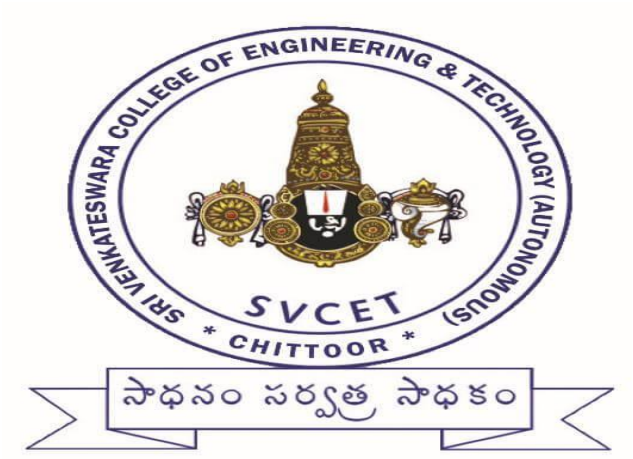


SRI VENKATESWARA COLLEGE OF ENGINEERING& TECHNOLOGY

(AUTONOMOUS)

R.V.S NAGAR, CHITTOOR – 517 127. (A.P)

(Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapuramu) (Accredited

By NBA, New Delhi & NAAC, Bengaluru)

(An ISO 9001:2000 Certified Institution)

2023-2024

PROJECT DETAILS:

CLASS : Third year ECE, second semester.

COURSE NAME: Applied industrial IOT

PROJECT NAME : Fire detection system.

BATCH: Batch 23(3)

1.Kurra Lavanya (21781A04A6)

2.Kasula Naga Neeraj (21781A0492)

3.Katha Jagan Mohan Reddy (21781A0493)

PROJECT

FIRE DETECTION SYSTEM

AIM :To Design and Develop a Fire Detection system using ESP8266 using temperature and smoke sensors, ensuring rapid response and enhanced safety to detect flames or smoke promptly.

PROBLEM STATEMENT: Fires pose a significant threat to both life and property, necessitating a reliable and rapid detection system. Traditional fire detection methods may lack real-time monitoring and remote alerting capabilities, limiting their effectiveness. The challenge is to design a Fire Detection System using ESP8266, integrating temperature and smoke sensors, to promptly and accurately detect flames or smoke, ensuring enhanced safety through rapid response.

SCOPE OF THE SOLUTION:

1. **Sensor Integration:**
   - Utilize temperature and smoke sensors to detect changes in the environment.
   - Ensure compatibility and seamless integration with ESP8266 for real-time data processing.
2. **Real-time Monitoring**:
   - Implement continuous monitoring of temperature and smoke levels.
   - Enable the system to identify abnormal patterns or sudden spikes indicative of a potential fire.
3. **Communication and Alerting:**
   - Establish a robust communication protocol for transmitting data to a centralized server or a cloud platform.
   - Integrate mechanisms for instant alerting, such as SMS, email, or push notifications, to notify users and authorities.
4. **Remote Access and Control**:
   - Facilitate remote access to the system for users through a secure interface.
   - Implement control features allowing users to remotely manage and monitor the system.
5. **Machine Learning Integration (Optional):**
   - Explore the possibility of integrating machine learning algorithms to enhance the system's ability to distinguish between false alarms and actual fire events.
6. **Power Efficiency:**
   - Optimize power consumption to ensure prolonged operation, even during power outages, through the use of efficient algorithms and low-power modes.
7. **Scalability and Adaptability**:
   - Design the system with scalability in mind, allowing for easy expansion to cover larger areas.
   - Ensure adaptability to different environments and scenarios.

8. **Emergency Response Integration:**
   - Collaborate with local emergency services to integrate the system with their response mechanisms, facilitating a faster and coordinated reaction to fire incidents.
9. **User-Friendly Interface:**
   - Develop a user-friendly interface for easy setup, configuration, and monitoring, catering to both technical and non-technical users.
10. **Compliance and Standards:**
    - here to safety and regulatory standards to ensure the system's effectiveness and compliance with legal requirements.

