



**United International University**

**Department of CSE**

**Course Code: CSE 4326 (B)**

**Course Name: Microprocessors and Microcontrollers  
Laboratory**

**Experiment no. 1(B)**

**Experiment Name: An introduction to Arduino , Interfacing  
of Gas Sensor using Arduino & Showing the Sensor Data in  
OLED Display**

**Submitted by: Blitzkrieg Enigma (Group 2)**

**Name: Emon Karmoker**

**Student ID: 011201386**

**Section: B**

**Date of Performance: 17.11.2024**

**Date of Submission: 1.12.2024**

## **Objective:**

- To learn about Arduino boards and Arduino IDE.
- Examples of hardware and software interface using Arduino board.
- Examples of emulating arduino projects in Proteus software.
- Interfacing gas sensor with arduino.
- Using an OLED display to show sensor data.

## **Apparatus/Instruments used in the Lab:**

### **Hardware:**

- 1) Arduino Uno board
- 2) LED
- 3) MQ-2 Gas Sensor
- 4) OLED Display (SSD1306)
- 5) Breadboard
- 6) Jumper wires
- 7) 10k ohm Resistor
- 8) Push button

Software:

1) Proteus

2) Arduino IDE

**Theory:** The MQ-2 gas sensor is a metal oxide semiconductor (MOS) type gas sensor. It is a low-cost, widely used gas sensor that detects multiple gas (but cannot identify them) including Liquefied petroleum gas (LPG), Methane (CH<sub>4</sub>), Carbon Monoxide(CO), Alcohol, Smoke (CO<sub>2</sub>), Hydrogen(H<sub>2</sub>) and Propane in the air concentrations ranging from 200 to 10000 ppm. The MQ-2 gas sensor works by measuring the change in electrical resistance of a sensing element when it is exposed to a gas. The change in resistance of the sensing element is measured using a simple voltage divider circuit. The output voltage of the voltage divider circuit is proportional to the concentration of the gas present in the air. MQ-2 Gas and smoke sensor module has four pins:

**VCC** supplies power to the module. Connect it to the 5V output of your Arduino.

**GND** is the ground pin.

**D0** indicates the presence of combustible gasses. D0 becomes LOW when the gas concentration exceeds the threshold value (as set by the potentiometer), and HIGH otherwise.

**A0** produces an analog output voltage proportional to gas concentration, so a higher concentration results in a higher voltage and a lower concentration results in a lower voltage.

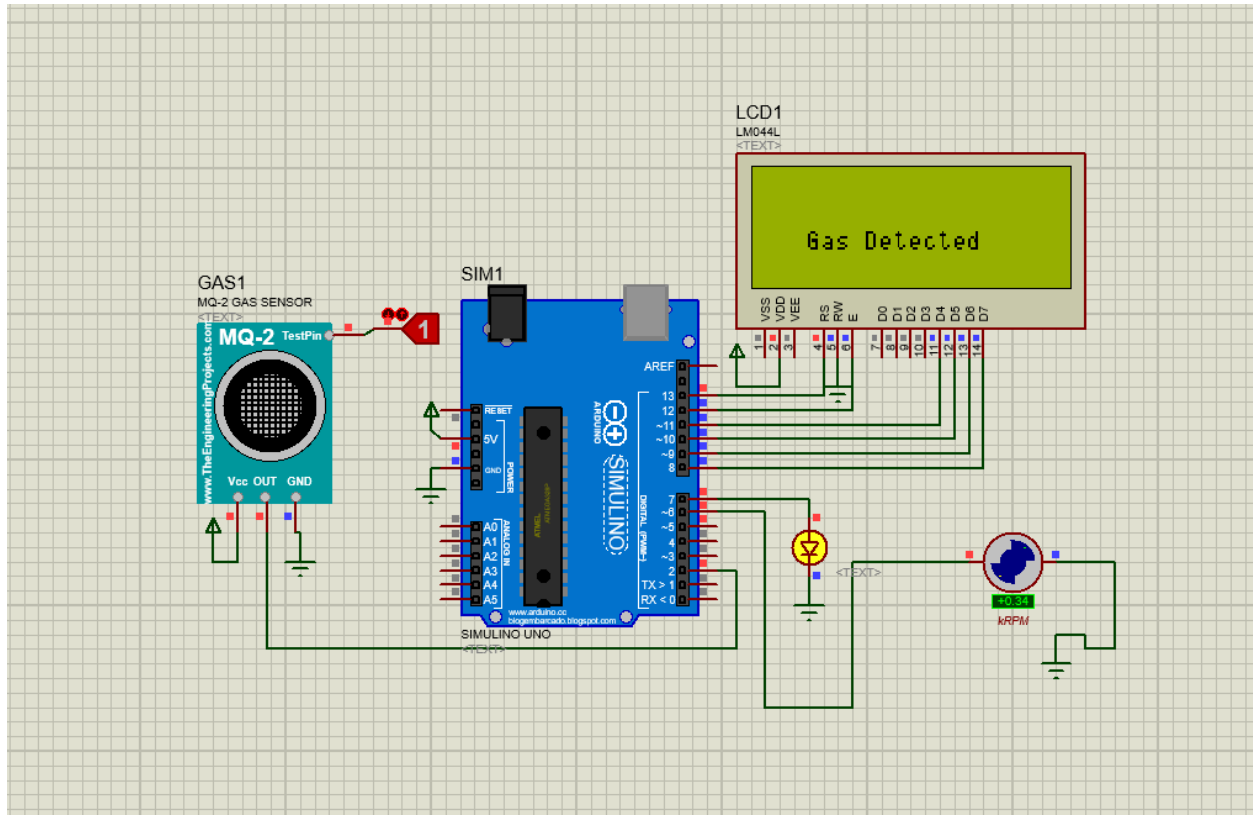
To power the circuit, the 5V pin of the Arduino is used because the operating voltage range of this module is 5V with  $\pm 0.1\%$  tolerance.

The module also has two onboard LEDs.

The power LED turns on when power is applied to the board and the Dout LED turns on when the trigger value set by the potentiometer is reached. This board also has a comparator OP-Amp onboard that is responsible for converting the incoming analog signal from the gas sensor to a digital signal. It also has a sensitivity adjustment Trim-pot, with that we can adjust the sensitivity of the device. Finally we have some resistor capacitors for decoupling and filtering. The MQ-2 gas sensor can be used in two different modes: ● Analog mode : In analog mode, the sensor output voltage is proportional to the concentration of gas present in the air. ● Digital mode : In digital mode, the sensor output voltage is a high or low signal depending on whether the gas concentration exceeds a certain threshold.

