PROBLEM STATEMENT:

Traditional fire alarm systems are often prone to false alarms due to dust, sudden temperature changes, or isolated sensor misfires. This leads to unnecessary panic, emergency service involvement, and loss of trust in the system. Additionally, most systems lack real-time remote alerting and cannot detect multiple fire indicators. There is a growing need for an intelligent, **multi-sensor, IoT-enabled fire alarm system** that ensures accurate fire detection, minimizes false alarms, and alerts users remotely through a mobile app.

OBJECTIVE:

1. To build a **smart, IoT-based fire alarm system** that detects:
   - Fire or flame (using **Flame Sensor**),
   - Abnormal **temperature and humidity** (using **DHT11/DHT22 Sensor**),
   - Presence of hazardous gases (using **MQ-2/MQ-135 Gas Sensor**).
2. To **minimize false alarms** using:
   - **Multi-sensor validation logic** – alarm is triggered only when at least **two conditions** are met (e.g., flame + gas, or high temp + gas).
   - **Threshold calibration** for temperature and gas concentration levels.
3. To integrate with the **Blynk IoT platform** for:
   - Real-time data monitoring on mobile,
   - Instant push notifications on fire detection,
   - System status visualization and reset/control options.
4. To provide a **low-cost, scalable solution** suitable for homes, industries, and public spaces.

DIFFERENCE BETWEEN THE SMART FIRE ALERT SYSTEM AND TRADITIONAL SYSTEMS:

| Aspect | Traditional Fire Alarm System | IoT-Based Multi-Sensor Fusion System |
|---|---|---|
| Sensor Type | Single sensor (e.g., smoke or heat sensor) | Multiple sensors: Flame, Gas (MQ), and Temperature/Humidity (DHT11) |
| Accuracy | Prone to false alarms due to single input | Higher accuracy with combined sensor inputs (sensor fusion) |

| | | |
|---|---|---|
| Real-Time Monitoring | No remote access | Real-time monitoring via Blynk app |
| Alert Mechanism | Local buzzer or siren only | Buzzer + Smartphone alert via Blynk |
| False Alarm Prevention | Not available | Sensor fusion and weighted logic minimize false alerts |
| Smart Decision Logic | None | Logical decision based on multiple thresholds and sensor weights |
| User Interface | Basic or no interface | Mobile dashboard with visual readings |
| Scalability and Customization | Fixed and hard to upgrade | Easily customizable and scalable with IoT and Arduino |

HARDWARE COMPONENTS:

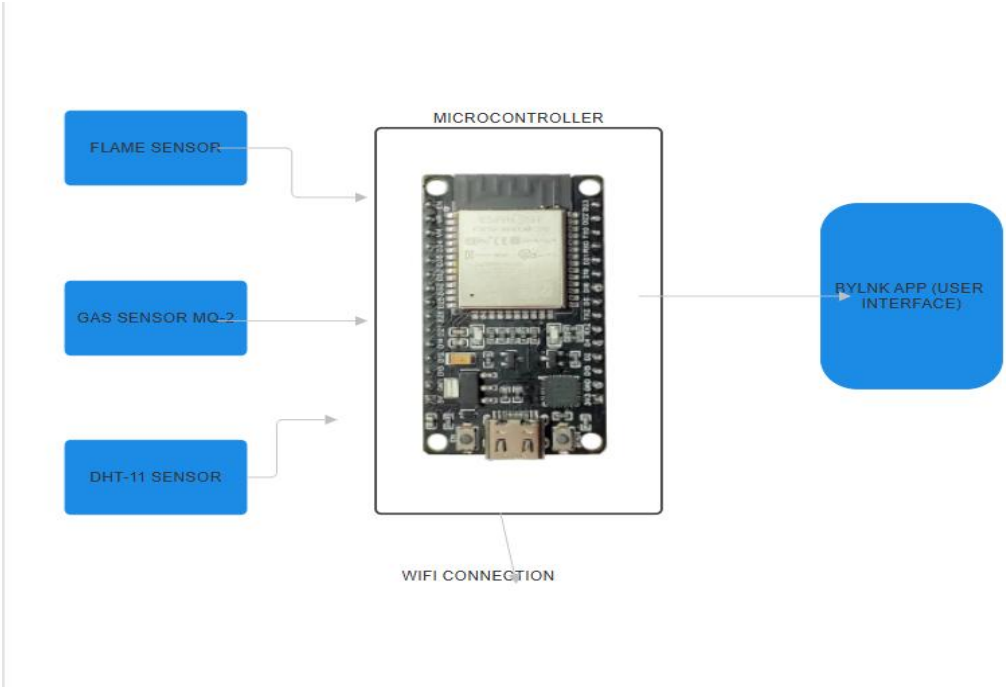
| Components | Description/purpose |
|---|---|
| Flame Sensor | Detects the presence of flame or fire through infrared light. |
| DHT11 Sensor | Measures temperature and humidity in the surrounding environment. |
| Gas Sensor | Detects gases like LPG, methane, carbon monoxide (CO), and smoke. |
| NodeMCU ESP32 | Main microcontroller with built-in Wi-Fi for connecting sensors and sending data to Blynk. |
| Power Supply / USB Cable | Provides power to the NodeMCU board (typically 5V via USB). |
| Jumper Wires | For making electrical connections between components. |
| Breadboard | Used to build and test the circuit without soldering. |
| Buzzer | Provides an audible alarm when fire or gas is detected. |

