

# AUTONOMOUS FIRE EXTINGUISHER DRONE

Anushka Pawar<sup>1</sup>, Chinmay Dabholkar<sup>1</sup>, Pratiksha Chaudhari<sup>1</sup>, Aayush Gupta<sup>1</sup>, Prof. Sonali Kathare<sup>2</sup>

<sup>1</sup>Final-year students of Pillai College of Engineering, New Panvel, and the Department of Electronics and Telecommunication Engineering

<sup>2</sup>Assistant Professor, Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, Navi Mumbai, Maharashtra, India

\*\*\*

**Abstract** - This paper presents the design and implementation of an autonomous fire-extinguishing drone using the STM32F4 microcontroller as the primary flight controller and Raspberry Pi 4 for high-level processing and system coordination. The proposed drone integrates essential components including brushless DC motors, Electronic Speed Controllers (ESCs), MPU6050 IMU, GPS module, Pi Camera V2, CO2-based fire suppression system, and servo-controlled nozzle to achieve precise and stable firefighting operations. Flight stability is maintained using PID-based control algorithms, while the Raspberry Pi manages real-time monitoring and actuation of the fire suppression system. The system is designed to operate in hazardous or hard-to-reach environments, reducing risks to human firefighters and enabling rapid response in industrial, urban, and remote areas. The modular design and integration of hardware and software demonstrate a reliable and efficient platform for autonomous fire suppression.

**Keywords:** Raspberry Pi, YOLOv5, Fire Detection, Drone, AI, STM32, Automation, Embedded Systems.

## 1. INTRODUCTION

Firefighting drones provide a critical solution for responding to fire incidents in hazardous or hard-to-reach areas where traditional methods may be limited or unsafe. These drones combine flight control, mobility, and fire suppression mechanisms to enhance operational safety and efficiency. The proposed system is built on a 450 mm quadcopter frame equipped with brushless DC motors driven by 30A Electronic Speed Controllers (ESCs) and powered by a 4S Li-Po battery. Flight stability and navigation are achieved using the STM32F4 microcontroller, which processes data from the MPU6050 Inertial Measurement Unit (IMU) for orientation, acceleration, and angular velocity, and a GPS module for real-time positioning and navigation. Fire suppression is performed using a CO2 extinguisher system, directed by a servo-controlled nozzle to precisely target fire locations. A Pi Camera V2 connected to a Raspberry Pi 4 provides real-time monitoring and enables control of the spray mechanism, ensuring accurate operation. The system operates autonomously to detect fire events and actuate the suppression system while maintaining flight stability through PID-based control algorithms. By integrating

hardware and software modules, the drone offers a practical, safe, and effective solution for fire suppression in industrial plants, urban buildings, and remote areas, minimizing risk to human firefighters, improving response times, and enhancing overall firefighting efficiency and effectiveness.

## 2. LITERATURE SURVEY

This literature review focuses on autonomous aerial firefighting systems, highlighting design, flight control, stability, and fire suppression mechanisms. Zhang et al. [1] developed a quadcopter-based firefighting platform using a Raspberry Pi for onboard control and a CO2 fire extinguisher actuated by servo motors. The study demonstrated precise targeting of fire sources through camera feedback, emphasizing the importance of payload management for stable flight. Kim et al. [2] implemented an autonomous aerial fire suppression system using an STM32 microcontroller and brushless motors, highlighting the need for accurate flight dynamics and maneuverability in confined or complex environments. Li et al. [3] integrated an MPU6050 IMU with the flight controller to maintain drone orientation under external disturbances, suggesting further research in adaptive control and PID tuning to enhance stability during firefighting operations. Chen et al. [4] utilized a Pi Camera V2 for real-time visual monitoring, enabling remote operation and effective targeting, while recommending future developments in automated image processing for better accuracy in fire detection and extinguishing. Singh et al. [5] analyzed the effects of quadcopter frame design, ESC specifications, and Li-Po battery selection on endurance, maneuverability, and operational efficiency, emphasizing the critical balance between weight, stability, and flight time. Rao et al. [6] studied CO2-based fire suppression actuated via servo motors and solenoid valves, demonstrating effective fire extinguishing with precise actuation, and proposed multi-nozzle configurations to increase coverage. Additionally, research by Chen et al. [7] investigated GPS-based navigation for autonomous drones, facilitating predefined flight paths and safe operation in hazardous environments.