By addressing these aspects, the proposed Fire Detection System aims to provide a comprehensive and technologically advanced solution to enhance fire safety measures, mitigating risks through swift detection and response.

REQUIRED COMPONENTS TO DEVELOP SOLUTIONS:

1) An ESP8266 microcontroller board
2) A temperature sensor
3) A smoke sensor
4) A breadboard
5) Jumper wires
6) A power supply
7) An LED (optional)
8) A buzzer (optional)

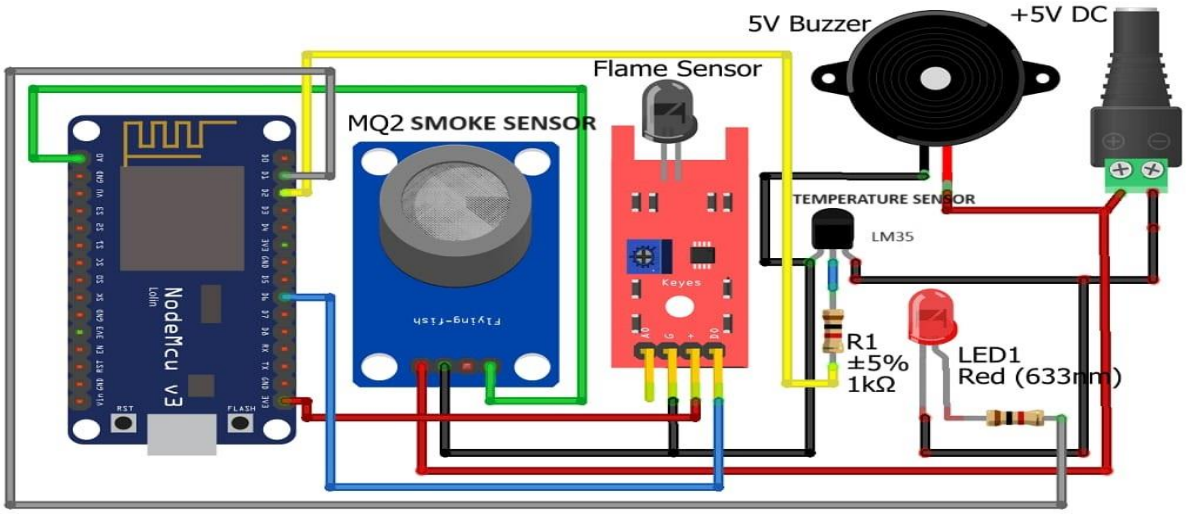
DESIGN PROCESS:

Connect the components.: Connect the temperature sensor and smoke sensor to the ESP8266 board according to the following diagram:

Temperature sensor: VCC to 3.3V, GND to GND, OUT to A0.

Smoke sensor: VCC to 3.3V, GND to GND, OUT to D1.

SIMULATED CIRCUIT :



Upload the code to the ESP8266 board : You can use the Arduino IDE to upload the code to the ESP8266 board.

Test the system : Once the code is uploaded, you can test the system by heating up the temperature sensor or blowing smoke into the smoke sensor. The system should send an alert to the cloud when a fire is detected.

Deploy the system : Once you are satisfied with the system, you can deploy it in your home or office. You can mount the ESP8266 board and sensors in a convenient location, and connect the power supply.

Here are some additional tips for designing and developing a fire detection system using ESP8266:

- Use a high-quality temperature sensor and smoke sensor.
- Place the sensors in strategic locations where they are likely to detect a fire early.
- Test the system regularly to make sure it is working properly.
- Consider adding an alarm or other notification system to the system so that you are alerted to a fire even if you are not home.