SOFTWARE COMPONENTS:

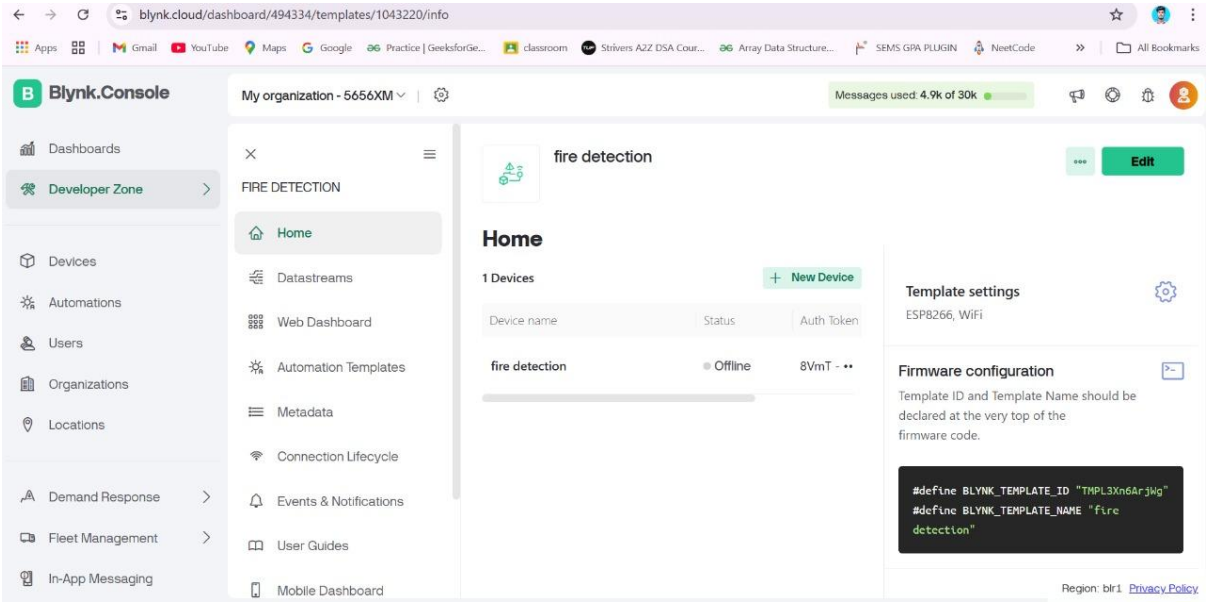
| Components | Description |
|---|---|
| Arduino IDE | Used to write, compile, and upload code to the NodeMCU microcontroller. |
| Blynk App (Mobile) | IoT platform that displays real-time sensor data and sends alerts to the user. |

