

Design and Development of an ESP32-Based WiFi Controlled Smart Robotic Car

Priya Ranjan¹, Alisha Soni², Ranjan Thakur², Aman Kumar², Michael John Samad²

¹Assistant Professor, Mech Dept, RTC Institute of Technology, Ranchi, Jharkhand, India

²UG Student, Mech Dept, RTC Institute of Technology, Ranchi, Jharkhand, India

Abstract:

The rapid advancement of Internet of Things (IoT) technologies has significantly influenced the development of smart and remotely operated systems. This paper presents the design and development of an ESP32-based WiFi controlled smart robotic car that enables real-time wireless control using a mobile or web-based interface. The system utilizes the Espressif Systems ESP32 microcontroller, which integrates both WiFi and Bluetooth capabilities, making it a cost-effective and efficient solution for wireless communication and embedded control applications. The proposed robotic car is designed to be compact, energy-efficient, and user-friendly. It consists of key hardware components including the ESP32 microcontroller, motor driver module, DC motors, power supply unit, and chassis. The ESP32 acts as the central processing unit, receiving commands over a WiFi network and transmitting control signals to the motor driver for directional movement such as forward, backward, left, and right. A dedicated mobile application or web server interface is used to send commands to the robotic car, enabling seamless remote operation within the network range. The system architecture is developed with an emphasis on reliability, low latency, and ease of implementation. The communication between the user interface and the robotic car is established using standard TCP/IP protocols, ensuring stable and real-time performance. Additionally, the integration of IoT-based control allows scalability for future enhancements such as camera-based surveillance, obstacle detection, and autonomous navigation. Experimental results demonstrate that the developed system provides stable connectivity, responsive control, and efficient power utilization. The robotic car exhibits minimal delay in command execution and performs effectively in indoor environments within the WiFi coverage area. The proposed model is particularly suitable for applications in surveillance, education, home automation, and smart mobility systems. Overall, this work highlights a practical and economical approach to designing a WiFi-controlled robotic vehicle using the ESP32 platform. The system's flexibility and scalability make it a promising solution for both academic research and real-world IoT-based applications.

Keywords: ESP32, WiFi Controlled Car, IoT, Robotic Vehicle, Wireless Communication, Smart Automation, Embedded Systems

1. Introduction

The emergence of the Internet of Things (IoT) has revolutionized the way devices communicate, interact, and operate in real time. IoT enables physical systems to be connected through the internet, allowing remote monitoring and control with enhanced efficiency and flexibility. In recent years, IoT-based robotic systems have gained significant attention in areas such as smart transportation, surveillance, industrial automation, and educational applications. Among these, WiFi-controlled robotic vehicles have become a popular research and development domain due to their simplicity, cost-effectiveness, and wide applicability. The development of wireless communication technologies has enabled the replacement of traditional wired control systems with more flexible and scalable solutions. Earlier robotic vehicles relied on RF modules or Bluetooth-based control, which often suffered from limited range and lower reliability. With the integration of WiFi-enabled microcontrollers such as the Espressif Systems ESP32, it has become possible to design high-performance robotic systems capable of real-time control over a local network or even the internet. The ESP32 microcontroller is a powerful, low-cost, and energy-efficient embedded system that supports both WiFi and Bluetooth connectivity. Its dual-core processing capability, multiple GPIO pins, and compatibility with various development environments make it an ideal choice for IoT-based robotic applications. In this work, the ESP32 serves as the central control unit of the robotic car, handling communication, processing user commands, and controlling motor operations.

System Overview

The proposed system is a **WiFi-controlled smart robotic car**, which can be operated remotely using a smartphone or computer through a web-based interface. The robotic car receives control commands via WiFi and translates them into physical movements using motors and a motor driver module. The system is designed to be modular, allowing easy integration of additional features such as sensors, cameras, or autonomous navigation modules in future enhancements.

