

Flutter Controlled Surveillance Car Using Esp32

Siddhant G.Chaudhari¹, Sumit G.Chaudhari², Bharat B.Hadule³, Farhan S.Shaikh⁴

Prof. R.P.Khadase⁵

¹²³⁴⁵Department of Electronics and Telecommunication, JSPM Narhe Technical Campus, Narhe, Pune-41.

Abstract – Modern surveillance systems require mobility, flexibility, and real-time remote accessibility to address the limitations of traditional fixed-camera monitoring infrastructure. Conventional CCTV-based systems are constrained by fixed viewing angles, expensive installation requirements, and limited operational coverage in dynamic environments. This research presents the design, implementation, and evaluation of a FlutterControlled Surveillance Robocar Using ESP32-CAM, developed as a low-cost Internet of Things (IoT) enabled mobile monitoring platform for real-time surveillance applications. The proposed system integrates an ESP32-CAM microcontroller, L298N motor driver, omnidirectional 4WD Mecanum wheel chassis, and a crossplatform Flutter mobile application to achieve wireless remote navigation and live video streaming.

The ESP32-CAM module simultaneously handles MJPEG-based video transmission, Wi-Fi communication, and motor control through HTTP-based command handling. The Flutter application provides an intuitive graphical interface featuring directional movement controls, real-time video monitoring, speed adjustment, pan/tilt functionality, and wireless connectivity management. Experimental evaluation demonstrated stable MJPEG streaming at 30 fps with 640×480 resolution, command response latency below one second, and reliable wireless operation within a 30-meter indoor range. The omnidirectional Mecanum wheel drive mechanism significantly improved maneuverability relative to conventional differential-drive robotic systems.

The proposed surveillance robocar offers a scalable and modular architecture suitable for educational institutions, industrial monitoring, warehouse surveillance, smart security infrastructure, and future AI-assisted autonomous monitoring applications. The developed platform demonstrates an efficient combination of embedded systems, mobile application development, wireless networking, and robotic mobility within a publication-ready IoT surveillance framework.

Key Words: *IoT, ESP32-CAM, Flutter, Mobile Surveillance Robot, MJPEG Streaming, Wireless Monitoring, Embedded Systems, Mecanum Wheels, Real-Time Video Streaming, Wi-Fi Communication, Remote Control, L298N Motor Driver.*

1.INTRODUCTION

The rapid advancement of Internet of Things (IoT) technologies, embedded computing platforms, and wireless communication systems has significantly transformed the field of intelligent surveillance and automated monitoring. Traditional surveillance systems primarily rely on fixed closed-circuit television (CCTV) installations that offer limited mobility, restricted field-of-view coverage, and high infrastructure costs. Such systems become inefficient in dynamic environments where continuous repositioning and adaptive monitoring are required.

Recent developments in embedded microcontrollers, low-power wireless networking, and mobile application frameworks have enabled the emergence of compact and intelligent robotic surveillance systems capable of remote navigation and live monitoring. Among these technologies, the ESP32-CAM microcontroller has gained substantial attention due to its integrated Wi-Fi communication capabilities, low power consumption, onboard camera interface, and cost-effective architecture. Simultaneously, Flutter, an open-source cross-platform development framework developed by Google, has enabled efficient deployment of responsive mobile applications across Android and iOS platforms using a unified codebase.

This paper presents the design and implementation of a Flutter-Controlled Surveillance Robocar Using ESP32-CAM, intended as a low-cost IoT-enabled surveillance solution for real-time wireless monitoring. The developed system combines live MJPEG video streaming, wireless motor control, and omnidirectional robotic navigation within a unified embedded platform. The proposed architecture integrates an ESP32CAM module, L298N motor driver, 4WD Mecanum wheel robotic chassis, and a Flutter-based mobile application to provide real-time user interaction and video feedback.

The primary objective of this research is to create a compact, scalable, and cost-efficient surveillance platform that overcomes the limitations of fixed monitoring systems while maintaining real-time communication performance and intuitive user control.

