**🔥 Air Alarm System – Detailed Documentation**

**✅ 1. Overview**

This project is a **Smoke and Air Detection Alarm System** using:

* **MQ2 Gas Sensor** to detect smoke/gas levels
* **Flame Sensor** (used as air/smoke detection)
* **Buzzer** to alert with sound
* **16x2 LCD with I2C** for visual feedback

When smoke exceeds a threshold or flame sensor detects smoke particles, the system triggers an alarm, displays warnings on the LCD, and logs values via Serial Monitor.

**📋 2. Components Used**

| **Component** | **Quantity** | **Description** |
| --- | --- | --- |
| Arduino Uno | 1 | Main microcontroller board |
| MQ2 Gas Sensor | 1 | Detects smoke, LPG, CO, methane etc. |
| Flame Sensor (DOUT) | 1 | Digital output sensor, used here for smoke/air |
| Buzzer | 1 | Emits beeping alert sound |
| 16x2 LCD (I2C) | 1 | Displays gas value and system status |
| Breadboard + Jumper Wires | As needed | For connections |
| USB Cable | 1 | Upload code + Serial Monitor |

**🔌 3. Circuit Diagram / Pin Configuration**

**(URL:-** [**https://app.cirkitdesigner.com/project/be00edef-4975-4b14-8539-0749b613fc7e**](https://app.cirkitdesigner.com/project/be00edef-4975-4b14-8539-0749b613fc7e)**)**

+---------------------+

| Arduino Uno |

| |

| [A0] <----------+ MQ2 Sensor (Analog Out)

| [D2] <----------+ Flame Sensor (Digital Out)

| [D5] ---------->+ Buzzer

| |

| SDA [A4]----------> I2C LCD (SDA)

| SCL [A5]----------> I2C LCD (SCL)

| |

+---------------------+

* **MQ2 Sensor:**
  + VCC → 5V
  + GND → GND
  + AOUT → A0
* **Flame Sensor:**
  + VCC → 5V
  + GND → GND
  + D0 → D2
* **Buzzer:**
  + Positive leg → D5
  + Negative leg → GND
* **I2C LCD:**
  + VCC → 5V
  + GND → GND
  + SDA → A4
  + SCL → A5

**🧾 Libraries You Need**

* LiquidCrystal\_I2C

***Install them via the Arduino Library Manager (Sketch → Include Library → Manage Libraries).***

**🧰 4. Arduino Code**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

// Fire & Gas Leakage Detection System

LiquidCrystal\_I2C lcd(0x27, 16, 2);

const int gasSensorPin = A0;

const int flameSensorPin = 2;

const int buzzerPin = 5;

const int gasThreshold = 400;

void setup() {

pinMode(gasSensorPin, INPUT);

pinMode(flameSensorPin, INPUT);

pinMode(buzzerPin, OUTPUT);

Serial.begin(9600);

Serial.println("SMOKE Alarm System Starting...");

lcd.init();

lcd.backlight();

lcd.setCursor(0, 0);

lcd.print("System Initializing");

delay(2000);

lcd.clear();

}

void loop() {

int gasValue = analogRead(gasSensorPin);

int airStatus = digitalRead(flameSensorPin);

Serial.print("Gas: ");

Serial.print(gasValue);

Serial.print(" | Air\uD83D\uDCA8: ");

Serial.println(airStatus == LOW ? "DETECTED" : "Safe\u2705");

if (gasValue > gasThreshold || airStatus == LOW) {

lcd.clear();

lcd.setCursor(0, 0);

if (gasValue > gasThreshold) {

lcd.print("!! SMOKE LEAK !!");

Serial.println("SMOKE LEAK DETECTED");

} else {

lcd.print("\*\* SMOKE DETECTED \*\*");

Serial.println("SMOKE DETECTED");

}

lcd.setCursor(0, 1);

lcd.print("Evacuate Now! ");

for (int i = 0; i < 10; i++) {

digitalWrite(buzzerPin, HIGH);

delay(500);

digitalWrite(buzzerPin, LOW);

delay(500);

}

lcd.setCursor(0, 0);

lcd.print("Rechecking... ");

lcd.setCursor(0, 1);

lcd.print("Please Wait... ");

delay(2000);

lcd.clear();

} else {

digitalWrite(buzzerPin, LOW);

lcd.setCursor(0, 0);

lcd.print("Smoke Level: ");

lcd.print(gasValue);

lcd.print(" ");

lcd.setCursor(0, 1);

lcd.print("System Normal ");

delay(1000);

}

}

**📊 5. Sample Output (Serial Monitor)**

SMOKE Alarm System Starting...

Gas: 368 | Air💨: Safe✅

Gas: 412 | Air💨: Safe✅

🚨 SMOKE LEAK DETECTED

Gas: 420 | Air💨: DETECTED

**✅ 6. System Behavior Summary**

| **Condition** | **Action** |
| --- | --- |
| Gas > Threshold (400) | LCD: "SMOKE LEAK" + Buzzer alert |
| Flame Sensor = LOW | LCD: "SMOKE DETECTED" + Buzzer alert |
| Normal Condition | LCD: Smoke level + "System Normal" |

**📌 7. Suggestions for Improvement**

* Add LED indicators for visual warning
* Add GSM module for SMS alerts
* Display temperature/humidity with DHT11
* Add fan or relay module for smoke evacuation