UNITED INTERNATIONAL UNIVERSITY (UIU)

Dept. of Computer Science & Engineering

Course No: CSE 4326

Course Title: Microprocessors and Microcontrollers Laboratory

**Experiment 1: Interfacing Air Humidity, Temperature, RH%, and Time using necessary sensors, Arduino & OLED Display**

# Objectives:

# In this lab, we will learn how to interface a DHT11 (or DHT22) sensor with an Arduino and display humidity, temperature, and relative humidity (RH%) on an OLED display. Additionally, an MQ-2 gas sensor will be used to measure air quality, and a real-time clock simulation will be implemented using the Arduino's millis() function. The final outcomes of this experiment will be:

* Understanding the working of DHT sensors.
* Interfacing DHT11/DHT22 and MQ-2 with Arduino.
* Displaying sensor data on an OLED display.
* Implementing a time simulation without an RTC module.
* Classifying air quality based on MQ-2 sensor readings.

# Components required:

**Hardware:**

1. Arduino Uno board
2. DHT11 or DHT22 Sensor
3. MQ-2 Gas Sensor
4. OLED Display (SSD1306)
5. Breadboard
6. Jumper wire

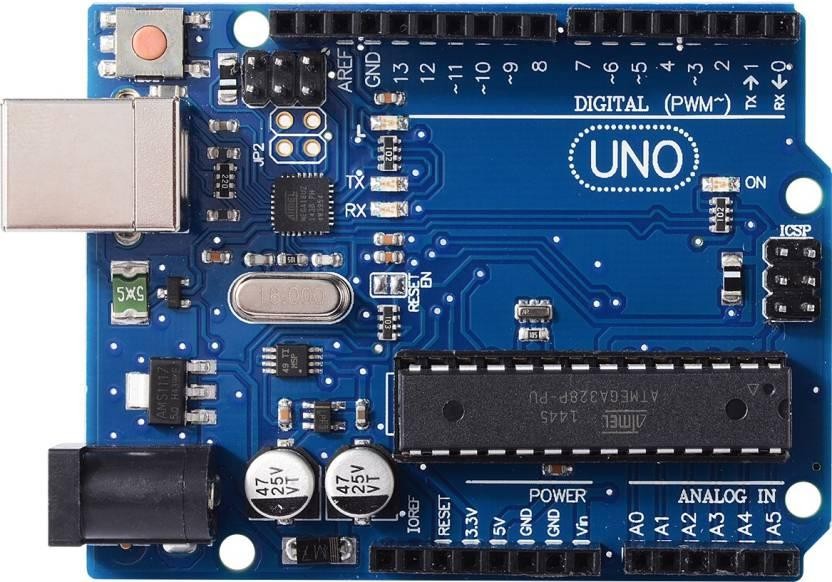
# Software:

1. Arduino IDE

# Introduction to Components:

1. **​Arduino Board:**

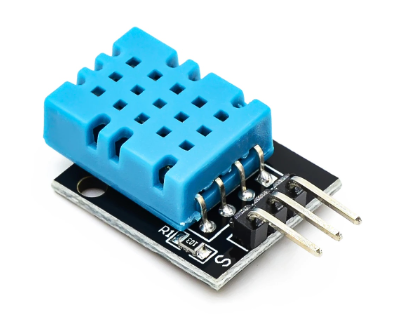
Arduino is an open-source microcontroller platform used for various embedded system applications. It consists of a microcontroller, input/output pins, power connections, and serial communication capabilities. The Arduino Uno, used in this experiment, is equipped with an ATmega328P microcontroller and features 14 digital I/O pins, 6 analog inputs, and communication interfaces such as I2C and SPI.



# Fig 1: Arduino Uno Board

1. **​DHT Sensor:**

The DHT11 and DHT22 are commonly used temperature and humidity sensors that provide calibrated digital output. The sensor consists of a capacitive humidity sensor and a thermistor to measure temperature.



# Fig 2: DHT11 Sensor

1. **MQ-2 Gas Sensor:**

The MQ-2 sensor is a metal oxide semiconductor (MOS) sensor used to detect gases such as LPG, methane, alcohol, and carbon monoxide. It provides an analog output based on the concentration of gases present in the air.

