Gas sensor based alert system using Arduino Uno

1. Aim of the Project

The aim of this project is to design a basic safety monitoring system using an Arduino Uno, analog sensor (e.g., gas or smoke sensor), 16x2 LCD display, LEDs, and a buzzer. The system continuously monitors the environment and displays the sensor readings. If the sensor detects a dangerous level (above a threshold), it alerts the user by activating a red LED and a buzzer and displaying warning messages on the LCD. Otherwise, it indicates a safe condition using a green LED.

2. Purpose of the Project

This project demonstrates how to interface an analog sensor, LCD, LED indicators, and a buzzer with Arduino Uno to develop a real-time environmental alert system. It provides a foundation for safety and monitoring applications in embedded systems, making it suitable for learning sensor interfacing, threshold-based decisions, and alert systems using microcontrollers.

3. Applications of the Project

- Fire and Smoke Detection Systems in homes and industries
- Gas Leak Detection in kitchens or factories
- Air Quality Monitoring Systems
- Safety Systems in Chemical Labs
- Smart Home Automation Alert Modules

4. Components Required

S.NO	Quantity	Component
1.	1	Arduino Uno
2.	1	Analog Gas/Smoke Sensor
3.	1	16x2 LCD Display
4.	1	Buzzer
5.	2	LEDs (Red & Green)
6.	5	Resistors (220KΩ or 1KΩ)
7.	1	Breadboard
8.	multiple	Jumper Wires

5. Some Key Concepts

Analog Sensor (e.g., MQ2 Gas Sensor):

Converts physical quantities like gas concentration into an analog voltage. The sensor output increases with higher gas concentration, which is read using Arduino's analog input pins.

LCD (16x2):

Used to show real-time messages and warnings. The LiquidCrystal library simplifies the interface.

Buzzer and LEDs:

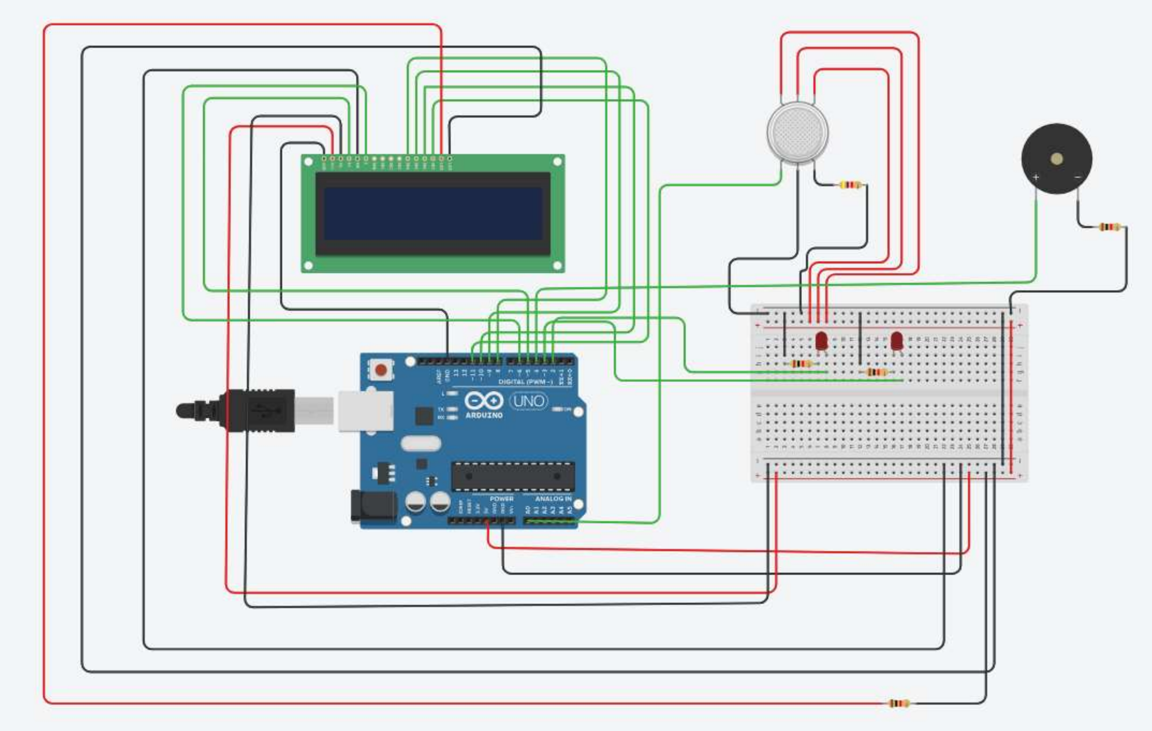
Visual and audio indicators to alert the user. Red LED + buzzer indicates danger, while green LED indicates safety.

6. Pin Configuration & Function

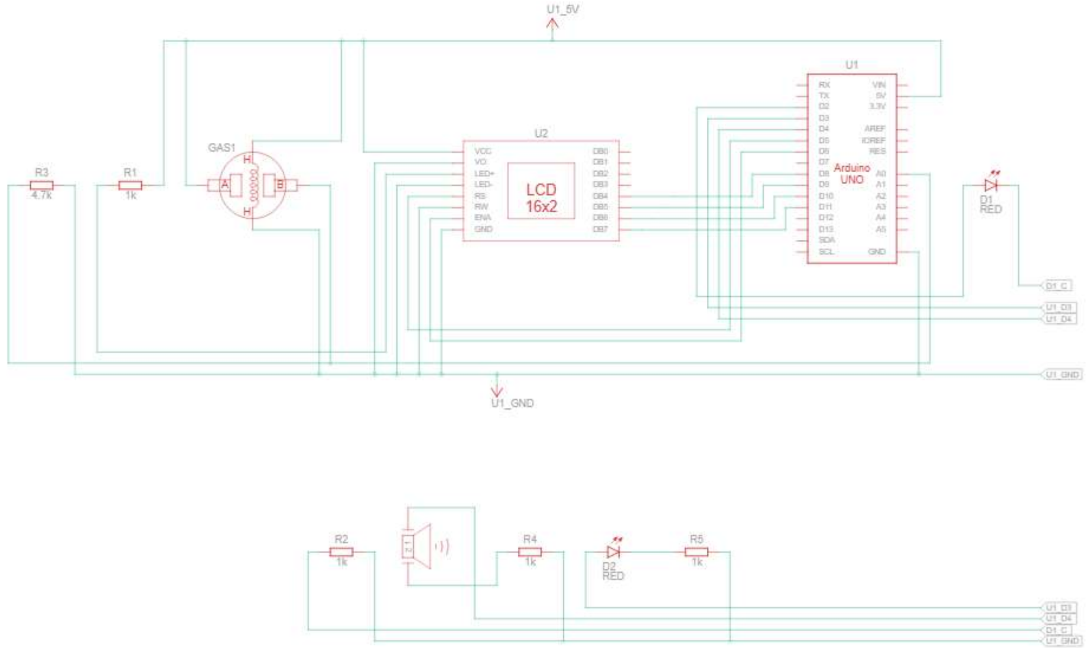
S.NO.	Component	Arduino Pin	Function
1.	Gas Sensor	A0	Analog input (sensor value)
2.	Red LED	D2	Danger Indicator
3.	Green LED	D3	Safe Indicator
4.	Buzzer	D4	Audible Alarm
5.	LCD RS	D5	Register Select
6.	LCD E	D6	Enable Pin
7.	LCD D4	D8	Data line 4
8.	LCD D5	D9	Data line 5
9.	LCD D6	D10	Data line 6
10.	LCD D7	D11	Data line 7

7. Circuit Diagram

the circuit diagram built on a breadboard, as shown in your simulation image. It shows correct connections between the sensor, LCD, Arduino, LEDs, buzzer, and power rails using jumper wires and resistors.



8. Layout (Tinkercad Simulation)



9. Code with Explanation

Below is the Arduino code used in this project:

```
#include<LiquidCrystal.h> // Include the LCD Library

LiquidCrystal lcd(5, 6, 8, 9, 10, 11); // LCD(rs, e, d4, d5, d6, d7)

int redled = 2;
int greenled = 3;
int buzzer = 4;
int sensor = A0;
int sensorThresh = 400;

void setup() {
  pinMode(redled, OUTPUT);
  pinMode(greenled, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(sensor, INPUT);
  Serial.begin(9600);
  lcd.begin(16,2);
}
```

```
void loop() {
 int analogValue = analogRead(sensor);
 Serial.println(analogValue);

 if(analogValue > sensorThresh) {
  digitalWrite(redled, HIGH);
  digitalWrite(greenled, LOW);
  tone(buzzer, 1000, 10000);
  lcd.clear();
  lcd.setCursor(0, 1);
  lcd.print("ALERT");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 1);
  lcd.print("EVACUATE");
  delay(1000);
 } else {
  digitalWrite(greenled, HIGH);
  digitalWrite(redled, LOW);
  noTone(buzzer);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("SAFE");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 1);
  lcd.print("ALL CLEAR");
  delay(1000);
 }
}
```

9. Justification

- #include<LiquidCrystal.h>: Imports the library to control the LCD screen.
- LiquidCrystal lcd(...): Initializes the LCD with pin connections to Arduino.
- analogRead(sensor): Reads analog value from the connected sensor.
- digitalWrite(...): Used to turn LEDs on/off based on sensor data.
- tone(...): Activates buzzer with a specified frequency and duration.
- lcd.print(...): Displays the alert or safe message to the LCD.

10. Conclusion

In this project, we built a sensor-based alert system using an Arduino Uno, an analog sensor, an LCD display, LEDs, and a buzzer. The system monitors environmental conditions in real time, and when a dangerous threshold is reached, it gives a visual and audible warning. This project helped us understand sensor interfacing, analog signal reading, and real-time alert feedback using Arduino. It provides foundational knowledge for developing more advanced embedded safety systems.