### **Solution of the mentioned problems:**

#### **When Turned On:-**



## **Recent PPM Readings:**

Reading 1: 20.50

Reading 2: 22.30

Reading 3: 19.85

Reading 4: 105.20

Reading 5: 110.45

Reading 6: 115.75

Reading 7: 95.60

Reading 8: 85.30

Reading 9: 80.00

Reading 10: 75.25

## **Codes:-**

### **1. Display the Gas Sensor data you measured using MQ-2 sensor on the OLED like this: “Gas sensor data: ppm value”.**

```
#include <Adafruit_SSD1306.h>
```

```
#include <Adafruit_GFX.h>
```

```
#include <Wire.h>
```

```
#define SCREEN_WIDTH 128
```

```
#define SCREEN_HEIGHT 64
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

```
const int AnalogSensorPin = A0; // MQ-2 analog pin for PPM calculation
```

```
void setup() {
```

```
  if (!display.begin(SSD1306_I2C_ADDRESS, 0x3C)) {
```

```
    Serial.println("OLED allocation failed");
```

```
    for (;;);
```

```
  }
```

```
  display.clearDisplay();
```

```
  display.display();
```

```
}
```

```
void loop() {
```

```
  int analogValue = analogRead(AnalogSensorPin); // Read MQ-2 analog output
```

```
  float ppm = map(analogValue, 0, 1023, 0, 1000); // Map analog value to PPM range
```

```
  display.clearDisplay();
```

```
  display.setTextSize(1);
```

```
  display.setTextColor(SSD1306_WHITE);
```

```
  display.setCursor(0, 0);
```

```
  display.print("Gas sensor data: ");
```

```
  display.print(ppm);
```

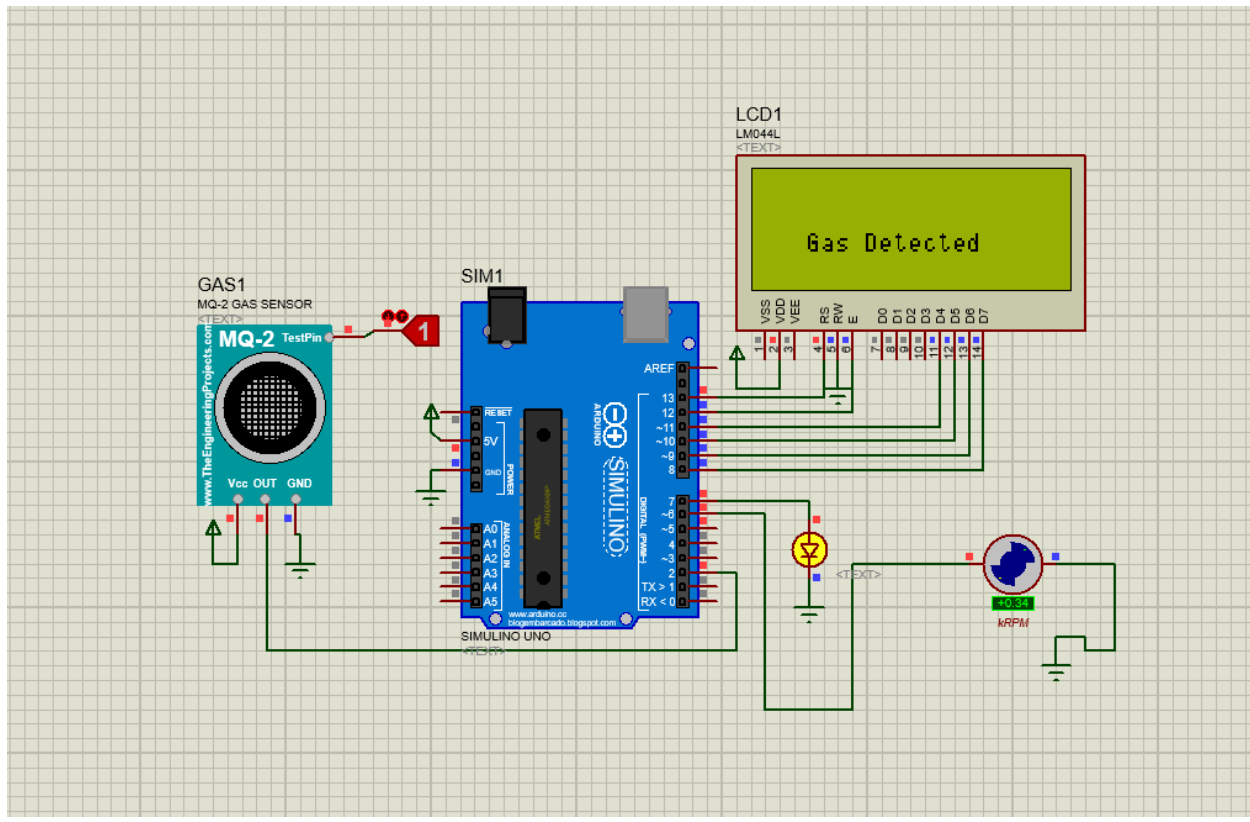
```
  display.println(" ppm");
```

```
  display.display();
```

```
  delay(500); // Update every 500ms
```

```
}
```

The MQ-2 gas sensor detects gas, and the system responds to the gas presence.



## 2. Turn on a buzzer if your gas sensor detects ppm above a certain threshold using Arduino.

```
const int BuzzerPin = 6; // Buzzer pin

float thresholdPPM = 300; // Define PPM threshold

void setup() {
  pinMode(BuzzerPin, OUTPUT);
  digitalWrite(BuzzerPin, LOW); // Ensure buzzer is off
}

