International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Surveillance Drone Using ESP32 and Raspberry Pi 3

Atharva Karambe[1], Sanjana Rani Sinha[2], Aryan Paul[3], Ananya Agrawal[4], Abhishek Kumar Dewangan[5]*

Department of Computer Science & Engineering (AIML), Shri Shankaracharya Technical Campus (CSVTU), Junwani, Chhattisgarh, India

*[1]E-mail: atharvakarambe262@gmail.com, [2]sanjanaranisinha@gmail.com, [3] www.aryanpaul858472@gmail.com, [4]aagrawal1104@gmail.com, [5] abhishek.dewangan@sstc.ac.in

ABSTRACT

This paper presents the comprehensive design, development, and implementation of a hybrid surveillance drone system that integrates an ESP32-based flight controller and a Raspberry Pi 3 functioning as a mobile cyber reconnaissance unit. The drone can execute both physical aerial operations and digital network surveillance, making it suitable for security-critical missions. The ESP32 microcontroller manages the drone's flight stability through PID control algorithms, interprets sensor data in real-time, and controls motor thrust via ESCs and PWM signals. Simultaneously, the Raspberry Pi 3 runs Kali Linux and executes Python-based cybersecurity scripts, including Wi-Fi scanning and webcam surveillance. Using Tail scale, a secure mesh VPN, the drone achieves global connectivity and remote access. The integration of embedded systems, open-source tools, and real-time video streaming demonstrates a scalable, cost-effective UAV capable of dual-purpose surveillance—both physical and digital.

1. Introduction

Unmanned Aerial Vehicles (UAVs) have transformed surveillance operations by offering high mobility, rapid deployment, and extended reach. Traditionally, drones have been used for photography, logistics, mapping, and environmental monitoring. However, the potential to merge aerial surveillance with cybersecurity functions remains underexplored. This project aims to bridge that gap by creating a hybrid drone system. The flight component is driven by the ESP32 microcontroller, chosen for its real-time processing power, PWM control, and sensor interfacing capabilities. Parallel to flight operations, the Raspberry Pi 3 embedded onboard runs Kali Linux—a penetration testing OS—to perform reconnaissance on nearby wireless networks. The resulting drone becomes a compact, mobile surveillance platform capable of scanning both physical and digital environments. Such functionality can be critical in operations such as counter-terrorism, border patrol, and network security audits in inaccessible locations.

2. Methodology

The drone is architected around a dual-module system to separate flight operations from cybersecurity functionalities:

A. Flight Control Module

- **Microcontroller:** The ESP32 is programmed in Embedded C and C++ using the Arduino IDE. It communicates with the MPU6050 IMU sensor using I²C to fetch real-time gyroscope and accelerometer data.
- **Control Logic:** PID algorithms are implemented to maintain flight stability. The system checks orientation and adjusts motor speeds approximately 250 times per second, allowing for responsive control over pitch, roll, and yaw.
- **Power Distribution:** ESCs connected to each brushless motor are controlled via PWM signals from the ESP32.

B. Surveillance Module

- **Computer Board:** The Raspberry Pi 3 runs a full version of Kali Linux, which includes built-in network reconnaissance tools such as Nmap, Wireshark, and Aircrack-ng.
- **Remote Access:** Using Tailscale, a mesh VPN, the Raspberry Pi can be accessed securely from any global location.
- **Webcam Integration:** A USB or PiCam module streams live footage to a remote server or terminal using Python-based scripts, enabling visual monitoring of the drone's surroundings.

Together, these modules allow simultaneous flight and surveillance capabilities, effectively creating a dual-purpose drone.

3. Literature Review

In recent years, drones have evolved beyond conventional aerial photography into tools capable of performing advanced surveillance, environmental monitoring, and cybersecurity reconnaissance. Numerous open-source drone frameworks, including ArduPilot and PX4, have contributed significantly to drone autonomy. However, most of these systems rely on high-end microcontrollers or expensive flight controllers and lack built-in support for cybersecurity tasks.

Previous research has primarily focused on drones with visual capabilities for crowd monitoring, disaster assessment, or agricultural data gathering. In contrast, only limited studies have examined UAVs as platforms for real-time cyber reconnaissance, especially in low-cost, embedded environments.

Our proposed system bridges this technological gap by integrating a dual-microcontroller architecture—leveraging the ESP32 for flight dynamics and Raspberry Pi 3 for network analysis. The ESP32 manages real-time control using PID algorithms, ensuring stable flight and sensor data integration. Meanwhile, the Raspberry Pi runs Kali Linux, allowing the drone to act as a mobile cyber toolkit capable of Wi-Fi sniffing, port scanning, and device fingerprinting. This dual-purpose approach enables rapid deployment in restricted or high-threat zones, making the system a practical tool for modern digital surveillance.

4. Components and Specifications

The following hardware and software components were selected based on performance, cost-efficiency, and compatibility:

- **ESP32 Microcontroller**: Dual-core processor with Wi-Fi/Bluetooth for real-time flight control and telemetry.
- **Raspberry Pi 3**: Runs Kali Linux and Python scripts for cybersecurity tasks.
- **ESCs with Brushless Motors (Sky fly 30A)**: High-efficiency propulsion for stable and responsive flight.
- **3500mAh 11.1V LiPo Battery**: High discharge capacity to power motors and computing units.
- **HC-05 Bluetooth Module**: Enables basic remote commands or tuning via mobile apps.
- **ESP8266 Wi-Fi Module**: Facilitates network-based control or data logging (used optionally).
- **2.4GHz Radio Transmitter**: Primary method of manual drone control during testing and operation.
- **Webcam**: Mounted on the drone, interfaced with the Raspberry Pi for live video feed and environment monitoring.
- **Software Stack**:
    - *Arduino IDE* for ESP32 programming
    - *Embedded C/C++* for flight logic
    - *Python 3* for scripting network tools and webcam streaming
    - *Tail scale VPN* for encrypted remote access
    - *Kali Linux* for executing penetration testing and surveillance utilities

5. System Architecture

The architecture of the proposed surveillance drone is logically divided into two major subsystems:

1. **Flight Control Subsystem**

   This subsystem is based on the ESP32 microcontroller, chosen for its dual-core processing and real-time capabilities. It interfaces with the MPU6050 sensor to read gyroscope and accelerometer values at a frequency of 250 Hz. This data is used in a PID control loop to compute motor outputs via PWM signals. ESCs (Electronic Speed Controllers) then modulate power delivery to brushless motors for precise motion control. The system also supports input from a radio-controlled transmitter, allowing manual override and directional commands.

2. Surveillance Subsystem

The Raspberry Pi 3 runs Kali Linux OS and hosts several cybersecurity tools (e.g., Nmap, Wireshark, Aircrack-ng). It connects to the drone’s on-board webcam for live video streaming and utilizes Tail scale to enable secure remote access over the internet. Python scripts handle automation tasks such as scanning local networks or capturing traffic logs. Power is supplied via a dedicated battery connection with voltage regulation to ensure stable operation.

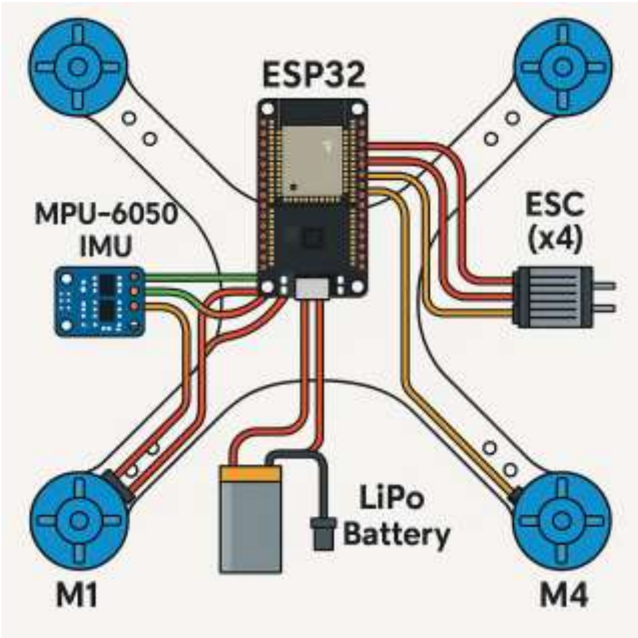
The integration between these two subsystems enables real-time coordination, with the ESP32 providing flight data to the Raspberry Pi and vice versa, supporting mission-driven surveillance operations.

6. Implementation

The implementation process began with the structural assembly of the drone frame, followed by the integration of the ESP32 flight controller and Raspberry Pi surveillance unit.

- **Flight Controller Setup**: The ESP32 was programmed using the Arduino IDE with integrated PID logic. Motor calibration and sensor tuning were performed during tethered tests.
- **Sensor Communication**: The MPU6050's gyroscopic and acceleration data were processed to calculate orientation angles, which the PID controller used to maintain stability.
- **PWM Signal Control**: The ESP32 issued PWM signals to ESCs, dynamically adjusting motor speeds to maintain balance and execute user commands.
- **Surveillance Setup**: Kali Linux was pre-installed on the Raspberry Pi 3. Python scripts were used to activate the webcam and scan for local Wi-Fi networks. This data was either stored locally or streamed via Tail scale.
- **Integration and Synchronization**: Both systems were powered by the same battery but operated independently, minimizing processing conflicts and improving real-time responsiveness.

Wiring Diagram –



7. Results and Analysis

- **Flight Performance**: The drone achieved a stable hover within 30 seconds of startup. PID tuning successfully reduced oscillation and improved responsiveness. The drone could manoeuvre on command and hold its position under minor wind conditions.
- **Cyber Surveillance**: The Raspberry Pi detected unsecured Wi-Fi networks and listed connected devices using Nmap and Air crack-ng. Live webcam streaming worked effectively over Tail scale with minimal latency within a 10-meter radius.
- **Combined Operation**: Both modules ran simultaneously without interference. This demonstrates that with proper power distribution and resource isolation, a multi-function drone can operate both physical and digital surveillance tools.
- **Challenges**:
  - Minor electromagnetic interference was observed near the ESCs.
  - The webcam resolution dropped during peak CPU load on the Pi.
  - Battery voltage fluctuated under heavy load, suggesting the need for a voltage regulator or dual-battery design.

8. Future Enhancements

There are multiple pathways to expand this system's capabilities:

- **GPS Integration:** For autonomous navigation and geo-fencing.
- **Barometric Altimeter:** To enhance altitude accuracy and auto-hover reliability.
- **Machine Learning Modules:** For real-time object detection and pattern recognition using onboard cameras.
- **Long-Range Communication:** By adding LoRa modules for extended-range telemetry.
- **Battery Management System (BMS):** To monitor and report battery health in-flight.
- **Solar Power Add-ons:** For prolonged operation in remote regions.

These improvements can evolve the drone from a semi-autonomous system into a fully intelligent UAV capable of real-time decision-making.

9. Conclusion and Future Scope

This research proves the viability of building a multi-role drone using low-cost, widely available components. By integrating a real-time flight controller (ESP32) and a fully functional minicomputer (Raspberry Pi 3), the system combines mobility and cyber surveillance.

Future enhancements include:

- GPS-based waypoint navigation for autonomous patrolling.
- AI-based face/object recognition using OpenCV and TensorFlow.
- Automatic return-to-home features on signal loss.
- Encrypted data transmission using SSH or MQTT protocols.
- Integration with cloud dashboards for live analytics.

Such systems could be revolutionary in law enforcement, disaster recovery, industrial inspection, and ethical hacking.

References

Espressif Systems, “ESP32 Series Datasheet,” 2022. [Online]. Available: https://www.espressif.com/en/products/socs/esp32

Raspberry Pi Foundation, “Raspberry Pi 3 Model B+ Documentation,” 2021. [Online]. Available: https://www.raspberrypi.com/documentation/

InvenSense, “MPU6050 Datasheet,” TDK Electronics.

Tailscale Inc., “Tailscale VPN Documentation,” [Online]. Available: https://tailscale.com/kb/

Offensive Security, “Kali Linux Tools Documentation,” [Online]. Available: https://www.kali.org/tools/