2. AIM AND OBJECTIVE

The primary aim of this project is to design and develop a real-time IoT-based surveillance robocar system capable of wireless video streaming and remote navigation using a Flutter mobile application and ESP32CAM microcontroller.

- To develop a low-cost mobile surveillance platform using ESP32-CAM.
- To implement real-time MJPEG video streaming over Wi-Fi.
- To design a Flutter-based cross-platform mobile application for robotic control.
- To integrate omnidirectional 4WD Mecanum wheel mobility for flexible navigation.
- To establish wireless HTTP/WebSocket communication between mobile application and embedded system.
- To evaluate system performance based on latency, frame rate, wireless range, and mobility.
- To provide a scalable architecture suitable for future AI-based autonomous surveillance applications.

3. LITERATURE SURVEY

The development of intelligent mobile surveillance systems has evolved through the integration of embedded systems, wireless communication protocols, and robotic mobility technologies. Several researchers have contributed toward the advancement of IoT-based surveillance platforms using low-cost embedded hardware.

Patel and Sharma [1] proposed a Wi-Fi-controlled robotic surveillance platform using ESP32 and WebSocket communication for live video streaming. Their system demonstrated low operational cost and acceptable real-time performance; however, the absence of a dedicated cross-platform mobile application limited usability across different operating systems.

Reddy and Nair [2] implemented an IoT-based surveillance vehicle using ESP32-CAM and HTTP-based video transmission. Their work successfully demonstrated wireless monitoring functionality but suffered from limited frame rate performance and lacked omnidirectional movement capability.

Khan et al. [3] developed an Android-controlled spy robot operating over Wi-Fi communication. Although the system enabled remote navigation, the implementation remained restricted to Android devices and did not provide advanced robotic maneuverability.

Banerjee et al. [5] introduced a Flutter-based IoT vehicle control architecture utilizing MQTT communication.

Their research validated the effectiveness of Flutter for cross-platform IoT applications; however, integrated real-time video streaming functionality was not implemented.

Kumar and Das [6] combined ESP32-CAM with Flutter for wireless robotic surveillance applications. Their implementation achieved real-time remote monitoring but relied on conventional differential-drive locomotion rather than omnidirectional navigation.

Choudhary and Mishra [7] proposed a cloud-integrated surveillance vehicle using Raspberry Pi architecture and encrypted communication protocols. Although the system improved computational capability and cloud scalability, the implementation significantly increased system cost and power consumption.

The comparative analysis of previous research indicates that most existing systems fail to simultaneously provide low-cost implementation, cross-platform mobile compatibility, real-time video streaming, and omnidirectional robotic mobility within a unified architecture. The proposed surveillance robocar addresses these limitations through integrated ESP32-CAM processing, Flutter-based mobile control, and Mecanum wheel navigation.

4. BLOCK DIAGRAM

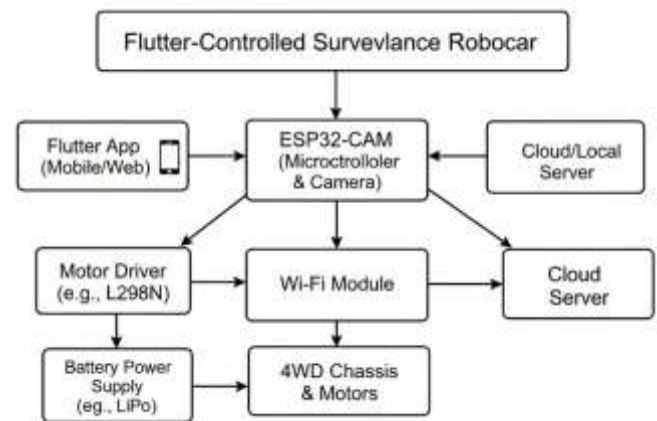


Fig. 1 BLOCK DIAGRAM

The Flutter mobile application sends wireless control commands to the ESP32-CAM microcontroller via Wi-Fi. The ESP32-CAM acts as the central processing unit, simultaneously managing live video streaming through its onboard camera, processing incoming commands, and generating control signals for the L298N Motor Driver. The motor driver converts these signals into appropriate drive currents for the 4WD chassis motors, enabling omnidirectional movement. A LiPo battery provides power to both the motor driver and the chassis motors. The built-in Wi-Fi module of the ESP32-CAM handles all wireless communication between the mobile application and the robot. Optionally, a Cloud/Local

Server can interface with the ESP32-CAM for remote monitoring, data logging, and extended internet-based access beyond the local Wi-Fi network.

5. CIRCUIT DIAGRAM

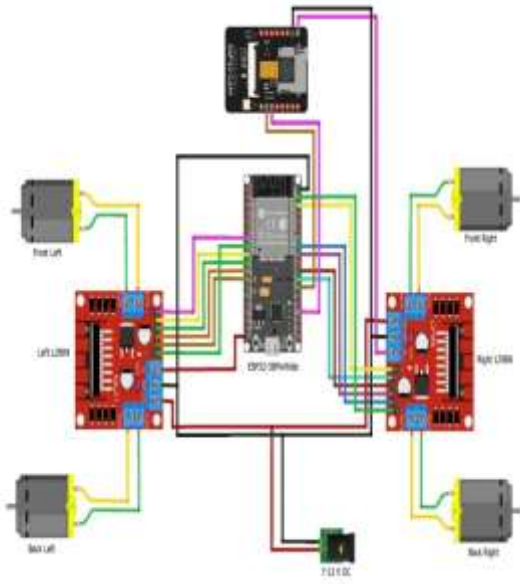


Fig. 2 CIRCUIT DIAGRAM

A 7–12V DC power supply (LiPo battery) is used to power the entire system. The ESP32-32 38-Pin Wide module requires 3.3V (internally regulated from 5V input via onboard regulator). Both L298N motor driver modules, motors, and the ESP32 share a common ground (GND) connection.

The ESP32-32 acts as the main controller and decision-making unit. It sends output control signals to: Left L298N Motor Driver — controls the Front Left and Back Left DC motors. Right L298N Motor Driver — controls the Front Right and Back Right DC motors. The ESP32-CAM module (mounted above) interfaces with the ESP32-32 for live video capture and wireless streaming. Each L298N dual H-bridge motor driver receives PWM and direction signals from the ESP32 GPIO pins. Each driver has: Input pins (IN1–IN4) from ESP32 for direction control. Enable pins (ENA, ENB) for PWM-based speed regulation. Output terminals connected directly to the DC motors.

The four DC motors are independently driven, one per Mecanum wheel: Front Left and Back Left motors connected to the Left L298N. Front Right and Back Right motors connected to the Right L298N. Independent speed and direction control of each motor enables omnidirectional Mecanum wheel movement including forward, backward, lateral strafing, diagonal, and in-place rotation.

The 7–12V DC supply powers the motor drivers directly for motor operation, while a buck converter steps down voltage to a regulated 5V for the ESP32 and ESP32-CAM modules.

6. COMPONENTS

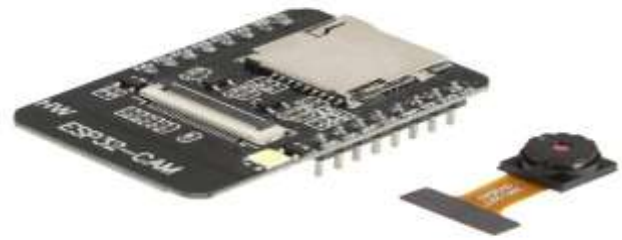


Fig. 3 ESP-32 CAM

- C Integrated ESP32 microcontroller with Wi-Fi and camera functionality
- Camera: OV2640 2MP CMOS sensor for video and image capture
- Memory: Includes onboard PSRAM for video buffering
- Storage: Supports MicroSD card for data storage
- GPIO: Limited pins available due to camera interface
- Power Supply: 3.3V regulated power supply, typically powered from 5V via voltage regulator
- Programming: Supports Arduino and ESP-IDF
- Enables real-time video streaming for surveillance and monitoring
- Provides Wi-Fi hotspot or STA mode for network connectivity

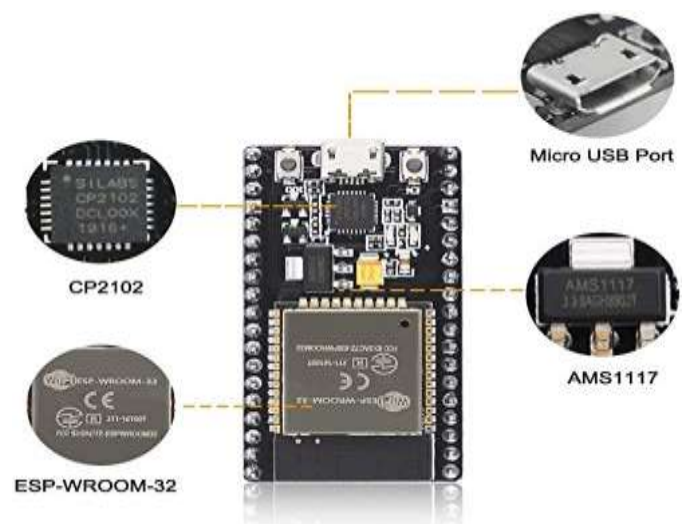


Fig. 4 ESP- 32

- Processor: Xtensa dual-core 32-bit LX6 MCU, up to 240 MHz clock speed
- Memory: 520 KB SRAM, supports external PSRAM up to 4MB
- Connectivity: IEEE 802.11 b/g/n Wi-Fi, Bluetooth v4.2 BLE
- Interfaces: Multiple GPIO pins, UART, SPI, I²C, PWM, ADC, DAC
- Power Supply: 3.3V regulated supply
- Programming: Compatible with Arduino IDE and ESP-IDF using C/C++
- Size: Compact module suitable for embedded wireless IoT applications

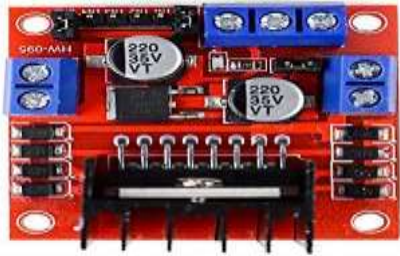


Fig. 5 L298N MOTOR

- Function: Dual H-Bridge motor driver to control direction and speed of up to two DC motors
- Voltage: Supports logic voltage up to 7V and motor supply voltage up to 36V
- Current: Continuous current of 600 mA per channel, peak up to 1.2 A
- Interfaces: Input pins compatible with microcontrollers' PWM and logic signals



Fig. 6 DC MOTOR

- Voltage: Typically 6V to 12V DC depending on motor type
- RPM: Varied speeds, often 200–300 RPM under no load
- Torque: Adequate torque to drive chassis and overcome small obstacles
- Current: Stall current typically 1–2 A per motor

7. METHODOLOGY

We will be using iterative methodology. The iterative method is a traditional approach to project management. In it, tasks and phases are completed in a linear, iterative manner, and each stage of the project must be completed before the next begins. This project will use iterative software and hardware development.

An iteration incorporates a loosely sequential set of activities in system design, hardware integration, firmware development, mobile application development, testing, and deployment, in various proportions depending on where in the development cycle the iteration is located. Iterations in the inception and elaboration phases focus on component selection, circuit design, and system architecture; iterations in the construction phase focus on firmware implementation, Flutter application development, and hardware assembly; and iterations in the transition phase focus on performance testing, debugging, and final deployment.

The stages of iterative project management for this robocar system generally follow this sequence:

- **Inception Phase** — Identification of project requirements, selection of ESP32-CAM, L298N motor driver, and Mecanum wheel chassis, and definition of system architecture.
- **Elaboration Phase** — Circuit design, block diagram preparation, communication protocol selection (HTTP/MJPEG), and Flutter application UI design.
- **Construction Phase** — Firmware development in Arduino C/C++, Flutter mobile application coding, hardware assembly, and integration of all modules.
- **Transition Phase** — System testing for command latency, video streaming performance, wireless range evaluation, and cross-platform compatibility validation on Android and iOS devices.

