FOREST FIRE DETECTOR

A Project Report

Submitted to the APJ Abdul Kalam Technological University

in partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Electronics and Communication Engineering

With Minor Degree

in

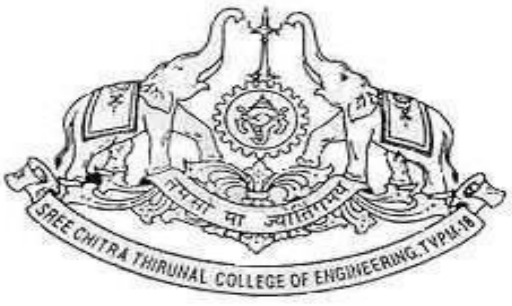
Networking

By

ALWIN JOHN THOMAS (SCT22EC025)

ANASWARA S KUMAR (SCT22EC031)

VAISHNA S P (SCT22BT055)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SREE CHITRA THIRUNAL COLLEGE OF ENGINEERING

THIRUVANANTHAPURAM

KERALA

OCTOBER 2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SREE CHITRA THIRUNAL COLLEGE OF ENGINEERING

THIRUVANANTHAPURAM

2025-2026

CERTIFICATE

This is to certify that the report entitled **FOREST FIRE DETECTOR** submitted by **ALWIN JOHN THOMAS (SCT22EC025), ANASWARA S KUMAR (SCT22EC031), VAISHNA S P (SCT22BT055),** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of B.Tech Degree in Electronics and Communication Engineering with Minor Degree in Networking is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Smt. Ammu Archa P
(Project Guide)
Assistant Professor
Department of CSE
SCTCE
Thiruvananthapuram

Smt. Merrin J
(Project Coordinator)
Assistant Professor
Department of CSE
SCTCE
Thiruvananthapuram

Dr. Soniya B
Professor and
Head of the Department
Dept. of CSE
SCTCE
Thiruvananthapuram

DECLARATION

We hereby declare that the project report **FOREST FIRE DETECTOR**, submitted for partial fulfillment of the requirements for the award of B.Tech Degree in Electronics and Communication Engineering with Minor Degree in Networking of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Smt. Ammu Archa P. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Alwin John Thomas
Anaswara S Kumar
Vaishna S P

Thiruvananthapuram

24-10-2025

ACKNOWLEDGEMENT

We take this opportunity to express our deepest sense of gratitude and sincere thanks to everyone who helped us to complete this work successfully. We express our sincere thanks to **Dr. Soniya B**, Head of Department, Computer Science and Engineering, Sree Chitra Thirunal College of Engineering, Thiruvananthapuram for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to **Smt. Merrin J**, Assistant Professor, Department of Computer Science and Engineering, Sree Chitra Thirunal College of Engineering, Thiruvananthapuram for the support and co-operation.

We would like to place on record our sincere gratitude to our project guide **Smt. Ammu Archa P**, Assistant Professor, Computer Science and Engineering, Sree Chitra Thirunal College of Engineering for the guidance and mentorship throughout the course.

Finally We thank our family, and friends who contributed to the successful fulfilment of this project work.

Alwin John Thomas
Anaswara S Kumar
Vaishna S P

ABSTRACT

Forest fires cause serious damage to the environment, wildlife, and nearby human areas. Early detection plays a major role in preventing the spread of such fires. This project aims to develop an IoT-based Forest Fire Detector that can monitor environmental conditions and detect early signs of fire using low-cost electronic components. The system uses an ESP32 microcontroller connected with various sensors such as DHT11 (temperature and humidity sensor), MQ-2 gas sensor (smoke detection), flame sensor, and PIR sensor (motion detection). The collected data is displayed on an LCD with an I²C interface, while a relay module acts as an indicator for sprinkler activation when a fire is detected. The sensor readings are sent to the Firebase Realtime Database, allowing users to view the data and alerts remotely through the internet. When the system detects abnormal conditions like high temperature, smoke, or flame, it immediately updates the database and shows an alert message "FOREST FIRE DETECTED." The project successfully demonstrates a simple, reliable, and low-cost IoT solution for early forest fire detection. It can be extended further by adding GSM or camera modules for larger and more practical outdoor applications.

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER-1
INTRODUCTION

1.1. Overview

Forest fires are one of the most devastating natural disasters that can cause severe damage to the environment, wildlife, and human property. Traditional fire detection systems rely on human observation or satellite imagery, which often lead to delays in detection and response. To minimize losses, an automated and real-time fire detection system is essential.This project focuses on developing an IoT-based Forest Fire Detection System that continuously monitors environmental parameters such as temperature, humidity, presence of smoke, flame, and human activity using multiple sensors. The collected data is transmitted to a cloud database for analysis and real-time alert generation.

1.2. Motivation

The increase in global temperatures and dry environmental conditions have made forests more vulnerable to fire outbreaks. Early detection and timely action can prevent the spread of fires and reduce environmental destruction. The motivation behind this project is to design a cost-effective and efficient system capable of detecting early signs of a fire using sensor data and triggering automatic safety responses, such as activating sprinklers, while simultaneously notifying users through cloud connectivity.

