ESIoT MINI PROJECT REPORT

On

IoT-Based Real-Time Fire Detection and Alert System Using ESP32 with Blynk Integration and Google Sheets Logging

Submitted in partial fulfilment for the completion of

BE-VI Semester

in

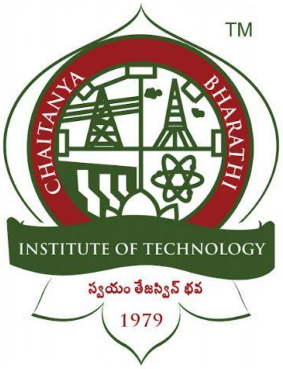
INFORMATION TECHNOLOGY

By

GUNTI SAMHITHA (160122737147)
SUDHA JENNIFER M (160122737150)

Under the guidance of

E. RAMALAKSHMI
Assistant Professor,
Dept. of IT



DEPARTMENT OF INFORMATION TECHNOLOGY

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
(Autonomous)
(Affiliated to Osmania University; Accredited by NBA(AICTE) and NAAC(UGC), ISO Certified 9001:2015)
GANDIPET, HYDERABAD – 500 075
Website: www.cbit.ac.in

2024-2025

CERTIFICATE

This is to certify that the project work entitled "**IoT-Based Real-Time Fire Detection and Alert System Using ESP32 with Blynk Integration and Google Sheets Logging**" submitted to **CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY,** in partial fulfilment of the requirements for the award of the completion of ESIoT Mini Project VI semester of B.E in Information Technology, during the academic year 2024-2025, is a record of original work done by **GUNTI SAMHITHA (160122737147),SUDHA JENNIFER M (160122737150)** during the period of study in Department of IT, CBIT, HYDERABAD, under our supervision and guidance.

Project Guide
E. RAMALAKSHMI
Asst. Professor, Dept. of IT,
CBIT, Hyderabad.

Head of the Department
Dr.M. Venu Gopalachari
Professor, Dept. of IT,
CBIT, Hyderabad.

ABSTRACT

This project presents a real-time fire detection and alert system based on the ESP32 microcontroller, designed to enhance fire safety in residential or industrial environments. The system utilizes an infrared (IR) flame sensor to detect the presence of fire. Upon detection, it activates an audible buzzer and visual LED alarm to notify people nearby. Simultaneously, it leverages the Blynk IoT platform to send instant alerts to a smartphone, providing remote monitoring capabilities.

In addition to real-time notifications via Blynk, the system also logs fire incidents to a Google Sheet through a webhook configured with Google Apps Script. This enables centralized and persistent record-keeping of critical fire events. To prevent redundancy, data logging is triggered only once per fire detection event, avoiding duplicate entries.

The ESP32's built-in Wi-Fi capability makes the system compact and ideal for wireless deployment. This project is a low-cost, scalable solution to improve early fire warning systems, making it particularly suitable for smart home, educational, and industrial safety applications.

TABLE OF CONTENTS

LIST OF FIGURES AND SCREENSHOTS:

LIST OF TABLES:

ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to **E. RAMALAKSHMI** our project guide, for her invaluable guidance and constant support, along with her capable instruction and persistent encouragement.

We are grateful to our Head of Department, **Dr. M. Venugopala Chari** for his steady support and for the provision of every resource required for the completion of this project.

We would like to take this opportunity to thank our principal, Prof. C.V. Narasimhulu, as well as the Management of the Institute, for having designed an excellent learning atmosphere.

Our thanks are due to all members of the staff and our lab assistants for providing us with the help required to carry out the groundwork of this project.

1. INTRODUCTION

Fire accidents pose a significant threat to life and property across homes, industries, and public spaces. Traditional fire detection systems, although effective, often lack remote monitoring capabilities and real-time notifications, limiting their usefulness in fast-response scenarios. In the age of the Internet of Things (IoT), smarter and more connected systems can be developed to enhance safety measures and reduce the risk of fire-related damage.

