Drone and Robotics Laboratory,

Air Quality G Environment Monitoring System

Group- DR-1 (Manas Kumar Singh, Ayush Kumar, Naman Jha)

Roll- SBU25000- 26, 15 and 34

January 15, 2026

Contents

---

1 Introduction

This project implements an Air Quality and Environmental Monitoring System using an Arduino Uno, DHT11 temperature–humidity sensor, MQ gas sensor, and a 128×64 SPI OLED display (SSD1306). The system continuously measures temperature, humidity, and air quality, and displays the values on a compact graphical OLED screen.

Unlike standard industrial monitors, this system provides a low-cost, portable solution suitable for indoor air monitoring applications. By utilizing the Arduino IDE and its simplified C++ libraries, the project demonstrates how complex sensor data can be visualized in real-time.

---

2 Components Description

2.1 Arduino UNO

The Uno is the central microcontroller of the system. It consists of 14 digital I/O pins and 6 analog inputs[3]. It processes the digital signals from the DHT11 and the analog voltage from the MQ sensor to generate a visual output for the OLED.

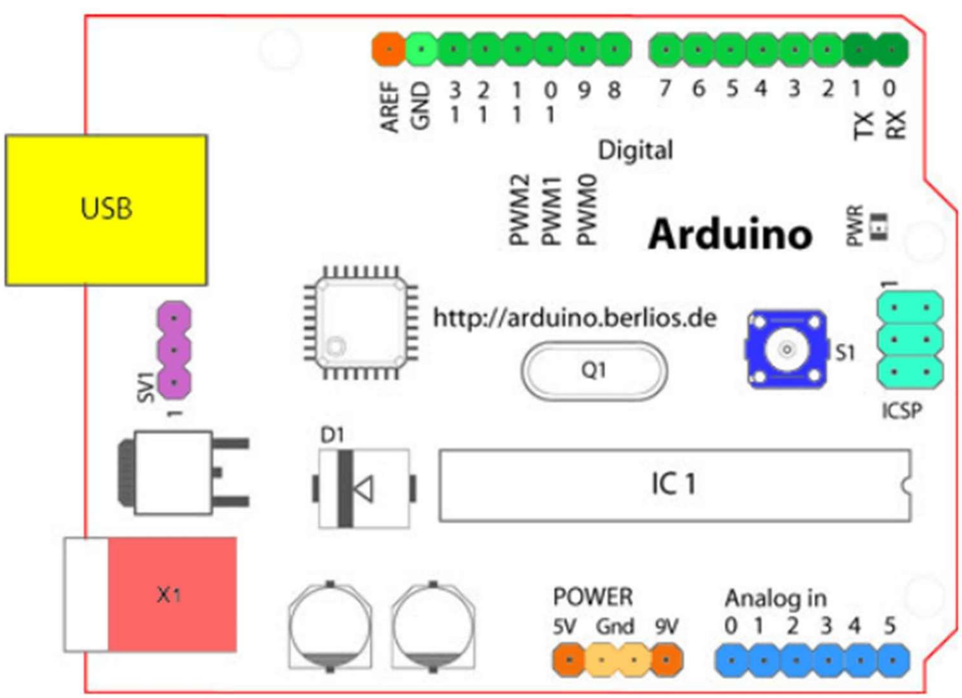
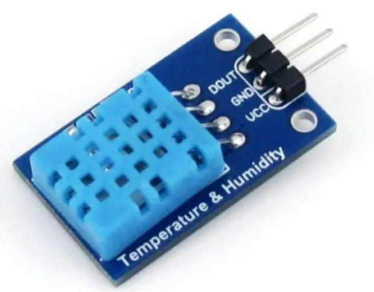


Figure 1: Common Components of Arduino Boards [1]

2.2 DHT11 Sensor

The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, spitting out a digital signal on the data pin.



2.3 MQ Gas Sensor (MQ-2 / MQ-135)

The MQ sensor series are used to detect gas concentrations. It outputs an analog voltage proportional to the concentration of smoke, CO, or other harmful gases in the air.

2.4 SSD1306 OLED Display (128x64 SPI)

A graphical display used to show real-time data. It is highly efficient and offers high contrast, making it superior to standard 16x2 LCDs for portable monitoring devices.

---

3 Project Implementation

3.1 Objective

The primary objective is to design a standalone embedded device that measures temperature, humidity, and air quality index (AQI) simultaneously and presents the data through a graphical interface.

