FRANKFURT
UNIVERSITY
OF APPLIED SCIENCES

Arduino-based Real-time air Quality and Pollution Monitoring System

Patel Harshkumar Ganpatbhai
Frankfurt University of Applied Sciences, Germany
harshkumar.patel@stud.fra-uas.de

Patel Parth
Frankfurt University of Applied Sciences, Germany
parth.patel@stud.fra-uas.de

Prajapati Riddhi
Frankfurt University of Applied Sciences, Germany
riddhi.prajapati@stud.fra-uas.de

Abstract: - In this study, carbon monoxide CO and air quality were measured using MQ135 and MQ7 Sensors, respectively. Measurement of air quality is a crucial step in raising public awareness of the need to ensure the health of future generations. Due to the recent rapid industrialization, there is an enormously growing need for individuals to monitor their local air quality in order to understand how they live and what they breathe. In this work, we suggested an Arduino-based real-time air quality and pollution monitoring system. This environmental air monitoring system is made to offer a reliable, simple and effective way to continually and in-the-moment check the air quality.

Keyword: - Air Pollution, Arduino, gas sensor, atmosphere, MQ-7, MQ-135, Temperature Sensor, Adapter.

1. INTRODUCTION

During the past few years, air pollution has grown to be a risky issue. The health of people is directly impacted by this element. Several countries are worried about global warming, one major issue is air pollution. The production of harmful gasses by companies, automobile emissions, and an increase in the amount of harmful chemicals and particulate matter in the atmosphere are all contributing to air growth, and automobile use, the level of pollution is increasing rapidly. Increased health effects of air Pollution include lung cancer, ischemic heart disease, asthma symptoms, and so on. The presence of one or more pollutants in the atmosphere, such as gases, in an amount that directly harms people, animals, and plants is referred to as air pollution[1]. Particle pollution is one of the most important factors causing a major rise in environmental pollution. This requires the measurement and analysis of real-time air quality monitoring in order to allow for prompt decision-making. Various air quality monitoring methods have been developed. However, the majority of these technologies are costly and unsuitable for long-term monitoring. The development of low-cost sensors and microcontrollers has enabled the development of low-cost and effective air quality monitoring devices. A low-cost gas sensor that is compact, reliable, and adaptable could be an equally effective option. Electrochemical, infrared, reactive particles, photo ionization, and solid-state monitoring methods for gases [2]. Visibility could be additionally limited by polluted air. One in eight premature deaths in the world each year, or 7 million people, are due to airborne pollutants [3]. IoT also enables the development of intelligent environments where objects communicate and work together [4]. In this research, we offer a real-time air quality and pollution monitoring system based on Arduino.

2. STATE OF THE ART

The Arduino platform provides an easy and affordable way to build devices for monitoring pollution and air quality. The utilization of different sensors, communication technology, and data analysis methods is the state of the art in this subject. Gas sensors, particle sensors, temperature and humidity sensors, and light sensors are some of the sensors utilized in air quality monitoring systems. These sensors are employed to determine the concentrations of several pollutants, including particulate matter, ozone, nitrogen oxides, sulfur dioxide, and carbon monoxide. Wi-Fi, Bluetooth, and GSM/GPRS are just a few of the communication technologies that are utilized to send the data collected by the sensors to a centralized monitoring system. Artificial intelligence and machine learning are used as data analysis tools to analyze the monitoring system's data.

3. METHODS AND MATERIALS

- Arduino Uno (Olimexino)
- MQ-135 (Air Quality and Hazardous Gas Sensor)
- MQ-7 (Carbon Monoxide Gas Sensor)
- Temperature Sensor
- Humidity Sensor
- Dust Sensor
- LCD

Arduino Uno

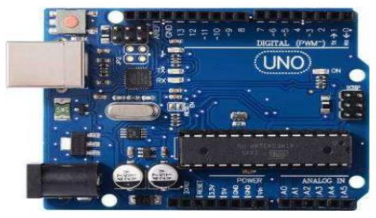


Fig 3.1 Arduino Uno (Micro-Controller)

The microcontroller Arduino can work with a number of different sensors. The board can be used with the greatest amount of effectiveness due to its simplicity and accessibility of a number of equipment extensions. This is connected to the board and LCD.

Breadboard

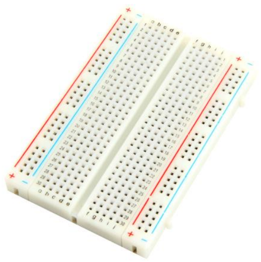


Fig 3.2 Breadboard

As electronics development and exploration, a breadboard is a common tool. It consists of a rectangular board with horizontal and vertical rows and columns of holes into which wires and electrical parts can be inserted and connected without the use of solder.

MQ-135(Air Quality and Hazardous Gas Sensor)

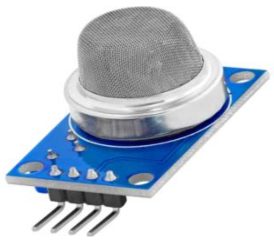


Fig 3.3 MQ-135(Air Quality & Hazardous Gas Sensor)

Ammonia, benzene, nitric oxide, smoke gases, argon gases, and many other gases are among the wide spectrum of chemicals that MQ135 is used to detect, making it possible to use this system both indoors and outdoors.

MQ-7 (Carbon Monoxide Gas Sensor)

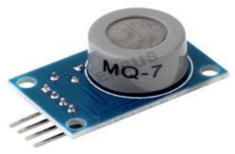


Fig 3.4 MQ-7(Carbon Monoxide Gas Sensor)

MQ7 is used to identify carbon monoxide, which is the primary cause of air pollution and the cause of many serious illnesses in humans. MQ7 Semiconductor sensor for Carbon Monoxide. In this system it connect with breadboard.

Temperature Sensor

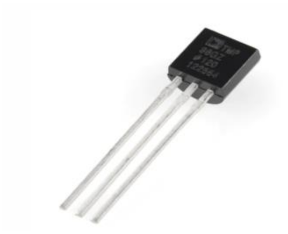


Fig 3.5 Temperature Sensor

In task mechanization, temperature is the procedure variable that is most frequently measured. The important instrument for monitoring temperatures in industrial systems is temperature sensors. It is used as a humidity sensor.

Humidity Sensor

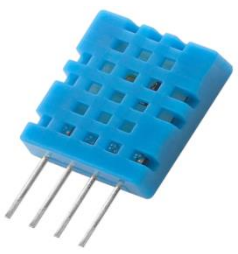


Fig 3.6 Humidity Sensor

A humidity sensor is a device that calculates the relative humidity in a specific area. Humidity sensors come in analog and digital varieties. This sensor works as a temperature sensor and it will detect particles in the air which are bad for human life.

Dust Sensor

Fig 3.7 Dust Sensor

It is used to identify the little particles in the air that will eventually cause larger contaminations. It is nowhere close to being capable of detecting the best molecules that are larger than 0.8µm, much like cigarette smoke.

LCD Display

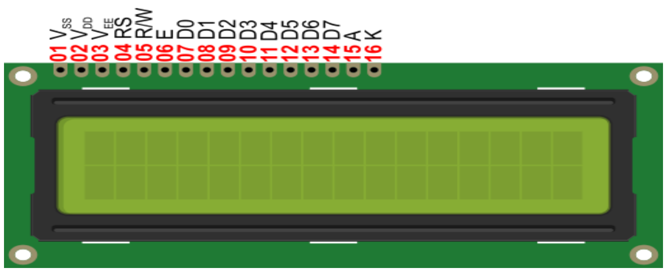


Fig 3.8 LCD Display

Liquid cell shows (LCDs) accustomed show of display of numeric and alphanumeric characters in matrix and segmental displays. The parallel interface of LCDs require simultaneous control of many interface pins by the microcontroller in order to control the display. The following pins contribute the connection:

- A register (RS) pin that controls where data is written to the LCD's memory.
- The Read/Write (R/W) pin, that chooses between reading and writing mode.
- The registers can be written to using an Enable pin.
- 8 Data pins.

4. BLOCK DIAGRAMS

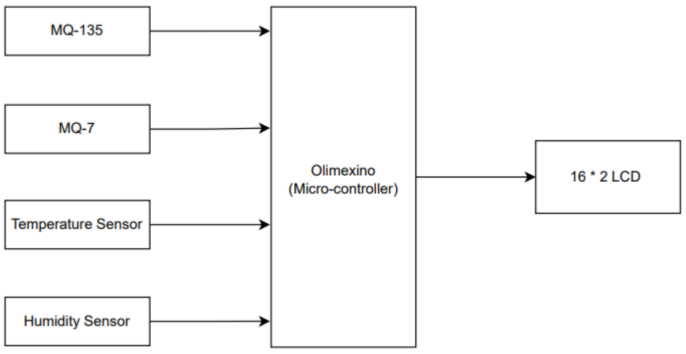


Fig 4. Block Diagrams

Above the system diagram Using the MQ-135, MQ-7, Temperature Sensor and Humidity Sensor are connected Olimexino Micro-controller board and these both are interconnected with LCD which shown output of the system. The system uses MQ-135 and MQ-7 gas sensors to measure the concentration of carbon dioxide, ammonia, nitrogen oxides, and carbon monoxide gases in the air. The sensors are connected to the analog input pins of the Arduino board, which processes the data and displays it on the LCD screen.

5. SYSTEM DESIGN

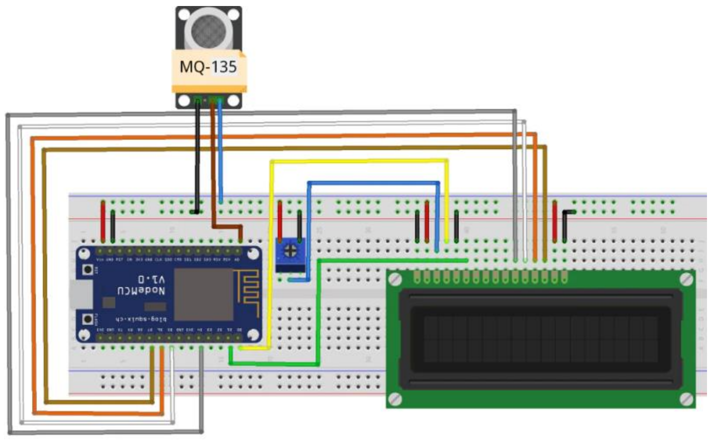


Fig 5. System Design

As shown in circuit, first we connect power to the Arduino board, using the MQ 135 sensor starts looking for gases like ammonia, benzene, carbon dioxide etc and it indicates the output voltage according to the different gas of particles in the air. In the process DH11 Humidity sensor will detect the temperature and relative humidity values and start sending it to the Arduino board. As the hardware implementation of the system involves connecting the sensors, microcontroller these are connected with each other using jumper wires. Connect the MQ-7 sensor to analog input A0 of the Arduino. Connect the MQ-135 sensor to analog input A1 of the Arduino. Connect the DHT22 sensor to digital pin D8 of the Arduino. The system is powered by USB Cable. The software implementation of the system involves programming the Arduino using the Arduino integrated development (IDE).

6. SYSTEM FLOWCHART

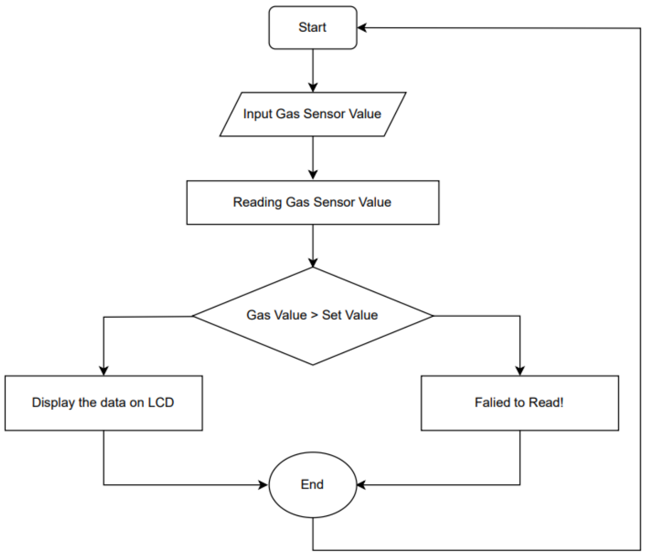


Fig 6.1 Gas Sensor Flowchart

The system will start first, and after that, we will provide the inputs. Sensors read the input and display the value, which we display as the value range from minimum to maximum gas air pollution. Which is good or bad.

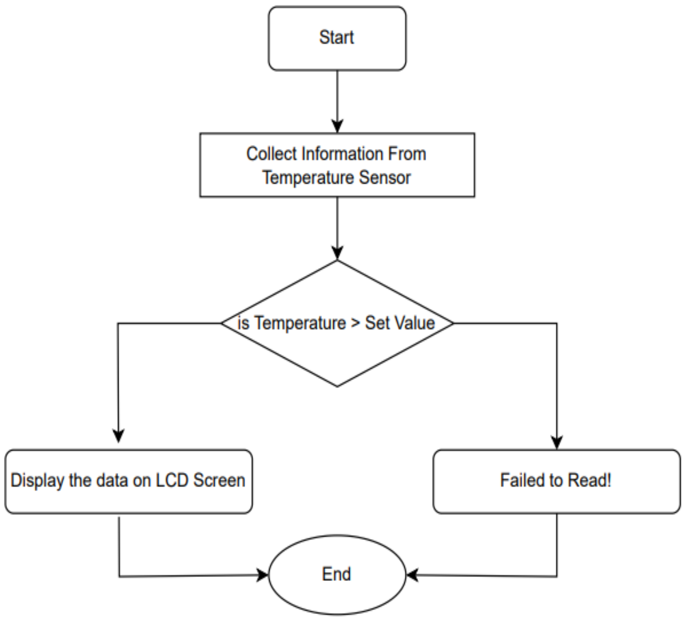
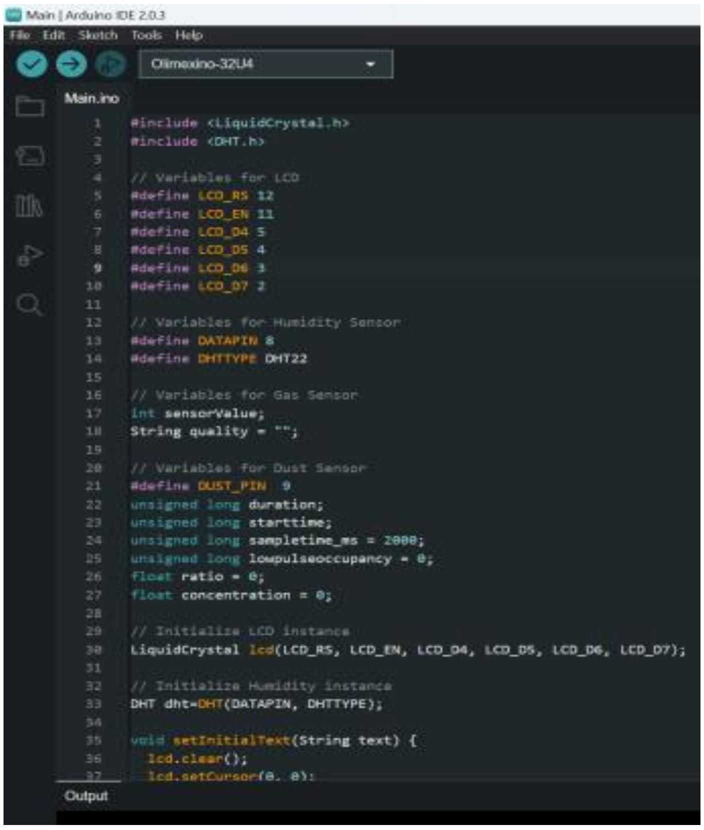

Start
Collect Information From Temperature Sensor
is Temperature > Set Value
Display the data on LCD Screen
Failed to Read!
End

Fig 6.2 Temperature Sensor Flowchart

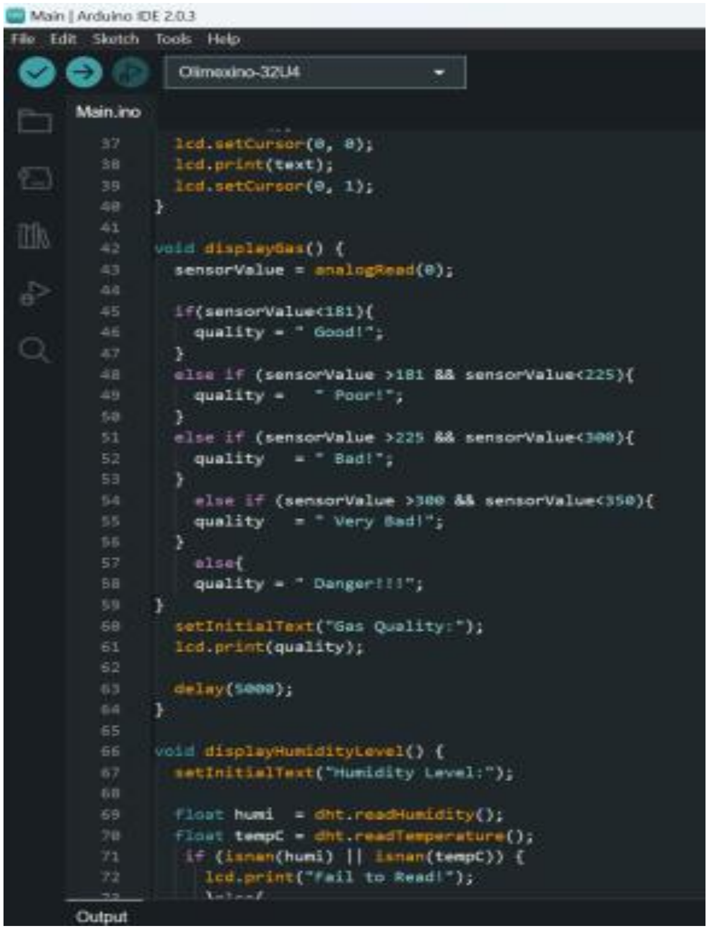
In the second flow chart, we calculate the temperature of air polluted. System will start to collect information from the air particles (dust particles), which is indicate the humidity level of air, and display the data or value on display of LCD. If the air was good, it indicates on the LCD screen and the current temperature of the weather.

7. SYSTEM CODING



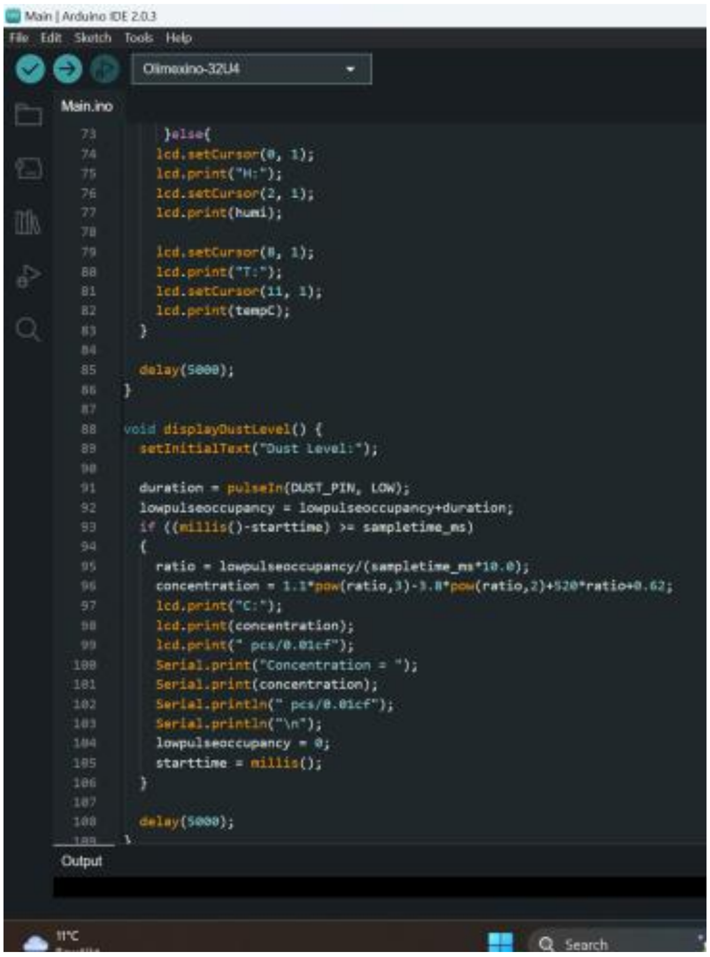
```
#include <LiquidCrystal.h>
#include <DHT.h>

// Variables for LCD
#define LCD_RS 12
#define LCD_EN 11
#define LCD_D4 5
#define LCD_D5 4
#define LCD_D6 3
#define LCD_D7 2

// Variables for Humidity Sensor
#define DATAPIN 8
#define DHTTYPE DHT22

// Variables for Gas Sensor
int sensorValue;
String quality = "";

// Variables for Dust Sensor
#define DUST_PIN  9
unsigned long duration;
unsigned long starttime;
unsigned long sampletime_ms = 2000;
unsigned long lowpulseoccupancy = 0;
float ratio = 0;
float concentration = 0;

// Initialize LCD instance
LiquidCrystal lcd(LCD_RS, LCD_EN, LCD_D4, LCD_D5, LCD_D6, LCD_D7);

// Initialize Humidity instance
DHT dht=DHT(DATAPIN, DHTTYPE);

void setInitialText(String text) {
  lcd.clear();
  lcd.setCursor(0, 0);
```

Fig 7.1 System Coding



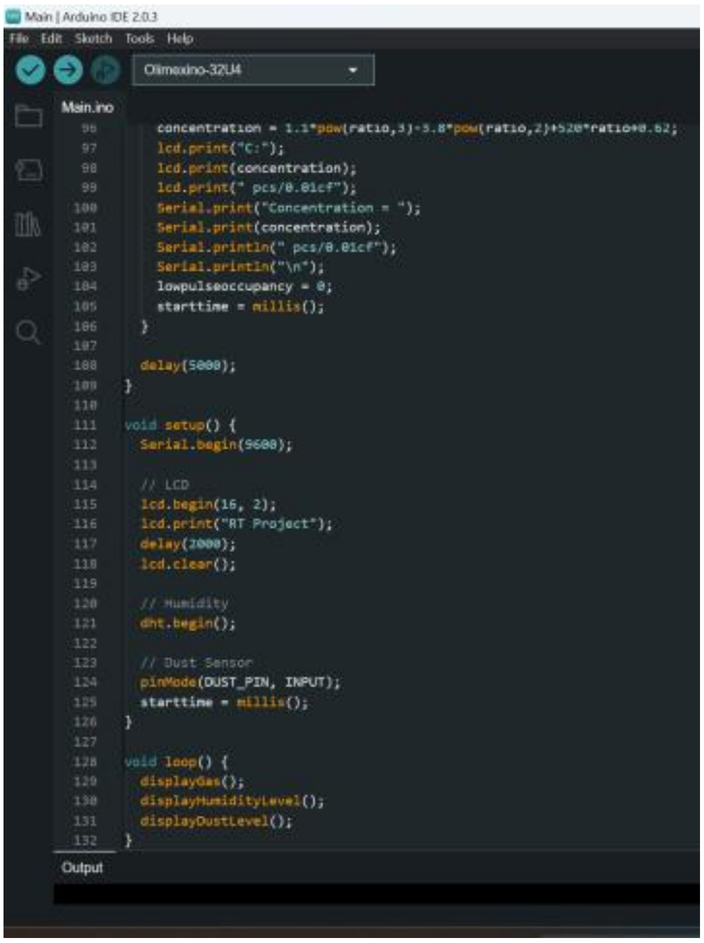
```
  lcd.setCursor(0, 0);
  lcd.print(text);
  lcd.setCursor(0, 1);
}

void displayGas() {
  sensorValue = analogRead(0);

  if(sensorValue<181){
    quality = " Good!";
  }
  else if (sensorValue >181 && sensorValue<225){
    quality =    " Poor!";
  }
  else if (sensorValue >225 && sensorValue<300){
    quality    = " Bad!";
  }
    else if (sensorValue >300 && sensorValue<350){
    quality    = " Very Bad!";
  }
    else{
    quality = " Danger!!!";
}
  setInitialText("Gas Quality:");
  lcd.print(quality);

  delay(5000);
}

void displayHumidityLevel() {
  setInitialText("Humidity Level:");

  float humi  = dht.readHumidity();
  float tempC = dht.readTemperature();
   if (isnan(humi) || isnan(tempC)) {
     lcd.print("Fail to Read!");
```

Fig 7.2 System Coding



```
    }else{
    lcd.setCursor(0, 1);
    lcd.print("H:");
    lcd.setCursor(2, 1);
    lcd.print(humi);

    lcd.setCursor(8, 1);
    lcd.print("T:");
    lcd.setCursor(11, 1);
    lcd.print(tempC);
  }

  delay(5000);
}

void displayDustLevel() {
  setInitialText("Dust Level:");

  duration = pulseIn(DUST_PIN, LOW);
  lowpulseoccupancy = lowpulseoccupancy+duration;
  if ((millis()-starttime) >= sampletime_ms)
  {
    ratio = lowpulseoccupancy/(sampletime_ms*10.0);
    concentration = 1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62;
    lcd.print("C:");
    lcd.print(concentration);
    lcd.print(" pcs/0.01cf");
    Serial.print("Concentration = ");
    Serial.print(concentration);
    Serial.println(" pcs/0.01cf");
    Serial.println("\n");
    lowpulseoccupancy = 0;
    starttime = millis();
  }

  delay(5000);
}
```

Fig 7.3 System Coding



```
    concentration = 1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62;
    lcd.print("C:");
    lcd.print(concentration);
    lcd.print(" pcs/0.01cf");
    Serial.print("Concentration = ");
    Serial.print(concentration);
    Serial.println(" pcs/0.01cf");
    Serial.println("\n");
    lowpulseoccupancy = 0;
    starttime = millis();
  }

  delay(5000);
}

void setup() {
  Serial.begin(9600);

  // LCD
  lcd.begin(16, 2);
  lcd.print("RT Project");
  delay(2000);
  lcd.clear();

  // Humidity
  dht.begin();

  // Dust Sensor
  pinMode(DUST_PIN, INPUT);
  starttime = millis();
}

void loop() {
  displayGas();
  displayHumidityLevel();
  displayDustLevel();
}
```

Fig 7.4 System Coding

8. RESULTS

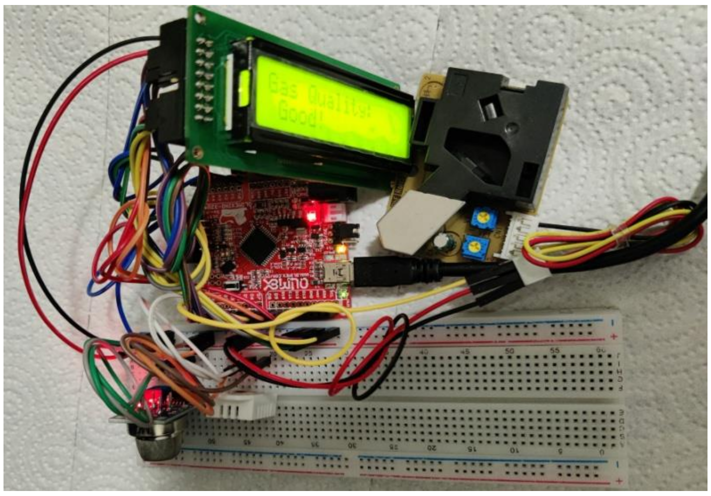


Fig 8 Result

According to the implementation, we connected an Arduino uno to the mentioned sensors, and they are all joined via an adapter, which is connected to a battery. As we supplied smoke or some gas to the MQ 7, it indicated how much good or bad gas was present in the air. We can analyze that normal mosquito coil is not too bad for humans, while some industrial coil. bad particles in the air indicate how much bad dust in the air, which is very harmful for human lives.we have defined system software and hardware both in this literature. It is efficient and user friendly air quality detection system.

9. CONCLUSIONS

A highly effective air quality monitoring system that we have created is an Arduino-based air pollution detector. We can conclude that it is based on the performance because Comparable in capability to more expensive current air pollution detectors, and simple to operate. It is a portable device using a microcontroller. The air quality detection system is effective and simple to operate.

10. ACKNOWLEDGMENTS

Prof. Dr. Matthias Deegener has been a great help to us during this study, and we would like to thank him. This research report was prepared as a part of the RT (Advanced Real Time Systems) course for the High Integrity Systems master's degree program.

11. REFERENCES

1. Arun Raj V., Priya R.M.P., and Meenakshi, V., "Air Pollution Monitoring In Urban Area," International Journal of Electronics and Communication Engineering , 2017 .

2. Nihal Kularatna, and B.H. Sudantha, (2008). An Environmental Air Pollution Monitoring System. IEEE 1451 Standard for Low Cost Requirements. IEEE Sensors Journal; 8(4), pp 415-422.

3. Nghi Dam, Andrew Ricketts, Benjamin Catlett, Justin Henriques , "Wearable Sensors for Analyzing Personal Exposure to Air Pollution," IEEE , 2017.

4. Priyanka V. Shitole, Dr. S. D. Markande2, "Review: Air Quality Monitoring System," International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 6, 2016.

5. W. Y. Yi, K. M. Lo, T. Mak, K. S. Leung, Y. Leung, and M. L. Meng, “A Survey of Wireless Sensor Network Based Air Pollution Monitoring Systems”, Sensors, vol. 15, No. 12, pp. 31392–31427. Dec. 2015, doi:10.3390/s151229859.

6. Perumal, B., Deny, J., Alekhya, K., Maneesha, V., & Vaishnavi, M. (2021, August). Air Pollution Monitoring System by using Arduino IDE. In 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 797-802). IEEE.