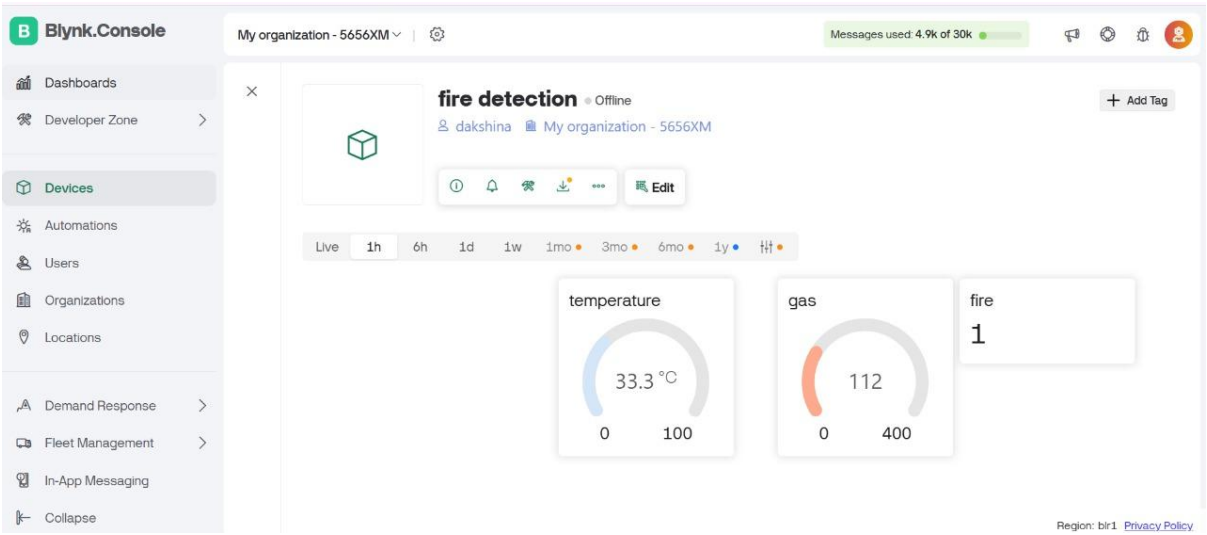
| | |
|---|---|
| Wi-Fi Router | Provides internet access to the NodeMCU for sending data to Blynk server. |
| Blynk Library | Arduino library required to connect NodeMCU to the Blynk cloud platform. |
| Blynk Dashboard UI | Mobile interface with widgets like gauge, graph, notification, and button. |

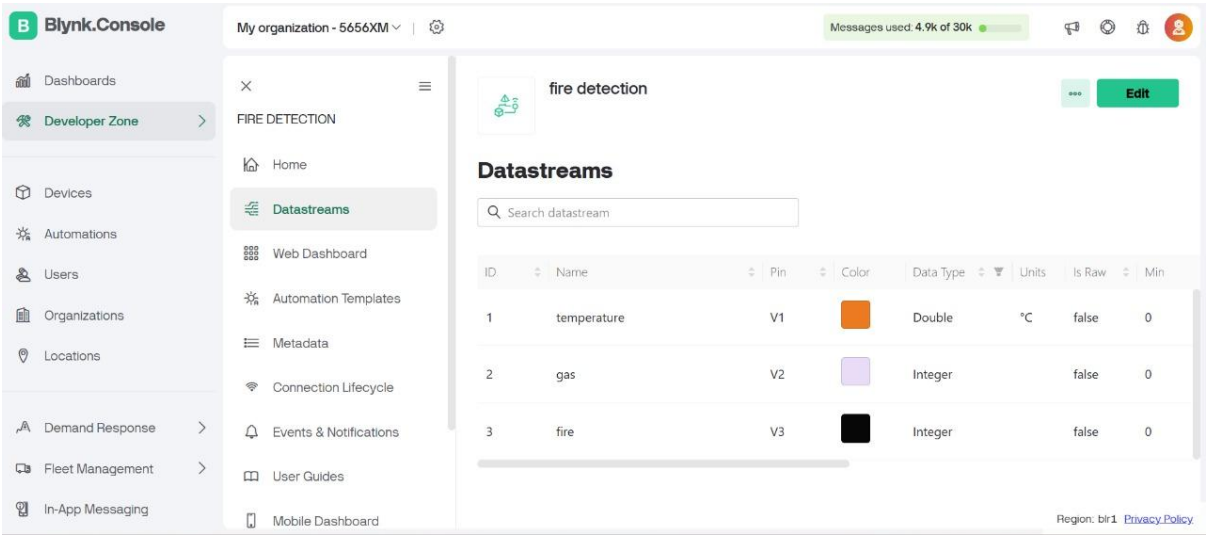
MODEL ARCHIECTURE:




BLYNK INTERFACE:






THRESHOLD VALUES:

| Sensor | Parameter | Threshold Value | Condition |
|---|---|---|---|
| Flame Sensor | Digital (0 or 1) | 1 = Fire detected | If Flame = 1 → Fire present |
| DHT11 Sensor | Temperature | > 50°C | Abnormally high temperature |
| DHT11 Sensor | Humidity | > 80% (optional) | Very high humidity (optional) |
| MQ Gas Sensor | Analog (0–1023) | > 400 | Gas leakage or smoke detected |

WEIGHTS ASSIGNEMENT AND LOGIC :

| Sensor Condition | Weight |
|---|---|
| Flame Detected | 0.5 |

| | |
|---|---|
| Gas Level > 400 | 0.3 |
| Temperature > 50°C | 0.2 |

The output calculation:

Score = (Flame * 0.5) + (GasCondition * 0.3) + (TempCondition * 0.2)

Where:

Flame = 1 if flame detected, else 0

GasCondition = 1 if gas level > threshold, else 0

TempCondition = 1 if temperature > threshold, else 0

The decision rule:

IF Score >= 0.7:

=> Fire Alarm = TRUE (Trigger Blynk Alert & Buzzer)

ELSE:

=> No Alarm (Possibly false trigger)

ALGORITHM:

1. **Start**
2. **Initialize** all sensors and Blynk connection.
3. **Loop begins**:
    - Read **Flame sensor** value → F
    - Read **Temperature** from DHT11 → T
    - Read **Gas sensor** value → G
4. Set conditions:
    - If F == 1, set FlameCondition = 1, else 0
    - If T > 50, set TempCondition = 1, else 0
    - If G > 400, set GasCondition = 1, else 0
5. **Calculate Score**:

Score = (FlameCondition * 0.5) + (GasCondition * 0.3) + (TempCondition * 0.2)

6. If Score ≥ 0.7:
    - Activate **buzzer**
    - Send **alert via Blynk**
7. Else:
    - No action (prevent false alarm)
8. **Repeat loop**

ARDUINO CODE:

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
char auth[] = " b1c2d3e4f5g6h7i8j9k0l1m2";
char ssid[] = " MonicaHomeWiFi";
char pass[] = "Monica@6#";
#define DHTPIN D3
#define DHTTYPE DHT11
#define FLAME_PIN D2
#define GAS_PIN A0
#define BUZZER_PIN D4
const int GAS_THRESHOLD = 400;
const float TEMP_THRESHOLD = 50.0;
const float FLAME_WEIGHT = 0.5;
const float GAS_WEIGHT = 0.3;
const float TEMP_WEIGHT = 0.2;
const float ALERT_THRESHOLD = 0.7;
DHT dht(DHTPIN, DHTTYPE);
bool alertActive = false;
void setup() {
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  dht.begin();
```

```
  pinMode(FLAME_PIN, INPUT);
  pinMode(GAS_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW);
}
void loop() {
  Blynk.run();
  int flame = digitalRead(FLAME_PIN);
  int gas = analogRead(GAS_PIN);
  float temp = dht.readTemperature();
  if (isnan(temp)) return;
  int flameCondition = (flame == 1) ? 1 : 0;
  int gasCondition = (gas > GAS_THRESHOLD) ? 1 : 0;
  int tempCondition = (temp > TEMP_THRESHOLD) ? 1 : 0;
  float score = (flameCondition * FLAME_WEIGHT) +
                (gasCondition * GAS_WEIGHT) +
                (tempCondition * TEMP_WEIGHT);

  Serial.print("Score: ");
  Serial.println(score);
  if (score >= ALERT_THRESHOLD && !alertActive) {
    digitalWrite(BUZZER_PIN, HIGH);
    Blynk.notify("⚠️ Fire Alert! Check your premises!");
    alertActive = true;
  }
  else if (score < ALERT_THRESHOLD && alertActive) {
    digitalWrite(BUZZER_PIN, LOW);
    alertActive = false;
  }
  delay(2000);
}
```

CONCLUSION:

The Smart Fire Alarm System developed using IoT technologies—integrating **Flame Sensor**, **DHT11 (temperature and humidity sensor)**, and **MQ Gas Sensor**, along with **Blynk App connectivity**—provides a reliable, real-time fire detection and alerting solution. By setting threshold values and incorporating weighted logic to minimize false alarms, the system ensures accurate hazard detection. The integration with **Blynk** enables live monitoring, remote alerts, and real-time updates on fire, gas, and temperature status, enhancing safety and responsiveness. With the addition of a **buzzer for local alerts**, the system serves as an effective early warning mechanism, ideal for homes, labs, and commercial spaces.