3.2 Necessary Components

- Arduino UNO
- SSD1306 128×64 SPI OLED
- DHT11 Sensor
- MQ Gas Sensor (MQ-2 / MQ-135)
- Breadboard and Jumper Wires
- USB Cable / 9V External Power Source

3.3 Experimental Planning

The system follows a sequential data acquisition cycle:

1. **Sensing**: The DHT11 measures environmental variables digitally, while the MQ sensor provides an analog voltage.
2. **Processing**: The Arduino reads these inputs and converts the analog air quality value into status categories (GOOD / MOD / BAD).
3. **Visualization**: The processed data is formatted into strings and sent to the OLED via SPI communication.

3.4 Experimental Circuit Setup

The components were connected on the breadboard following the pin configurations below:

OLED Pin	**Arduino Pin**	**DHT11/MQ Pin**	**Arduino Pin**
GND	GND	DHT VCC / MQ VCC	5V
VCC	5V	DHT DATA	D2
SCL / SDA	D13 / D11	MQ Analog (A0)	A0
CS / DC / RES	D10 / D9 / D8	GND	GND

---

4 Observation

During the experiment, the system successfully initialized both sensors and the OLED screen. As shown in the Serial Plotter graphs (or on the OLED display), temperature and humidity remained stable in a room environment, while the Air Quality value fluctuated when exposed to different air samples.

The OLED displayed the following status based on the MQ sensor's analog reading:

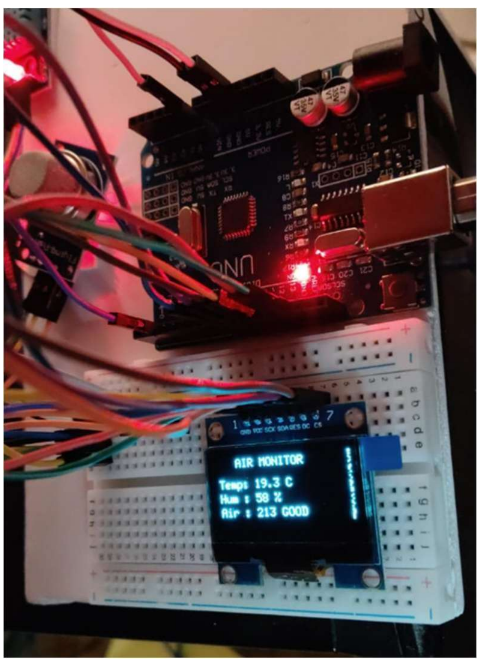
- **< 300**: GOOD
- **300 - 600**: MODERATE
- **> 600**: BAD

Discussion

1. **Sensitivity**: The MQ sensor requires a "warm-up" period of about 60 seconds to provide accurate readings.
2. **SPI Logic**: Using SPI for the OLED ensures a faster refresh rate compared to I2C, allowing for smoother data updates.
3. **Future Enhancements**: The addition of a buzzer alarm or a Wi-Fi module (like the ESP8266) would allow for remote monitoring and pollution alerts.

Conclusion

The project successfully demonstrates an embedded environmental monitoring system. According to the observations, the software and hardware synced effectively. This system serves as a reliable stepping stone for smart home integration and pollution awareness devices.



---

8 Appendices

8.1 Appendix 1: Complete Arduino Code

C++

```cpp
#include <U8g2lib.h>

#include <DHT.h>
```

```
/* OLED Display */
U8G2_SSD1306_128X64_NONAME_1_4W_HW_SPI u8g2(U8G2_R0, 10, 9, 8);

/* Sensors */
#define DHTPIN 2
#define DHTTYPE DHT11
#define MQ_PIN A0

DHT dht(DHTPIN, DHTTYPE);

void setup() {
 dht.begin();
 u8g2.begin();
}

void loop() {
 float temp = dht.readTemperature();
 float hum  = dht.readHumidity();
 int air   = analogRead(MQ_PIN);

 u8g2.firstPage();
 do {
  u8g2.setFont(u8g2_font_6x12_tf);
  u8g2.drawStr(23, 10, "AIR MONITOR");
  u8g2.setCursor(5, 28); u8g2.print("Temp: "); u8g2.print(temp, 1); u8g2.print(" C");
  u8g2.setCursor(5, 40); u8g2.print("Hum : "); u8g2.print(hum, 0); u8g2.print(" %");
  u8g2.setCursor(5, 52); u8g2.print("Air : "); u8g2.print(air);
```

```
    if (air < 300) u8g2.print(" GOOD");
    else if (air < 600) u8g2.print(" MOD");
    else u8g2.print(" BAD");
  } while (u8g2.nextPage());
  delay(2000);
}
```