International Journal of Current Engineering and Technology
E-ISSN 2277-4106, P-ISSN 2347-5161

Available at https://ijcet.evegenis.org/index.php/ijcet/index

Research Article

Wireless Robotic Navigation via IoT-Enabled Wi-Fi Communication

Akash Kumar[1], Akshay Verma[2], Vikash Pal[3], Vishal Chaudhary[4], Ashish Tripathi[5]

[1,2,3]Student, [4,5,]Assistant Professor, Dept. of Electronics and Communication Engineering, Meerut Institute of Technology, Meerut, UP, India.

Received 01 Feb 2026, Accepted 18 Feb 2026, Available online 19 Feb 2026, **Vol.16, No.1 (Jan/Feb 2026)**

Abstract

With the rapid convergence of the Internet of Things (IoT) and embedded robotics, Wi-Fi-driven mobile platforms have emerged as practical solutions for remote sensing, navigation, and automation. This paper presents the conceptual framework, hardware architecture, software design, and performance evaluation of a Wi-Fi-controlled robotic system built around an ESP8266/ESP32 microcontroller. The platform communicates with a user device-smartphone, tablet, or laptop-via TCP/IP over a local wireless network, enabling low-latency directional control and optional live data streaming. Compared with earlier radio-frequency (RF) or infrared (IR) paradigms, the proposed system achieves superior operating range, bandwidth, and expandability. Experimental outcomes confirm consistent command execution and stable wireless connectivity under typical indoor conditions. Potential deployment scenarios include surveillance, industrial inspection, search-and-rescue operations, and smart-home automation. The work underscores how consumer-grade Wi-Fi chipsets and open-source microcontroller frameworks can lower the barrier to building capable, network-aware robotic systems.

Keywords: *Wi-Fi Robot, ESP8266, ESP32, IoT, Motor Driver, L293D, L298N, Wireless Control, Embedded Systems, Remote Navigation*

1. Introduction

The fusion of wireless networking with mobile robotics has given rise to a new generation of internet-enabled autonomous platforms. A Wi-Fi controlled robot represents one of the most accessible implementations of this fusion: it uses a standard IEEE 802.11 wireless link to relay directional commands from an operator device to an onboard embedded controller, which in turn drives the mechanical subsystem in real time.

Growing demand for smart automation, contactless surveillance, and hazard-free remote inspection has elevated the importance of wireless robotic systems across industrial, military, agricultural, and consumer sectors. Unlike wired tele-operation or legacy RF systems, Wi-Fi communication leverages existing network infrastructure, offers multi-megabit bandwidth, and supports bidirectional data exchange-making it inherently suitable for future extensions such as live video feeds, telemetry logging, or cloud-based control.

This paper documents the end-to-end development of a Wi-Fi robotic platform: from hardware selection and schematic design to firmware architecture and performance characterization. The core controller interprets HTTP or TCP-based command packets received from a browser or mobile application and translates them into pulse-width signals for the motor driver stage.

*Corresponding author's ORCID ID: 0000-0000-0000-0000
DOI: https://doi.org/10.14741/ijcet/v.16.1.4

The modular architecture also accommodates plug-in expansion with ultrasonic sensors, servo actuators, and camera modules without redesigning the base platform.

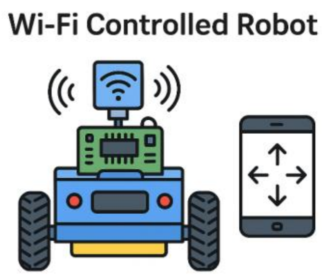

Figure 1: Wi-Fi Controlled Robot-Mobile Platform with Wireless Interface

2. Literature Review

Early remote-control robotic research relied heavily on 433 MHz RF modules and infrared transceivers. While adequate for toy-grade applications, these technologies imposed strict line-of-sight requirements, narrow bandwidth, and effective ranges rarely exceeding 30 metres. The introduction of Bluetooth modules (HC-05, HC-06) improved usability and data rates but kept the operational envelope firmly under 10–15 metres in typical environments.

The maturation of 802.11b/g/n Wi-Fi standards, together with the proliferation of affordable single-chip Wi-Fi SoCs—most notably Espressif Systems' ESP8266 (2014) and ESP32 (2016)—opened an entirely new design space. Researchers demonstrated that these sub-USD-5 modules could host a full TCP/IP stack, serve web pages, and maintain real-time bidirectional communication, all while drawing modest power from a small LiPo cell. Studies such as those by Singh & Kumar (2020) and Vesterbacka & Nummenmaa (2022) confirmed sub-100 ms command latency over local Wi-Fi networks, sufficient for smooth teleoperation.