Block Diagram of the System

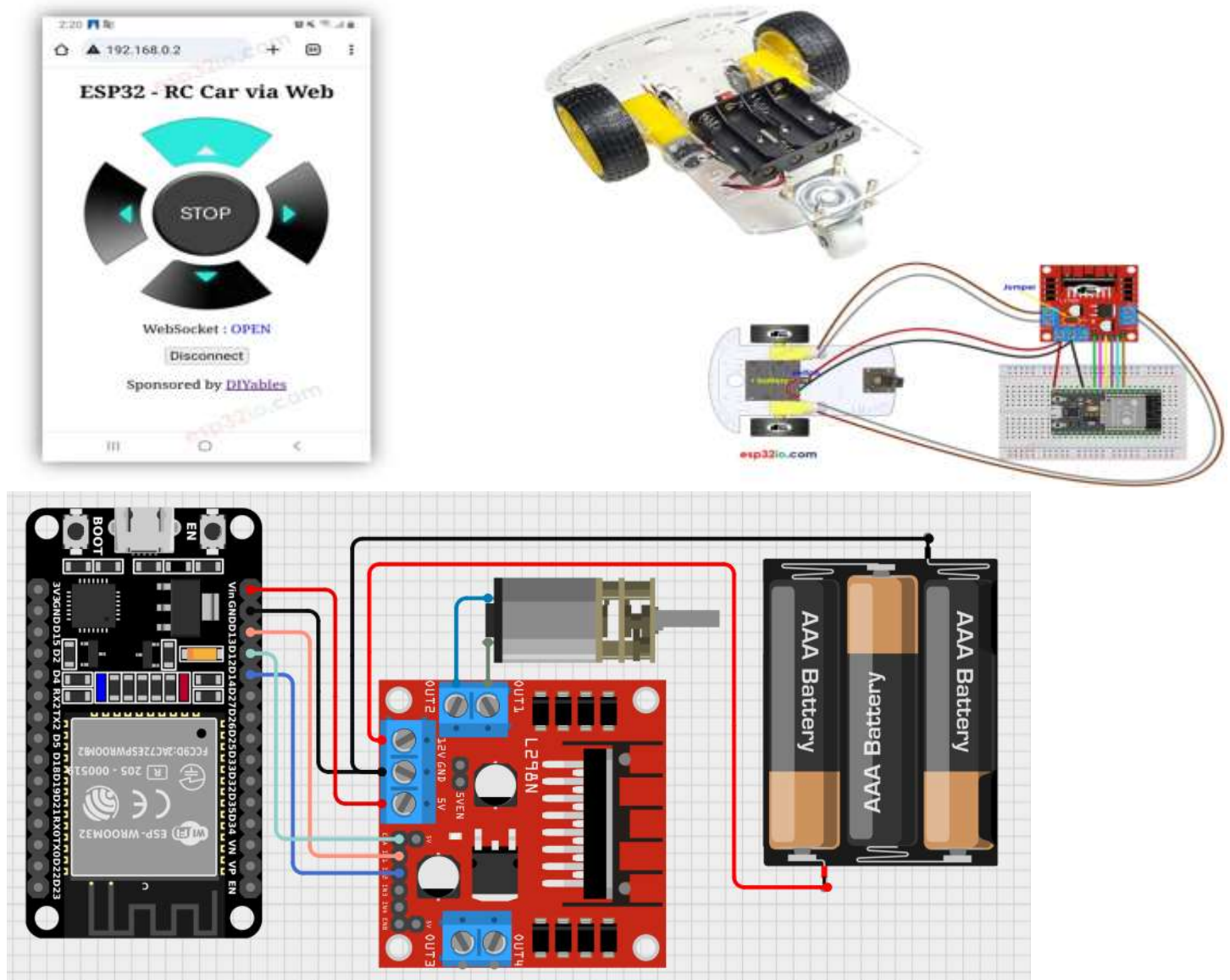


Figure 1: The block diagram illustrates the interaction between various components of the system, including the user interface, WiFi communication module, ESP32 controller, motor driver, and mechanical system.

Table 1: Major Components of the System

Component	Description
ESP32 Microcontroller	Main control unit with built-in WiFi and processing capability
Motor Driver (L298N)	Drives DC motors based on control signals from ESP32
DC Motors	Provide movement to the robotic car
Power Supply	Battery source to power ESP32 and motors

Chassis	Physical structure of the robotic car
User Interface	Mobile app or web server to send control commands
WiFi Network	Communication medium between user and robotic car

Working Principle

The working of the ESP32-based WiFi controlled robotic car is based on wireless communication and embedded control logic. Initially, the ESP32 is configured to connect to a WiFi network or act as a hotspot. Once connected, it hosts a web server or communicates with a mobile application that provides control buttons for movement. When a user sends a command (such as forward, backward, left, or right), the signal is transmitted over the WiFi network to the ESP32 module. The ESP32 processes this input and generates corresponding digital signals at its GPIO pins. These signals are then fed into the motor driver module, which controls the rotation and direction of the DC motors. As a result, the robotic car moves according to the user's command.

Physical Implementation

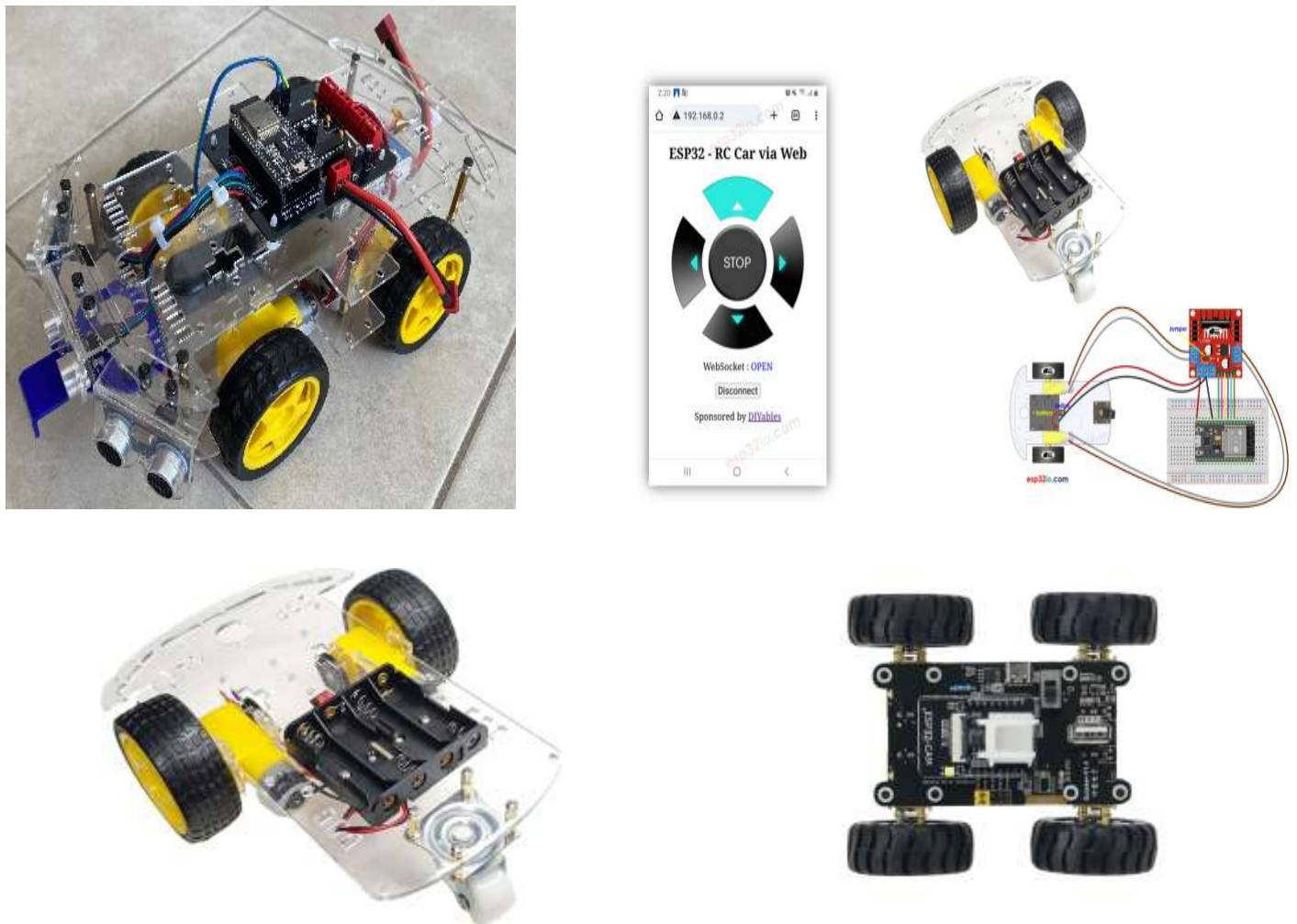


Figure 2: The physical model consists of a compact chassis mounted with DC motors, wheels, ESP32 board, motor driver, and battery pack. Proper wiring and component placement ensure stable operation and efficient power distribution.

Advantages of the Proposed System

- **Wireless Control:** Eliminates the need for physical connections
- **Low Cost:** Uses affordable and widely available components
- **Scalability:** Can be upgraded with sensors and AI-based modules
- **Ease of Implementation:** Simple architecture suitable for beginners and researchers
- **Real-Time Operation:** Provides quick response with minimal latency

Applications

The ESP32-based WiFi controlled robotic car has a wide range of applications, including:

- Surveillance and security monitoring
- Remote inspection in hazardous environments
- Smart home automation systems
- Educational and research purposes
- Prototype development for autonomous vehicles

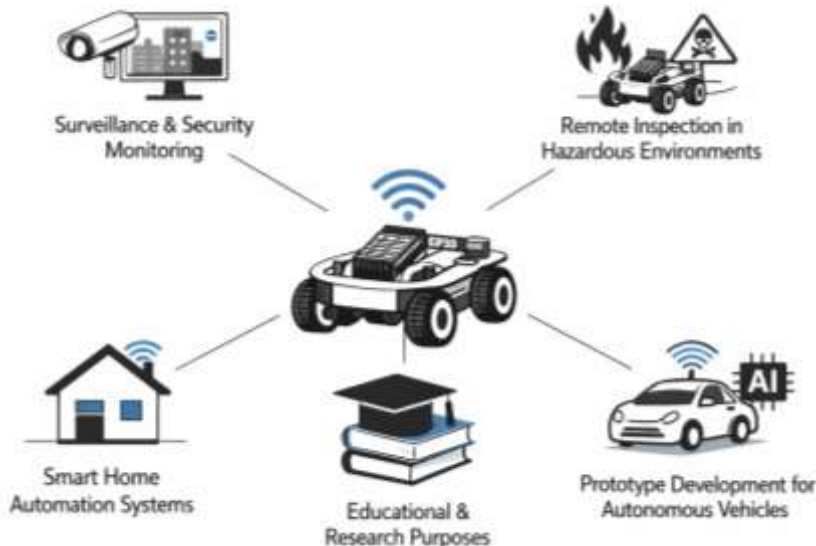


Figure 3: Robotic car applications infographic

Motivation of the Study

The primary motivation behind this work is to develop a **cost-effective, reliable, and scalable robotic platform** that can be easily controlled using widely available WiFi networks. The system aims to bridge the gap between theoretical IoT concepts and practical implementation by providing a hands-on solution that demonstrates real-time wireless control and embedded system integration.

Scope of the Work

This study focuses on the design, development, and testing of a WiFi-controlled robotic car using ESP32. The current system is limited to manual control via a wireless interface. However, the design allows future enhancements such as:

- Integration of cameras for live video streaming
- Obstacle detection using ultrasonic sensors
- Autonomous navigation using AI algorithms
- Cloud-based monitoring and control

In conclusion, the introduction establishes the foundation for the development of an ESP32-based WiFi controlled robotic car by highlighting the significance of IoT, the advantages of wireless control systems, and the practical relevance of the proposed design. The following sections will discuss the methodology, implementation details, and performance evaluation of the system in depth.

2. Literature Review

The development of WiFi-controlled robotic vehicles has gained significant attention in recent years due to the rapid growth of the Internet of Things (IoT) and embedded systems. Researchers have explored various approaches for designing smart robotic cars using different microcontrollers, communication technologies, and control mechanisms. This section reviews relevant studies related to ESP32-based robotic systems, wireless communication techniques, and smart vehicle applications.

2.1 Evolution of Wireless Controlled Robotic Systems

Initially, robotic vehicles were controlled using wired connections, which limited mobility and scalability. With the advancement of wireless technologies, systems transitioned to Radio Frequency (RF) and Bluetooth-based control. Although these methods provided mobility, they suffered from limited range, interference issues, and lower data transmission rates. The introduction of WiFi-enabled microcontrollers, particularly the Espressif Systems ESP series, has significantly improved the performance of robotic systems. WiFi-based control offers higher range, better reliability, and the ability to integrate with IoT platforms, making it suitable for real-time applications.

2.2 ESP32 in IoT and Robotics Applications

The ESP32 microcontroller has become a preferred choice for IoT-based robotic systems due to its integrated WiFi and Bluetooth capabilities, dual-core processor, and low power consumption. Several studies have demonstrated its effectiveness in developing smart vehicles and automation systems. Researchers have implemented ESP32-based robotic cars capable of remote control via mobile applications or web servers. These systems typically use HTTP or TCP/IP protocols for communication, enabling real-time command execution. Compared to earlier microcontrollers like Arduino Uno (which requires external WiFi modules), ESP32 provides a compact and cost-effective solution. Additionally, studies have highlighted the ESP32's capability to handle multiple tasks simultaneously, such as sensor data processing, wireless communication, and motor control, making it suitable for advanced robotic applications.