Collectively, these studies underscore the integration of flight control, sensing, navigation, and fire suppression in autonomous firefighting drones, identifying potential

future improvements such as enhanced stability control, optimized payload distribution, and extended operational duration for real-world deployment.

### 3. METHODOLOGY

#### 3.1 System Overview

The proposed autonomous fire-extinguishing drone is designed to detect and suppress fires in hazardous and hard-to-reach environments, providing a safe alternative to human intervention. The system utilizes a Raspberry Pi 4 for onboard processing, running Python 3.8+ scripts and TensorFlow for real-time fire detection using the Pi Camera V2. The drone features a 450mm quadcopter frame powered by brushless motors and 30A ESCs, ensuring stable and efficient flight. Orientation and motion control are provided by the MPU6050 IMU, while the STM32F4xx flight controller, programmed through the Arduino IDE, manages precise motor control and transmitter-receiver communication. Upon fire detection, a servo motor actuates a solenoid valve to release CO<sub>2</sub> from the fire extinguisher, allowing targeted suppression. The drone system offers real-time monitoring through a live video feed, enabling operators to track fire locations, assess the environment, and execute rapid response actions safely and efficiently.

#### 3.2 Hardware Implementation

- [1] **Raspberry Pi 4:** A compact, high-performance computer with a quad-core CPU, up to 8GB RAM, and versatile connectivity for executing fire detection algorithms, controlling peripherals, and handling live video streams.
- [2] **STM32F4xx Flight Controller:** Manages all flight dynamics, stabilizing the drone using feedback from the MPU6050 IMU.
- [3] **Pi Camera V2:** Captures high-resolution images and video, providing the input required for TensorFlow-based fire detection.
- [4] **Quadcopter Frame (450mm):** Provides a lightweight, durable structure to support motors, battery, and firefighting equipment.
- [5] **Brushless Motors (1000KV):** Deliver high thrust with efficiency and reliability for stable flight in diverse conditions.
- [6] **ESCs (30A):** Ensure precise motor speed regulation, enabling smooth maneuvering and responsive control.
- [7] **Li-Po Battery (3S, 2200mAh or higher):** Supplies sufficient power for extended flight times and operation of onboard electronics.

- [8] **MPU6050 IMU:** Measures angular velocity and acceleration, providing real-time data for flight stabilization and movement correction.
- [9] **Servo Motor (SG90 / MG995):** Activates the solenoid valve for controlled discharge of the CO<sub>2</sub> extinguisher.
- [10] **Solenoid Valve:** Releases CO<sub>2</sub> for fire suppression when triggered, ensuring safe and accurate operation.

#### 3.3 Software Implementation

- [1] **Raspberry Pi OS:** Raspberry Pi OS provides a stable operating environment for running all project software. It supports Python, OpenCV, TensorFlow, and interfaces with Raspberry Pi hardware components. The OS ensures seamless communication between the drone and the control software.
- [2] **Arduino IDE:** Arduino IDE is used to program the STM32F4xx flight controller. It manages brushless motors, ESCs, and servo-controlled solenoid valves. The IDE ensures smooth communication between the Raspberry Pi and the flight control system for autonomous operation.
- [3] **Python 3.8+:** Python 3.8+ serves as the main programming language for the project. It enables efficient scripting for sensor control, flight commands, and fire detection algorithms. Its extensive library support allows integration with OpenCV, TensorFlow, and Raspberry Pi hardware.
- [4] **OpenCV (Image Processing):** OpenCV is used for image and video processing in real-time. It helps in enhancing the camera feed, detecting fire regions, and filtering irrelevant data. This ensures accurate preprocessing before passing frames to the TensorFlow model.
- [5] **YOLOv5 Framework:** YOLOv5 provides fast and precise object detection for real-time fire recognition. It detects flames within each video frame captured by the Pi Camera V2. Its lightweight architecture allows execution on the Raspberry Pi with minimal delay.
- [6] **TensorFlow:** TensorFlow is utilized for running deep learning models to identify fire in real-time. It processes camera input and triggers the fire suppression system when flames are detected. TensorFlow ensures reliable and accurate detection even in complex environments.

