EMBEDDED FIRE DETECTION AND ALARM NOTIFICATION SYSTEM

TEAM MEMBERS:

DHARSHINI S S

ANGELIN PRESENCIA P

GAYATHRI D

ANANYA C V

AIM:

The aim of the project "Embedded Fire Detection and Alarm Notification System" is to design and implement a smart, low-cost, user-friendly system that can rapidly and accurately detect fires at an early stage, issue timely alerts, and minimize the risk of property loss and harm to individuals through automatic notifications and alarms.

COMPONENTS REQUIRED:

- ESP32 (microcontroller)
- Flame sensor
- Smoke sensor
- Temperature sensor(DHT22)
- Buzzer
- LED

ESP32 (MICROCONTROLLER):

- Dual-core 32-bit processor, 240 MHz.
- Wi-Fi and Bluetooth built-in.
- Operates 2.7V–3.6V, -40°C to 85°C.

FLAME SENSOR(YG1006):

- Detects IR light (760–1100 nm).
- Flame detection range: up to 0.8–1 meter.
- Detection angle: ~60°, sensitivity adjustable.

SMOKE SENSOR:

- Detects smoke and flammable gases.
- Detection range: 300–10,000 ppm.
- Operating voltage: 5V (use level shifter with ESP32).

TEMPERATURE SENSOR (DHT22):

- Measures temperature from -40°C to 80°C (accuracy: ±0.5°C).
- Measures humidity from 0% to 100% RH (accuracy: ±2% RH).
- Operating voltage: 3.3V to 6V, digital output.

BUZZER:

- Provides audible alert (sound).
- Typical voltage: 3–12V.
- Sound: 80–100dB at 10cm.

AVAILABLE COMPONENTS:

- ESP32 (Microcontroller)
- Breadboard
- Red led
- Piezo buzzer
- Resistor
- DHT11
- Flame sensor

PIN CONFIGURATION:

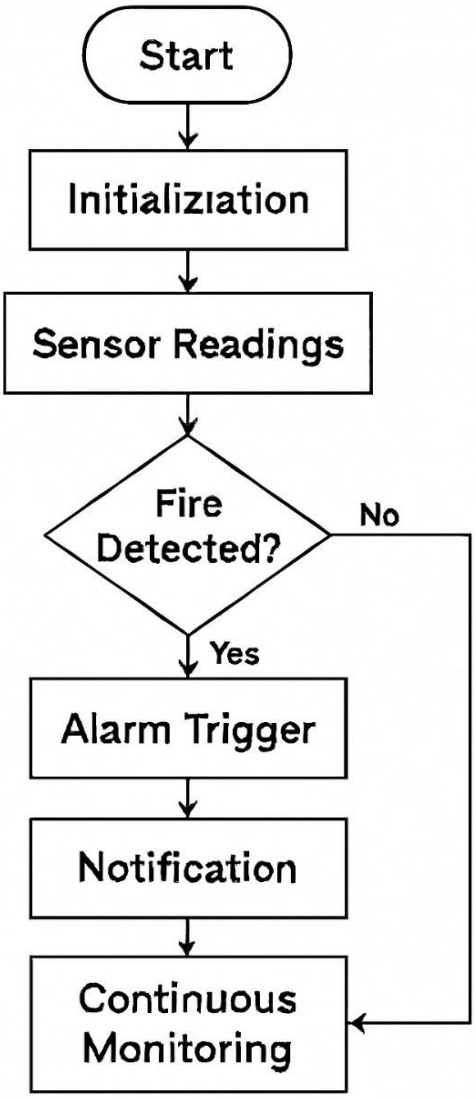
COMPONENTS	**ESP32**
DHT11: VCC GND DATA	 3.3V GND GPIO4
FLAME SENSOR: VCC GND DO	 3.3V GND GPIO5
RED LED: ANODE CATHODE	 GPIO2 GND
PIEZO BUZZER: POSITIVE NEGATIVE	 GPIO18 GND

PROCEDURE:

- Connect the flame sensor to GPIO27, MQ-2 to GPIO34, DHT11 to GPIO4, buzzer to GPIO18, and LED to GPIO19 of ESP32, with a common ground.
- Install Arduino IDE, add ESP32 board support, and install the required libraries (DHT, Blynk if IoT is used).