void loop() {
  int analogValue = analogRead(AnalogSensorPin); // Read MQ-2 analog output
  float ppm = map(analogValue, 0, 1023, 0, 1000); // Map analog value to PPM range
```



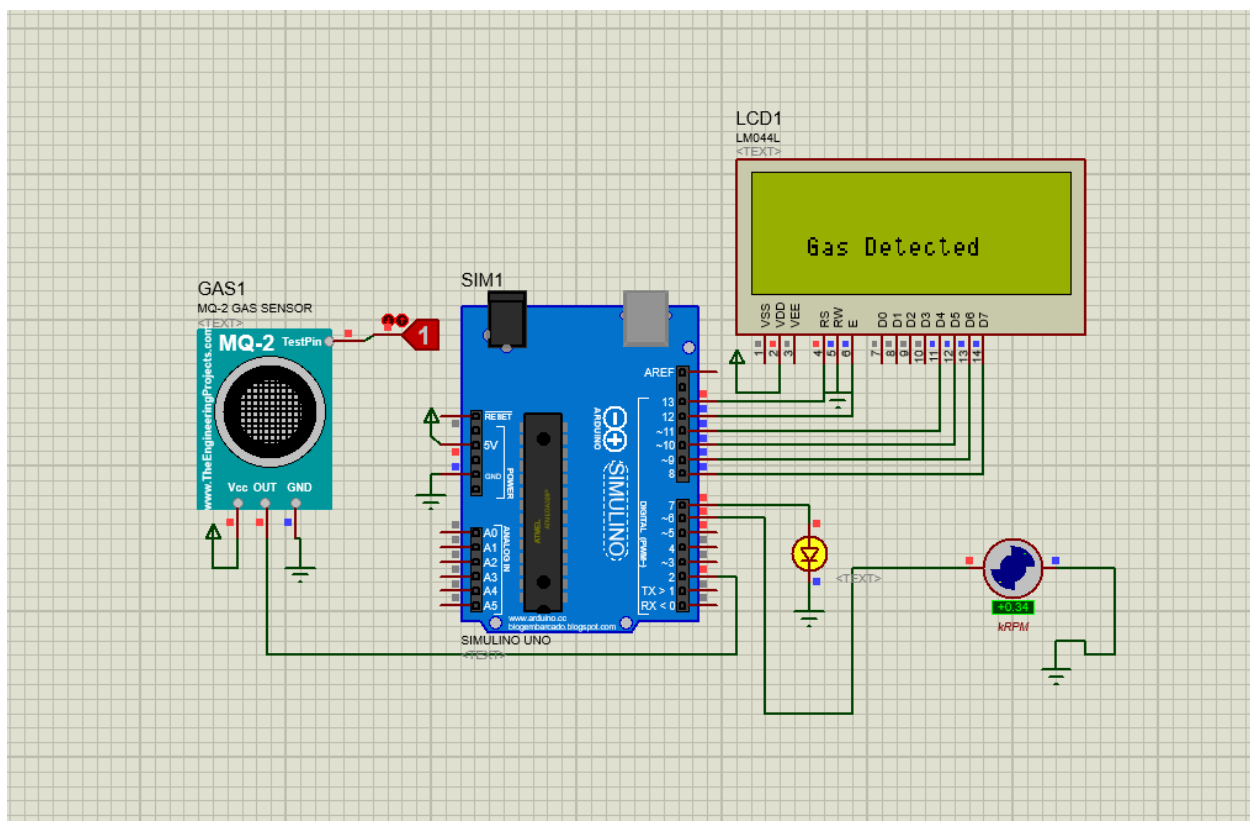
```

if (ppm > thresholdPPM) {
    digitalWrite(BuzzerPin, HIGH); // Turn on buzzer
} else {
    digitalWrite(BuzzerPin, LOW); // Turn off buzzer
}

delay(100); // Rapid detection
}

```

The MQ-2 gas sensor detects gas, and the system responds to the gas presence.



**3. Setup a gas sensing system so that if it detects gas, a message- “Gas is detected” is shown on the OLED screen.**

```

void loop() {

```

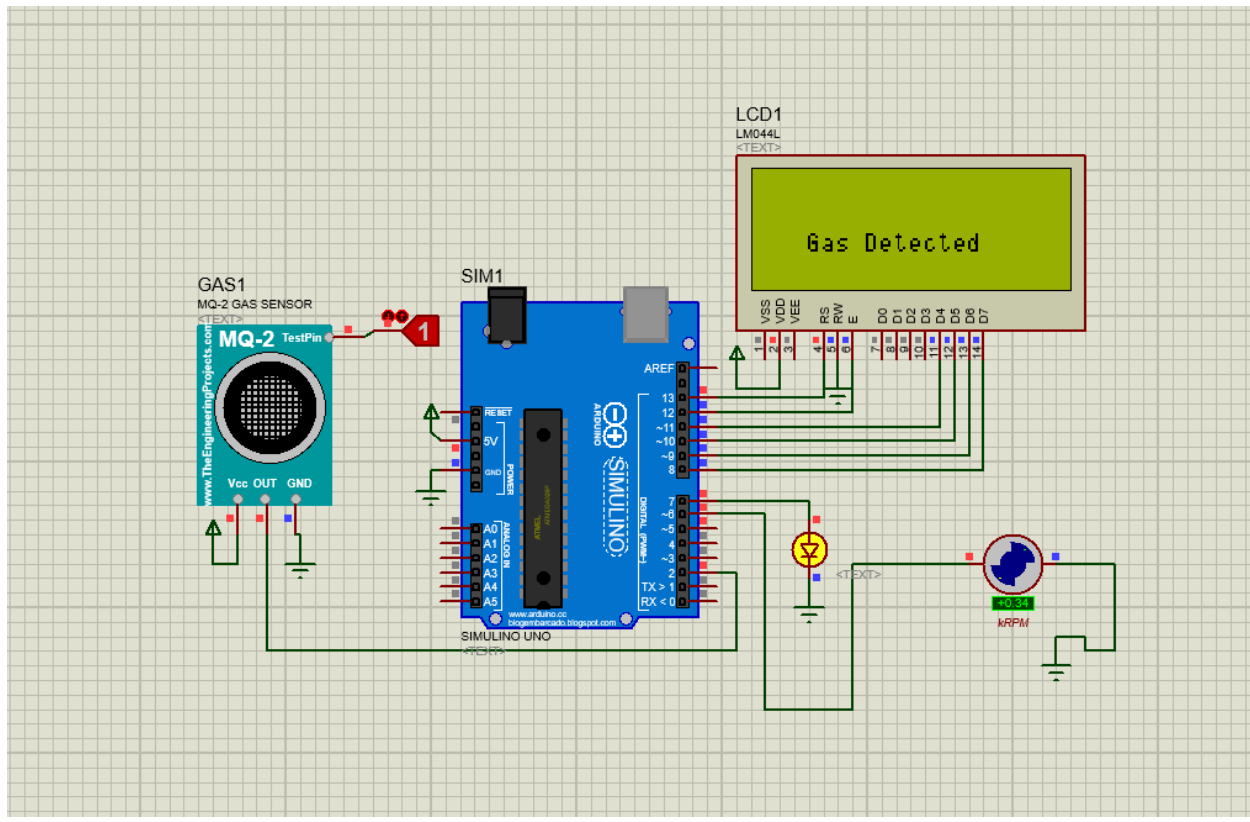
```
int analogValue = analogRead(AnalogSensorPin); // Read MQ-2 analog output
float ppm = map(analogValue, 0, 1023, 0, 1000); // Map analog value to PPM range

display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);

if (ppm > thresholdPPM) {
    display.setCursor(0, 0);
    display.println("Gas is detected!");
} else {
    display.setCursor(0, 0);
    display.println("No Gas Detected");
}

display.display();
delay(500);
}
```

The MQ-2 gas sensor detects gas, and the system responds to the gas presence. The LCD displays "Gas Detected", indicating that the gas concentration exceeds the threshold.



**4. A gas sensing based system should be implemented such that if gas is detected, an alarm/buzzer is triggered instantly to notify the incident.**