VIDEO OF THE DEMO :

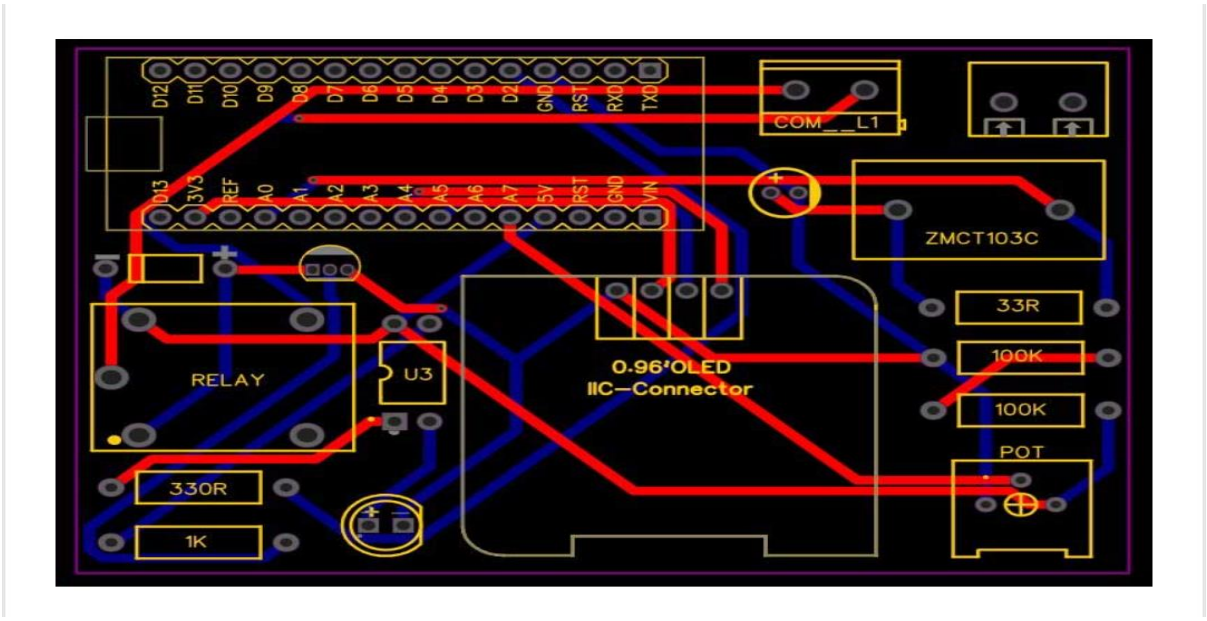
SMOKE DETECTION :

https://drive.google.com/file/d/18W1W7LaGBmbhY03JFZOQ3JYLqMYZNfxZ/view?usp=drivesdk

FIRE DETECTION:

https://drive.google.com/file/d/19-0vMgylL-rnq8RtNHVtdwAnX2AUkfZ6/view?usp=drivesdk

GERBER FILE :



CODE FOR THE SOLUTION :

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>  // You'll need to install FirebaseArduino library

#define FIREBASE_HOST "your-firebase-url"
#define FIREBASE_AUTH "your-firebase-auth-token"
#define WIFI_SSID "your-wifi-ssid"
#define WIFI_PASSWORD "your-wifi-password"

#define SMOKE_SENSOR_PIN A0
#define TEMPERATURE_SENSOR_PIN A1

void setup() {
  Serial.begin(115200);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(250);
    Serial.print(".");
  }
  Serial.println("");

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
  int smokeValue = analogRead(SMOKE_SENSOR_PIN);
  int temperatureValue = analogRead(TEMPERATURE_SENSOR_PIN);

  // Convert analog readings to meaningful values (you may need calibration)
```

```
  float smokePercentage = map(smokeValue, 0, 1023, 0, 100);
  float temperatureCelsius = map(temperatureValue, 0, 1023, 0, 100);

  Serial.print("Smoke Percentage: ");
  Serial.print(smokePercentage);
  Serial.print("%, Temperature: ");
  Serial.print(temperatureCelsius);
  Serial.println("°C");

  // Upload data to Firebase or take necessary actions based on your requirements
  // Firebase.setInt("smokePercentage", smokePercentage);
  // Firebase.setInt("temperatureCelsius", temperatureCelsius);

  delay(5000);  // Adjust the delay based on your requirements
}
```

CONCLUSION :

A fire detection system can be designed and developed using ESP8266, temperature and smoke sensors. The system can be used to detect fire in homes, offices, and other buildings. The system can also be used to monitor the temperature and smoke levels in a building. The system is a low-cost and easy-to-build solution for fire detection