1.3. Objectives

The main objectives of this project are:

- To design a real-time forest fire monitoring system using an ESP32 microcontroller.
- To integrate multiple sensors such as temperature, humidity, smoke, flame, and motion sensors for comprehensive detection.
- To send the collected data to Firebase Cloud for remote monitoring and analysis.
- To activate a relay-controlled sprinkler system automatically upon fire detection.
- To provide visual output on an OLED display for local status monitoring.

1.4. System Overview

The system utilizes an ESP32 microcontroller as the central processing unit. It interfaces with a temperature sensor (DHT11) to measure ambient temperature and humidity, a flame sensor to detect open flames, an MQ-2 gas sensor to identify smoke or combustible gases, and a PIR sensor to monitor movement in the vicinity. The OLED display (with I2C interface) is used to present real-time sensor readings, while a relay module controls the sprinkler system. When the ESP32 detects conditions indicating a fire, it triggers the relay, thereby activating the sprinkler. Simultaneously, all sensor data and alerts are uploaded to Firebase, which serves as the cloud database for real-time monitoring and further analysis on a connected laptop or other devices.

1.5. Advantages

- Early detection of forest fires through real-time monitoring.
- Automated response mechanism using a sprinkler system.
- Remote access and monitoring through Firebase integration.
- Cost-effective and scalable design.

CHAPTER-2
LITERATURE REVIEW

2.1. Introduction

Detecting forest fires early is crucial to prevent large-scale damage. Recent research focuses on two main approaches: (a) camera-based systems with deep learning for accurate fire detection over large areas, and (b) small sensor networks for fast, local detection. Each approach has benefits and limitations. Camera-based deep learning models are accurate but need more computation and reliable network connectivity. Sensor networks are fast and low-cost but cover only the immediate area and can produce false alarms [1][2][3].

2.2. Surveys and Reviews

Comprehensive reviews indicate a growing trend toward combining traditional sensor networks, unmanned aerial vehicles (UAVs), and deep learning for wildfire surveillance [1]. Saleh et al. provide a broad review of deep-learning methods for forest-fire surveillance and show that while DL models (classification, detection, segmentation) have achieved high accuracy, challenges remain in data scarcity, small-spot detection and real-time constraints [1]. These observations motivate hybrid systems that fuse lightweight local sensing with cloud analytics to enable both fast local response and more accurate central analysis.

Table 2.1: Literature Survey

Name of the Author / Source	Name of Journal / Work	Contribution
Saleh et al.[1]	IEEE Access: "Forest Fire Surveillance Systems: A Review of Deep Learning Methods"	Reviewed deep-learning methods for forest-fire detection. It showed they are accurate but face issues with limited data, detecting small fires, and working in real-time.

Tong et al. [2]	IEEE Access: "Real-Time Wildfire Monitoring Using Low-Altitude Remote Sensing Imagery"	Created FireFormer, a model for UAV images that accurately detects small fire spots in real-time using advanced image processing.
Avazov et al. [3]	IEEE Access: "Forest Fire Detection and Notification Method Based on AI and IoT Approaches"	Built an AI and IoT system using YOLOv5 and sensors to detect fires accurately and reduce false alarms.
Lee et al. [4]	IEEE Access: "Development of IoT-Based Real-Time Fire Detection System Using Raspberry Pi and Fisheye Camera"	Built an IoT system with a Raspberry Pi and camera that sends fire detection data to the cloud in real-time for low-cost monitoring.
Ghali et al. [5]	IEEE Access: "Deep Learning and Transformer Approaches for UAV-Based Wildfire Detection and Segmentation "	Showed that advanced deep-learning models improve detection of small fires in UAV images for better wildfire monitoring.

2.3. Image-based and UAV-based Approaches

High-resolution aerial imagery from drones or satellites is often processed with deep learning to detect small fires. For example, FireFormer uses a combination of convolutional and transformer networks to detect fires in drone images with high accuracy [2]. Other research shows that vision-transformer-based models improve fine-grain fire detection [5]. While these methods are accurate, they require significant computing power and may not work in areas with poor connectivity.

2.4. Sensor-based IoT Systems and Cloud Integration

IoT sensor networks measure temperature, smoke, gas, and motion to detect fires locally. Combining multiple sensors can reduce false alarms. For example, Avazov et al. developed a system that combines AI with IoT sensors to improve fire detection accuracy [3]. Lee et al. built a field-tested system using Raspberry Pi and Firebase for real-time cloud monitoring [4]. These studies show that combining local sensing with cloud analytics provides a low-cost and reliable early warning solution.

2.5. Key Findings & Gaps

From the surveyed literature the following key points are evident:

- Image-based deep learning provides high accuracy but needs more computation and connectivity.
- Local sensor networks are fast and low-cost but cover limited areas and can give false alarms.
- Combining both approaches is effective: sensors detect fires quickly, and cloud storage allows remote monitoring and analysis.

2.6. How This Project Builds on Previous Work

Our project uses an ESP32 with multiple sensors (DHT11, flame sensor, MQ-2, PIR) and an OLED display. The ESP32 triggers a relay to turn on a sprinkler when a fire is detected, while sending data to Firebase for remote monitoring and analysis. This hybrid approach provides fast local response and cloud-based tracking, addressing limitations of both image-based and sensor-only systems [1][3][4][5].

