United International University

Course Code: CSE 4326
Section: B

Course Name
Microprocessors and Microcontrollers Laboratory

Experiment No: 01

Experiment Name
Interfacing of Gas Sensor using Arduino
& Showing the Sensor Data in OLED Display

Submitted by (Group 4):

Muhammad Saadman Sakib Hossain - 0112230100
Jahidul Islam - 0112230654
Abrar Habib Adib - 0112230312
A. M. Zayed Abdullah - 011221127
Samira Bintey Haque - 011221128

Date of Performance: 29/07/2025
Date of Submission: 12/08/2025

1. Objective:

In this lab, we will learn how to interface a **gas sensor (MQ-2)** with an **Arduino Uno R3** and display the detected gas concentration data on an **OLED display**. We will use the MQ-2 gas sensor to detect the presence of gases such as LPG, CO, and smoke in the environment. The data collected from the sensor will then be displayed on an OLED display in real-time. Through this experiment, we will learn how to integrate sensors and displays with Arduino and create a basic safety system useful in many real-life applications.

Final outcomes of this experiment will be:

1. To learn about Arduino boards and the Arduino IDE.
2. To understand how gas sensors detect gases and provide output signals.
3. To interface the MQ-2 gas sensor with Arduino.
4. To use an OLED display to show live gas concentration in PPM.
5. To simulate the gas sensing system in Proteus.

Expected Outcomes:

1. To develop a functional system that displays gas concentration data (in ppm) on an OLED display.
2. To trigger a buzzer when gas levels exceed a critical safety threshold.
3. To display the message "Gas is detected" on the OLED screen upon gas detection.

2. Apparatus:

Hardware:

1. Arduino Uno Board (Uno R3)
2. MQ-2 Gas Sensor
3. OLED Display (SSD1306)
4. Breadboard
5. Jumper Wires
6. 10k Ohm Resistor
7. Buzzer (Optional, for alert system)

Software:

1. Arduino IDE
2. Proteus Design Suite (for simulation)

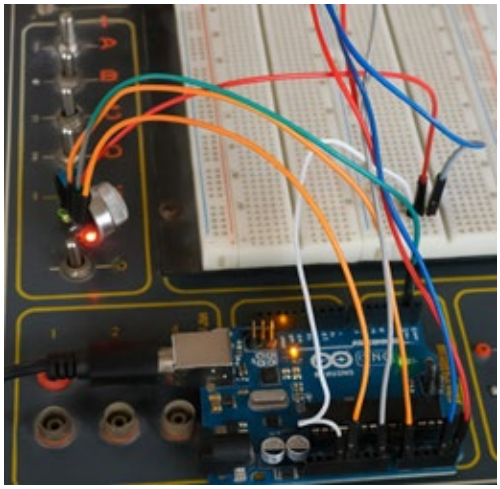
3. Theory:

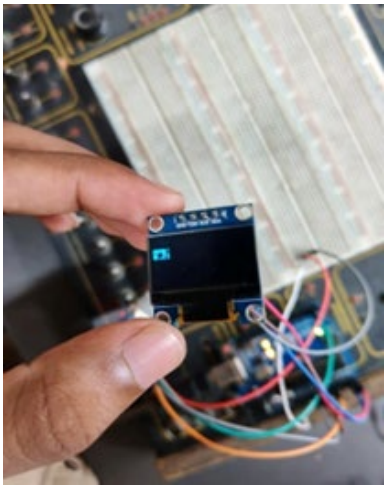
1. The **MQ-2 gas sensor** is a metal oxide semiconductor (MOS) sensor that detects combustible gases like LPG, smoke, CO, and methane. It changes its resistance when exposed to gas, and this change is read as analog voltage using Arduino's analog pins.

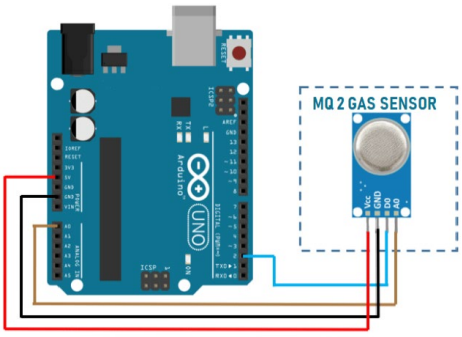
2. The **sensor output (A0)** gives a variable voltage (0-5V) based on gas concentration. Arduino reads this via analog pin A0 and maps it to a numerical value between 0-1023. With calibration, this can be translated to an estimated value in **PPM (parts per million)**.

3. The **OLED display (SSD1306)** is used to show the gas concentration values in real-time. It communicates with the Arduino using **I2C protocol** with only two data lines (SDA & SCL). This makes the setup efficient and compact.

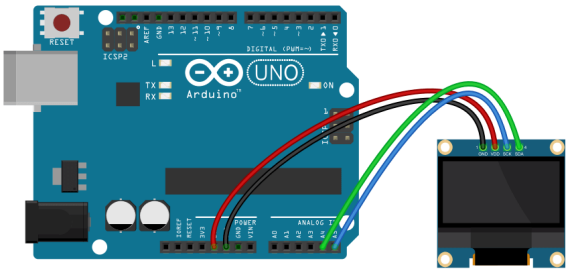
4. A **buzzer** is added to produce a sound alarm if the gas level crosses a specific limit, simulating a real-world gas leakage alert system. The buzzer provides an audible alert when unsafe gas levels are detected.







