

- Drive motors respond to movement commands (forward, reverse, turn).
- Elevation motor activates when a lift or lower command is received.

- Manual push buttons can also trigger elevation locally.

4) No Command / Timeout

- If no input is detected for a set duration, motors stop automatically.
- System returns to idle, maintaining WiFi connection.
- ESP32 awaits the next user command.

The car moved forward, backward, left, and right smoothly via WiFi commands.

VIII. RESULT

The Elevating WiFi Car using ESP32 was successfully tested, and key outcomes include:

- Movement Control: Incorporates cost-effective and readily accessible components such as the ESP32 and DC motors.
- Elevation Mechanism: Servo/linear actuators successfully elevated and lowered the car as intended.

- Response Time: Commands from the smartphone interface were executed within 100–200 ms.
- User Feedback: The interface provided real-time status updates for movement and elevation.

- Reliability: The car maintained stable operation over a 10–15 meter WiFi range without interruptions

IX. APPLICATIONS

1) Remote-Controlled Material Handling
   - Can be used to lift and transport small loads in warehouses or labs using WiFi control.

2) Smart Mobility Platforms

- Useful in assistive technologies where elevation is needed for accessibility or loading purposes.

3) IoT-Based Automation Projects

- Demonstrates wireless control and mechanical automation, ideal for smart system prototypes.

4) Educational Robotics

- Serves as a practical project for students learning about embedded systems, WiFi communication, and motor control.

5) Prototype Development

- Can be adapted for future innovations like delivery robots, inspection bots, or smart service carts

X. ADVANTAGES

- Low Cost:
  Incorporates cost-effective and readily accessible components such as the ESP32 and DC motors.

- Wireless Control:
  Operates fully through WiFi, eliminating the need for physical remotes or Bluetooth.

- Compact Design:

  Fits all hardware into a small chassis, making it suitable for tight or indoor environments.

- Dual Control Options:
  Offers both web-based and manual push-button control for flexibility.

- Expandable Platform:
  Can be enhanced with sensors, cameras, or automation features for advanced applications.

XI. LIMITATIONS AND FUTURE SCOPE

A. Limitations

- Limited Range of WiFi:

  WiFi control is restricted to the range of the network or hotspot.

- No Sensor Feedback:

  Lack of obstacle or position sensors limits autonomous safety features.

- Manual Elevation Duration:

  The elevation system operates on a timed basis without real-time positional feedback, potentially resulting in inaccurate movement control.

B. Future Enhancements

1) Obstacle Detection Integration:
Add ultrasonic or IR sensors to avoid collisions and enable semi-autonomous navigation.

2) Mobile App Interface:
Develop a custom Android/iOS app for enhanced control and real-time status display.

3) GPS-Based Tracking:
Add a GPS module for remote location monitoring and outdoor mobility.

4) Automated Path Navigation:
Integrate line-following or path-planning features for autonomus movement in structured environments.

5) Load Detection Mechanism:
Include weight sensors to prevent overloading during elevation.

XII. CONCLUSION

The Elevating WiFi Car using ESP32 project successfully demonstrates wireless control of a robotic car with an elevation mechanism. The system allows smooth movement in all directions and precise elevation control through a smartphone interface. It operates reliably over a WiFi connection and provides responsive real-time control. This project lays a foundation for further enhancements, such as autonomous navigation, obstacle avoidance, and integration with smart applications, making it a versatile platform for future robotic and IoT-based developments.

ACKNOWLEDGMENT

This project was carried out under the guidance of Prof. Shivarathnamma, Assistant Professor, Tontadarya College of Engineering, Gadag. We express our sincere gratitude for her valuable support, guidance, and encouragement throughout the development of this project

REFERENCES

[1] S. Patil and R. Deshmukh, “Wi-Fi Controlled Robotic Car Using ESP32,” International Journal of Innovative Research in Technology (IJIRT), vol. 7, no. 8, pp. 45–48, Aug. 2020.

[2] A. K. Sharma and M. Pandey, “IoT Based Smart Car Using ESP32 with Elevation Mechanism,” International Journal of Engineering Research Technology (IJERT), vol. 9, no. 11, Nov. 2021.

[3] T. Joshi and K. Mehta, “ESP32 Web Server Controlled Car with Lift System,” IEEE Conference on Emerging Technologies, 2022.

[4] L293D Motor Driver IC Datasheet – Texas Instruments.

[5] “ESP32-WROOM-32 Datasheet,” Espressif Systems.

[6] “ESP32 DevKit V1 Pinout, Features and Uses,” Electronics Hub.

[7] “Web-Based Robot Control Using ESP32,” Arduino Project Hub.

[8] “Push Button Switch Overview,” SparkFun.