This project introduces an IoT-based real-time fire detection and alert system using the ESP32 microcontroller, integrated with the Blynk IoT platform and Google Sheets for cloud-based data logging. The system uses an infrared (IR) flame sensor to continuously monitor the environment for signs of fire. When a flame is detected, it triggers both audible and visual alerts through a buzzer and an LED, respectively. Simultaneously, the system sends instant notifications to the user's smartphone via the Blynk app, ensuring that users are promptly alerted even if they are not physically present at the location.

To support long-term analysis and auditing of fire incidents, the system also logs fire detection events into a Google Sheet using a webhook created through Google Apps Script. This log is maintained only when an actual fire is detected, minimizing unnecessary data and improving system efficiency.

The choice of ESP32, with its built-in Wi-Fi capability, ensures the system remains compact and cost-effective while offering flexibility in deployment. This project demonstrates how IoT can be leveraged to create a smart, responsive, and scalable fire alert system suitable for homes, schools, offices, and industrial setups.

2. PROJECT DESCRIPTION

2.1 System Overview

The IoT-Based Fire Alert System is a smart, real-time fire detection solution that leverages the capabilities of the ESP32 microcontroller, infrared flame sensor, Blynk IoT platform, and Google Sheets integration. The core objective of the system is to detect the presence of fire and provide immediate alerts both locally, using visual and audio indicators, and remotely through mobile notifications. The system uses an IR flame sensor to identify the presence of fire and activates a buzzer and LED for on-site warning. For remote access and monitoring, the system is connected to the Blynk platform via Wi-Fi, enabling real-time updates and alert notifications on the user's smartphone. Additionally, a webhook is used to log significant fire events to Google Sheets for record-keeping and analysis. This compact and cost-effective system is ideal for environments such as homes, schools, labs, and small-scale industries, offering 24/7 surveillance with reliable alert mechanisms.

2.2 Working Principle

The operation of the Fire Alert System is based on continuous monitoring of the surroundings using the IR flame sensor. When no fire is detected, the sensor output remains HIGH, indicating normal conditions. Once the sensor detects a fire, its output goes LOW, signalling the ESP32 to trigger the alert mechanism. The microcontroller responds by turning ON the LED and buzzer to notify people nearby and simultaneously sends a fire alert to the Blynk mobile app, including a message like " Fire Detected! Please take immediate action." It also updates virtual widgets and logs the event to the Blynk cloud using Blynk.logEvent(). At the same time, the ESP32 sends a POST request through a webhook to Google Sheets to store the fire status. To avoid duplicate entries, the system uses a flag that ensures only one log per detection event. Once the fire is no longer detected, the output goes HIGH again, and the system turns OFF the buzzer and LED, resets the alert status, and updates the user that the environment is safe. This efficient loop ensures real-time monitoring, rapid response, and reliable data storage for improved safety management.

3. HARDWARE REQUIREMENTS

3.1 ESP32 Microcontroller

The ESP32 is a powerful, low-cost, and energy-efficient microcontroller that serves as the central control unit in this Fire Alert System. It features integrated Wi-Fi and Bluetooth capabilities, making it an ideal choice for Internet of Things (IoT) applications. In this project, the ESP32 reads input from the flame sensor, controls the buzzer and LED for local alerts, and communicates with the Blynk IoT platform and Google Sheets for remote notifications and data logging. Its dual-core processor, ample GPIO pins, and built-in networking stack enable smooth multitasking and real-time performance, ensuring that fire detection and alert transmission occur without delay. The ESP32's versatility and reliability make it a core component of this responsive and connected safety system.