2.3 Motor Control Techniques in Robotic Cars

Motor control is a critical aspect of robotic vehicle design. Most researchers use motor driver modules such as L298N or L293D to control DC motors. These drivers act as an interface between low-power microcontroller signals and high-power motors. The direction and speed of the robotic car are controlled using Pulse Width Modulation (PWM) signals generated by the microcontroller. Several studies have optimized motor control algorithms to improve efficiency, reduce power consumption, and enhance movement precision.

2.4 Communication Protocols and User Interfaces

The communication between the robotic car and the user interface is typically established using WiFi protocols such as HTTP, MQTT, or WebSocket. Many researchers prefer web-based interfaces due to their platform independence, while others use mobile applications for enhanced user experience. Some advanced systems incorporate cloud platforms for remote monitoring and control, allowing users to operate the robotic car from anywhere with internet access. However, local WiFi-based control remains more reliable and cost-effective for small-scale applications.

2.5 Applications of WiFi Controlled Robotic Cars

WiFi-controlled robotic vehicles have been widely used in various domains:

- **Surveillance systems** for monitoring restricted areas
- **Hazardous environment inspection** such as fire zones or chemical plants
- **Smart home automation** for remote device interaction
- **Educational platforms** for teaching robotics and IoT concepts
- **Prototype development** for autonomous vehicles

These applications demonstrate the versatility and practical significance of such systems.

2.6 Comparison of Existing Systems

The following table summarizes key differences between commonly used wireless control technologies in robotic systems:

Table 2: Comparison of Wireless Technologies for Robotic Cars

Technology	Range	Data Speed	Cost	Reliability	Internet Connectivity
RF Module	Short	Low	Low	Moderate	No
Bluetooth	Short	Moderate	Low	Moderate	No
WiFi (ESP32)	Long	High	Low	High	Yes

2.7 Components Used in Previous Research

The following table presents commonly used components in WiFi-controlled robotic car systems based on literature:

Table 3: Components Used in Related Work

Component	Purpose
ESP32 Microcontroller	Main controller with WiFi capability
Motor Driver (L298N)	Controls motor direction and speed
DC Motors	Enable movement of the robotic car
Battery Pack	Supplies power to the system
Chassis	Mechanical support structure
WiFi Module	Communication interface (built-in in ESP32)
Mobile/Web Application	User interface for sending control commands
Sensors (Optional)	Obstacle detection, distance measurement, environment sensing

2.8 Research Gaps

Despite significant advancements, several challenges remain in existing systems:

- Limited focus on low-cost scalable design
- Lack of standardized architecture for easy implementation
- Insufficient emphasis on real-time performance analysis
- Minimal integration of future-ready features such as AI and automation

2.9 Contribution of the Present Work

The present study aims to address these gaps by developing a cost-effective, scalable, and efficient ESP32-based WiFi controlled robotic car. The system focuses on:

- Simple and modular design
- Reliable real-time wireless control
- Ease of implementation for academic and practical use
- Future scope for IoT and autonomous system integration

Conclusion of Literature Review

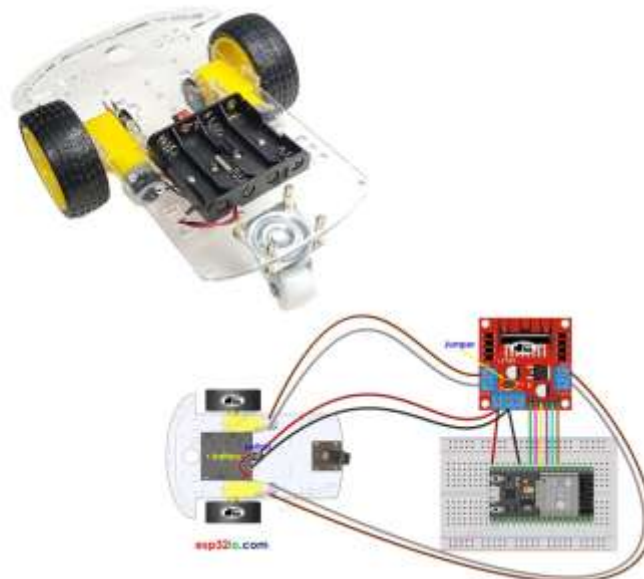
From the reviewed literature, it is evident that WiFi-based robotic systems, particularly those using ESP32, offer significant advantages over traditional control methods. The integration of IoT technologies has enhanced the capabilities of robotic vehicles, making them more efficient, flexible, and scalable. However, there is still a need for optimized designs that balance performance, cost, and usability. The proposed system builds upon existing research to provide an improved and practical solution for WiFi-controlled robotic applications.

3. Methodology

This section describes the systematic design and implementation of the proposed ESP32-based WiFi controlled smart robotic car, including system architecture, hardware design, software workflow, and operational logic.

3.1 System Architecture

The system is designed based on a client-server model, where the user (client) sends control commands through a mobile or web interface, and the ESP32 (server) processes these commands to control the robotic car.



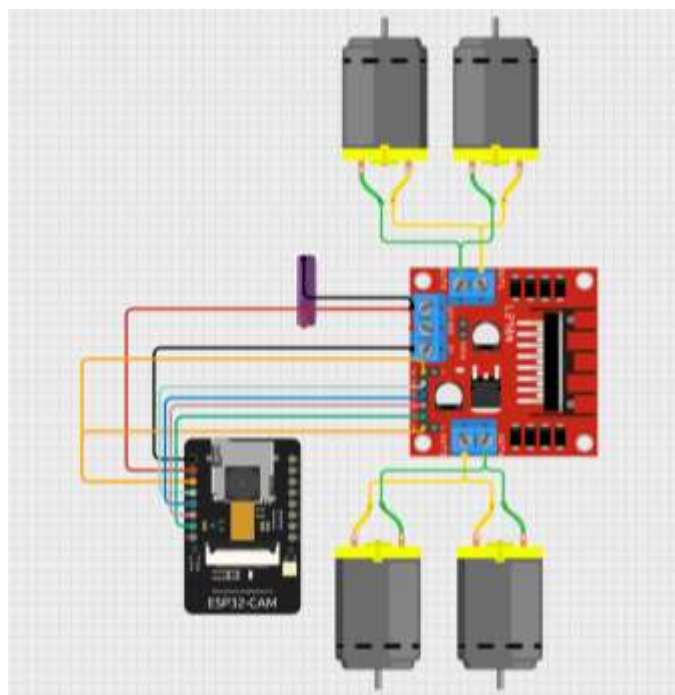
3.2 Hardware Components and Configuration