```
void loop() {

  int analogValue = analogRead(AnalogSensorPin); // Read MQ-2 analog output
  float ppm = map(analogValue, 0, 1023, 0, 1000); // Map analog value to PPM range

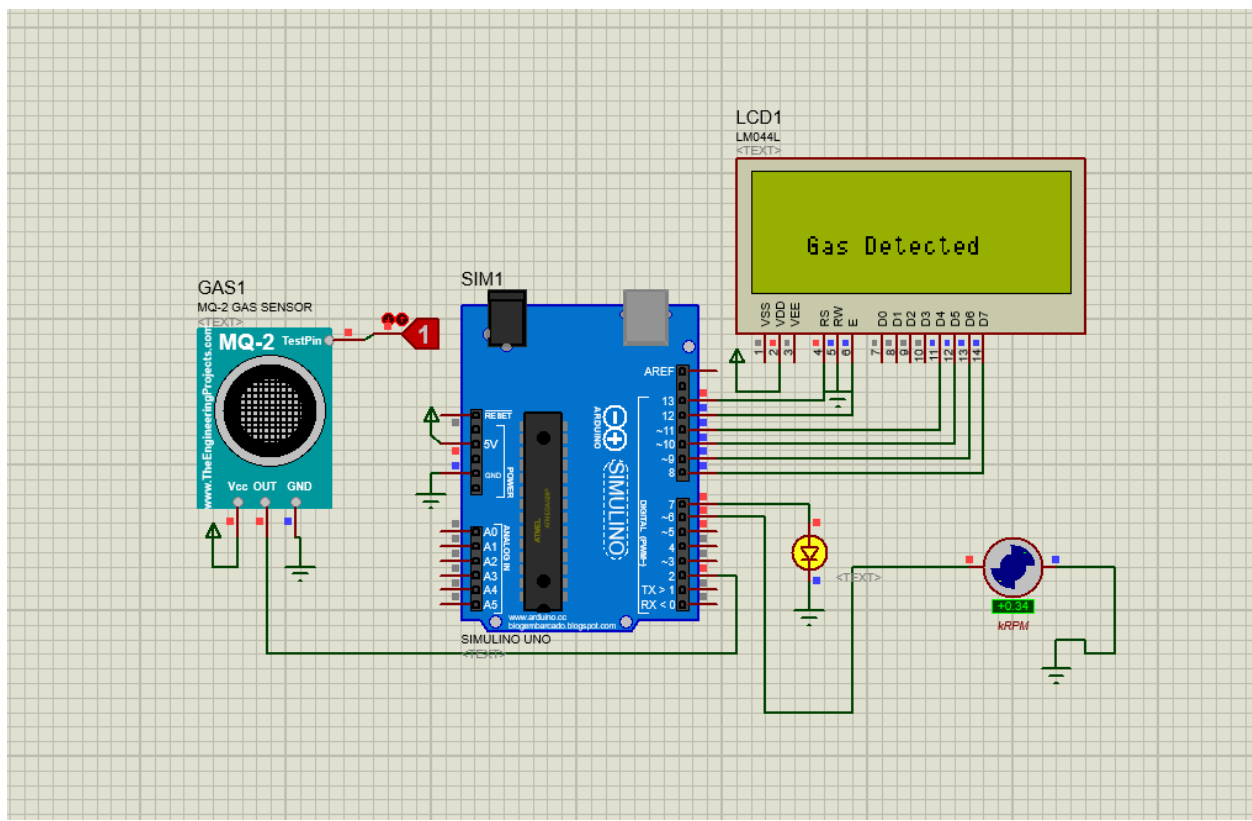
  if (ppm > thresholdPPM) {
    digitalWrite(BuzzerPin, HIGH); // Trigger buzzer
  } else {
    digitalWrite(BuzzerPin, LOW); // Turn off buzzer
  }
}
```

```

delay(100); // Rapid reaction
}

```

The MQ-2 gas sensor detects gas, and the system responds to the gas presence. The buzzer is activated, emitting a sound as an alert. The LED is also on, signaling the gas detection visually.



**5. A gassensing based system should be implemented such that if gas is detected, a motorized exhaust fan is triggered to keep out the fumes.**

```
const int FanPin = 5; // Motorized fan relay pin

void setup() {
  pinMode(FanPin, OUTPUT);
  digitalWrite(FanPin, LOW); // Ensure fan is off initially
}

void loop() {
  int analogValue = analogRead(AnalogSensorPin); // Read MQ-2 analog output
  float ppm = map(analogValue, 0, 1023, 0, 1000); // Map analog value to PPM range

  if (ppm > thresholdPPM) {
    digitalWrite(FanPin, HIGH); // Turn on fan
  } else {
    digitalWrite(FanPin, LOW); // Turn off fan
  }

  delay(100); // Rapid detection and response
}
```

The fan is turned on to ventilate the area, as indicated by the connected motor.



```

pinMode(SensorPin, INPUT); // Gas sensor as input

pinMode(LEDPin, OUTPUT); // LED as output

pinMode(BuzzerPin, OUTPUT); // Buzzer as output

pinMode(FanPin, OUTPUT); // Fan relay as output


// Turn off outputs initially
digitalWrite(LEDPin, LOW);

digitalWrite(BuzzerPin, LOW);

digitalWrite(FanPin, LOW);


// Initialize the LCD
lcd.begin(20, 4); // Set up the LCD (20 columns, 4 rows)
lcd.setCursor(0, 0);
lcd.print(" THE BRIGHT LIGHT ");
lcd.setCursor(0, 1);
lcd.print(" GAS DETECTION SYS ");
delay(2000); // Display welcome message for 2 seconds
lcd.clear();
}

void loop() {
    // Read gas sensor digital output
    if (digitalRead(SensorPin) == HIGH) {
        // Gas detected

        digitalWrite(LEDPin, HIGH); // Turn on LED
        digitalWrite(BuzzerPin, HIGH); // Turn on Buzzer
        digitalWrite(FanPin, HIGH); // Turn on Fan


        // Display "Gas Detected" with animation
        lcd.setCursor(0, 2);
        lcd.print(" Gas Detected ");
    }
}

```

```
delay(100);

lcd.setCursor(0, 2);

lcd.print("  Gas Detected. ");

delay(100);

lcd.setCursor(0, 2);

lcd.print("  Gas Detected.. ");

delay(100);

lcd.setCursor(0, 2);

lcd.print("  Gas Detected... ");

delay(100);