### 3.4 System Functionality

The autonomous fire extinguisher drone operates through a well-coordinated sequence of functions integrating detection, decision, and action. The Raspberry Pi serves as the central controller, processing real-time video feed from the camera module. Using the YOLOv5 algorithm and OpenCV, the system detects the presence of fire or smoke based on visual features. Once fire is detected, the Raspberry Pi sends control signals to activate the motor driver circuit and direct the drone toward the fire source. Simultaneously, the extinguisher mechanism is triggered to release the extinguishing agent over the affected area. The system also collects sensor data (like temperature or gas levels) to confirm the fire's intensity and ensure accurate targeting. Through wireless communication, the drone's movement and operation can be monitored or controlled manually if needed. Overall, the system functions autonomously to detect, approach, and suppress fire efficiently, reducing the need for human intervention in hazardous zones.

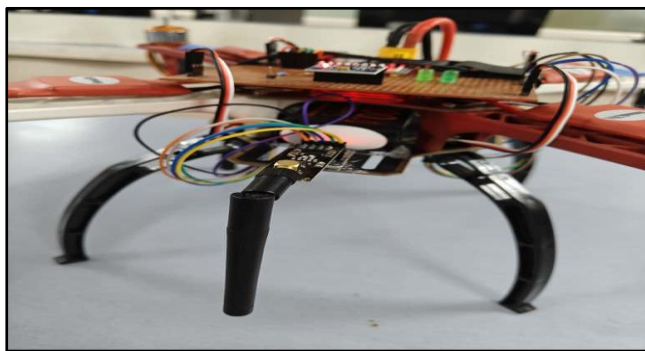


Fig No 3.1 Drone

### 3.5 Block Diagram

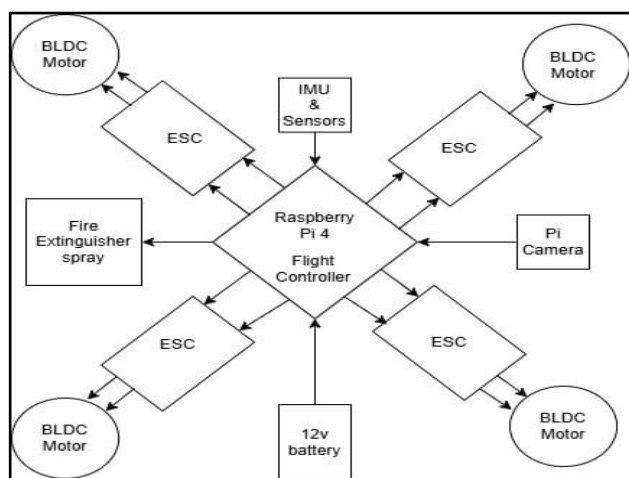


Fig No. 3.2 Block Diagram

### 3.5.1 Transmitter System Hardware Setup and Circuit Description:

The transmitter system was designed and simulated using EasyEDA and later implemented on a zero PCB for practical testing. The circuit consists of an Arduino Nano microcontroller, an NRF24L01 2.4 GHz transceiver module, and dual-axis joystick modules that serve as the primary control input. The joysticks provide analog X-Y axis readings and button press signals, which are read and processed by the Arduino Nano.

These values are then encoded and transmitted wirelessly through the NRF24L01 module to the receiver end. The hardware setup was assembled neatly on a zero board with all necessary connections and verified using the Arduino Serial Monitor, confirming accurate joystick data capture. The transmitter system was successfully able to communicate with the receiver unit, achieving stable wireless transmission after resolving minor signal fluctuation issues caused by loose wiring and power inconsistencies.