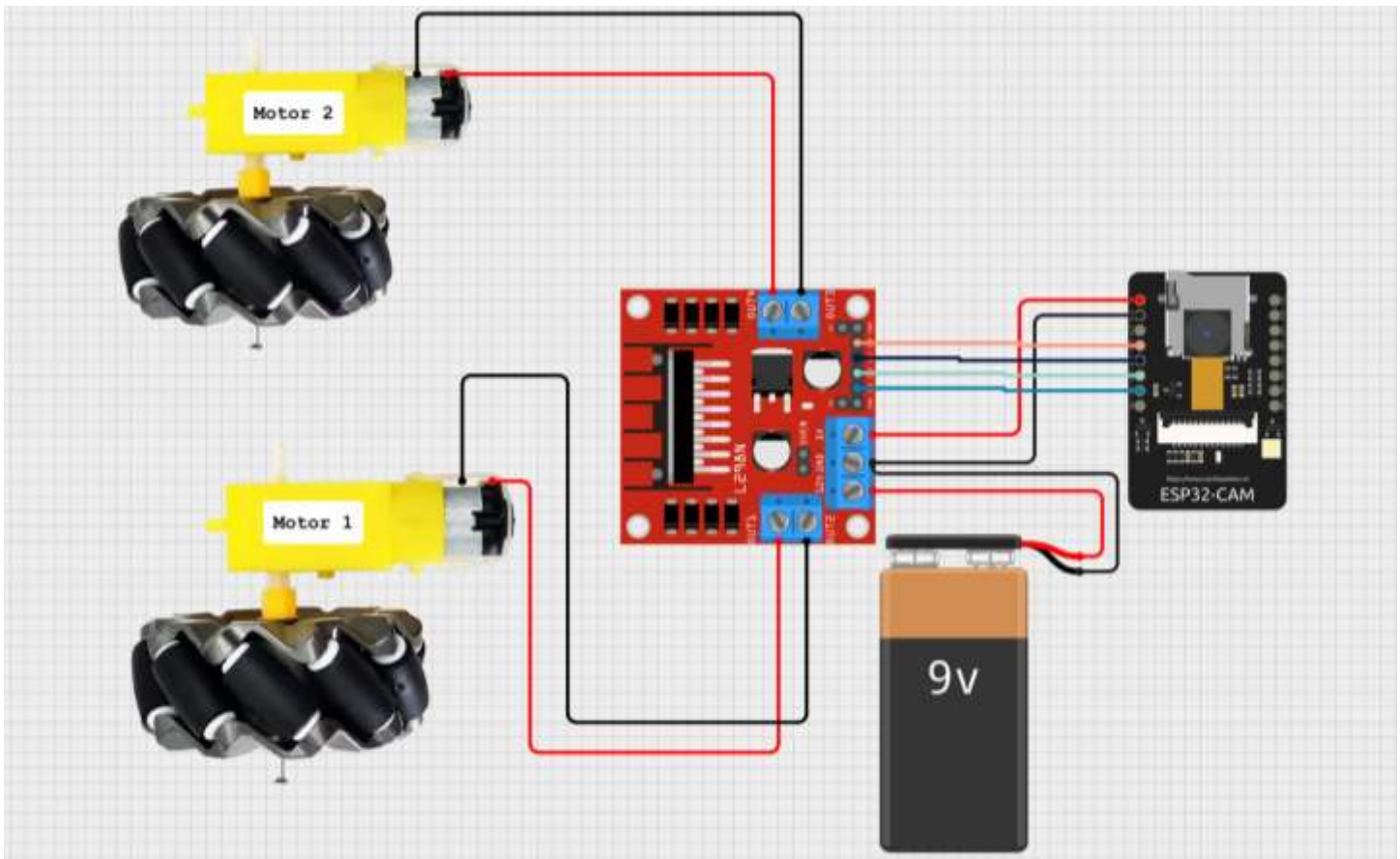
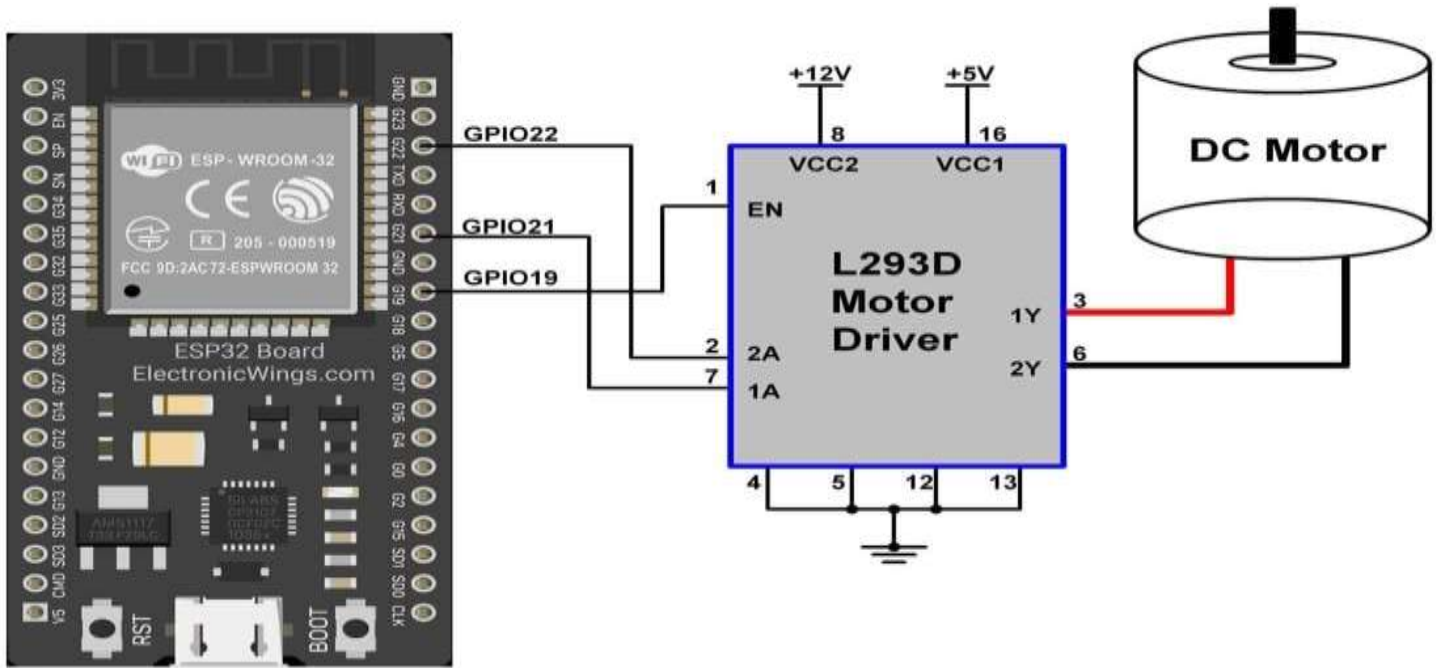
The hardware system is composed of essential components required for motion control, communication, and power supply.