} else {

  // No gas detected

  digitalWrite(LEDPin, LOW);    // Turn off LED

  digitalWrite(BuzzerPin, LOW); // Turn off Buzzer

  digitalWrite(FanPin, LOW);    // Turn off Fan


  // Display "No Gas" on LCD

  lcd.setCursor(0, 2);

  lcd.print("    No Gas    ");

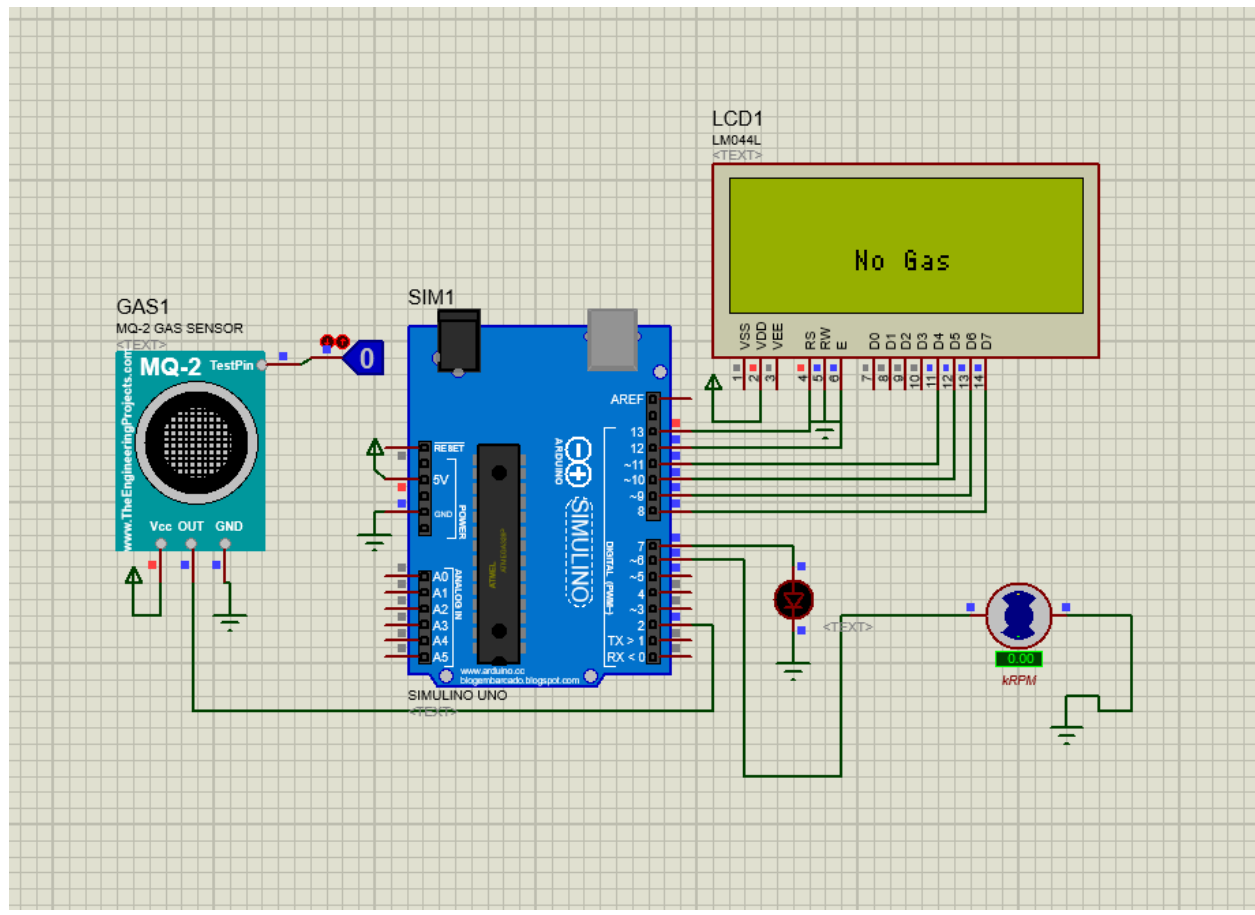
  delay(500); // Refresh display every 500ms

}

}
```



## When Turns Off



## Codes:-

## **1. The PPM value will decrease as the sensor detects less gas.**

```
#include <Adafruit_SSD1306.h>

#include <Adafruit_GFX.h>

#include <Wire.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

const int AnalogSensorPin = A0; // MQ-2 analog pin for PPM calculation

void setup() {
  if (!display.begin(SSD1306_I2C_ADDRESS, 0x3C)) {
    Serial.println("OLED allocation failed");
    for (;;);
  }
  display.clearDisplay();
  display.display();
}

void loop() {
  int analogValue = analogRead(AnalogSensorPin); // Read MQ-2 analog output
  float ppm = map(analogValue, 0, 1023, 0, 1000); // Map analog value to PPM range

  display.clearDisplay();
```

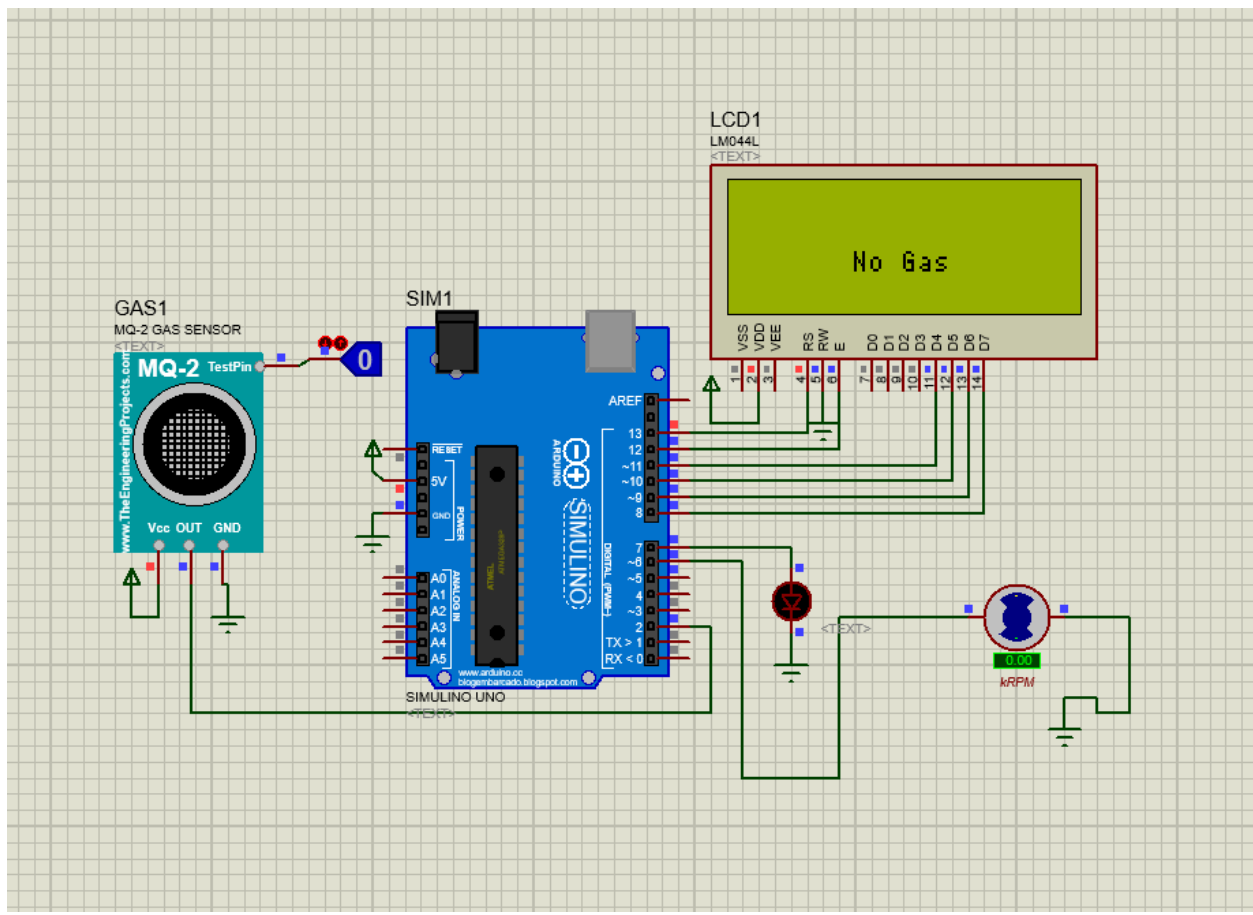
```

display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);
display.print("Gas sensor data: ");
display.print(ppm);
display.println(" ppm");
display.display();

delay(500); // Update every 500ms
}

```

The message will change to indicate "No Gas Detected."



## **2. The buzzer will turn off automatically when the PPM value falls below the threshold.**

```
const int BuzzerPin = 6;  // Buzzer pin

float thresholdPPM = 300; // Define PPM threshold

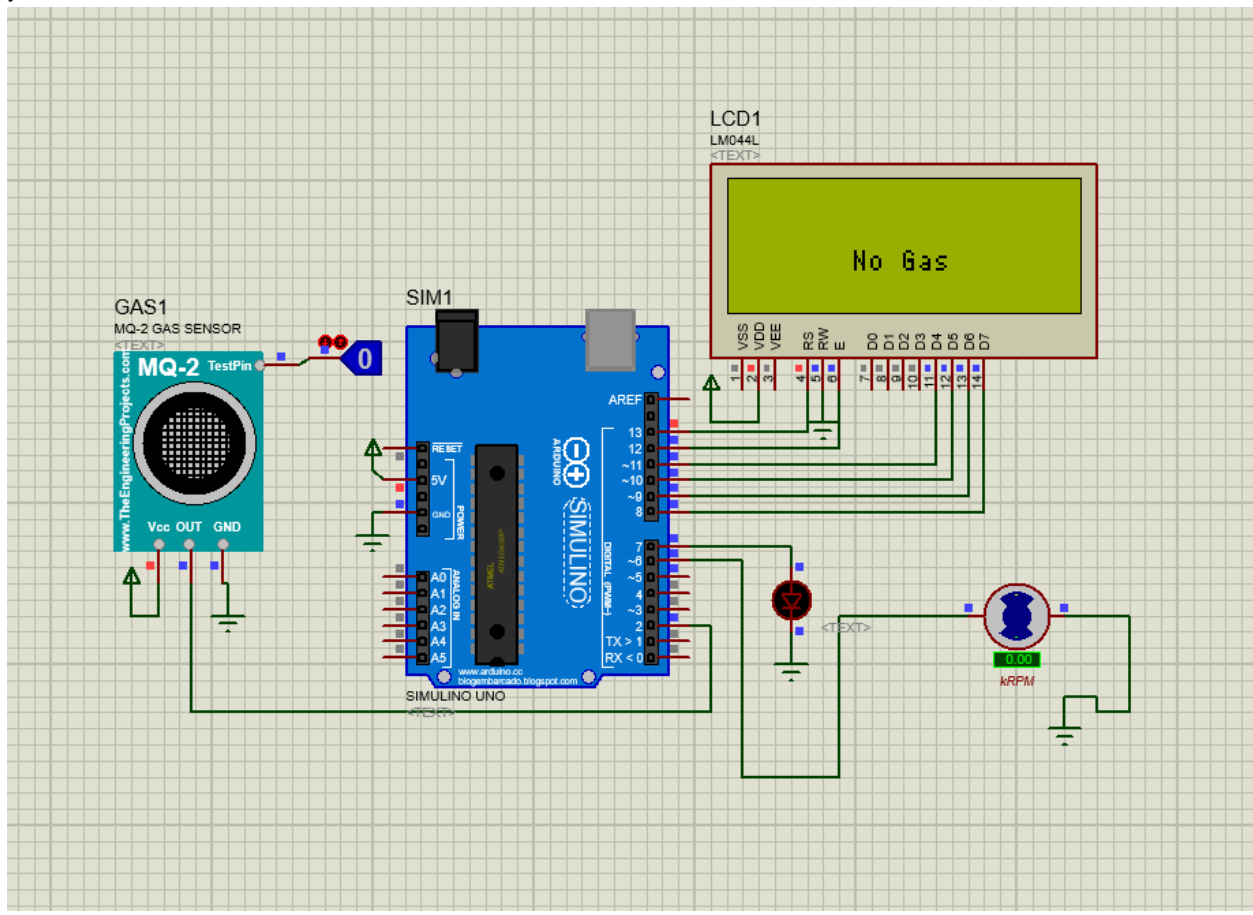
void setup() {
  pinMode(BuzzerPin, OUTPUT);
  digitalWrite(BuzzerPin, LOW); // Ensure buzzer is off
}

void loop() {
  int analogValue = analogRead(AnalogSensorPin); // Read MQ-2 analog output
  float ppm = map(analogValue, 0, 1023, 0, 1000); // Map analog value to PPM range

  if (ppm > thresholdPPM) {
    digitalWrite(BuzzerPin, HIGH); // Turn on buzzer
  } else {
    digitalWrite(BuzzerPin, LOW); // Turn off buzzer
  }

  delay(100); // Rapid detection
```

}



The buzzer will turn off automatically when the PPM value falls below the threshold.

### **3. The message will change to indicate "No Gas Detected."**