2.7. Summary

Recent studies show two main ways to detect forest fires: (a) using cameras and deep learning for accurate, wide-area monitoring, and (b) using small sensor networks for fast, local detection. The best systems combine both approaches: sensors at the site for quick action, and cloud storage for tracking and analysis. Our ESP32-based system follows this idea — it triggers the sprinkler immediately when needed and sends data to Firebase for remote monitoring and further analysis on a laptop.

pattern-recognition strength of deep learning models with the qualitative, contextual analysis capabilities of LLMs. Furthermore, there is a clear trend towards developing systems that are not only accurate but also usable and educational for a broader audience, including novice investors. Fin-Sight is positioned within this research trajectory. It synthesizes these findings by proposing a client-side, web-integrated system that employs a TensorFlow.js model for technical analysis and the Gemini AI API for intelligent interpretation, thereby creating a practical and insightful tool for short-term market forecasting.

CHAPTER 3
SYSTEM DESIGN AND COMPONENTS

3.1. Introduction

This chapter explains the hardware and software architecture of the proposed IoT-based Forest Fire Detection System. It describes the main components used, their interfacing with the ESP32 microcontroller, and the data flow from sensor nodes to Firebase for cloud storage and monitoring. The system is designed to continuously collect environmental parameters, analyze sensor readings, and activate an alert mechanism in the event of a fire hazard.

3.2. System Architecture

The overall system architecture consists of four main stages:

1. **Sensing Layer:**
   The SensorS DHT11 (temperature and humidity), Flame sensor, MQ-2 (smoke/gas), and PIR module (motion detection) are used to monitor environmental conditions.

2. **Processing Layer:**
   The ESP32 microcontroller acts as the core control unit. It reads real-time data from all sensors through analog/digital pins, processes the data, and makes decisions based on predefined threshold values.

3. **Communication Layer:**
   The ESP32 uses its built-in Wi-Fi module to send sensor data to Firebase Cloud, where it can be accessed and analyzed remotely on the laptop.

4. **Actuation Layer:**
   The Relay Module is connected to the ESP32 which indicates a sprinkler system. When a fire event is detected, the relay is activated to trigger the sprinkler automatically. The LCD Display provides local visualization of sensor data and alerts.

3.3. Block Diagram

The system's functional flow can be described as follows:

1. The sensors continuously sense temperature, humidity, flame, gas, and motion.
2. The ESP32 processes these signals and checks for threshold exceedance.
3. If a fire is suspected (e.g., high temperature, smoke, or flame detection), the relay activates the sprinkler system.
4. Data from all sensors are sent to Firebase using the ESP32's Wi-Fi interface.
5. The OLED screen displays live readings and system status.
6. The stored data in Firebase can later be visualized and analyzed to observe patterns or predict future risks [3].

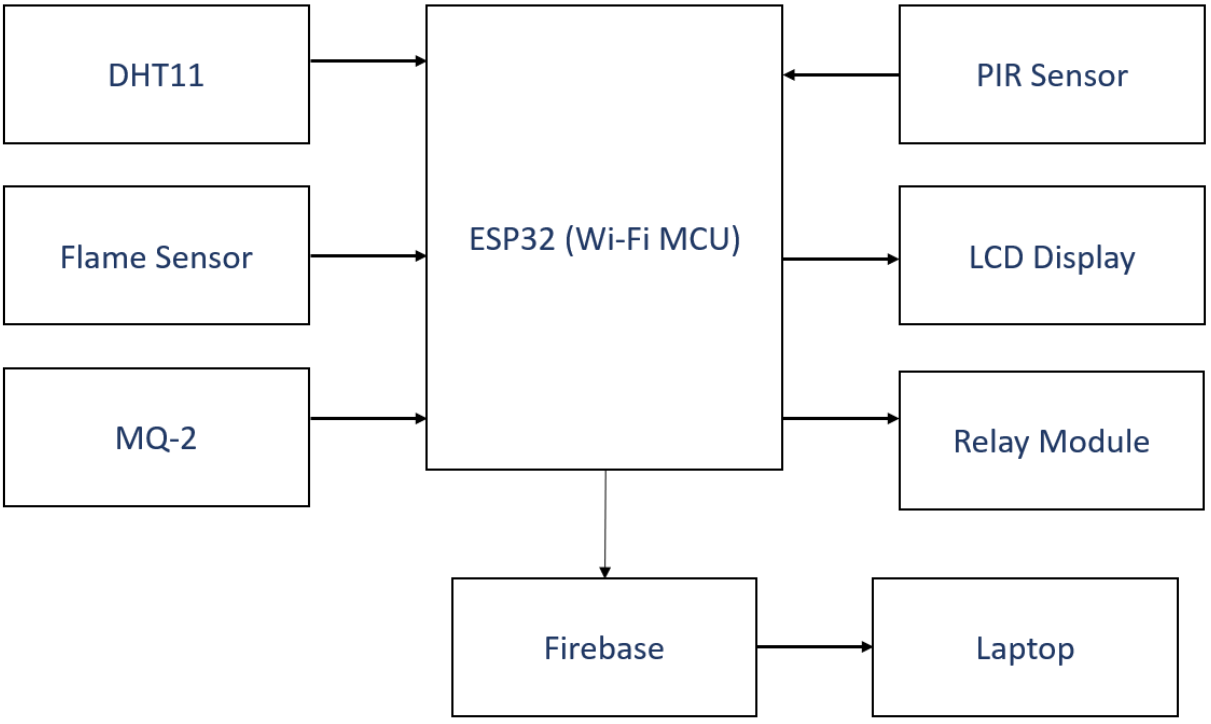


Fig. 3.1: Block Diagram

3.1. Hardware Components

3.4.1. ESP32 Microcontroller

The ESP32 is a low-cost, low-power microcontroller equipped with Wi-Fi and Bluetooth capabilities. It provides high processing power for IoT applications and is ideal for real-time sensing and data transmission tasks. The ESP32 replaces the need for an external communication module, reducing cost and complexity.