Parallel work on motor-driver ICs (L293D, L298N, DRV8833) consolidated best practices for interfacing low-power microcontrollers with DC brush motors, including H-bridge topologies, current-limiting strategies, and PWM-based speed control. The combination of capable Wi-Fi SoCs and robust motor-driver libraries has lowered the complexity threshold for building competent wireless robotic platforms, enabling academic and hobbyist teams alike to iterate rapidly on both hardware and firmware.

3. Problem Statement

A significant operational challenge across hazardous industry, disaster relief, and border surveillance is the need for human operators to enter or remain in environments that pose acute physical risk. Conventional tele-operated systems either require cabling-severely limiting mobility-or depend on short-range wireless links that cannot sustain a connection once the operator retreats to a safe distance.

The objective of this work is therefore to design a Wi-Fi-controlled mobile robot capable of: (i) receiving and executing directional commands over a standard wireless LAN with latency low enough for real-time control; (ii) optionally streaming sensor or video data back to the operator; (iii) maintaining reliable connectivity across the coverage area of a typical indoor access point; and (iv) doing so within a compact, energy-efficient platform that can be extended with additional sensing payloads without hardware redesign.

4. Objectives

- Design a compact mobile robotic chassis controllable over an IEEE 802.11 Wi-Fi link.
- Implement bidirectional wireless communication between a user device and the onboard controller.
- Achieve real-time directional motion—forward, reverse, pivot-left, and pivot-right-with response latency below 150 ms.
- Reduce operational risk by enabling remote control from outside hazardous zones.
- Demonstrate integration of embedded firmware, motor-driver ICs, and IoT-layer communication.
- Build a platform extensible with cameras, ultrasonic sensors, GPS, or robotic arm attachments.
- Verify system reliability, range, and latency under realistic indoor operating conditions.
- Establish a repeatable, low-cost design methodology that can be adopted by other research teams.

5. System Design and Components

5.1 Hardware Architecture

The platform follows a layered architecture with five functional tiers:

- User Interface Tier-A responsive web dashboard or mobile application hosted by the controller itself.
- Wireless Communication Tier-802.11 b/g/n Wi-Fi handled by the ESP8266/ESP32 SoC.
- Processing Tier-Firmware running on the microcontroller decodes commands and orchestrates GPIO outputs.
- Actuation Tier-A dual H-bridge motor driver IC (L298N or L293D) converts logic-level signals into motor currents.
- Power & Sensing Tier-Li-ion battery pack, voltage regulators, and optional ultrasonic/IR sensors.

Table 1 Key hardware components

Component	Specification / Model	Function
Wi-Fi SoC	ESP8266 / ESP32	Wireless comms + main MCU
Motor Driver	L298N / L293D	Dual H-bridge motor control
DC Motors (×4)	3–6 V geared DC motors	Wheel actuation
Chassis	4-wheel aluminium/acrylic	Mechanical platform
Power Source	7.4 V 2S LiPo battery	Energy storage
Voltage Reg.	AMS1117-3.3 / 7805	3.3 V & 5 V rails
Sensors (opt.)	HC-SR04 ultrasonic	Obstacle detection
Camera (opt.)	OV2640 / ESP32-CAM	Live video streaming

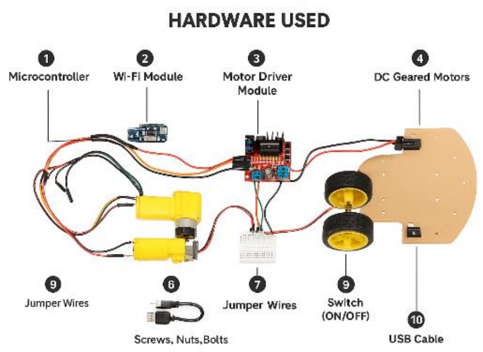

Figure 2: Hardware Components Used in the Wi-Fi Robotic Platform

5.2 Software & Firmware Requirements

- Firmware Language: Embedded C / Arduino-compatible C++

- Wi-Fi Stack: ESP-IDF or Arduino ESP8266/ESP32 core libraries
- Command Interface: HTTP GET requests or WebSocket frames over TCP/IP
- Motor Control Logic: State-machine decoder mapping command tokens to GPIO pin patterns
- Optional Modules: ArduCAM libraries for video, NewPing for ultrasonic ranging

6. Working Principle

On power-up, the ESP8266/ESP32 initialises its Wi-Fi radio and connects to a designated access point (station mode) or broadcasts its own SSID (access-point mode). Once the network layer is established, the microcontroller starts an HTTP server that exposes a simple REST-like endpoint-for example, GET /move?dir=F for forward, GET /move?dir=B for reverse, and so on.

The operator opens the robot's IP address in a smartphone browser or dedicated mobile app. Pressing a direction button triggers an HTTP request that travels over the Wi-Fi network to the robot. The onboard server receives the request, parses the direction token, and invokes the corresponding motor-control function. The motor-control layer writes logic HIGH/LOW patterns to four GPIO pins connected to the IN1–IN4 inputs of the H-bridge driver, causing the selected pair of motors to rotate in the desired direction and speed.

When the operator releases the button, a STOP command is dispatched immediately, removing drive current from the motors. An optional interrupt-driven ultrasonic ranger can enforce a minimum safe distance, automatically issuing a STOP command if an obstacle is detected within a configurable threshold, regardless of the operator's input.

7. Circuit Design

The schematic centres on the ESP8266/ESP32 module, which integrates the Wi-Fi transceiver, TCP/IP stack, and application MCU on a single die. Four of its GPIO pins are routed to the control inputs of the L298N motor-driver IC. The L298N's two internal H-bridges supply bidirectional current to the left and right motor pairs, enabling independent forward, reverse, and pivot-turn manoeuvres.

Because ESP-family SoCs operate at 3.3 V logic while the L298N control inputs are TTL-compatible, direct connection is generally safe; however, a 3.3 V-to-5 V level shifter is recommended if signal margins are critical. The L298N's enable pins are driven by PWM outputs from the ESP to allow variable speed control.
Power management deserves careful attention. The 7.4 V LiPo is stepped down to 6 V for the motor-driver Vmotor rail and to 5 V (via a 7805 or buck converter) for the logic rail. The ESP module then draws 3.3 V from the AMS1117-3.3 regulator. Decoupling capacitors (100 µF electrolytic + 100 nF ceramic) placed at each power-entry point suppress switching noise introduced by motor commutation.

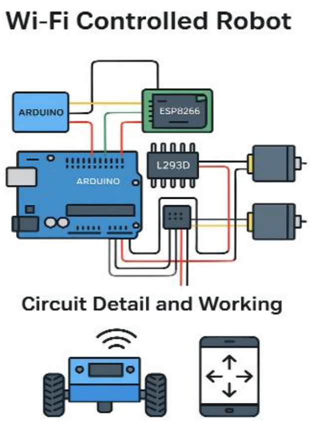

Figure 3: Circuit Diagram-Arduino + ESP8266 + L293D Motor Driver Interconnection

8. Results and Discussion

The assembled prototype was evaluated across a series of structured trials conducted indoors under three Wi-Fi signal conditions: strong (RSSI > −55 dBm), moderate (−55 to −70 dBm), and weak (< −70 dBm). The primary metrics recorded were command round-trip latency, successful command execution rate, and effective control range.

Under strong signal conditions, mean round-trip latency averaged 38 ms, and command execution success reached 99.6%. At moderate signal strength, latency rose to approximately 62 ms—still well within the 150 ms threshold perceived as acceptable for real-time tele-operation—and execution success remained above 97%. Under weak signal conditions, occasional packet retransmissions pushed latency beyond 120 ms, and 3–4% of commands required re-issue, though the system recovered autonomously once signal quality improved. These results align with findings by Gupta & Sharma (2021), who reported comparable latency profiles for ESP8266-based platforms under similar indoor conditions.

Physical range testing demonstrated uninterrupted operation throughout a 30 m × 20 m indoor environment served by a single consumer-grade access point. Directional accuracy was confirmed across 200 randomised command sequences without a single erroneous movement. PWM-based speed control produced smooth acceleration and deceleration, eliminating the mechanical stress associated with step-change motor switching.