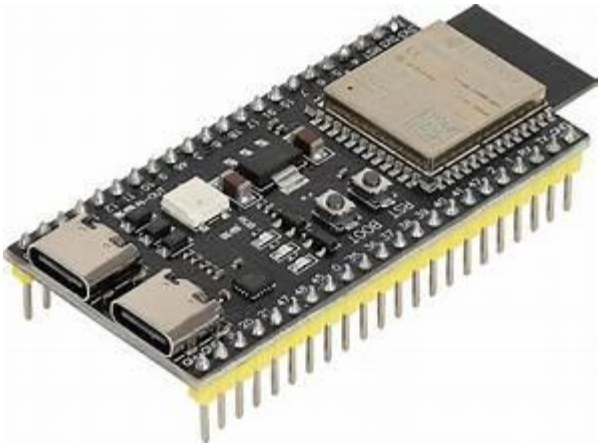


Fig.3.1 ESP32 Microcontroller

3.2 IR Flame Sensor Module

The IR Flame Sensor is a crucial component used to detect fire by sensing infrared radiation emitted by flames. It has an infrared light detector and an emitter, allowing it to identify the unique wavelengths produced by flames. When a flame is detected, the sensor outputs a LOW signal to the ESP32, triggering the alert mechanism. In the absence of a flame, the sensor outputs a HIGH signal, indicating safe conditions. This sensor is highly sensitive and can detect fires from a considerable distance, making it ideal for monitoring environments where fire safety is a concern.

*Fig.3.2.**IR Flame Sensor Module***

3.3 LED

The LED provides a visual alert for fire detection. When the flame sensor detects a fire, the ESP32 turns ON the LED to indicate the presence of a fire. The LED's bright illumination serves as an immediate visual warning for individuals nearby, helping them identify potential fire hazards without delay. The LED is an important part of the system's local alert mechanism, offering an easily visible signal that complements the auditory buzzer, providing redundancy in case one form of alert is missed.

Fig.3.3.LED

3.4 Piezo Buzzer

The Buzzer serves as an auditory alert device that produces a loud sound whenever a fire is detected. It is activated by the ESP32 in response to a LOW signal from the flame sensor. The buzzer plays an essential role in notifying individuals in the immediate vicinity of a fire emergency, ensuring that local personnel are made aware of the situation even in noisy environments. Its straightforward operation makes it a reliable component for generating immediate and attention-grabbing alerts during critical situations.

Fig.3.4. Piezo Buzzer

3.5 Breadboard and Jumper Wires

The Breadboard and Jumper Wires are essential tools for prototyping and assembling the Fire Alert System in its early stages. The breadboard provides a convenient platform for placing and connecting various electronic components, such as the ESP32, IR flame sensor, LED, and buzzer, without the need for soldering. It allows for quick and easy modifications during testing and troubleshooting, enabling a flexible and reusable setup. Jumper Wires are used to make electrical connections between the components on the breadboard, linking the ESP32 to the flame sensor, LED, and buzzer. These wires are flexible, color-coded, and come with pre-fitted connectors, making it simple to establish reliable connections for signal transmission. Together, the breadboard and jumper wires facilitate the construction of a compact and functional prototype for the Fire Alert System, ensuring that all components are properly connected and that the system operates as intended.

Fig.3.5. Breadboard and Jumper Wires

4. Software Requirements

The Fire Alert System involves various software tools and libraries to ensure proper functionality and interaction between the hardware components. Below is a list of the key software requirements:

1. Arduino IDE
    - The Arduino IDE is used for writing and uploading the code to the ESP32 microcontroller. It provides a user-friendly interface for coding, compiling, and uploading the program to the board. The IDE supports the ESP32 platform, allowing you to integrate libraries and communicate with the hardware components efficiently.
2. ESP32 Board Package
    - To program the ESP32, the ESP32 Board Package must be installed within the Arduino IDE. This package includes all necessary drivers, core libraries, and tools to program and interface with the ESP32 microcontroller. It enables communication between the development environment and the hardware.
3. Blynk Library
    - The Blynk Library is used to integrate the ESP32 with the Blynk IoT platform. This library facilitates communication between the microcontroller and the Blynk app, allowing for remote monitoring, control, and notifications related to fire detection. It enables the system to send real-time updates to the user's smartphone through virtual widgets in the Blynk app.
4. HTTPClient Library
    - The HTTPClient Library is used to send HTTP POST requests to external servers. In this project, it enables the ESP32 to send fire detection status to Google Sheets through a webhook. The library manages the HTTP communication, ensuring data is sent in the correct format and the response is processed correctly.
5. Google Sheets Web App
    - A Google Apps Script is required to create a Web App that will receive data from the ESP32 through a webhook. This script is hosted on Google's servers and allows the ESP32 to log fire detection events (status and value) directly into a Google Sheets document. The Google Sheets integration ensures that all fire events are recorded for future analysis and monitoring.
6. Google Apps Script
    - The Google Apps Script is used to handle the data received from the ESP32 and write it into Google Sheets. This script is a simple web service that listens for

POST requests from the ESP32 and updates the Google Sheets document accordingly. It also manages the authentication and security for external access.