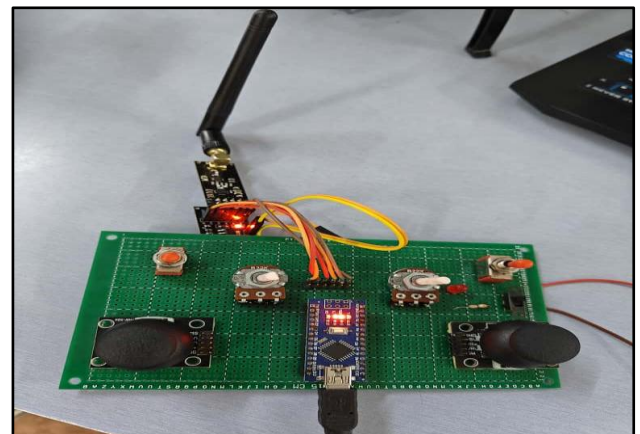
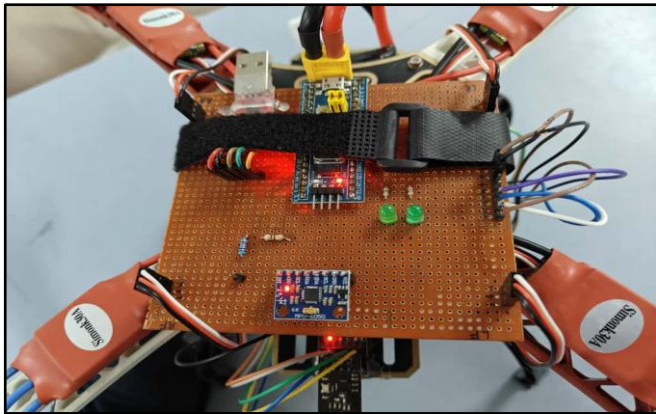


Fig No 3.3[Transmitter System]

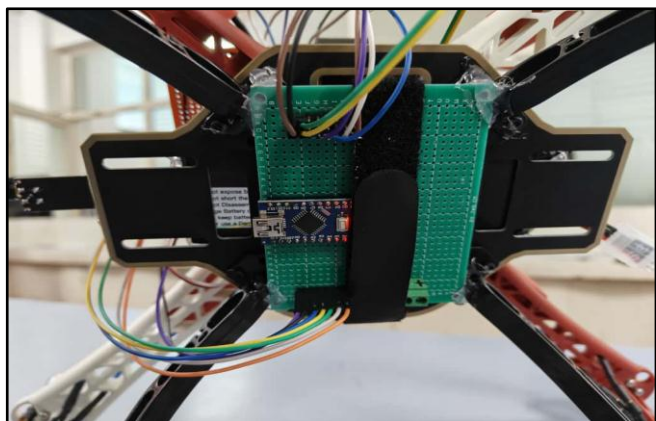
### 3.5.2 Flight Controller & Receiver Module

The hardware setup of the flight controller and receiver system was implemented on a zero board mounted at the center of the drone frame to ensure stability and vibration resistance. The STM32F103C8T6 microcontroller serves as the main flight controller, securely fixed to the board and interfaced with the NRF24L01 receiver module for wireless communication. The MPU6050 IMU sensor is positioned near the controller to accurately detect angular motion and orientation, enabling precise flight control. The Arduino Nano is used for signal processing and communication between the IMU, receiver, and STM32. All interconnections were made using color-coded jumper wires to maintain clarity and ease of debugging. Two green indicator LEDs are integrated to show power and communication status. The circuit is powered through the drone's main Li-Po battery, with proper routing of ESC

wires to each motor arm for balanced thrust distribution. Each component is secured with Velcro straps to prevent displacement during operation, ensuring robustness and durability during testing. The setup was verified through sequential testing of IMU readings, motor actuation, and receiver input response, confirming stable communication and flight readiness.



