Design and Implementation of Arduino Based Air Quality Meter Using Blynk

Abigail Frimpong
Queen Mary University
Corresponding email: a.frimpong@se24.qmul.ac.uk

Abstract – Air pollution is the contamination of the atmosphere, primarily due to human activities. Due to the lack of awareness, the air is being polluted by various sources; in recent years, it has become a major concern, largely driven by the growing number of factories, vehicles and mills. These sources have contributed significantly to air pollution by releasing smoke and toxic gases like Nitrogen dioxide (NO_2) and Particulate Matter (PM2.5 & PM10).

This project presents an Internet of Things (IoT) – based pollution monitoring system to measure hazardous gas levels in the air.

The data collected by the system can be viewed through a smartphone application; in this project we used Blynk.

Keywords: Internet of Things (IoT), Humidity sensor, Gas Sensor, Air Quality, Arduino Uno, Blynk Software.

1. Introduction

Air quality indicates to the condition or cleanliness of the air, often it is measured by the concentration of its constituent pollutants and harmful gasses like Nitrogen Dioxide (NO_2), Particulate Matter (PM), Carbon Monoxide (CO), Sulphur Dioxide (SO_2), Ozone (O_3), and more. Air quality is good when the air is free from harmful levels of these pollutants, which is crucial for human health and the environment. [1]It is important to monitor the air quality to identify pollution sources, track the air quality levels in particular areas and identify the implications on human health. Furthermore, air pollution also contributes to global warming [2]; which could cause acid rains and/or the melting of ice in the North Pole, which could lead to major floods.

To address the problem of toxic gas exposure in the environment, it is necessary to develop an air quality monitoring system that can provide real-time notifications about air quality levels.

This paper is presenting an economical, energy-efficient, IoT based air quality monitoring system using an Arduino Uno R4 WIFI microprocessor integrated with the Blynk App. The IoT system includes embedded air quality sensors, MCUs and software frameworks for data exchange and sharing information. Furthermore, the collected data can be utilized to offer instant updates, in a digital dashboard, on a smartphone, displaying real-time air quality readings while monitoring the effects of air quality in specific area. The final product can be used to analyse and compare the air quality levels of different areas such as

indoor, villages, city, industrial, haze, over a period of time.

2. System Specification

2.1 Temperature Sensor Module (DHT11)

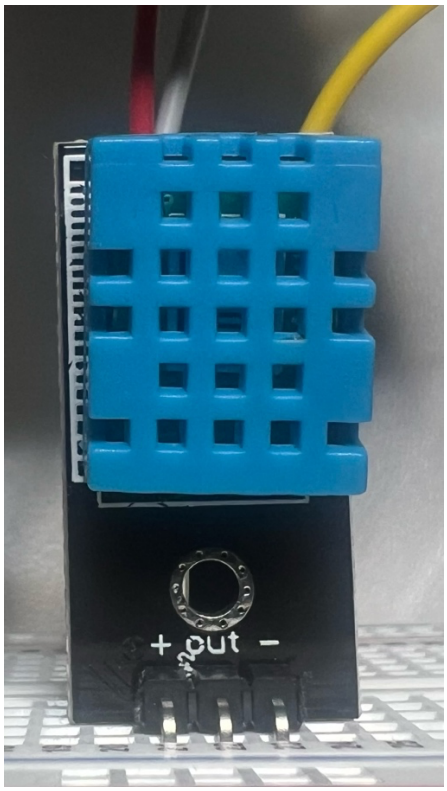

Figure 1: Temperature Sensor Module (DHT11)

The DHT11 Temperature & Humidity sensor consists of a sensor complex capable of measuring ambient temperature and humidity with a calibrated digital signal output. DHT11 provides high reliability and long-term stability due to its digital-signal-acquisition technique and temperature and humidity sensing ability. [3] The design of the temperature sensor consists of an NTC temperature measurement component and a resistive-type humidity measurement component, that connect to an 8-bit microcontroller with high-performance, providing fast response, excellent quality and cost-effectiveness. [3]

The sensor can be used in any microcontroller, such as Arduino Uno and Raspberry pi; and still giving rapid result.

The humidity sensor has two electrodes with moisture holding substance between them. With the change of humidity, the conductivity of the substrate changes or resistance between these electrodes changes; these are then processed by the IC. NTC temperature sensor or a thermistor is what measures the temperature. [3]

2.2 Gas Sensor Module (MQ-2)

The MQ-2 gas sensor operates on 5V DC and is capable of detecting Smoke, Alcohol, Carbon Monoxide concentrations, Propane and LPG. The MQ-2 is capable of detecting the gases but is unable to identify them. [4]

The MQ-2 is a heather-driven driven sensor covered by the Anti-explosion Network (two layers of fine stainless-steel mesh); the latter prevents explosion due to the flammable gasses and protects the sensing element and connecting legs from suspended particles.

Figure 2: Gas Sensor Module (MQ-2)

2.3 I2C LCD 2004

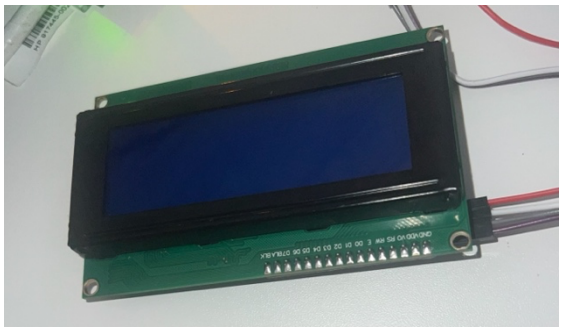

Figure 3: I2C LCD 2004

The LCD2004 consist of an I2C bus to prioritise high to low-speed device synchronization, necessary in multiple host system. There is a blue potentiometer on the I2C, used to regulate the backlight for the display on the LCD. [5]

2.4 Arduino UNO R4 WiFi

The Arduino UNO R4 Wifi has a 32-bit microcontroller and an ESP32-S3 Wi-Fi module (ESP3-S3-MINI-1-N8). [6]

The UNO R4 WiFi's main MCU is the R7FA4M1AB3CFM#AA0, often referred to as RA4M1, is connected to all pin headers and all communication buses of the board. [6]

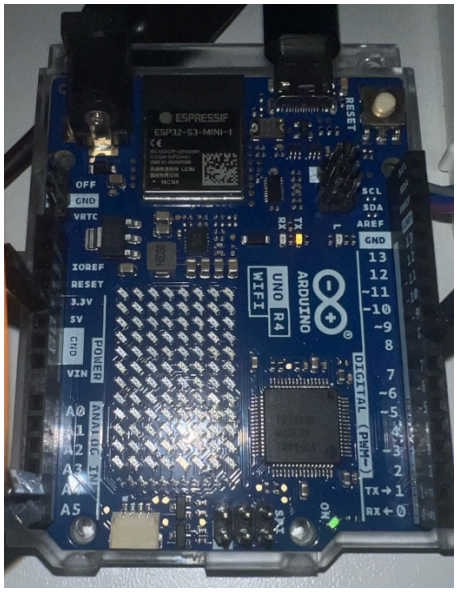

Figure 4: Arduino UNO R4 WIFI

3. Design & Architecture

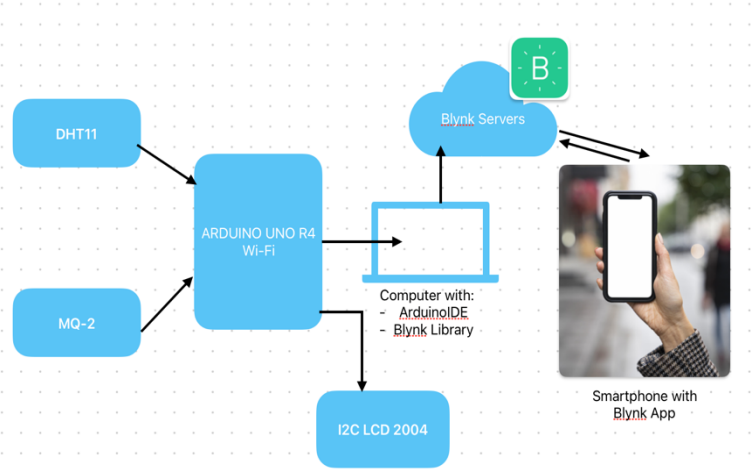

Figure 5: Proposed Design

3.1 Blynk

Blynk is a free software platform with iOS and Android applications to easily build mobile and web IoT applications and deploy and manage Arduino, Raspberry Pi projects.

It is used to display sensor data, store and visualize it, among different features.

Blynk offers both Cloud and local server storing. Blynk Libraries help the communication with the server and process ongoing and incoming commands. [7] A Pro subscription is required to obtain a wide range of features and accessibility.

3.2 Hardware Connections Design

1. **Sensing - DHT11 Temperature/ Humidity**:
   - **VCC**: 5V (Arduino)
   - **GND**: GND (Arduino)
   - **Data Pin**: Any digital pin (e.g., D2)
2. **Sensing - MQ-2 Gas Sensor**:
   - **VCC**: 5V (Arduino)
   - **GND**: GND (Arduino)
   - **A0 (Analog Pin)**: Analog input pin on Arduino (e.g., A0)
3. **Display- I2C LCD 2004**:

- **VCC**: 5V (Arduino)
- **GND**: GND (Arduino)
- **SDA**: SDA pin (on Arduino UNO: A4)
- **SCL**: SCL pin (on Arduino UNO: A5)

4. **Controller & Communication - Arduino UNO R4 WiFi**:
   - No specific connection to power or GND, just connect the required sensors and the LCD to the board pins as mentioned.

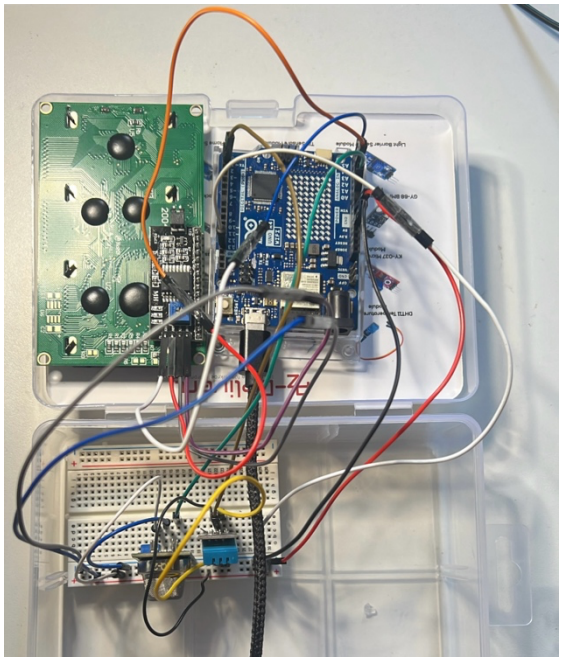

Figure 6: Picture of the hardware connections

3.3 Software Development

The air quality monitoring system consists of the peripheral devices with software.

The monitoring system includes several sensors, like the Gas and temperature sensors, MCUs and connections for communication that can be programmed by the Arduino UNO R4 WiFi.

After that, all virtual and hardware connections will be implemented and synchronized with the Blynk application; the latter controls the Arduino microcontroller on the internet. [7]

We will use Arduino Uno IDE application to edit the developed code to allow functionality upon detection of an air quality condition. The Arduino Uno Ide will check if the code is compiling successfully and is verifying. Upon success, the code will be uploaded on the Arduino Uno via its serial peripheral interface (SPI) bus.

Arduino Uno R4 Wi-fi will collect all the data from the sensors, process it, and upload it to the Blynk app over its integrated ESP32-S3 module which allows users to connect to Wi-Fi networks and perform network operations.

Then the data is sent to the I2C LCD 2004 module and Blynk applications.

The received data will be analysed and utilised on data visualisation for the end-user to receive the air quality updates and notifications based on the notification alerts (functions presents with the Blynk app on a smartphone).

3.4 Software Libraries

It is required to download some software libraries to connect both the hardware devices and to link Arduino to Blynk interface.

To connect the hardware devices to the Arduino UNO R4 Wifi, download the following libraries to Arduino IDE:

- $LiquidCrystal - I2C$ [8] available on Github. This provides the interface to connect the Arduino UNO to the LCD.
- $DHT11$ [9] also available to download on Github and on the Arduino IDE Library Manager [10].
- $AnalogRTCLib$ by Analog devices [10]. This library helps as an

interface for Analog devices (like the MQ-2 Gas Sensor) Real time clocks.

To implement and link the sensors output to Blynk download the following libraries to Arduino IDE:

- *Blynk* by Volodymyr Shymanskyy, downloadable from GitHub [11]. This allows the connectivity of the Arduino UNO R4 Wi-fi board with Ethernet, Wi-Fi, Cellular. This is necessary in order to connect the Arduino's IP address to the Wi-fi in use. The Arduino's IP address can be obtained by searching through the connected devices on the Hub Manager of the Wi-Fi router being used.

4. Design Implementation

Sign up to Blynk [7] or log in if you already have an account.

In the Developer Zone create a new device selecting Arduino Uno.

The Software will scan for all Wi-Fi connected devices and detected the Arduino UNO R4 Wifi, thus providing the Authorisation Token (see Figure 7), in this project $1D6yDP3tpZ7fqumRHDEpAD4\_lh9Ue9Qy$; the Blynk Template ID $TMPL5M0UMyNnd$; and the Blynk Template Name $UNO\ R4$; these are necessary Firmware Configuration (see Figure 8) to link our Arduino data to our Blynk app, offering Security, thus achieving a private/ secure IoT air Quality monitoring meter.

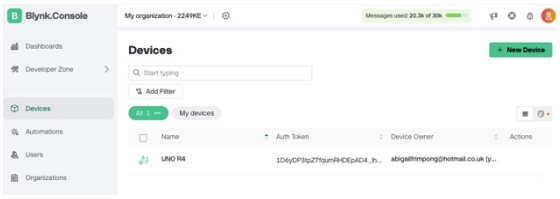

Figure 7:Blink Authorisation Token of the device

Figure 8: Firmware Configuration from Blink

After implementing the final code (Table 1 at the bottom of the page) on the Arduino Uno IDE, and checking successful connectivity to Blynk Cloud (through the output of the Arduino Uno IDE's Serial Monitor – see Figure 9), it was possible to proceed and visualise the data from the air monitoring devise on the smartphone Blynk App (downloadable on any Apple or Android smartphone); and on the Blynk.Console website. [7]

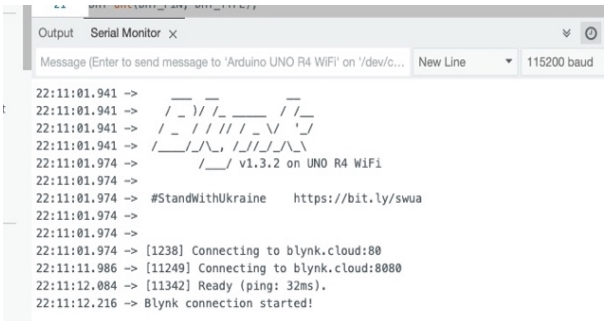

Figure 9 Output of the Serial Monitor confirming Success Wi-Fi and Blynk.Cloud connectivity

5. Testing and Visualisation

To test the Air Quality Monitor, the LCD is used to display the Real-Time data from the temperature and gas sensors (see picture 10 below for LCD view).

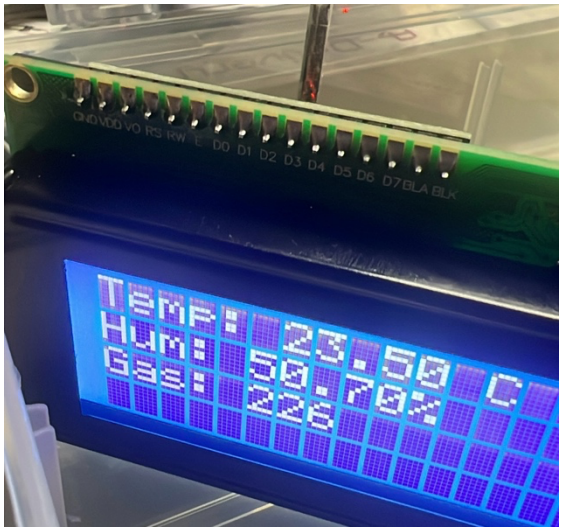

Figure 10 LCD view of output data of Air quality

Since it is required to visualise this in an IoT System, the Blynk widget features in the Web Dashboard is utilised to perform such task.

- *Temperature visualisation dashboard* was achieved by using the Labeled Value Widget Box. It is important to set the threshold between -8° and 35° C as any values outside these ranges aren't suitable living conditions for humans. Additionally, temperature above 30°-35° C is optimal for the growth of microbial organisms like Bacteria, Fungi, Moss. These affect the air quality, thus affecting the human health.
- *Humidity visualisation dashboard* was implemented with the Gauge Widget Box, at a threshold of max 80%. Any value after this would generate microbial growth in indoor applications and would be suffocating.
- *Gas visualisation dashboard* was achieved with the use of a Radial Widget Box, with threshold of 200. The value was obtained after monitoring the data over a long period of time, with different testing conditions. Any value above 200 implies the presence of air pollutants in high concentration.

Data was sent with a 2seconds interval (implemented in the code – Table1). This is to give the Blynk server adequate number of data/ requests without overflooding.

Blynk's Web view dashboard (Figure 11) provides a more ease implementation, whilst App view dashboard gives a more accessible and comfortable approach.

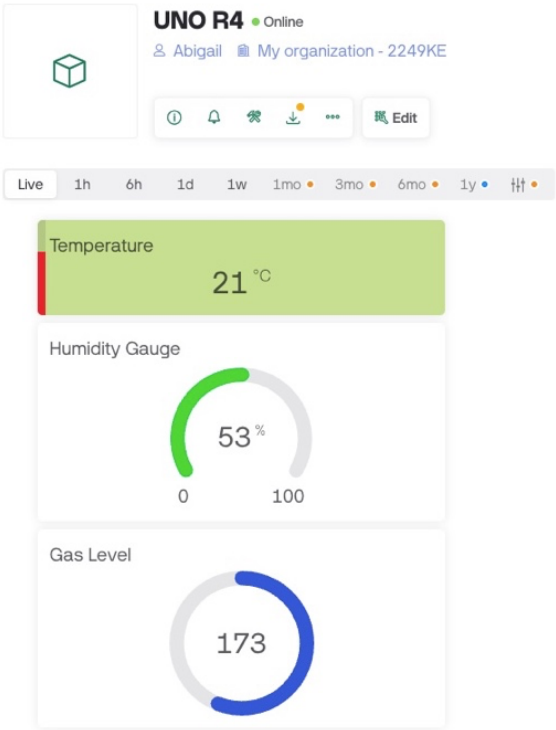

Figure 11Blynk's Web view dashboard for the Air Quality Monitoring Meter

With the smartphone application, it is possible to view the data in real time, at any

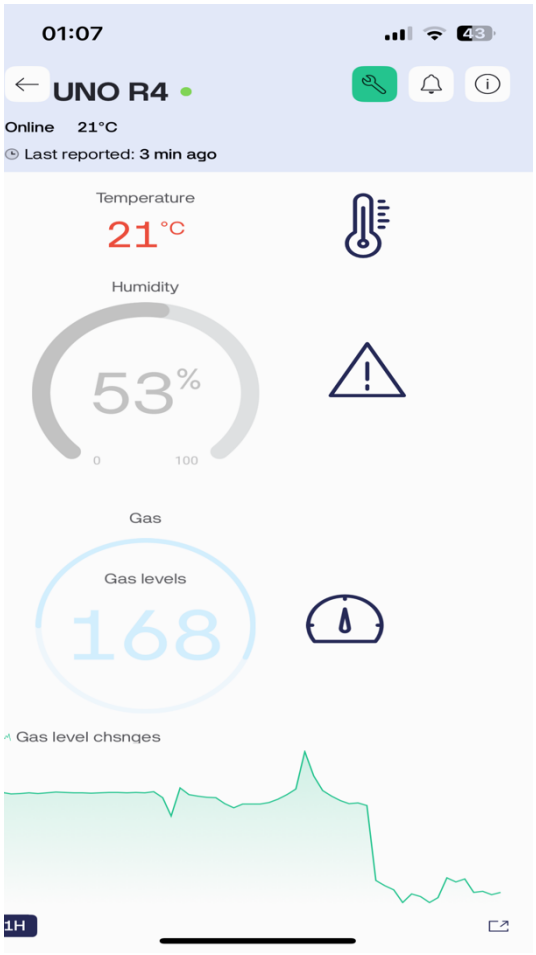

Figure12: Blynk Smartphone application air quality monitoring

location. It is possible to add Widgets to give the status of the device, the connectivity strength, the Gas level changes over a time (see Figure 12).

From the Blynk's widget dashboards it is possible to analyse and monitor the changes in air quality over a range of time, depending on preference and intended application.

To test the Air quality in an indoor environment, it is possible to place the device in an environment with high humidity such as in the kitchen or bathroom. It is visible through Simple Chats the changes that occur over time.

Below is a figure of the Air quality monitoring data over an hour period. The peak observed occurred by introducing heat generated by a lighter to the MQ-2 sensor.

Figure 12 Air quality changes over an hour

6. Conclusion

Air quality in this day and age is of notable consideration due to the rate of air pollution and the environmental and health implications caused by this.

Air quality monitoring helps improve the quality of air by eliminating the observed causes of changes in the air quality.

Air quality monitoring can be expensive but with this IoT implementation system, it is offered an accessible, cost effective and efficient way of keeping track of air pollution and environmental changes.

With the use of Blynk's dashboards on the smartphone, users experience a system that allow air quality monitoring in a target area, thus taking any actions or precautions to keep the air quality under control or to avoid the area entirely if unsafe to visit.

This is a highly user-friendly, economical and with urgent necessity application.

```
#define BLYNK_PRINT Serial
// Blynk credentials from Blynk app
#define BLYNK_TEMPLATE_ID "TMPL5M0UMyNnd"
#define BLYNK_TEMPLATE_NAME "UNO R4"
#define BLYNK_AUTH_TOKEN "1D6yDP3tpZ7fqumRHDEpAD4_lh9Ue9Qy" // Get from Blynk app
char auth[] = "1D6yDP3tpZ7fqumRHDEpAD4_lh9Ue9Qy";

#include <SPI.h>
#include <WiFiS3.h>              // Wi-Fi library for Arduino UNO R4 WiFi
#include <BlynkSimpleWiFiS3.h>   // Blynk library for Wi-Fi
#include <DHT.h>                 // DHT sensor library
#include <LiquidCrystal_I2C.h>   // LCD library for I2C display

// Wi-Fi credentials from my router
char ssid[] = "BT-J7C2TJ";
char pass[] = "GhYmhCDJCbPV79";

// DHT Sensor Setup (DHT11 or DHT22)
#define DHT_PIN 2                // Pin for DHT sensor
#define DHT_TYPE DHT11           // DHT11 or DHT22
DHT dht(DHT_PIN, DHT_TYPE);

// MQ-2 Gas Sensor Setup
#define MQ2_PIN A0               // Pin for MQ-2 sensor

// LCD Setup
LiquidCrystal_I2C lcd(0x27, 20, 4);  // 0x27 is the I2C address for many LCD screens, adjust if
needed

void setup() {
  Serial.begin(115200);
  delay(1000);

  Serial.println("Connecting to WiFi...");

  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }

  Serial.println("\nConnected to WiFi!");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());

  // Connect to Blynk
  Blynk.begin(auth, ssid, pass);
  Serial.println("Blynk connection started!");

  // Initialize DHT sensor
  dht.begin();

  // Initialize LCD
  lcd.begin();
  lcd.print("Air Quality");
  lcd.setCursor(0, 1);
  lcd.print("Temp: --- C  Hum: ---%");
}
```

```
void loop() {
  // Run Blynk
  Blynk.run();

  // Read temperature and humidity from DHT sensor
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  // Read gas levels from MQ-2 sensor (adjust this based on your needs)
  int mq2Value = analogRead(MQ2_PIN);

  // Send data to Blynk app
  Blynk.virtualWrite(V1, temperature);  // Virtual Pin V1 for temperature
  Blynk.virtualWrite(V2, humidity);     // Virtual Pin V2 for humidity
  Blynk.virtualWrite(V3, mq2Value);    // Virtual Pin V3 for MQ-2 value

  // Display the temperature snd humidity concentration on LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temperature);
  lcd.print(" C");
  lcd.setCursor(0, 1);
  lcd.print("Hum: ");
  lcd.print(humidity);
  lcd.print("%");

  // Display gas level
  lcd.setCursor(0, 2);
  lcd.print("Gas: ");
  lcd.print(mq2Value);

  delay(2000);  // Delay between updates
}
```

Table 1: Code implementation of IoT Air Quality Monitoring meter

Works Cited

[1] A. N. X. K. a. Á. L. P. Völgyesi, “Air Quality Monitoring with SensorMap,,” in *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008),*, St. Louis, MO, USA, 2008.

[2] W. H. Organization, “Air quality, energy and health,” [Online]. Available: https://www.who.int/teams/environment-climate-change-and-health/air-quality-energy-and-health/health-impacts/climate-impacts-of-air-pollution#:~:text=Air%20pollutants%2C%20such%20as%20methane,than%20carbon%20dioxide%20(CO2)..

[3] Mouser Electronics, “DHT11 Humidity & Temperature Sensor,” [Online]. Available: https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf?srsltid=AfmBOore7i-CPKXx9JpyeCpzyQEOG2VMSgXQavscNkc2ltzN0Cy_Az-N.

[4] “How MQ2 Gas/Smoke Sensor Works? & Interface it with Arduino,” [Online]. Available: https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/.

[5] “ I2C LCD2004,” [Online]. Available: http://wiki.sunfounder.cc/index.php?title=I2C_LCD2004.

[6] “Arduino® UNO R4 WiFi,” [Online]. Available: https://docs.arduino.cc/resources/datasheets/ABX00087-datasheet.pdf.

[7] “Blynk,” [Online]. Available: https://blynk.io/.

[8] j. a. fdebrabander, “fdebrabander / Arduino-LiquidCrystal-I2C-library,” [Online]. Available: https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library.git.

[9] dhrubasaha08, “dhrubasaha08 / DHT11,” [Online]. Available: https://github.com/dhrubasaha08/DHT11.git.

[10] Arduino, “Arduino IDE,” [Online]. Available: https://www.arduino.cc/en/software/.

[11] V. Shymanskyy, “blynkkk / blynk-library,” Github, [Online]. Available: https://github.com/blynkkk/blynk-library.