Embedded Fire Detection and Alarm notification system

TEAM MEMBERS:

Hilary Royson CB

Nitish Kumar J

Allan christ B

Ahash J

AIM:

To design and develop an embedded fire detection and alarm notification system using ESP32 that continuously monitors for fire or smoke and triggers an alarm in real time to ensure safety.

COMPONENTS REQUIRED:

1. ESP-32
2. MQ-2 Fire/Smoke Sensor
3. OLED
4. LED
5. Resistor
6. Breadboard

ESP-8266:

The ESP32 is a powerful and versatile microcontroller developed by Espressif Systems. It features built-in Wi-Fi and Bluetooth, making it ideal for Internet of Things (IoT) applications. The chip includes dual-core processing**,** ample memory, and supports multiple communication protocols like UART, SPI, and I²C**.**

MQ-2 Sensor:

A flame or smoke sensor that detects flammable gases and smoke particles in the environment. It outputs analog signals that indicate the presence and intensity of smoke or flame.

OLED:

An energy-efficient screen that visually shows information such as turbidity levels and alerts. It communicates with the microcontroller using the I2C protocol and requires minimal power.

LED:

A light-emitting component that signals changes in water quality. When turbidity crosses a set threshold, the LED glows to alert users of possible contamination.

Resistor:

A passive component used to limit the current flowing to the LED, ensuring that it operates safely without overheating or burning out.

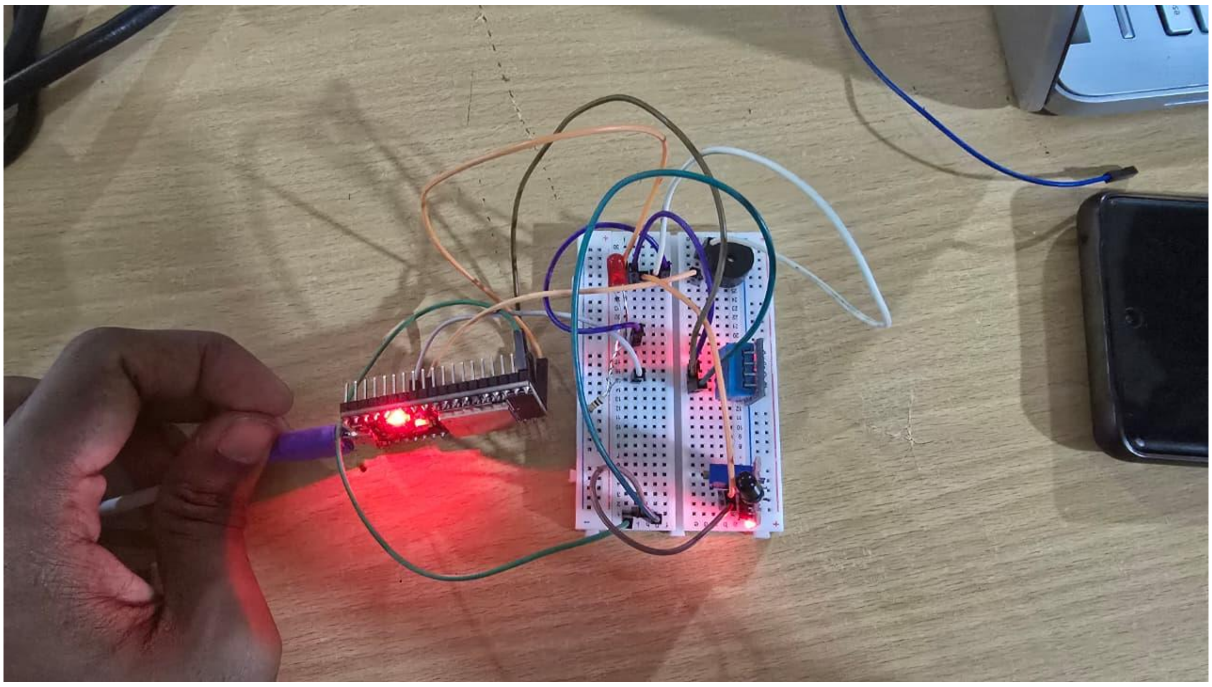
Breadboard:

A reusable prototyping board that allows electronic components to be connected without soldering. It is mainly used for testing, learning, and quick modification of circuits