7. Wi-Fi Network
    - A Wi-Fi network is required to connect the ESP32 to the internet. This allows the ESP32 to communicate with the Blynk IoT platform and send data to Google Sheets. A stable and reliable internet connection is crucial for ensuring that alerts and logs are transmitted without delay.

5. SYSTEM DIAGRAM

5.1. Pin Connections

Component	Component Pin	ESP32 Pin	Description
Flame Sensor (IR)	Output (D0)	Pin 18	Detects the presence of fire by sensing infrared radiation.
LED	Anode (+)	Pin 19	Provides a visual alert by lighting up when fire is detected.
Buzzer	Positive (+)	Pin 21	Emits an audible sound when a fire is detected.

Table.5.1. Pin Connection

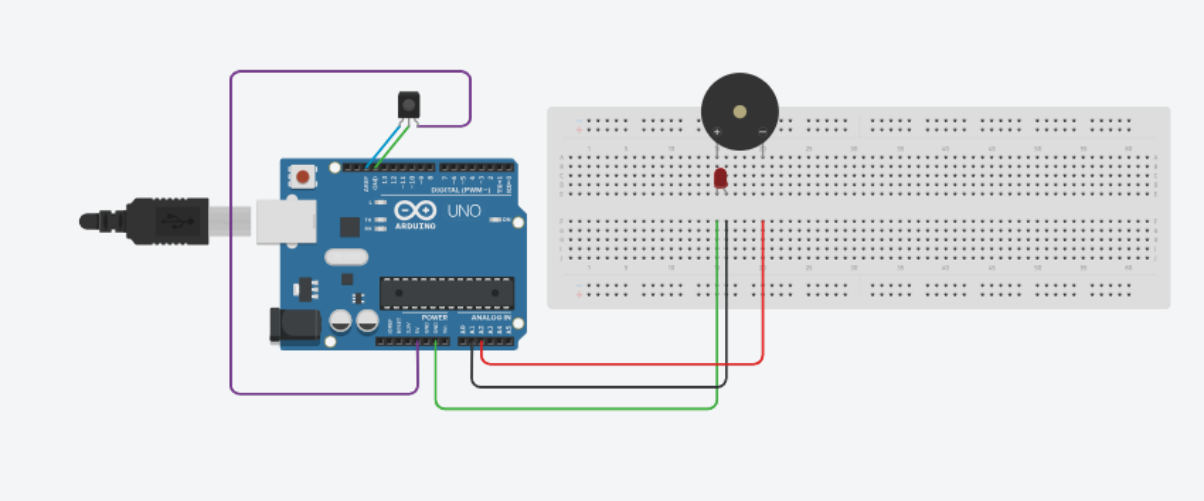


Fig.5.1. Circuit Diagram

6. IMPLEMENTATION

The Fire Alert System is implemented using the ESP32 microcontroller, an IR flame sensor, an LED, and a buzzer. The system is designed to detect fire and alert users both locally and remotely. Below is an overview of the implementation process:

1. **Hardware Setup:**

• The ESP32 is programmed to monitor the state of the flame sensor. When a fire is detected, the ESP32 triggers the LED and buzzer to provide a local alert.

• The flame sensor is connected to Pin 18 of the ESP32, and its output is read to determine the presence of fire based on infrared radiation.

• The LED is connected to Pin 19, and it lights up when fire is detected, acting as a visual indicator of the alert.

• The buzzer is connected to Pin 21, which is activated when the flame sensor detects fire, providing an audible alert.

2. **Software Libraries:**

• The Blynk Library is used to connect the ESP32 to the Blynk IoT platform, enabling remote monitoring and control via the Blynk app. The app is configured to display real-time fire detection alerts and system status.