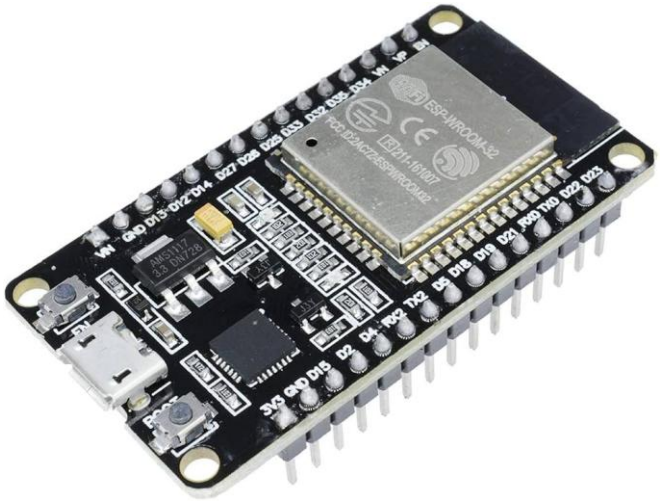


Fig. 3.2: ESP32 Microcontroller

3.4.2 DHT11 : Temperature and Humidity Sensor

The DHT11 sensor measures both temperature and relative humidity. These parameters are crucial for identifying dry conditions that increase fire risks. It outputs calibrated digital signals, which are processed by the ESP32 through a single data pin.

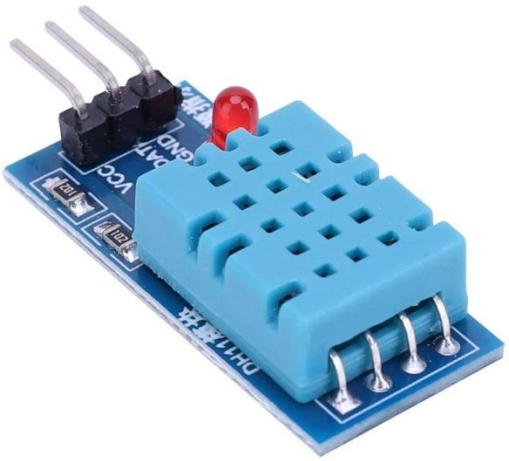


Fig. 3.3: DHT11

3.4.3 Flame Sensor

This sensor detects infrared light emitted by a flame. It helps identify the presence of fire at an early stage by sensing changes in IR radiation levels .

Fig. 3.4: Flame Sensor

3.4.4 MQ-2 Gas Sensor

The MQ-2 sensor detects the presence of smoke, methane, LPG, and other combustible gases. It is analog-based and provides readings proportional to the smoke or gas concentration in the air. It is particularly effective for detecting early smoke formation before flames are visible.

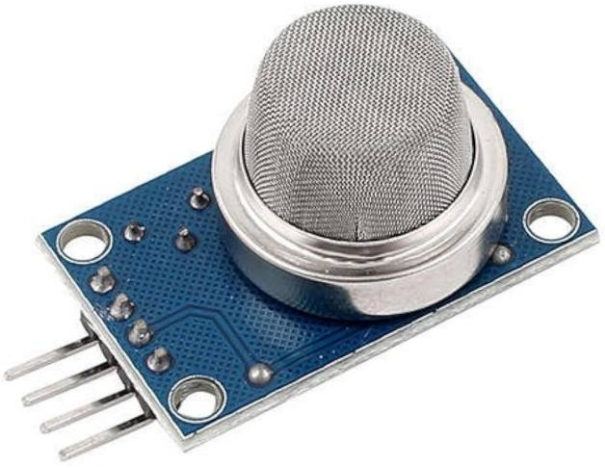


Fig. 3.5: MQ-2 Gas Sensor

3.4.5 PIR Motion Sensor

The Passive Infrared (PIR) sensor detects movement of warm bodies (like animals or humans) within its range. This data can be useful for understanding human activity or wildlife presence near the fire-prone area.

Fig. 3.6: PIR Motion Sensor

3.4.6 Relay Module

The relay module acts as a switch controlled by the ESP32. When the system detects a fire event, the ESP32 energizes the relay, which in turn activates the sprinkler system to suppress the flames.

Fig. 3.7: Relay Module

3.4.7 LCD Display (I2C Interface)

A compact LCD display module is used to show live temperature, humidity, and alert status. Its I2C interface minimizes wiring complexity and enables simultaneous operation with other components.