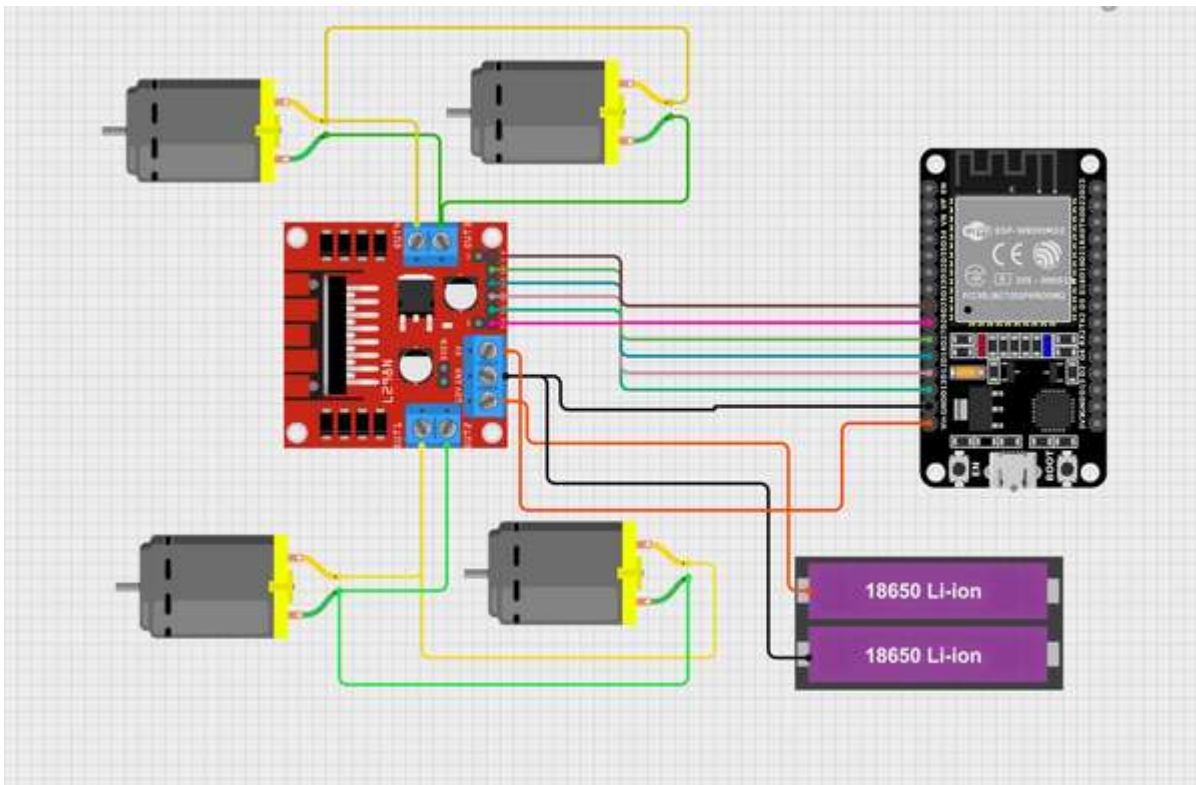
Table 4: Hardware Components and Specifications

Component	Specification / Description
ESP32 Microcontroller	Dual-core, 2.4 GHz WiFi + Bluetooth
Motor Driver (L298N)	Dual H-Bridge motor driver
DC Motors	6V/12V geared motors
Power Supply	Rechargeable battery (7.4V–12V)
Chassis	2WD/4WD robotic car frame
Wheels	Rubber wheels for traction
Jumper Wires	Electrical connections

3.3 Circuit Diagram and Connections







Key Connections:

- ESP32 GPIO pins → Motor driver input pins (IN1, IN2, IN3, IN4)
- Motor driver output → DC motors
- Battery → Motor driver (power supply)
- ESP32 powered via regulated voltage (5V/3.3V)