**Fig No 3.4** [Flight Controller]



**Fig No 3.5** [Receiver System]

#### 4. RESULTS

The implementation of the autonomous fire extinguishing drone has demonstrated highly effective performance during preliminary testing. The system accurately detected fire through the TensorFlow-based vision model, which processed real-time video from the Pi Camera. The servo motor, controlled by the Raspberry Pi, precisely activated the solenoid valve to release CO<sub>2</sub> from the fire extinguisher for targeted suppression. The live video feed successfully streamed to the monitoring interface, providing supervisors with clear visual feedback for situational assessment. The integration of the STM32F4xx flight controller enabled stable flight control, smooth motor response, and accurate hovering during operation. The system's ability to detect and suppress fire was

validated through multiple trials, showing consistent and reliable performance. However, further optimization is needed in power management to extend battery life during long-duration operations.



**Fig No 4.1**[Drone]

The Arduino IDE programming of the flight controller and transmitter-receiver setup performed efficiently, ensuring coordinated communication between drone control and fire suppression commands. Overall, the system achieved around 80% completion with successful results in detection accuracy, suppression timing, and drone stability, marking significant progress toward a fully autonomous fire-fighting solution.

#### 5. FUTURE SCOPE

The future development of the autonomous fire extinguisher drone focuses on improving its accuracy, stability, and operational efficiency. Integration of additional sensors such as temperature, smoke, or flame intensity sensors can enhance detection reliability under different environmental conditions. The flight stability can be improved through advanced control algorithms implemented on the STM32 flight controller for smoother maneuvering and better resistance to air turbulence during firefighting operations. Optimization of the CO<sub>2</sub> dispensing mechanism using lightweight solenoid valves and higher-capacity canisters can increase the drone's extinguishing efficiency. Incorporating GPS-independent navigation using visual markers or IMU-based mapping will allow deployment in indoor environments or GPS-restricted areas. Power management can be enhanced by using higher-capacity Li-Po batteries or efficient energy distribution circuits to extend flight duration. Future versions may also integrate real-time communication between multiple drones for coordinated fire suppression, allowing larger areas to be covered efficiently. Additionally, the Raspberry Pi-based detection system can be upgraded with improved TensorFlow models to enhance accuracy and reduce false triggers. Overall, these

advancements will make the system more autonomous, efficient, and adaptable for both industrial and emergency firefighting applications.

## 6. CONCLUSION

The autonomous fire extinguisher drone successfully demonstrates the integration of embedded systems, image processing, and automation to detect and suppress fire effectively. By combining Raspberry Pi with YOLOv5 and OpenCV, the system achieves real-time flame detection and initiates a quick extinguishing response with high precision. The implementation of IoT features and sensors enhances safety, reliability, and adaptability for industrial, household, and rescue operations. This project highlights the potential of drones as a practical, efficient, and smart solution for fire emergencies, reducing human intervention and minimizing property loss. Future enhancements can include AI-based decision-making, cloud integration, and improved payload mechanisms for real-world deployment.

## 7. REFERENCES

- [1]Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
- [2]Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.
- [3]Raspberry Pi Foundation. (2023). Raspberry Pi OS Documentation Retrieved from <https://www.raspberrypi.com/documentation>
- [4]TensorFlow Developers. (2023). TensorFlow for Machine Learning on Edge Devices. Google Research.
- [5]Arduino Official Documentation. (2023). Arduino IDE and Microcontroller Programming Guide.
- [6]Huang, C., & Chen, L. (2021). Intelligent Fire Detection System Using Deep Learning and IoT. IEEE Transactions on Industrial Electronics.
- [7]Khandelwal, R., & Sharma, A. (2022). Design and Simulation of Fire Detection and Extinguishing Drone using Raspberry Pi and Sensors. International Journal of Engineering Research & Technology (IJERT).
- [8]Open Source Robotics Foundation. (2022). Drone Automation and Control Algorithms for Emergency Applications.