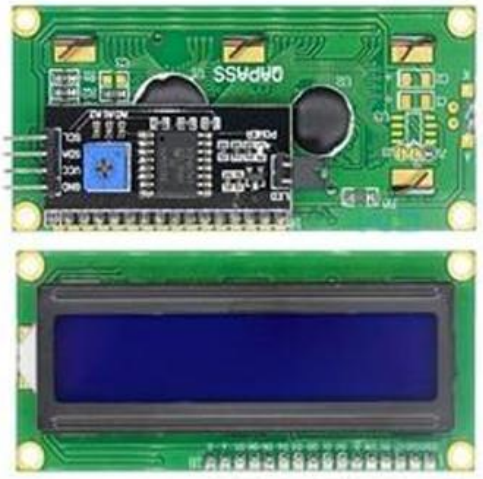


Fig. 3.8: LCD Display (I2C Interface)

3.5 Software Design

The software system is divided into three functional sections:

1. Sensor Data Acquisition: Reading analog and digital values from the connected sensors.
2. Decision Logic: Implementing threshold-based algorithms for detecting fire-like conditions.
3. Data Transmission: Sending sensor data and alerts to Firebase for real-time monitoring and visualization.

3.6 Data Flow

1. The datas from the sensors are collected by ESP32 through analog/digital pins.
2. Data is processed and compared against preset limits.
3. Fire condition triggers relay activation which indicates sprinkler response.
4. Processed data is formatted and uploaded to Firebase Cloud.
5. Firebase synchronizes data with the analysis dashboard on the laptop.

CHAPTER-4
IMPLEMENTATION AND WORKING

4.1. Introduction

This chapter explains how the *Forest* Fire Detection System was built and how it works in real time. The project uses the ESP32 microcontroller as the main controller and connects different sensors to monitor environmental conditions. The readings are displayed on an LCD display (I²C type) and uploaded to Firebase for remote viewing. The relay module is used to indicate a sprinkler activation, showing that the system has detected a possible fire situation.

4.2. System Setup

All the sensors—DHT11 (temperature and humidity), flame sensor, MQ-2 gas sensor, and PIR sensor—are connected to the ESP32 board. The ESP32 provides both processing and Wi-Fi connectivity. The LCD display with I²C interface is used to show real-time data such as temperature, humidity, smoke level, and flame detection status. The relay module is connected to a digital pin of the ESP32 and is used as an indicator that turns ON when the system detects a fire.

The data from the sensors is updated on the Firebase Realtime Database, which can be accessed on a laptop or phone for live monitoring. This setup helps in detecting possible fire conditions early and alerting the user immediately.

4.3. Flow of Working

The system works in a step-by-step process, as shown in the flow diagram below.

Step 1: All sensors start reading data continuously after the ESP32 is powered ON.
Step 2: The ESP32 receives the data and checks for abnormal values such as:

- High temperature or low humidity from DHT11,
- Detection of smoke/gas from MQ-2,
- Detection of flame from the flame sensor.

Step 3: If one or more conditions go beyond the preset limit, the ESP32 assumes that

a fire might be starting.

Step 4: The relay module turns ON, indicating that the sprinkler would have been activated in a real system.

Step 5: At the same time, all sensor readings are sent to Firebase using Wi-Fi.

Step 6: The LCD display shows the latest readings and the fire status.

This process repeats continuously to ensure that any small change in the environment is quickly detected

4.4. Firebase Integration

The Firebase Realtime Database is used as the cloud platform. The ESP32 connects to Firebase using Wi-Fi and uploads sensor readings at regular intervals. Each parameter such as temperature, humidity, flame, and smoke is stored under separate fields. When a fire-like condition is detected, an alert message is also uploaded. This allows real-time monitoring from any device with internet access, similar to the IoT monitoring setup proposed by Lee et al. [4].

4.5. Working Description

When the system is powered, the ESP32 begins reading data from all sensors. If the temperature increases sharply, the humidity drops, or if smoke or flame is detected, the system identifies that as a fire warning condition. The relay module turns ON as an indicator (acting as a sprinkler signal), and “FOREST FIRE DETECTED” is shown on the LCDscreen.

The same data is then uploaded to Firebase for storage and remote viewing. If all readings are within normal range, the relay remains OFF, and “No fire detected – All clear” is displayed.

This real-time detection and cloud update make the system fast and reliable for fire monitoring in forest-like environments.

4.6. Advantages of Implementation

- Quick detection: Immediate alert when abnormal readings occur.
- Easy monitoring: Live data on LCD and Firebase.
- Low cost: Uses common sensors and ESP32 board.

- Safe simulation: Relay acts as sprinkler indicator, no real sprinkler needed.
- Simple operation: Works automatically once powered on.

4.7. Summary

The implementation successfully combines different sensors with ESP32 to create a working prototype of a forest fire detection system. The LCD helps in local display, and Firebase enables cloud-based data access. The relay serves as a safe indicator for sprinkler activation. This design shows that an IoT-based system can give quick and reliable fire alerts while staying cost-effective and simple to operate.

CHAPTER-5

RESULTS AND DISCUSSION

5.1. Introduction

This chapter presents the results obtained after implementing and testing the IoT-based Forest Fire Detection System. The performance of each sensor, the relay indication, and the data updates on Firebase were observed under different environmental conditions. The main aim of testing was to verify that the system could correctly detect changes in temperature, smoke, or flame and give alerts instantly on both the LCD and Firebase platform.