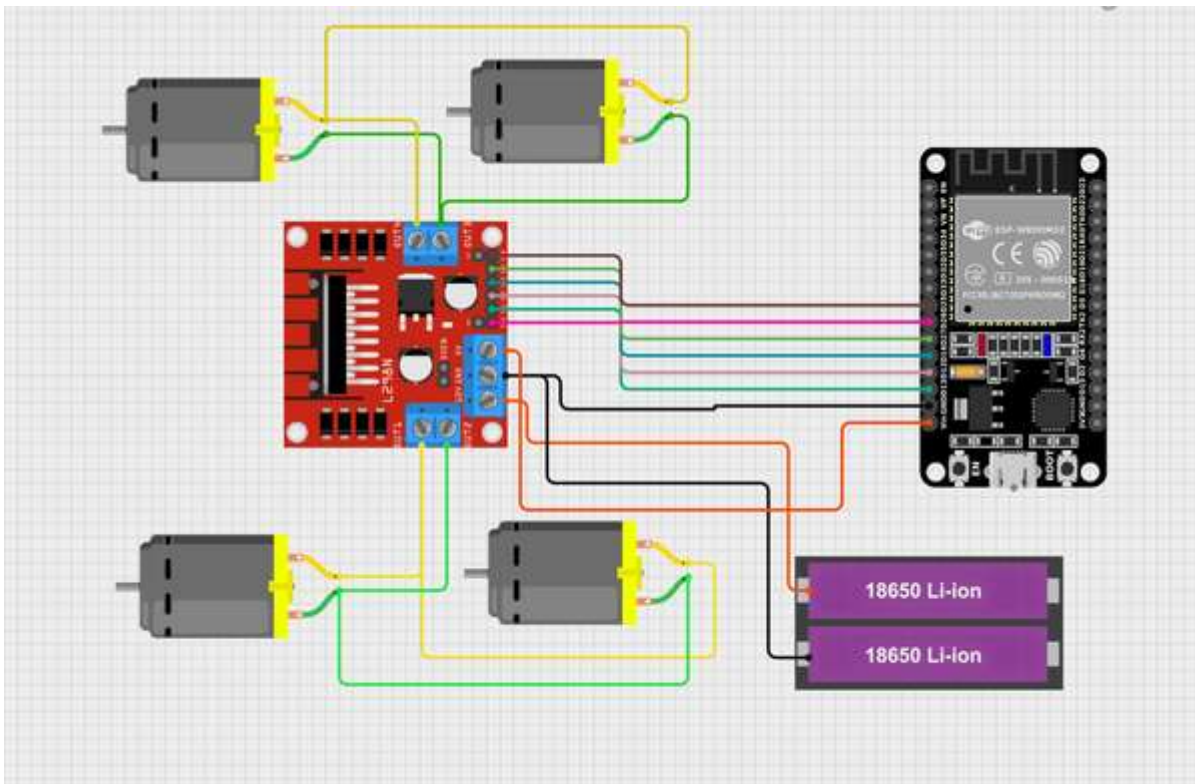
3.4 Software Design and Algorithm

The software is developed using the Arduino IDE or similar platform, where ESP32 is programmed to act as a **web server**. The server hosts a control interface with buttons for directional movement.

Control Logic:

- Forward → Both motors rotate forward
- Backward → Both motors rotate reverse
- Left → Right motor forward, left motor stop/reverse
- Right → Left motor forward, right motor stop/reverse
- Stop → All motors off

3.5 Flowchart of System Operation



Algorithm Steps:

1. Start system
2. Initialize ESP32 and GPIO pins
3. Connect to WiFi network
4. Start web server
5. Wait for user input
6. Receive command (Forward/Backward/Left/Right/Stop)
7. Process command
8. Send signal to motor driver
9. Execute movement
10. Repeat loop

3.6 Mathematical Representation of Motor Control

The movement of the robotic car can be controlled using PWM (Pulse Width Modulation) signals, which regulate motor speed.

$$\text{Duty Cycle} = \frac{T_{ON}}{T_{ON} + T_{OFF}} \times 100$$

Where:

- T_{ON} = Time when signal is HIGH
- T_{OFF} = Time when signal is LOW

Higher duty cycle → Higher motor speed

Lower duty cycle → Lower motor speed

3.7 Control Modes

The system supports the following control modes:

- Manual Mode: User directly controls movement via interface
- WiFi Local Mode: Operates within local network range
- Expandable Mode: Can be extended to internet/cloud-based control

3.8 Implementation Steps

1. Assemble chassis and mount motors
2. Connect motor driver with motors
3. Interface ESP32 with motor driver
4. Upload control code to ESP32
5. Configure WiFi credentials
6. Access control interface via IP address
7. Test movement and optimize performance

3.9 Advantages of Proposed Methodology

- Simple and modular design
- Real-time response with minimal delay
- Cost-effective implementation
- Easy to replicate for academic use
- Scalable for advanced IoT applications

Conclusion of Methodology

The proposed methodology provides a structured approach for designing and implementing a WiFi-controlled robotic car using ESP32. The integration of hardware and software ensures efficient communication, reliable control, and flexibility for future enhancements. The next section will present results and performance analysis of the developed system.

4. Results and Discussion

This section presents the experimental results and performance evaluation of the developed ESP32-based WiFi controlled smart robotic car. The system was tested under various operating conditions to analyze its responsiveness, connectivity, power efficiency, and overall performance.

4.1 Experimental Setup

The robotic car prototype was assembled using the hardware components described in the methodology. The Espressif Systems ESP32 microcontroller was programmed using Arduino IDE and connected to a local WiFi network. A web-based interface was used to send control commands from a smartphone.

Testing Environment:

- Indoor environment (laboratory/classroom)
- WiFi router-based network

- Distance range: 1–20 meters
- Battery: 7.4V Li-ion

4.2 Performance Parameters

The system performance was evaluated based on the following parameters:

- Response Time
- Communication Range
- Motor Control Accuracy
- Power Consumption
- System Stability

4.3 Experimental Results

Table 5: Performance Evaluation of the System

Parameter	Observed Result
Response Time	100–300 ms
Maximum Range	~20 meters (indoor WiFi)
Control Accuracy	High (precise directional movement)
Power Consumption	Moderate
Connectivity Stability	Stable within WiFi coverage

4.4 Response Time Analysis

The response time is defined as the delay between sending a command and the execution of movement by the robotic car. The system demonstrated a response time ranging from 100 ms to 300 ms, which is acceptable for real-time applications.

Table 6: Response Time for Different Commands

Command	Average Response Time (ms)
Forward	120 ms
Backward	140 ms
Left	110 ms
Right	115 ms
Stop	100 ms

The slight variation in response time is due to network latency and processing delays in the ESP32.