```
void loop() {  
    int analogValue = analogRead(AnalogSensorPin); // Read MQ-2 analog output  
    float ppm = map(analogValue, 0, 1023, 0, 1000); // Map analog value to PPM range  
  
    display.clearDisplay();  
    display.setTextSize(1);
```

```
display.setTextColor(SSD1306_WHITE);
```

```
if (ppm > thresholdPPM) {
```

```
    display.setCursor(0, 0);
```

```
    display.println("Gas is detected!");
```

```
} else {
```

```
    display.setCursor(0, 0);
```

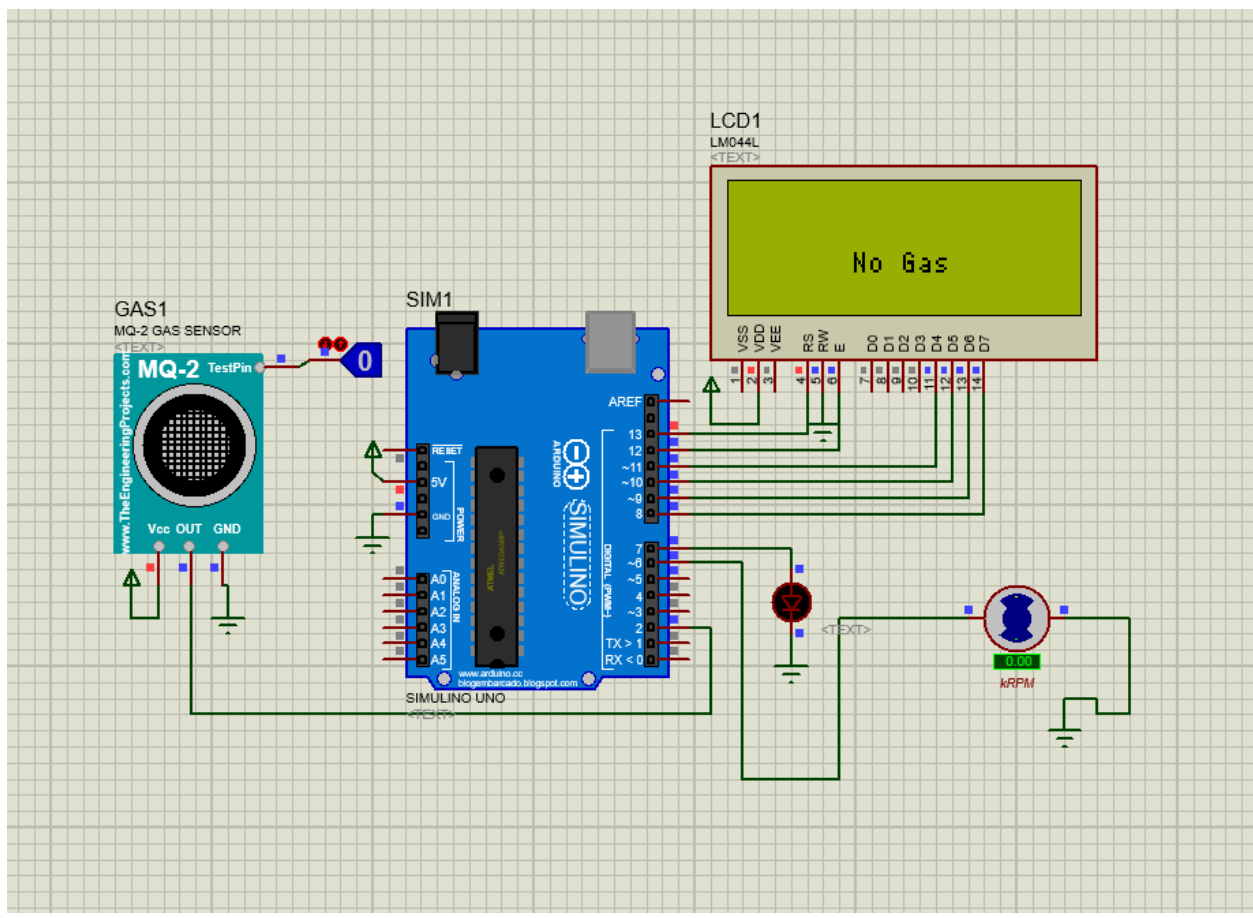
```
    display.println("No Gas Detected");
```

```
}
```

```
display.display();
```

```
delay(500);
```

```
}
```

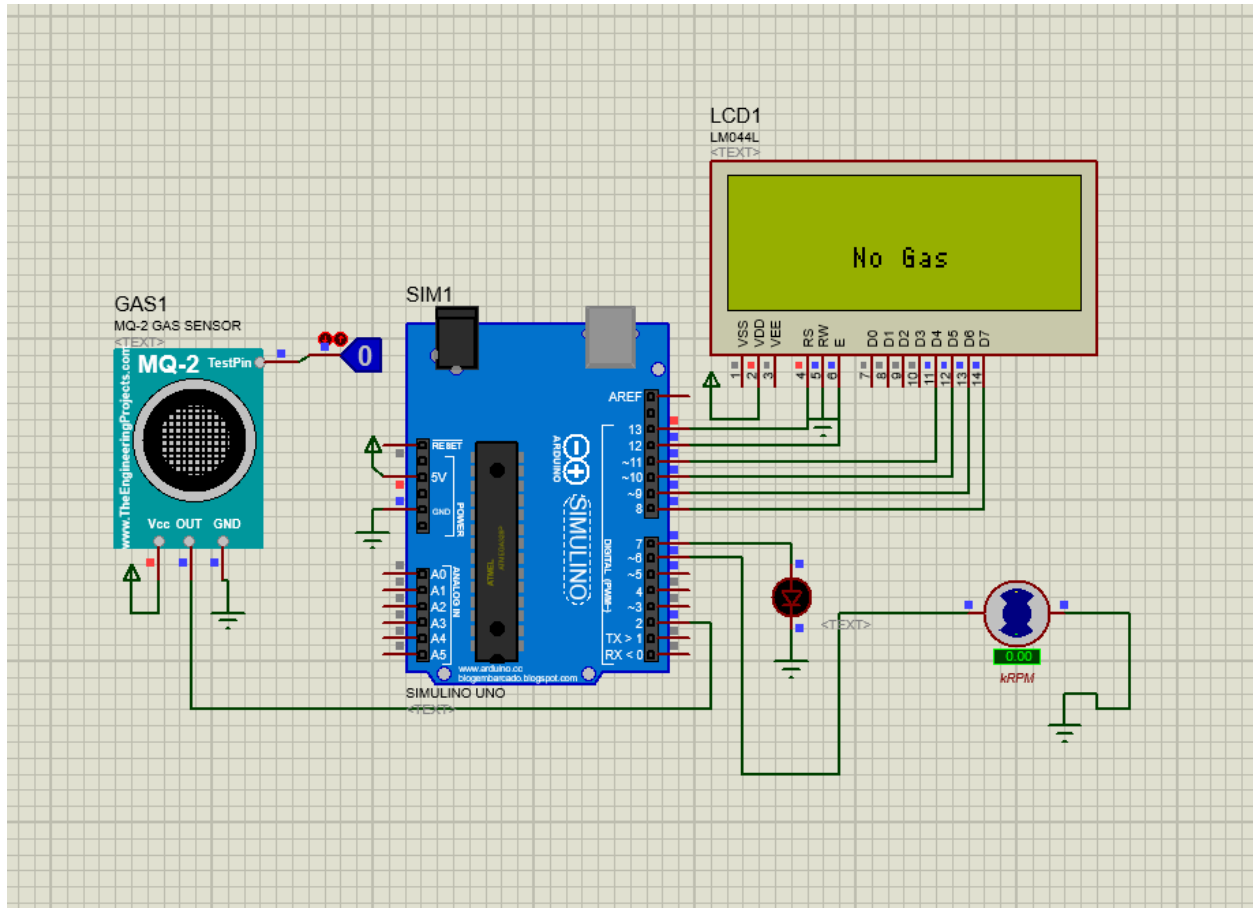


The MQ-2 gas sensor don't detects gas. The LCD displays "No Gas ", indicating that the PPM value will decrease as the sensor detects less gas

#### **4. The buzzer will turn off automatically when the PPM value falls below the threshold.**