5.2. Project Setup

The complete hardware setup of the project is shown below.

It includes the ESP32 microcontroller, DHT11 sensor, Flame sensor, MQ-2 gas sensor, PIR sensor, LCD with I2C interface, and Relay module.

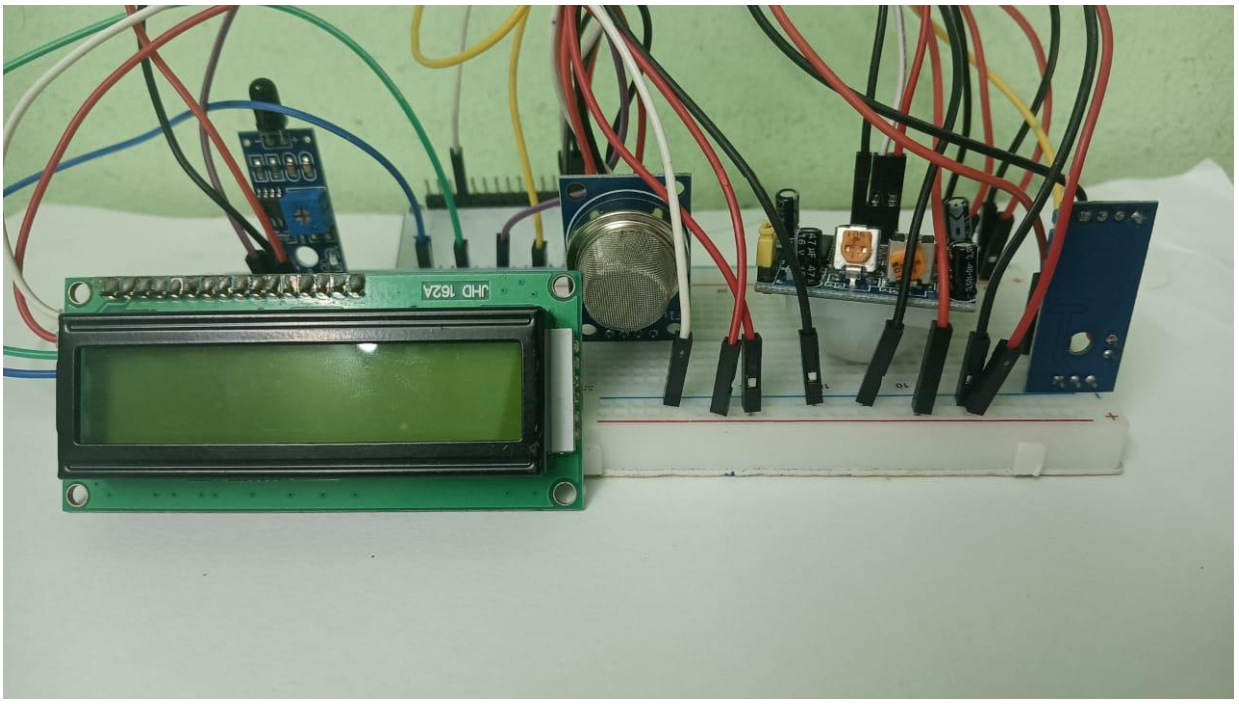


Fig. 5.1: Project Setup

Description of the setup:

- The ESP32 microcontroller acts as the main controller that collects readings from all connected sensors.
- The DHT11 sensor measures temperature and humidity.

- The MQ-2 gas sensor detects the smoke concentration in the air.
- The Flame sensor identifies the presence of open fire or flame.
- The PIR sensor senses human or animal movement in the area.
- The LCD display (I2C) shows real-time sensor readings and system status.
- The Relay module is connected to show an indication that a sprinkler would turn ON in case of fire detection.
- Data from the ESP32 is sent to the Firebase Realtime Database through Wi-Fi.

5.3. Firebase Data Display

All sensor readings were uploaded to Firebase Realtime Database in real time.

Whenever a fire-like condition was detected, Firebase displayed an alert message along with live sensor data.

Below is the Firebase output from the project:

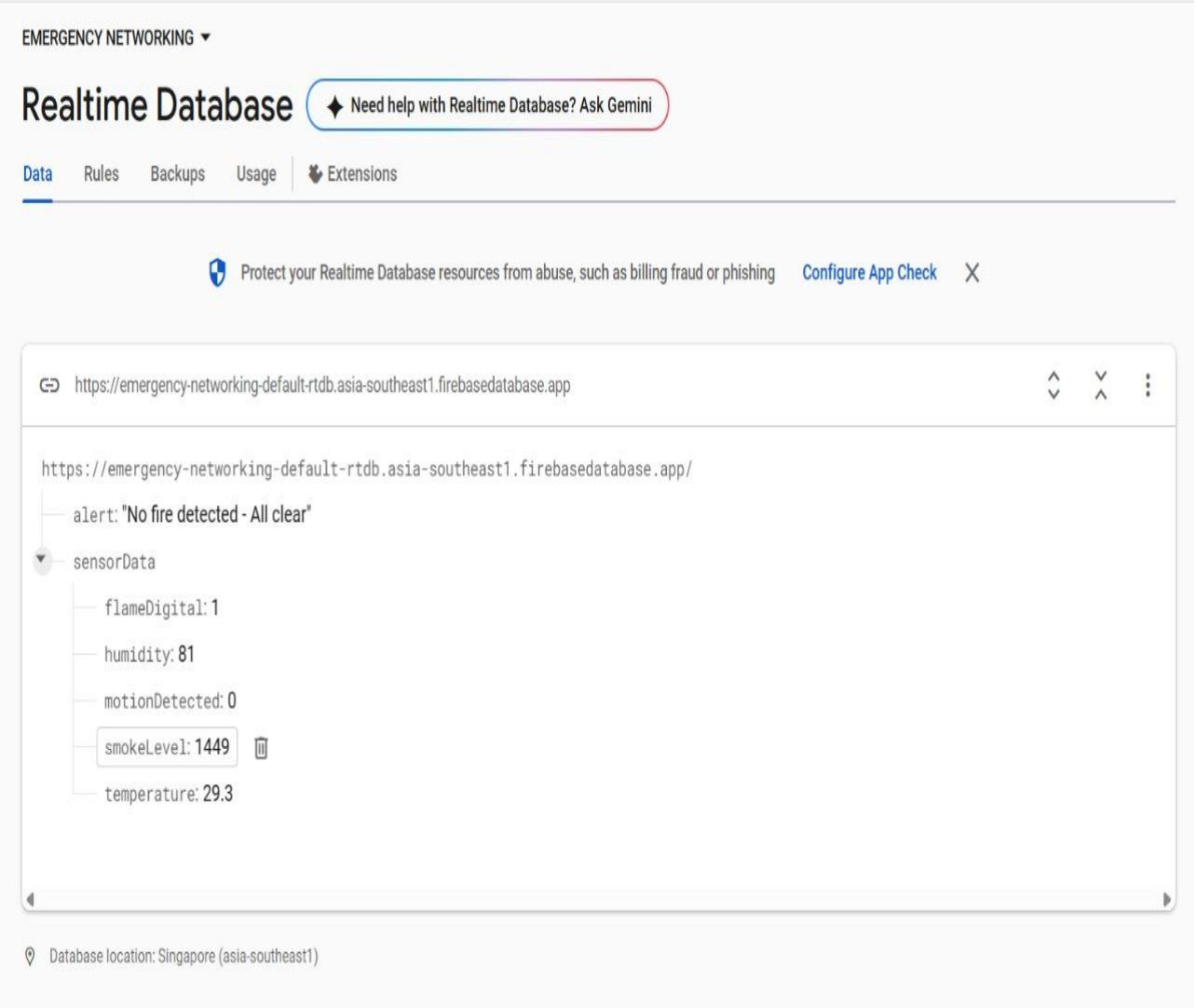


Fig. 5.2 : Firebase Data Display (Before Fire Detection)