Gas sensor's wire connection



MQ-2, OLED & Buzzer with Arduino

4. Circuit Description:

The system includes **three (03)** main components:

1. MQ-2 Gas Sensor Connection to Arduino Uno R3

- VCC → 5V
- GND → GND
- A0 → A0 (Analog signal to Arduino)

2. OLED Display Connection to Arduino Uno R3

- VCC → 5V
- GND → GND
- SDA → A4
- SCL → A5

3. Buzzer Connection to Arduino Uno R3

- Positive leg → D9
- Other leg → GND

The components are powered by Arduino and mounted on a breadboard. This creates a compact, reliable, and modular circuit.

5. Procedure:

1. Connect the MQ-2 gas sensor to the Arduino Uno R3 as per the circuit description:
   - VCC to 5V, GND to GND, analog output to A0.

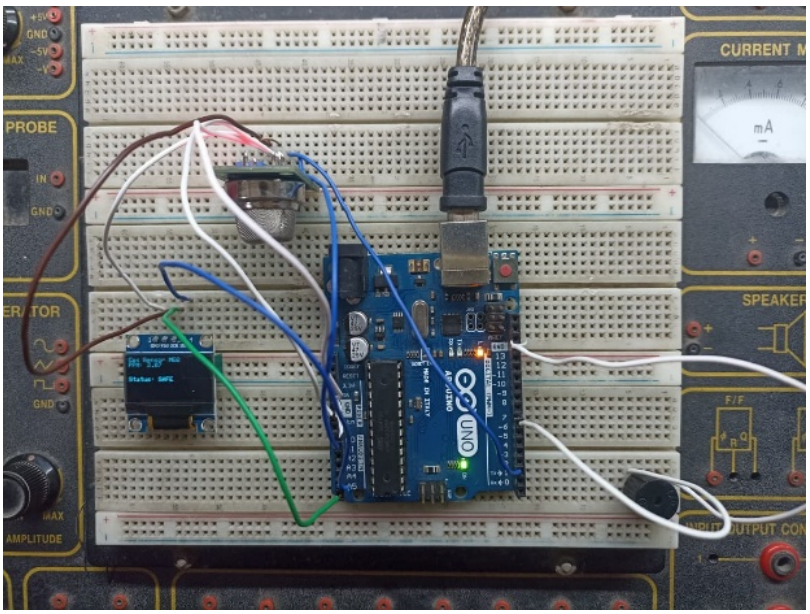
2. Connect the OLED display (SSD1306) to Arduino's I2C pins:
   - VCC to 5V, GND to GND, SDA to A4, SCL to A5.

3. Connect the buzzer's positive leg to digital pin 9 and the other leg to GND.

4. Write and upload the Arduino code that:

   - Reads the analog value from the MQ-2 sensor.
   - Converts the raw value into an approximate ppm measurement.
   - Displays the gas concentration data on the OLED screen in the format: **"Gas sensor data: <ppm>"**.
   - Turns on the buzzer when the ppm value exceeds the safety threshold (400 ppm).
   - Displays the message **"Gas is detected"** on the OLED when gas concentration is high; otherwise, displays **"Safe"**.
   - Observe the OLED display for real-time gas concentration and buzzer activation when gas is detected.

6. Final Setup:



Final setup

7. Program Code:

🔗 **GitHub Repository:** [Experiment No. 01]

8. Result & Observations:

Condition	PPM Value	OLED Display Message	Buzzer Status
No gas present	< 400	"Safe"	OFF
Gas detected (unsafe)	> 400	"Gas is detected!"	ON (beeping)

- The system successfully displayed gas concentration in PPM on the OLED.
- The buzzer was activated when the PPM crossed the safety threshold.
- The system behaved as expected both in physical testing and Proteus simulation.

📽 **Demonstration Video:** [Experiment No. 01]

9. Discussion:

This experiment successfully demonstrated how to interface an MQ-2 gas sensor with an Arduino Uno R3 and visualize the gas concentration using an OLED display. The system accurately displayed sensor readings in PPM and triggered a buzzer when the value exceeded a critical safety threshold. This mimics a real-life gas leakage detection system, which is widely used in industrial and household safety devices.

Some key takeaways from this experiment include:

1. Understanding analog sensor readings and how to convert them into meaningful real-world values like PPM.

2. Practical use of I2C communication for OLED display interfacing.

3. Implementation of conditional logic in code to respond to sensor data dynamically (e.g., triggering buzzer or changing display messages).

During the experiment, minor challenges included:

1. Properly calibrating the sensor for accurate readings, as gas sensors can be sensitive to environmental factors.

2. Ensuring the correct library installation and wiring for OLED display communication.

3. Handling minor fluctuations in sensor data and ensuring the threshold logic is not too sensitive.

Despite these, the final implementation worked as expected. The simulation in Proteus also helped in visualizing the hardware behavior before physical deployment. This experiment reinforced fundamental skills in embedded systems, sensor integration, and real-time data display.

10. Conclusion

The experiment was successful in achieving the following:

1. Understanding the working principles of MQ-2 gas sensors.
2. Interfacing an OLED display with Arduino using I2C.
3. Developing a working system that detects gases and displays PPM values.
4. Providing real-time alerts using a buzzer upon unsafe gas levels.
5. Simulating the system in Proteus before hardware implementation.

This foundational work can be built upon for more advanced safety and environmental monitoring systems.

11. References:

- Arduino Documentation: Arduino
- Adafruit SSD1306 Library: Library of Adafruit_SSD1306
- GitHub Code Repository: **GitHub (Micro Lab)**