- Select ESP32 board in Arduino IDE and upload the fire detection code.
- Power ON the system and open Serial Monitor to observe sensor readings.
- When fire, smoke, or high temperature is detected, the buzzer and LED are activated.
- In IoT version, a notification is also sent to the mobile app (Blynk).
- Once the fire/smoke is removed, the system resets and the alarm turns off automatically.

FLOWCHART:

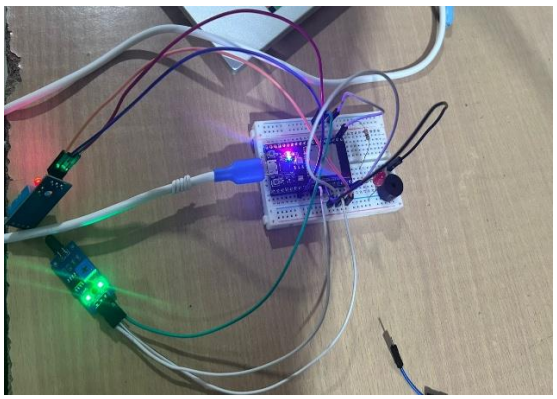


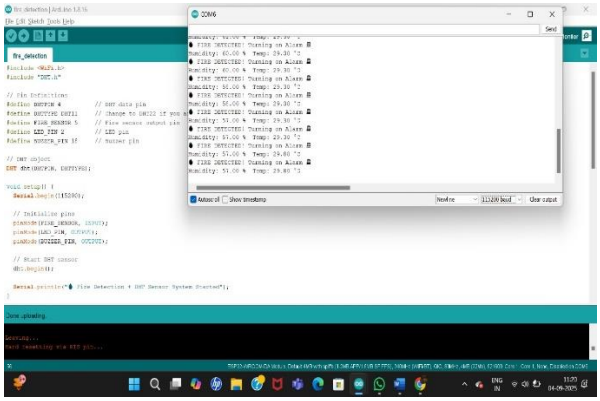
PROGRAM:

```
#include <WiFi.h>
#include "DHT.h"
#define DHTPIN 4          // DHT data pin
#define DHTTYPE DHT11     // Change to DHT22 if you are using it
#define FIRE_SENSOR 5     // Fire sensor output pin
#define LED_PIN 2         // LED pin
#define BUZZER_PIN 18     // Buzzer pin
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(115200);
  pinMode(FIRE_SENSOR, INPUT);
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  dht.begin();
  Serial.println("Fire Detection + DHT Sensor System Started");
}
void loop() {
  // Read temperature & humidity
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  // Check fire sensor
  int fireStatus = digitalRead(FIRE_SENSOR);
  if (t >= 29) {
    // Fire detected
    Serial.println(" FIRE DETECTED! Turning on Alarm ");
    digitalWrite(LED_PIN, HIGH);
```

```
    digitalWrite(BUZZER_PIN, HIGH);
  } else {
    // No fire
    digitalWrite(LED_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);
  }
  if (!isnan(h) && !isnan(t)) {
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print(" %  Temp: ");
    Serial.print(t);
    Serial.println(" °C");
  } else {
    Serial.println(" Failed to read from DHT sensor!");
  }
  delay(1000); // Update every second
}
```

EXECUTION:





RESULT:

The experiment demonstrated that the ESP32-based fire detection system effectively detects fire and smoke in real time, providing rapid alerts with sound and visual alarms. The system's IoT capability enables timely remote notifications, enhancing overall safety and response speed.