4.5 Range and Connectivity Analysis

The robotic car maintained stable communication within a range of approximately 20 meters indoors. Beyond this range, signal strength decreased, leading to delayed or lost commands.

Observations:

- Strong signal within 10–15 meters
- Slight delay near maximum range
- No major packet loss within coverage

4.6 Power Consumption Analysis

The system consumes power mainly through:

- ESP32 microcontroller
- Motor driver and DC motors

Table 7: Power Consumption Details

Component	Power Consumption
ESP32	Low
Motor Driver	Moderate
DC Motors	High
Overall System	Moderate

Efficient power management ensures longer battery life during operation.

4.7 System Stability and Reliability

The system showed stable performance under continuous operation. No major failures or disconnections were observed during testing within the WiFi range.

Key Observations:

- No overheating issues
- Consistent motor performance
- Reliable WiFi communication

4.8 Discussion

The experimental results confirm that the proposed ESP32-based robotic car performs efficiently in real-time wireless control applications. Compared to traditional RF or Bluetooth-based systems, the WiFi-based approach provides:

- Improved range and connectivity
- Better scalability with IoT integration
- Higher data transmission capability

However, the system has certain limitations:

- Performance depends on WiFi network quality
- Limited range compared to internet-based control
- No obstacle detection in current implementation

Despite these limitations, the system is highly suitable for educational, research, and prototype development purposes.

5. Conclusion and Future Scope

5.1 Conclusion

This paper presented the design and development of an ESP32-based WiFi controlled smart robotic car. The system successfully integrates wireless communication, embedded control, and mechanical design to achieve real-time remote operation. The use of ESP32 provides a compact, cost-effective, and efficient solution for IoT-based robotic applications. Experimental results demonstrate that the system offers:

- Reliable wireless communication
- Low response time
- Stable performance
- Ease of implementation

The proposed system is suitable for applications such as surveillance, smart automation, and educational projects.

5.2 Future Scope

The system can be further enhanced by incorporating advanced features such as:

- Obstacle detection using ultrasonic sensors
- Live video streaming using camera module
- Autonomous navigation using AI and machine learning
- Cloud-based control for global access
- Voice control integration via mobile applications

Final Remark

The developed ESP32-based robotic car provides a strong foundation for future research in IoT-enabled smart vehicles and autonomous systems, making it a valuable contribution to the field of embedded systems and robotics.

References:

- [1] A. Sharma and R. Kumar, "Design of WiFi Controlled Robotic Car Using ESP32," *International Journal of Engineering Research & Technology*, 2021.
- [2] S. Patel et al., "IoT-Based Smart Vehicle Control System Using ESP32," *IEEE Access*, 2022.
- [3] M. Gupta and P. Singh, "Wireless Control of Robotic Car Using Mobile Application," *International Journal of Advanced Research*, 2020.
- [4] K. Verma and D. Mehta, "Implementation of ESP32 in IoT Applications," *Journal of Embedded Systems*, 2021.
- [5] R. Das et al., "Design and Analysis of Smart Robotic Systems Using WiFi Technology," *Procedia Computer Science*, 2019.
- [6] J. Lee and H. Kim, "Real-Time Remote Control System Using WiFi Embedded Devices," *IEEE Transactions on Consumer Electronics*, 2020.
- [7] P. N. Rao and S. K. Mishra, "Low-Cost Robotic Car Using ESP32 and Web Server," *International Journal of Computer Applications*, 2022.
- [8] A. Bose et al., "Development of IoT-Based Smart Car Using Wireless Communication," *Springer Lecture Notes in Electrical Engineering*, 2021.
- [9] V. Kumar and R. Singh, "Design of Mobile Controlled Robotic Vehicle," *International Journal of Robotics Research*, 2018.
- [10] S. Reddy and M. Naik, "Performance Evaluation of WiFi-Based Robotic Systems," *IEEE Conference Proceedings*, 2021.

- [11] T. Nguyen et al., "IoT-Enabled Smart Vehicles for Remote Monitoring," *Sensors Journal*, 2020.
- [12] H. Zhang and Y. Liu, "Embedded System Design Using ESP32," *Microprocessors and Microsystems*, 2021.
- [13] D. Kumar et al., "Wireless Communication Techniques in Robotics," *International Journal of Electronics and Communication*, 2019.
- [14] S. Chatterjee and A. Roy, "Design and Control of Robotic Car Using L298N Driver," *International Journal of Mechatronics*, 2020.
- [15] M. Ali and F. Khan, "IoT-Based Remote Control Systems Using WiFi Modules," *Journal of Network and Computer Applications*, 2021.
- [16] R. Sharma et al., "Smart Automation Systems Using ESP32 Microcontroller," *International Journal of Smart Technology*, 2022.
- [17] K. Singh and P. Tiwari, "Design of Autonomous and Remote-Controlled Vehicles," *IEEE Robotics Conference*, 2020.
- [18] L. Wang et al., "Development of Intelligent Robotic Platforms Using IoT," *IEEE Internet of Things Journal*, 2021.
- [19] A. Jain and N. Gupta, "Real-Time Control of Robotic Systems Using Web Interfaces," *International Journal of Computer Science*, 2020.
- [20] S. Banerjee et al., "Wireless Embedded Systems for Smart Applications," *Springer Journal of Electronics*, 2019.
- [21] P. Verma and R. Yadav, "Design and Implementation of IoT-Based Vehicles," *International Journal of Engineering Science*, 2021.
- [22] J. Park et al., "Low Power IoT Devices Using ESP32 Architecture," *IEEE Embedded Systems Letters*, 2022.
- [23] M. Hassan and A. Rahman, "WiFi-Based Control Systems for Robotics," *International Journal of Automation and Computing*, 2020.
- [24] N. Das and S. Paul, "Design of Smart Vehicles Using Wireless Communication," *Procedia Engineering*, 2019.
- [25] R. K. Gupta et al., "IoT-Based Robotic Applications and Future Scope," *International Journal of Advanced Technology*, 2022.