• The HTTPClient Library is used to send data to Google Sheets via an HTTP POST request. This data is used for logging fire detection status and flame sensor values for further analysis and tracking.

3. **Code Logic:**

• The system continuously checks the state of the flame sensor. If fire is detected (sensor reads LOW), the LED and buzzer are activated. Simultaneously, an alert is sent to the Blynk app and a log entry is added to Google Sheets via a webhook.

• When no fire is detected (sensor reads HIGH), the LED and buzzer are turned off, and the system sends a “No Fire” status to both Blynk and Google Sheets.

• The sendToGoogleSheet function is used to send the fire status to a Google Sheets document, providing a log of all events.

4. **Wi-Fi Connection:**

• The ESP32 connects to the local Wi-Fi network using the provided SSID and password. This enables the ESP32 to communicate with the Blynk app and Google Sheets through the internet.

How it works:

• The ESP32 is initialized and configured to connect to a Wi-Fi network. It checks the flame sensor's state in a continuous loop.

• When fire is detected (flame sensor output is LOW), the LED lights up, and the buzzer sounds. An alert is sent to the Blynk app and a log entry is recorded in Google Sheets via the sendToGoogleSheet function.

• If no fire is detected (flame sensor output is HIGH), the LED and buzzer are turned off, and a "No Fire" status is logged both in the Blynk app and Google Sheets.

7. RESULT

The Fire Alert System successfully integrates multiple components, including the ESP32 microcontroller, flame sensor, LED, buzzer, Blynk app, and Google Sheets for real-time fire detection and logging. Below is a detailed explanation of the expected results during system operation:

1. Fire Detection:

When the flame sensor detects infrared radiation (typically from a fire), it outputs a LOW signal (indicating the presence of fire). Upon this detection:

- The LED connected to Pin 19 lights up, providing a visual indication that fire has been detected.
- The buzzer connected to Pin 21 sounds, alerting anyone nearby with an audible warning.
- The Blynk app receives a notification with the message "Fire Detected!" and the LED widget in the app turns on (corresponding to the LED in the physical setup).
- The Google Sheets log is updated with the status "Fire Detected!" and the sensor value (flame sensor reading of 1).

2. No Fire Detection:

When the flame sensor no longer detects infrared radiation (indicating that the fire is no longer present), it outputs a HIGH signal:

- The LED turns off, indicating that no fire is present.
- The buzzer stops sounding, indicating the absence of a fire threat.
- The Blynk app shows "No Fire," and the LED widget is turned off in the app.
- The Google Sheets log is updated with the status "No Fire" and the sensor value (flame sensor reading of 0).

3. Real-Time Data and Monitoring:

- Blynk App: The user can remotely monitor the fire status in real-time through the Blynk app. Whenever the flame sensor detects fire or no fire, the app updates immediately to show the status, LED widget, and alert messages.
- Google Sheets: All data regarding the fire detection status and flame sensor readings are logged in Google Sheets. This provides a historical log of events, which can be used for further analysis or review.

4. System Performance and Stability:

- The system continuously monitors the state of the flame sensor and responds accordingly to any changes in status. Each detection is communicated via Blynk and logged to Google Sheets with minimal delay, providing fast and efficient alerts.
- The ESP32 microcontroller operates efficiently, maintaining a stable connection to Wi-Fi and Blynk, and ensuring continuous communication with Google Sheets.

Summary of Results:

- Visual and Audible Alerts: The system successfully triggers both visual and audible alerts when fire is detected.
- Remote Monitoring: The integration with Blynk ensures that users can monitor the fire detection status remotely via a mobile app.
- Data Logging: The Google Sheets integration provides a valuable log of fire detection events for future reference and analysis.
- System Efficiency: The system performs well with minimal latency between detection, alerting, and logging, demonstrating its reliability for fire safety monitoring.

