# Embedded Fire Detection and Alarm Notification System

***TEAM MEMBER:***

JOVITHA S

MAADHINI K

## AIM:

"To design and implement an embedded fire detection and alarm notification system using ESP32, temperature and flame sensors, buzzer, and LED, which can monitor environmental conditions in real time, detect fire hazards quickly, and provide immediate audio-visual alerts to ensure safety."

## COMPONENTS REQUIRED:

ESP-WROOM-32 module / dev board (3.3V I/O).

DS18B20 Temperature Sensor (digital, 1-wire).

Flame Sensor Module (digital output).

Active Buzzer (5V type, driven via transistor or directly if 3.3V rated).

LED + 220Ω resistor.

4.7kΩ resistor (for DS18B20 pull-up)

## COMPONENTS SPECIFICATION:

### ESP-WROOM-32 (ESP32 Board)

Operating Voltage: 3.3V (powered via 5V USB).

Dual-core 32-bit processor with Wi-Fi + Bluetooth.

DS18B20 Temperature Sensor

Range: –55 °C to +125 °C with ±0.5 °C accuracy.

Digital 1-Wire output, works on 3.0–5.5V.

### Flame Sensor Module

Detects IR light (760–1100 nm) from flames.

Provides Digital (D0) and Analog (A0) outputs.

### Active Buzzer

Operates on 3.3–5V with ~85 dB sound output.

Needs only ON/OFF signal to produce sound.

### LED (Red)

Forward Voltage: ~2V, Current: 20 mA.

Used with 220Ω resistor for safe operation.

### Resistors

4.7kΩ (pull-up for DS18B20).

220Ω (current limiting for LED).

### PIN CONFIGURATION:

| Temperature sensor | ESP32 |
|---|---|
| VCC | 3.3V |
| GND | GND |
| Data | GPIO21 |

| Flame Sensor | ESP32 |
|---|---|
| VCC | 3.3V |
| GND | GND |
| DO | GPI035 |

| Buzzer | ESP32 |
|---|---|
| Positive | GPIO25 |
| Negative | GND |

| LED | ESP32 |
|---|---|
| Anode | GPIO26 |
| Cathode | GND |

**Procedure:**

Collect all required components – ESP-WROOM-32, DS18B20 temperature sensor, flame sensor, buzzer, LED with resistor, breadboard, and jumper wires

Connect the DS18B20 sensor to ESP32: VCC → 3.3V, GND → GND, Data → GPIO21 with a 4.7kΩ pull-up resistor.

Connect the flame sensor: VCC → 3.3V (or 5V if supported), GND → GND, D0 → GPIO34, and optionally A0 → GPIO35 for analog intensity.

Connect the buzzer: Positive terminal to GPIO25 and negative terminal to GND (via transistor if using a 5V buzzer).

Connect the LED: Anode through a 220Ω resistor to GPIO26, cathode to GND.

Power the ESP32 through USB or a regulated 5V power supply.

Open Arduino IDE, install ESP32 board support, and add DallasTemperature & OneWire libraries for the DS18B20 sensor.

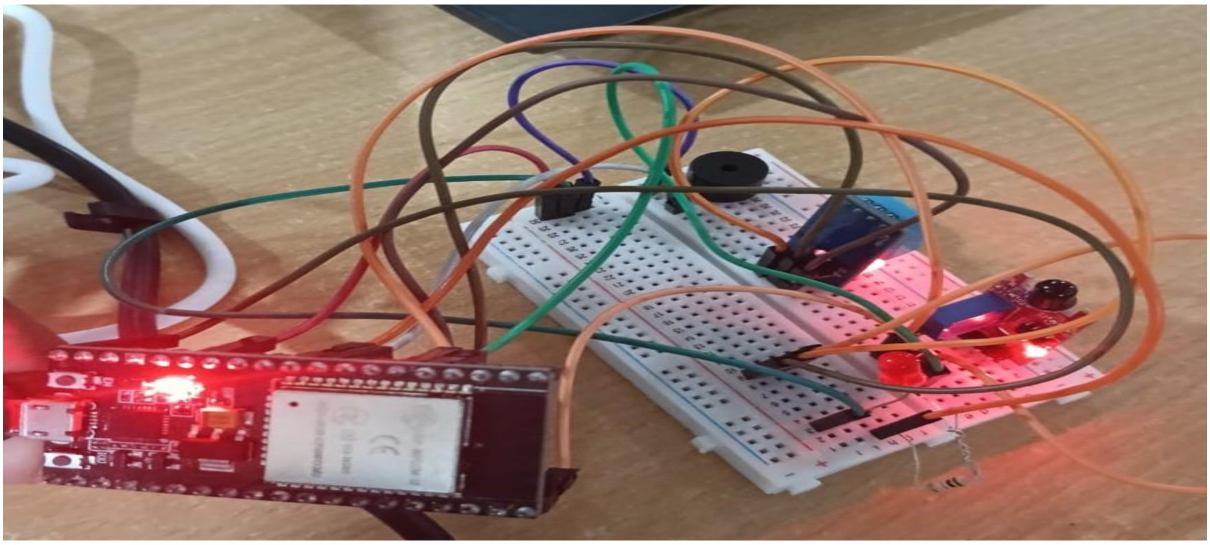
Upload the program to read sensor values, compare them with thresholds, and trigger the buzzer and LED when a fire condition is detected.