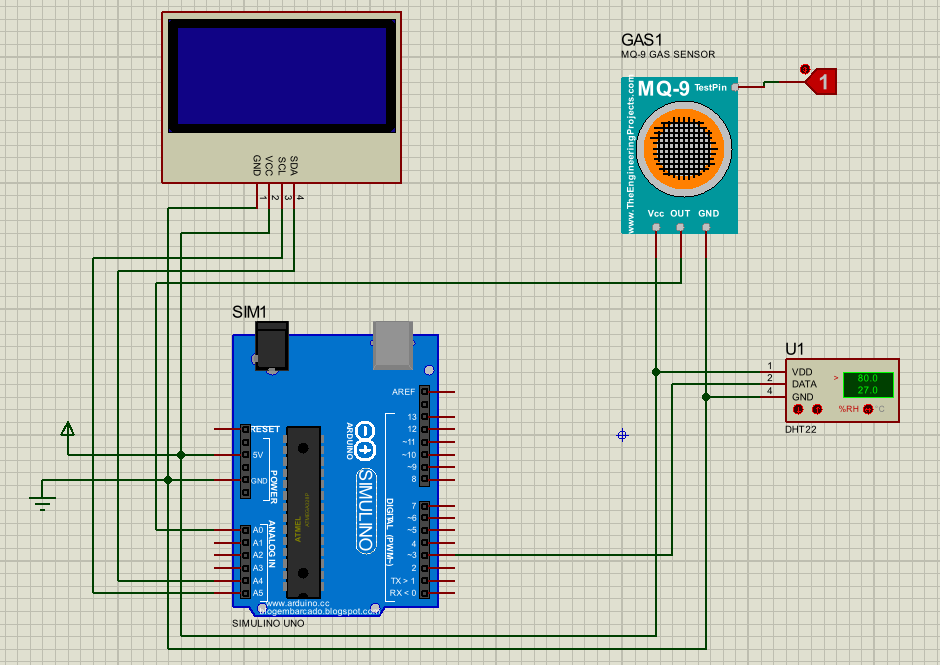


# Fig 3: MQ-2 Gas Sensor

**Pin Configurations:**

* **DHT Sensor:**
  + **VCC:** Power supply (3.3V - 5V)
  + **GND:** Ground
  + **Data:** Outputs temperature and humidity data
* **MQ-2 Sensor:**
  + **VCC:** 5V Power Supply
  + **GND:** Ground
  + **A0:** Analog output signal

**Procedure:**



# Fig 4: Pin Diagram

**Step 1: Connect the DHT Sensor**

1. Connect the VCC pin of the DHT sensor to the 5V pin of the Arduino.
2. Connect the GND pin of the DHT sensor to the GND pin of the Arduino.
3. Connect the Data pin of the DHT sensor to digital pin 3 of the Arduino.

**Step 2: Connect the MQ-2 Gas Sensor**

1. Connect the VCC pin of the MQ-2 sensor to the 5V pin of the Arduino.
2. Connect the GND pin of the MQ-2 sensor to the GND pin of the Arduino.
3. Connect the A0 pin of the MQ-2 sensor to A0 on the Arduino.

**Step 3: Connect the OLED Display**

1. Connect the VCC pin of the OLED display to the 5V pin of the Arduino.
2. Connect the GND pin of the OLED display to the GND pin of the Arduino.
3. Connect the SDA pin of the OLED display to A4 (SDA) of the Arduino.
4. Connect the SCL pin of the OLED display to A5 (SCL) of the Arduino.

**Step 4: Upload the Code to Arduino**

1. To interface the experiment, upload the provided Arduino code to the Arduino IDE and flash it onto the microcontroller.

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <TimeLib.h> // Time management library

#include <DHT.h> // Include the DHT library

#define MQ2\_PIN A0 // Pin for the MQ2 gas sensor

#define DHT\_PIN 3 // Pin for the DHT sensor (connected to D3 on Arduino)

#define DHT\_TYPE DHT11 // Change to DHT22 if using that sensor

DHT dht(DHT\_PIN, DHT\_TYPE); // Initialize DHT sensor

// OLED setup

#define OLED\_RESET -1

#define SCREEN\_ADDRESS 0x3C

Adafruit\_SSD1306 display(128, 64, &Wire, OLED\_RESET);

// Time variables

unsigned long previousMillis = 0; // Store last time update

const long interval = 1000; // Interval to update time every 1 second

int hours = 3; // Set initial time to 3:00:00

int minutes = 3;

int seconds = 0;

void setup() {

Serial.begin(9600);

if (!display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS)) {

Serial.println(F("SSD1306 allocation failed"));

for (;;);

}

dht.begin(); // Initialize the DHT sensor

display.clearDisplay();

display.setTextColor(SSD1306\_WHITE);

display.setTextSize(1);

display.display();

}

void loop() {

// Read current time every second

unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {

previousMillis = currentMillis;

// Update the time every second

seconds++;

if (seconds >= 60) {

seconds = 0;

minutes++;

}

if (minutes >= 60) {

minutes = 0;

hours++;

}

if (hours >= 24) {

hours = 0;

}

}

// Read the DHT sensor data

float temperature = dht.readTemperature(); // Temperature in Celsius

float humidity = dht.readHumidity(); // Humidity in percentage

// If reading fails, display error

if (isnan(temperature) || isnan(humidity)) {

display.clearDisplay();

display.setCursor(0, 0);

display.println(F("Failed to read from DHT sensor"));

display.display();

return;

}

int gasSensorValue = analogRead(MQ2\_PIN);

String airQuality = getAirQuality(gasSensorValue);

display.clearDisplay();

display.setCursor(0, 0);

// Display air quality

display.print("Air quality: ");

display.println(airQuality);

// Display temperature

display.print("Temperature: ");

display.print(temperature);

display.println(" C");

// Display RH%

display.print("RH%: ");

display.print(humidity);

display.println("%");

// Display real-time (HH:MM:SS)

display.print("Time: ");

display.print(hours);

display.print(":");

display.print(minutes);

display.print(":");

display.println(seconds);

display.display();

// Print gas sensor value to serial monitor

Serial.print("Gas Sensor Value: ");

Serial.println(gasSensorValue);

delay(100); // Small delay to avoid overwhelming the display

}

// Function to determine air quality based on MQ2 sensor value

String getAirQuality(int value) {

if (value < 185) return "Very Good";

else if (value < 190) return "Good";

else if (value < 195) return "Bad";

else return "Very Bad";

}

**Demonstration Video:**

A video of the project in action can be accessed at the following link: [Insert Video Link Here]

**Expected Results:**

1. The OLED display should show real-time temperature, humidity, and air quality readings.
2. The serial monitor should print the same values.
3. The readings should update every second.
4. The time simulation should increment HH:MM:SS correctly.

**Conclusion:**

This experiment successfully interfaced a DHT sensor and an MQ-2 gas sensor with an Arduino to measure environmental parameters. The data was displayed on an OLED screen along with a simulated real-time clock. This experiment provides a foundation for environmental monitoring applications in IoT and smart systems.