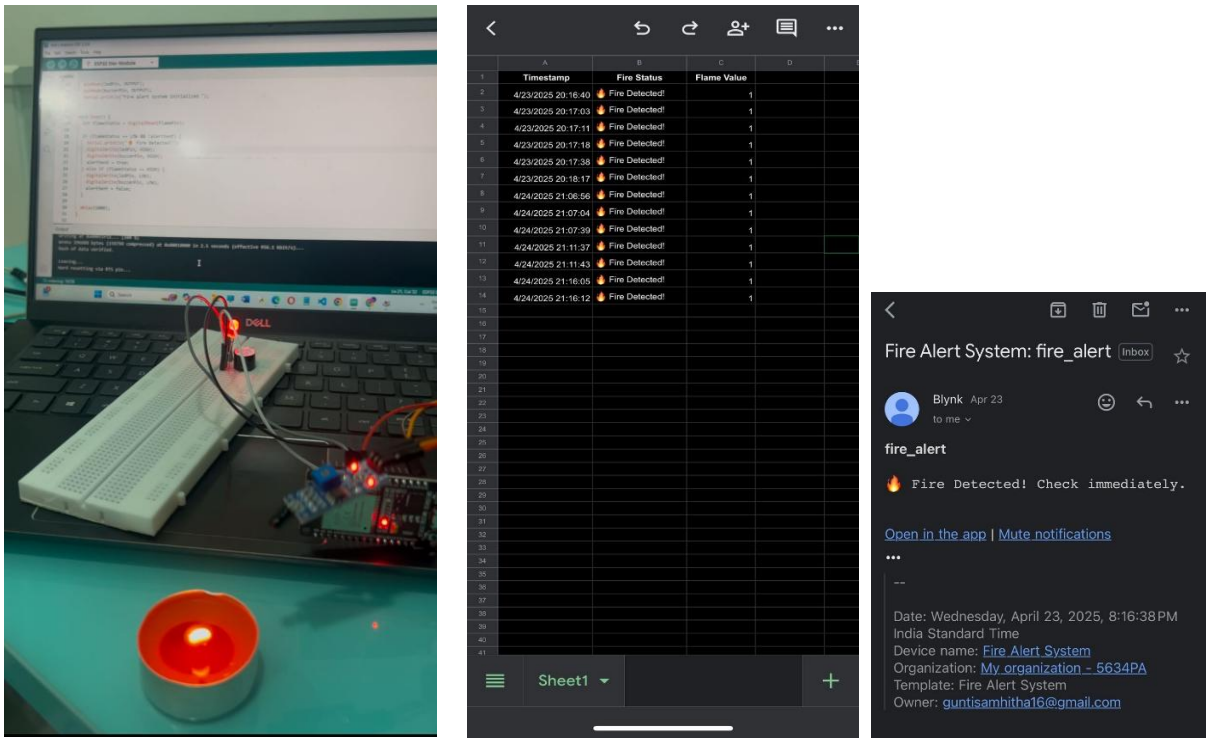


Fig.7.1. Result

8. CONCLUSION

The Fire Alert System developed using the ESP32 microcontroller effectively demonstrates a reliable and efficient method for early fire detection and alerting. By integrating hardware components such as a flame sensor, LED, and buzzer with cloud-based services like Blynk and Google Sheets, the system ensures both local and remote monitoring capabilities. This real-time detection and response mechanism enhances safety by promptly notifying users of potential fire hazards and logging the data for future reference.

The use of IoT technologies enables the system to be scalable and adaptable for various environments such as homes, offices, or industrial areas. Its low cost and ease of deployment make it an ideal solution for improving fire safety in everyday settings. The project highlights the practical application of embedded systems and IoT to address real-world problems, combining automation, connectivity, and data logging into a compact and user-friendly design.

9. REFERENCES

[1] R. Kumar and M. P. Singh, “Fire detection system using IoT and machine learning,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 5, pp. 626–630, 2020.

[2] M. A. Mazidi, R. D. McKinlay, and D. Causey, *Raspberry Pi and IoT: A Practical Guide to the Internet of Things*, Prentice Hall, 2019.

[3] A. Sharma and K. Jain, “Implementation of fire detection system using flame sensor and Raspberry Pi,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 6, pp. 2441–2444, 2020.

[4] S. Sharma and R. Singh, “Design and Development of IoT Based Smart Fire Detection and Notification System,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 8, pp. 182–186, 2020.