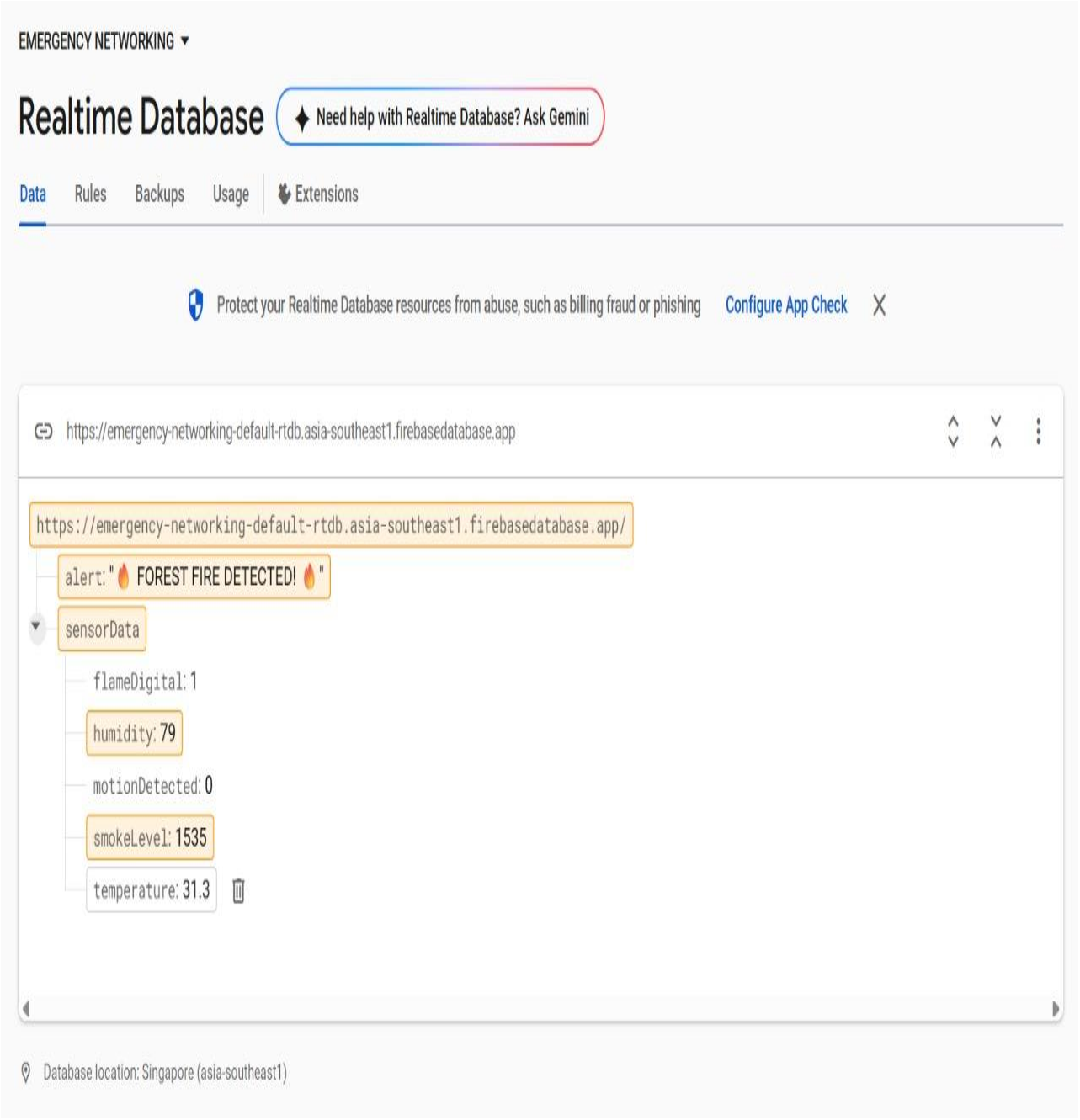


Fig. 5.3 : Firebase Data Display (After Fire Detection)

From the database, it can be seen that the system correctly detected a fire and immediately updated the alert message. The ESP32 continuously sent data through Wi-Fi, and the Firebase dashboard updated automatically.

This confirmed that the communication between ESP32 and Firebase was working perfectly and the cloud data reflected real-time conditions accurately

5.4. LCD Display Output

The LCD display (connected through I²C) continuously showed the live sensor readings. During normal conditions, it displayed temperature, humidity, and the message “Status: Normal.”

When the flame or smoke sensor was triggered, the LCD displayed the alert message “FIRE DETECTED,” making it easy to understand the system status even without internet connectivity.

Fig. 5.4 : LCD Display Output (Before Fire Detection)

Fig. 5.5 : LCD Display Output (After Fire Detection)

5.5. Summary

Include The system was able to detect early signs of fire accurately and transmit data to Firebase in real time. The LCD provided clear local readings, and the relay indicated the fire condition successfully. This proves that the proposed design can be used as a low-cost, IoT-based forest fire detection prototype for small forest regions and environmental monitoring.

CHAPTER-6

CONCLUSION

6.1. Conclusion

This project successfully demonstrates a working IoT-based forest fire detection system using the ESP32 microcontroller and Firebase cloud. The system monitors environmental conditions using multiple sensors: DHT11 for temperature and humidity, MQ-2 for smoke, flame sensor for fire, and PIR for motion detection.

During testing, the system reliably detected potential fire hazards. When unusual conditions occur, it triggers local alerts on the LCD and sends notifications to Firebase. The relay module shows when a sprinkler would activate, proving the concept works.

Our results show that the system meets its main goals: continuous monitoring, real-time data upload to Firebase, instant alerts, and clear visual feedback via the LCD and relay. It is simple, affordable, and made from easily available components.

However, there are some limitations. The system relies on Wi-Fi, which may not be available in remote forests. A single sensor unit covers only a small area, and the relay currently only indicates a sprinkler rather than controlling a real one.

Future improvements could include using GSM or LoRa for remote communication, adding a camera for visual confirmation, connecting real sprinklers or alarms, developing a mobile app or dashboard, and using solar panels for energy independence.

In short, this project provides a simple, affordable, and functional prototype for forest fire detection. With further improvements, it could become a useful tool for protecting forests and nearby communities.

CHAPTER 7

REFRENCES

[1] A. Saleh, M. A. Zulkifley, H. H. Harun, F. Gaudreault, I. Davison and M. Spraggon, “Forest fire surveillance systems: A review of deep learning methods,” *Heliyon*, vol. 10, no. 1, e23127, Dec. 2023, doi: 10.1016/j.heliyon.2023.e23127.

[2] H. Tong et al., “Real-Time Wildfire Monitoring Using Low-Altitude Remote Sensing Imagery,” *Remote Sensing*, vol. 16, no. 15, Art. no. 2827, Aug. 2024, doi: 10.3390/rs16152827.

[3] K. Avazov, A. E. Hyun, A. A. Sami S., A. Khaitov, A. B. Abdusalomov and Y. I. Cho, “Forest Fire Detection and Notification Method Based on AI and IoT Approaches,” *Future Internet*, vol. 15, no. 2, Art. no. 61, Jan. 2023, doi: 10.3390/fi15020061.

[4] C.-H. Lee, W.-H. Lee and S.-M. Kim, “Development of IoT-Based Real-Time Fire Detection System Using Raspberry Pi and Fisheye Camera,” *Applied Sciences*, vol. 13, no. 15, Art. no. 8568, Jul. 2023, doi: 10.3390/app13158568.

[5] R. Ghali, M. A. Akhloufi and W. S. Mseddi, “Deep Learning and Transformer Approaches for UAV-Based Wildfire Detection and Segmentation,” *Sensors*, vol. 22, no. 5, Art. no. 1977, Mar. 2022, doi: 10.3390/s22051977.