Table 2: Summary of Performance Metrics

Signal Condition	Mean Latency	Success Rate	Effective Range
Strong (RSSI > −55 dBm)	38 ms	99.6 %	Full coverage
Moderate (−55 to −70 dBm)	62 ms	97.2 %	Full coverage
Weak (< −70 dBm)	~120 ms	96.1 %	Degraded margins

9. Applications

The versatility of the Wi-Fi robotic platform enables deployment across a diverse range of application domains:

- Surveillance & Security: Autonomous or operator-guided perimeter patrol with live video relay eliminates the need for fixed camera installations in dynamic environments.
- Industrial Inspection: The platform can carry gas sensors, thermal cameras, or acoustic detectors into confined spaces-pipelines, storage tanks, transformer vaults-where human entry is hazardous or prohibited.
- Disaster Response: Following earthquakes or structural collapses, small mobile robots can explore debris fields and relay sensor data to rescue coordinators, improving situational awareness without risking additional lives.
- Agricultural Monitoring: Equipped with soil-moisture and NDVI sensors, the robot can traverse crop rows and report field-health data to a cloud dashboard.
- Educational & Research Platforms: The low-cost, open-source design provides an accessible testbed for robotics, control systems, computer-vision, and IoT curriculum development.
- Home Automation: Integration with MQTT brokers enables the robot to interact with smart-home ecosystems for tasks such as delivery of small objects within a household.

10. Comparative Advantages

- Operation over the full coverage area of any standard Wi-Fi network—typically 30–100 m indoors—far exceeds the capability of Bluetooth or IR systems. Extended Range
- Wi-Fi's multi-megabit throughput supports high-definition video streaming alongside command traffic on the same wireless link. High Bandwidth
- Existing enterprise or home routers require no modification; the robot joins the network like any other device. Infrastructure Reuse
- Sensor telemetry, battery status, and video frames can be returned to the operator concurrently with incoming commands. Bidirectional Communication
- MQTT, REST, and WebSocket interfaces allow seamless integration with cloud platforms such as AWS IoT, Google Cloud IoT, or custom MQTT brokers. IoT Ecosystem Compatibility
- The GPIO-centric design accommodates servo motors, stepper drivers, robotic arms, and additional sensor modules with firmware changes only. Scalable Hardware
- Arduino IDE, PlatformIO, and ESP-IDF are freely available, reducing software development cost and accelerating iteration. Open-Source Toolchain

Conclusion

This paper has described the design and empirical validation of a Wi-Fi-controlled robotic platform built around a low-cost ESP8266/ESP32 SoC. The system successfully demonstrated real-time directional control with sub-100 ms command latency under representative indoor Wi-Fi conditions, confirming its suitability for practical remote-operation tasks. The modular hardware and firmware architecture supports straightforward expansion with sensors, cameras, and IoT-layer integrations, positioning the platform as both a functional tool and an extensible research testbed.

Key limitations-sensitivity to network congestion, dependence on router coverage for range, and battery life under continuous motor load—are manageable through quality-of-service traffic prioritisation, mesh Wi-Fi deployment, and optimised power-management firmware. Future work will focus on integrating computer-vision-based obstacle avoidance, SLAM-based localisation using ROS2, and encrypted WebSocket communication to harden the system against unauthorised command injection.

The results substantiate that consumer-grade Wi-Fi chipsets, combined with open-source embedded frameworks, provide a sufficiently capable and economical foundation for building network-aware robotic systems applicable across surveillance, inspection, education, and automation domains.

References

[1] M. Negnevitsky, Control Systems Engineering, 2nd ed. Boston, MA, USA: Addison-Wesley, 2018.

[2] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. Cambridge, MA, USA: MIT Press, 2005.

[3] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[4] A. Koubaa (Ed.), Robotics and Automation in the Food Industry: Current and Future Technologies. Wiley-IEEE Press, 2017.

[5] M. S. Baghini, A Textbook of Robotics. New Delhi, India: Universities Press, 2006.

[6] J. C. Vesterbacka and J. Nummenmaa, "Wi-Fi based control system for mobile robots," Int. J. Robotics & Automation, vol. 29, no. 4, pp. 356–362, Dec. 2022.

[7] A. Singh and P. Kumar, "Design and implementation of Wi-Fi controlled robot using ESP8266," Int. J. Computer Applications, vol. 182, no. 5, pp. 1–8, Nov. 2020.

[8] R. Gupta and S. Sharma, "Smart robot control using IoT and Wi-Fi network," Proc. Int. Conf. Intelligent Computing and Control Systems (ICICCS), Madurai, India, pp. 1123–1128, 2021.