Test the system by applying safe heat near the temperature sensor and a small flame near the flame sensor.

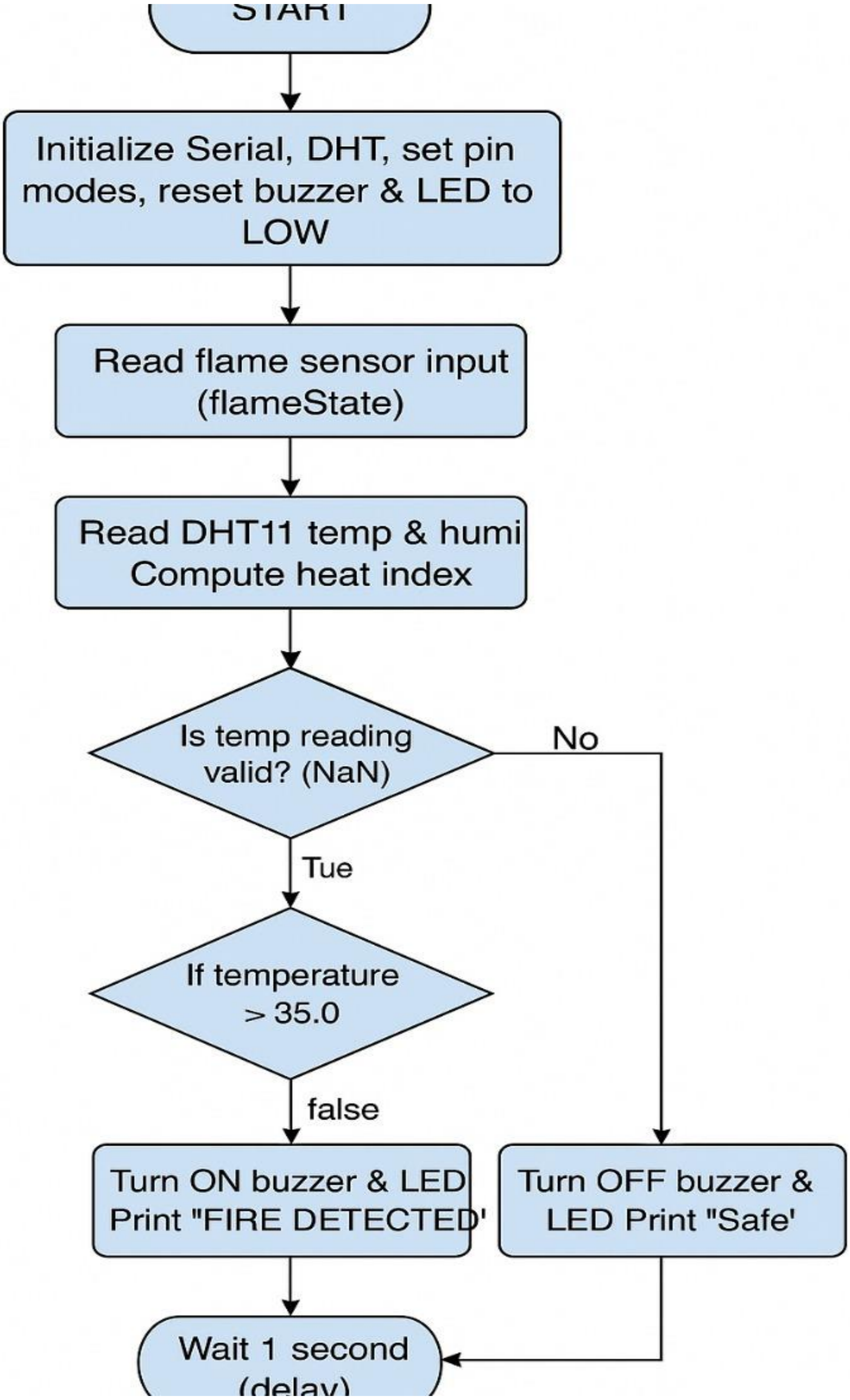
Observe the readings on the Serial Monitor and check whether the buzzer and LED activate during fire conditions.

Calibrate the flame sensor sensitivity using its onboard potentiometer and adjust the temperature threshold in code for accurate detection.

## CIRCUIT CONNECTION:



**FLOWCHAT**


START
Initialize Serial, DHT, set pin modes, reset buzzer & LED to LOW
Read flame sensor input (flameState)
Read DHT11 temp & humi Compute heat index
Is temp reading valid? (NaN)
No
Tue
If temperature > 35.0
false
Turn ON buzzer & LED Print "FIRE DETECTED'
Turn OFF buzzer & LED Print "Safe'
Wait 1 second (delay)

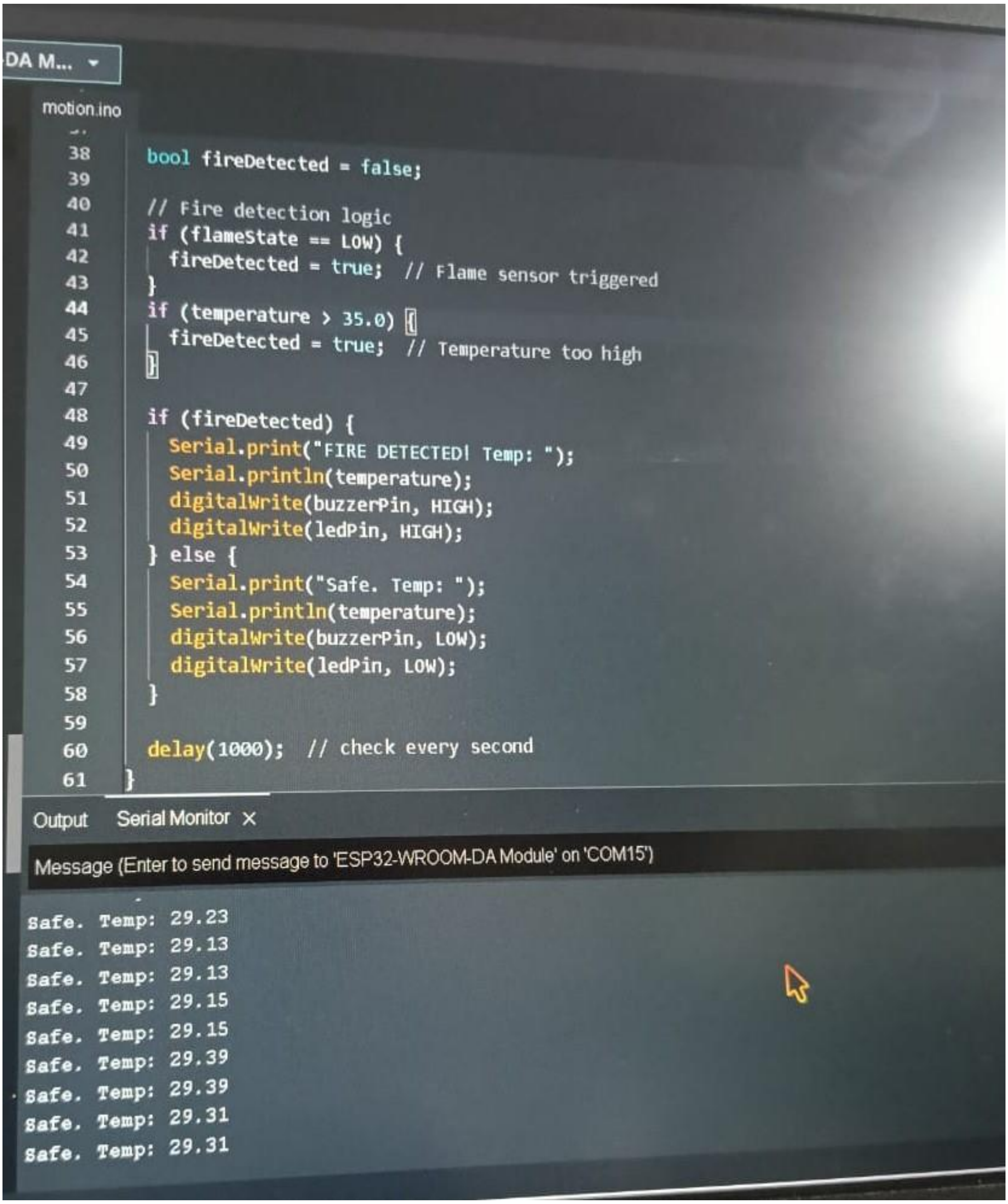
**PROGRAM:**

```
#include "DHT.h"

#define DHTPIN 21

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

const int flameSensorPin = 35;

const int buzzerPin = 25;

const int ledPin = 26;

void setup() {

  pinMode(flameSensorPin, INPUT);

  pinMode(buzzerPin, OUTPUT);

  pinMode(ledPin, OUTPUT);

  digitalWrite(buzzerPin, LOW);

  digitalWrite(ledPin, LOW);

  Serial.begin(115200);

  dht.begin();

}

void loop() {

  int flameState = digitalRead(flameSensorPin);

  float t = dht.readTemperature();

  // Read temperature as Fahrenheit (isFahrenheit = true)

  float h = dht.readHumidity();

  float temperature = dht.computeHeatIndex(t, h, false);

  if (isnan(temperature)) {

    Serial.println("Failed to read from DHT sensor!");

    delay(500);

    return;

  }

  bool fireDetected = false;

  if (flameState == LOW) {

    fireDetected = true;  // Flame sensor triggered
```

```
 }
 if (temperature > 35.0) {
   fireDetected = true;  // Temperature too high
 }
 if (fireDetected) {
   Serial.print("FIRE DETECTED! Temp: ");
   Serial.println(temperature);
   digitalWrite(buzzerPin, HIGH);
   digitalWrite(ledPin, HIGH);
 } else {
   Serial.print("Safe. Temp: ");
   Serial.println(temperature);
   digitalWrite(buzzerPin, LOW);
   digitalWrite(ledPin, LOW);
 }
 delay(1000);  // check every second
}
```

## EXECUTION:

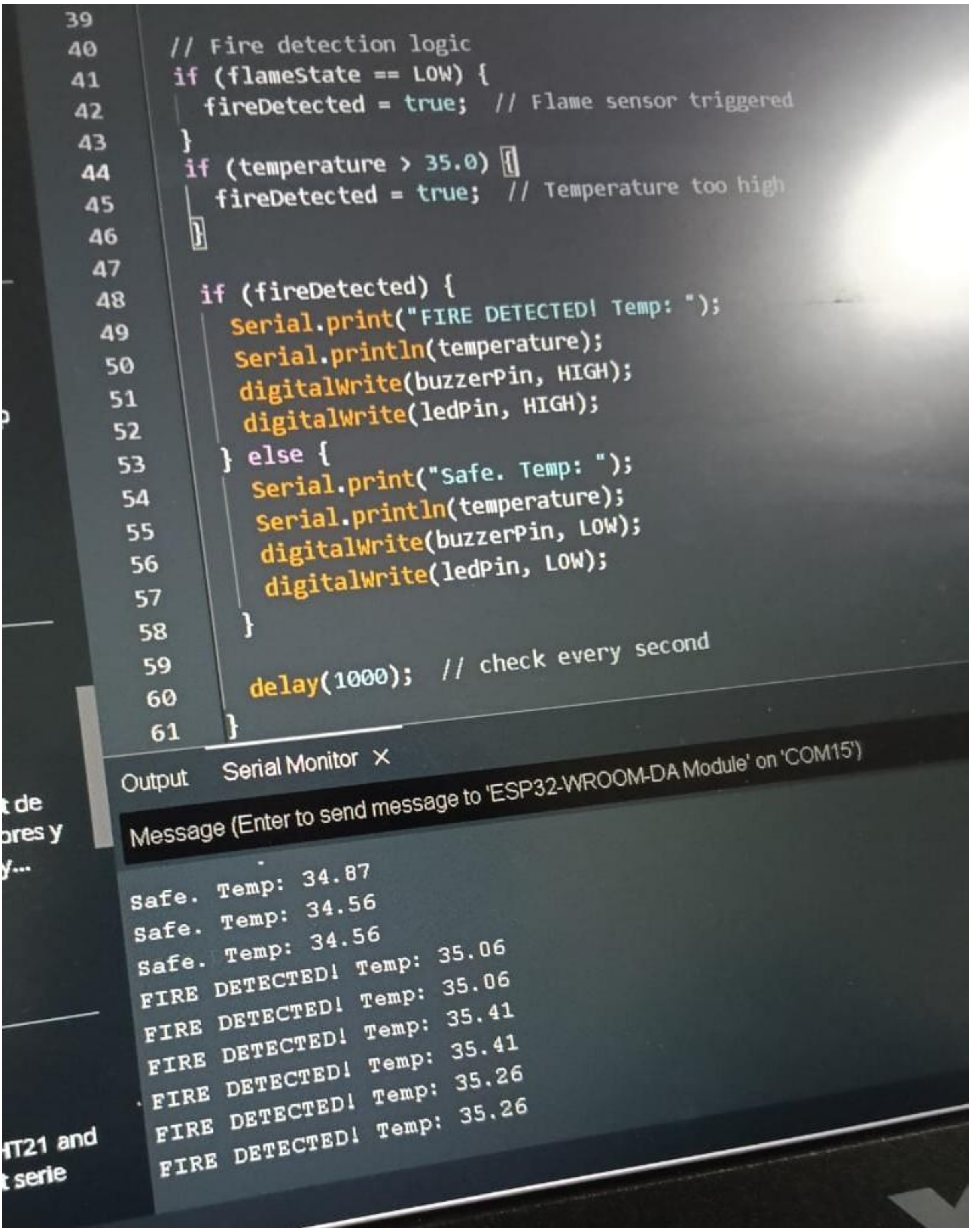
The fire detection and alarm system is executed by interfacing a DHT11 temperature sensor, a flame sensor, a buzzer, and an LED with the ESP32. When the system is powered on, the ESP32 initializes all sensors and peripherals. It continuously monitors the flame sensor and reads the temperature and humidity values from the DHT11. If the flame sensor detects fire (logic LOW) or the temperature rises above 35 °C, the system identifies it as a fire condition. In such a case, the buzzer is activated and the LED turns on, providing both sound and light alerts. If no fire or abnormal temperature is detected, the buzzer and LED remain off, and the system indicates a safe condition. This cycle repeats every second, ensuring real-time fire monitoring



```
  bool fireDetected = false;

  // Fire detection logic
  if (flameState == LOW) {
    fireDetected = true;  // Flame sensor triggered
  }
  if (temperature > 35.0) {
    fireDetected = true;  // Temperature too high
  }

  if (fireDetected) {
    Serial.print("FIRE DETECTED! Temp: ");
    Serial.println(temperature);
    digitalWrite(buzzerPin, HIGH);
    digitalWrite(ledPin, HIGH);
  } else {
    Serial.print("Safe. Temp: ");
    Serial.println(temperature);
    digitalWrite(buzzerPin, LOW);
    digitalWrite(ledPin, LOW);
  }

  delay(1000);  // check every second
}
```

Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM15')