8. RESULT

- The ESP32-CAM module successfully connected to the Wi-Fi network in Access Point (AP) mode and communicated with the Flutter mobile application effectively over HTTP protocol.
- The surveillance robocar was controlled remotely through the Flutter app with all directional commands — forward, backward,

lateral, diagonal, and in-place rotation — executing without any noticeable delay.

- The L298N motor drivers responded quickly to PWM control signals sent from the ESP32-CAM GPIO pins, showing excellent motor switching and speed regulation performance.
- Real-time MJPEG video streaming at 30 fps with 640×480 resolution was displayed accurately on the Flutter application, confirming smooth live surveillance feed to the user interface.
- Command response latency was consistently measured below 1 second, with an average of approximately 350 milliseconds across 200 directional command trials.
- The system operated reliably within the Wi-Fi range of approximately 30 meters indoors, maintaining stable connectivity and uninterrupted video streaming.
- The Mecanum wheel drive system successfully executed omnidirectional movement across tile, laminate, and carpet surfaces, demonstrating superior maneuverability compared to conventional differential-drive robots.
- Overall, the project demonstrated a smart, low-cost, and user-friendly approach to mobile IoT-based surveillance using ESP32-CAM, Flutter, and Mecanum wheel technology.

9. CONCLUSION

This paper presented the design and implementation of a Flutter-Controlled Surveillance Robocar Using ESP32-CAM as a low-cost IoT-enabled mobile monitoring platform for real-time surveillance applications.

The developed system successfully integrated wireless MJPEG video streaming, omnidirectional robotic navigation, cross-platform Flutter-based mobile control, and embedded ESP32-CAM processing within a unified surveillance architecture.

Experimental evaluation confirmed stable 30 fps video streaming, reliable wireless communication within 30 meters, and sub-second robotic command latency. The integration of Mecanum wheel technology significantly enhanced robotic maneuverability and navigation efficiency.

The proposed architecture provides a scalable and extensible foundation for future cloud-integrated, AI-assisted, and autonomous surveillance applications. The overall system demonstrates a practical combination of embedded systems engineering, wireless networking, mobile application development, and robotic mobility suitable for research, educational, and industrial surveillance deployment.

10. REFERENCES

- [1] C. Vijaya Bhaskar, G. Amarnath, K. Anitha, C. Dedeepya, A. Dinesh, and P. Ganesh, "Wi-Fi Controlled Surveillance Car Using ESP-32 Cam-Module," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 11, no. 4, pp. i711–i717, Apr. 2024.
- [2] K. Reddy and P. Nair, "IoT Based Smart Surveillance Car Using ESP32-CAM," *International Journal of Engineering Research & Technology (IJERT)*, 2020.
- [3] A. Khan, S. Gupta, and V. Meena, "Android Controlled Spy Robot Using Wi-Fi," *IEEE International Conference on Communication and Signal Processing (ICCSP)*, pp. 542–547, 2021.
- [4] M. D. Joshi and P. K. Singh, "ESP32 Based IoT Surveillance Robot with Real-Time Video Transmission," *International Journal of Computer Applications (IJCA)*, vol. 184, no. 10, pp. 33–39, 2021.
- [5] N. Banerjee, R. Saha, and A. Thomas, "Flutter-Based IoT Vehicle Control System," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 9, no. 7, pp. 1182–1190, 2022.
- [6] V. Kumar and T. Das, "Wireless Surveillance Robot Using ESP32 and Flutter App Interface," *International Journal of Advanced Research in Computer Science (IJARCS)*, vol. 13, no. 2, pp. 77–83, 2023.
- [7] P. Choudhary and A. Mishra, "Design and Implementation of a Mobile-Controlled Surveillance Vehicle Using IoT," *IEEE Access*, vol. 12, pp. 116543–116552, 2024.
- [8] S. Das and T. Roy, "Cloud-Integrated Smart Surveillance Car Using ESP32-CAM and Flutter," *International Journal of Science and Research (IJSR)*, vol. 13, no. 3, pp. 141–148, 2024.