```
void loop() {  
    int analogValue = analogRead(AnalogSensorPin); // Read MQ-2 analog output  
    float ppm = map(analogValue, 0, 1023, 0, 1000); // Map analog value to PPM range  
  
    if (ppm > thresholdPPM) {  
        digitalWrite(BuzzerPin, HIGH); // Trigger buzzer  
    } else {  
        digitalWrite(BuzzerPin, LOW); // Turn off buzzer  
    }  
  
    delay(100); // Rapid reaction  
}
```

The buzzer automatically turn off when the gas concentration decreases.



## 5. The fan will stop running when the gas concentration drops below the threshold.

```
const int FanPin = 5; // Motorized fan relay pin
```

```
void setup() {
  pinMode(FanPin, OUTPUT);
  digitalWrite(FanPin, LOW); // Ensure fan is off initially
}
```

```
void loop() {
```



```

int analogValue = analogRead(AnalogSensorPin); // Read MQ-2 analog output

float ppm = map(analogValue, 0, 1023, 0, 1000); // Map analog value to PPM range

if (ppm > thresholdPPM) {
    digitalWrite(FanPin, HIGH); // Turn on fan
} else {
    digitalWrite(FanPin, LOW); // Turn off fan
}

delay(100); // Rapid detection and response
}

```

The fan will stop running when the gas concentration drops below the threshold.

## **Integrated Full Code:-**

```

// Include the library for LCD
#include <LiquidCrystal.h>

// Initialize the library with the numbers of the interface pins
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

// Define pin connections
const int SensorPin = 2; // MQ-2 sensor digital output pin
const int LEDPin = 7;    // LED pin for visual alert
const int BuzzerPin = 6; // Buzzer pin
const int FanPin = 5;    // Fan relay pin

void setup() {

```

```

// Set pin modes

pinMode(SensorPin, INPUT); // Gas sensor as input
pinMode(LEDPin, OUTPUT); // LED as output
pinMode(BuzzerPin, OUTPUT); // Buzzer as output
pinMode(FanPin, OUTPUT); // Fan relay as output


// Turn off outputs initially

digitalWrite(LEDPin, LOW);
digitalWrite(BuzzerPin, LOW);
digitalWrite(FanPin, LOW);


// Initialize the LCD
lcd.begin(20, 4); // Set up the LCD (20 columns, 4 rows)
lcd.setCursor(0, 0);
lcd.print(" THE BRIGHT LIGHT ");
lcd.setCursor(0, 1);
lcd.print(" GAS DETECTION SYS ");
delay(2000); // Display welcome message for 2 seconds
lcd.clear();
}


void loop() {
    // Read gas sensor digital output
    if (digitalRead(SensorPin) == HIGH) {
        // Gas detected

        digitalWrite(LEDPin, HIGH); // Turn on LED
        digitalWrite(BuzzerPin, HIGH); // Turn on Buzzer
        digitalWrite(FanPin, HIGH); // Turn on Fan


        // Display "Gas Detected" with animation
        lcd.setCursor(0, 2);
    }
}

```

```

lcd.print(" Gas Detected ");
delay(100);
lcd.setCursor(0, 2);
lcd.print(" Gas Detected. ");
delay(100);
lcd.setCursor(0, 2);
lcd.print(" Gas Detected.. ");
delay(100);
lcd.setCursor(0, 2);
lcd.print(" Gas Detected... ");
delay(100);
} else {
  // No gas detected
  digitalWrite(LEDPin, LOW);    // Turn off LED
  digitalWrite(BuzzerPin, LOW); // Turn off Buzzer
  digitalWrite(FanPin, LOW);    // Turn off Fan

  // Display "No Gas" on LCD
  lcd.setCursor(0, 2);
  lcd.print(" No Gas ");
  delay(500); // Refresh display every 500ms
}
}

```

## Behavior When Gas Emission Stops

1. **OLED Display:**
  - The PPM value will decrease as the sensor detects less gas.
  - The message will change to indicate "No Gas Detected."
2. **Buzzer:**
  - The buzzer will turn off automatically when the PPM value falls below the threshold.
3. **Exhaust Fan:**
  - The fan will stop running when the gas concentration drops below the threshold.

## **Discussion:**

Understood how the MQ-2 sensor detects gas and measures concentration in PPM.

Learned to interface sensors with Arduino and control components like LCD, buzzer, LED, and fan.

Gained experience in real-time monitoring, threshold-based alerts, and system calibration.

Sensor readings were initially inaccurate due to insufficient preheating time.

Faced wiring errors with the LCD and peripherals.

Had to troubleshoot calibration issues to ensure accurate PPM values.

Can be used in kitchens, industries, or laboratories to detect gas leaks and prevent hazards.

Suitable for safety systems in homes, workplaces, or vehicles to monitor air quality.

Please note that, your lab report must have these 5 sections. If you take image/information from internet, add reference of that image/information. Properly name all the figures you add with figure number and label of the figure. The report should look clean. For any confusion, you can check the lab report guidelines pdf.