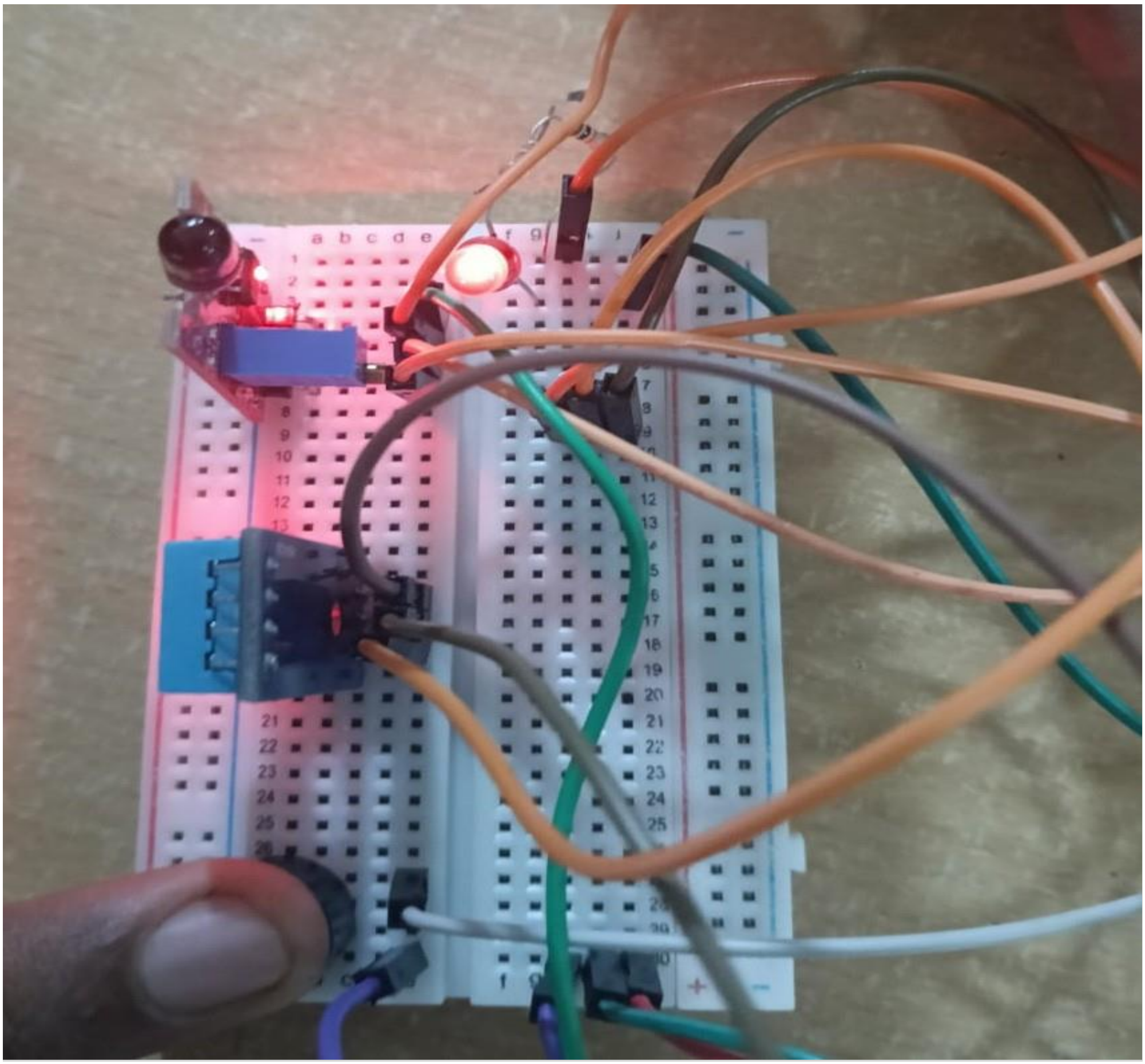
```
Safe. Temp: 29.23
Safe. Temp: 29.13
Safe. Temp: 29.13
Safe. Temp: 29.15
Safe. Temp: 29.15
Safe. Temp: 29.39
Safe. Temp: 29.39
Safe. Temp: 29.31
Safe. Temp: 29.31
```



```
  // Fire detection logic
  if (flameState == LOW) {
    fireDetected = true;  // Flame sensor triggered
  }
  if (temperature > 35.0) {
    fireDetected = true;  // Temperature too high
  }

  if (fireDetected) {
    Serial.print("FIRE DETECTED! Temp: ");
    Serial.println(temperature);
    digitalWrite(buzzerPin, HIGH);
    digitalWrite(ledPin, HIGH);
  } else {
    Serial.print("Safe. Temp: ");
    Serial.println(temperature);
    digitalWrite(buzzerPin, LOW);
    digitalWrite(ledPin, LOW);
  }

  delay(1000);  // check every second
}
```

Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM15')

```
Safe. Temp: 34.87
Safe. Temp: 34.56
Safe. Temp: 34.56
FIRE DETECTED! Temp: 35.06
FIRE DETECTED! Temp: 35.06
FIRE DETECTED! Temp: 35.41
FIRE DETECTED! Temp: 35.41
FIRE DETECTED! Temp: 35.26
FIRE DETECTED! Temp: 35.26
```



## RESULT:

The system successfully detected fire and high temperature conditions using the flame sensor and DHT11. During detection, the buzzer produced sound and the LED provided a visual alert. In safe conditions, both the buzzer and LED remained off, ensuring reliable monitoring and timely warning of fire hazards.