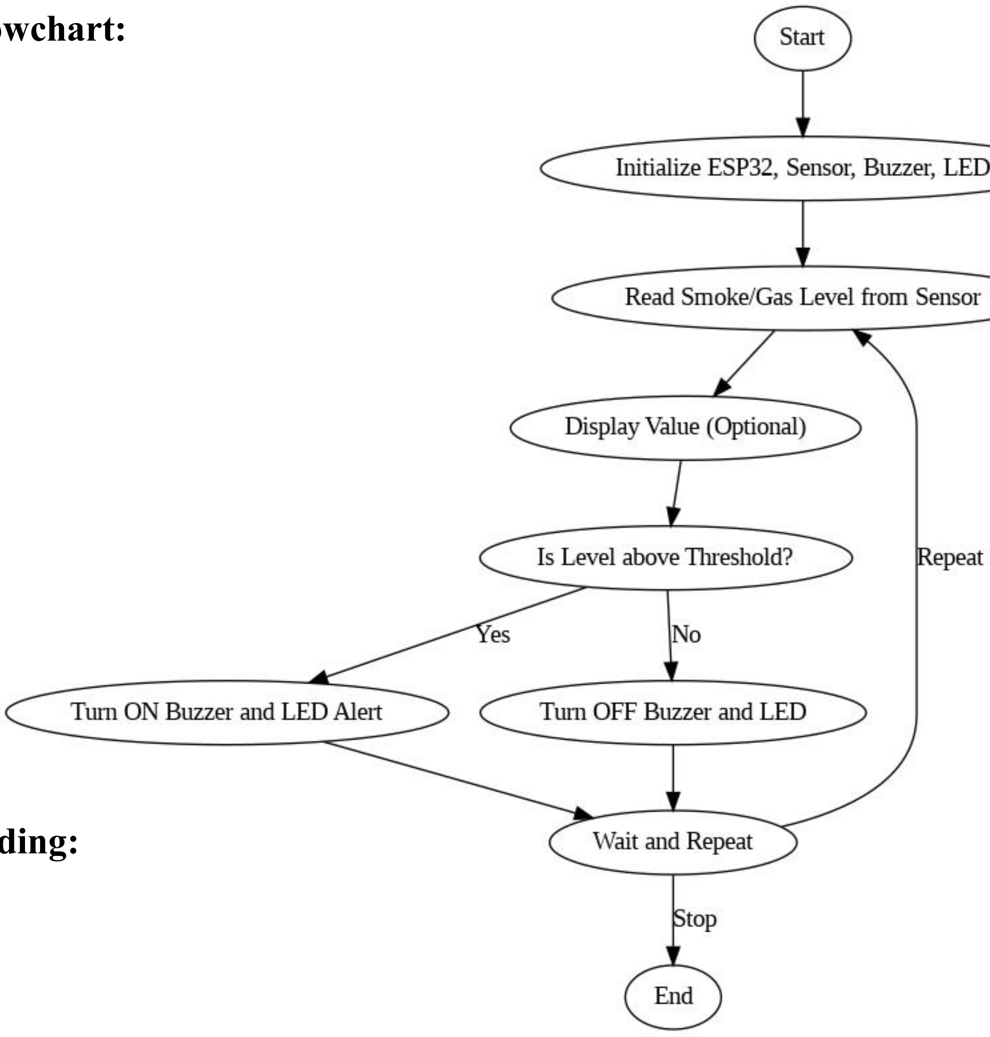
Pin Table:

Component	**Pin on Component**	**ESP32 Pin**	**Description**
MQ-2 Sensor	VCC	3.3V or 5V	Power supply to the sensor
MQ-2 Sensor	GND	GND	Ground connection
MQ-2 Sensor	A0 (analog out)	GPIO 34 (ADC)	Reads analog value indicating smoke level
Buzzer	+ (positive)	GPIO 15	Digital output to activate buzzer
Buzzer	– (negative)	GND	Ground connection
LED	+	GPIO 2	Digital output to control LED
LED	–	GND	Ground connection
OLED (opt.)	VCC	3.3V	Power supply to the display
OLED (opt.)	GND	GND	Ground connection

Circuit Connection:



Flowchart:


Start
Initialize ESP32, Sensor, Buzzer, LED
Read Smoke/Gas Level from Sensor
Display Value (Optional)
Is Level above Threshold?
Repeat
Yes
No
Turn ON Buzzer and LED Alert
Turn OFF Buzzer and LED
Wait and Repeat
Stop
End

Coding:

```
// Define Pins
const int fireSensorPin = 4; // GPIO4 connected to fire sensor OUT
const int buzzerPin = 13;    // GPIO13 connected to buzzer +
const int ledPin = 12;       // GPIO12 connected to LED +

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

  // Initialize pins
  pinMode(fireSensorPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);

  // Initially turn off buzzer and LED
  digitalWrite(buzzerPin, LOW);
  digitalWrite(ledPin, LOW);

  Serial.println("Fire Detection System Initialized");
}

void loop() {
  int fireState = digitalRead(fireSensorPin);

  if (fireState == HIGH) {  // Fire detected
    Serial.println("    Fire Detected!");
    digitalWrite(buzzerPin, HIGH); // Turn ON buzzer
    digitalWrite(ledPin, HIGH);    // Turn ON LED
  } else {
    Serial.println("✅ No Fire");
    digitalWrite(buzzerPin, LOW);  // Turn OFF buzzer
    digitalWrite(ledPin, LOW);     // Turn OFF LED
  }

  delay(1000); // Check every 1 second
}
```

Execution: