GOVERNMENT COLLEGE OF ENGINEERING, ERODE

அரசினர் பொறியியல் கல்லூரி, ஈரோடு
Government College of Engineering, Erode
(Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai)

B.E Electronics and Communication Engineering

AIR QUALITY MONITORING

Done By

MITHUN KUMARAN G - 731121106032

Under the mentor of **Dr.M.SATHYAKALA**

Department of Information Technology (IT)

Department of Electronics and Communication Engineering.

Government College of Engineering

Erode ,PO ,near Vasavi College,TamilNadu-638316,Affiliated to Anna University ,Chennai.

AIR QUALITY MONITORING

Introduction:

Continued exposure to environments with poor air quality is a major public health concern in developed and developing countries. It is estimated that the pollutants responsible for poor air quality cause nearly 2.5 million premature deaths per year world-wide. Significantly, around 1.5 million of these deaths are due to polluted indoor air, and it is suggested that poor indoor air quality may pose a significant health risk to more than half of the world's population. Due to its link with industrialisation, societal health problems associated with poor air quality disproportionately affects developed and developing nations – it is estimated that air pollution is responsible for the premature deaths. Remedial action to improve air quality is often easy to implement once airborne pollutants have been detected.

Need for Air Quality Monitoring:

Air pollution has critical impacts on our health and the state of our environment, and it is responsible for significant economic consequences due to these harms. Thus, working towards cleaner air through effective air quality monitoring and management is an essential part of building a healthier and more sustainable future.

Consequences of Poor Air Quality:

- Air pollution has significant economic consequences due to the variety of ways that poor air quality impacts human and environmental health.

- Data from the World Health Organization (WHO) and the Organisation for Economic Cooperation and Development (OECD) in 2015 found an estimated USD 1.6 trillion in costs due to premature death and disability from air pollution in Europe.

- According to data from Stanford University, air pollution costs the US about 5% of its GDP in damages, which totalled to $790 billion in 2014.

- The biggest air pollution-related damages are concentrated in the sectors of agriculture, utilities, manufacturing, and transportation.

- **Types of air quality monitoring systems:**

❖ Air quality is measured at a variety of levels — including federal, state, regional, and local levels. Air quality can change greatly between relatively close geographic locations, making it important to have a dense air pollution measurement network that reports air quality at both a regional and smaller scale.

❖ At the federal level, reference-grade monitors often make up the national air quality monitoring network.

- ❖ State-level networks monitor air pollution at the state level. In California, the state-wide ambient air quality monitoring network has more than 250 monitoring stations placed across the state.

Objectives of Air Quality Monitoring:

- ✓ Characterize the nature and severity of air pollution in the state
- ✓ Measure air quality against the Ambient Air Quality Standards and determine attainment status
- ✓ Identify trends in pollutant concentrations
- ✓ Provide regularly updated air quality information
- ✓ Determine levels of community exposure to harmful pollutants

IOT device Setup:

Temperature and Humidity sensor (DHT11)

• Air Quality sensor (MQ 135)

• 2n2222 Transisitor

• DC Fan

• Potentiometer

• 16x2 LCD Panel

• NodeMCU

• Arduino Uno

For Power Supply:-

• Step down transformer (12-0-12 V,1 A)

• Diodes

• Voltage Regulator (7805)

• Capacitors (0.01 micro Farad, 470 micro Farad)

• Wires

SOFTWARE REQUIREMENTS:

- Arduino (Version 1.8.2)
- THINGSPEAK

DEVICE WORKING:

- ✓ The design included various units mainly: sensing unit, processing unit, power unit, display unit and communication unit.
- ✓ The system operates over an existing WIFI wireless network utilizing the MQTT protocol.
- ✓ It is capable of monitoring the indoor air quality as well as controlling an air purifier to regulate the particulate matters concentration.

DATASET:

AQI LEVEL	AQI RANGE	Description of Air Quality
Good	0 – 50	Air quality is satisfactory, and air pollution poses little or no risk.
Moderate	51 – 100	Air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution.
Unhealthy for Sensitive Groups	101 – 150	Members of sensitive groups may experience health effects. The general public is less likely to be affected.
Unhealthy	151 – 200	Some members of the general public may experience health effects; members of sensitive groups may experience more serious health effects
Very Unhealthy	201 – 300	Health alert; The risk of health effects is increased for everyone.
Hazardous	301 above	Health warning of Emergency conditions everyone is more likely to be affected.

CODING:

```
#define BLYNK_TEMPLATE_ID "TMPLgwKssgggsnFXp"
#define BLYNK_DEVICE_NAME "Air Quality Monitoring"
#define BLYNK_AUTH_TOKEN "k03gT6nJosdsfsffesrJV_S5SXEAdgdsdghhgPZvXEwSKDfj"
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <BlynkSimpleEsp8266.h>

#include <DHT.h>
```

```
//#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
   byte degree_symbol[8] =
{
                0b00111,
                0b00101,
                0b00111,
                0b00000,
                0b00000,
                0b00000,
                0b00000,
                0b00000
              };
char auth[] = BLYNK_AUTH_TOKEN;
 char ssid[] = "";  // type your wifi name
char pass[] = "";  // type your wifi password

BlynkTimer timer;
 int gas = A0;
int sensorThreshold = 100;

#define DHTPIN 2 //Connect Out pin to D2 in NODE MCU
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);


void sendSensor()
{    float h =
dht.readHumidity();   float t =
dht.readTemperature(); // or
dht.readTemperature(true) for
Fahrenheit

     if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
return;   }
   int analogSensor = analogRead(gas);
  Blynk.virtualWrite(V2, analogSensor);
  Serial.print("Gas Value: ");
    Serial.println(analogSensor);
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
    Blynk.virtualWrite(V0, t);
    Blynk.virtualWrite(V1, h);

    Serial.print("Temperature : ");
    Serial.print(t);
    Serial.print("    Humidity : ");
```

```
    Serial.println(h);

  } void
setup()
{

  Serial.begin(115200);
 //pinMode(gas, INPUT);
Blynk.begin(auth, ssid, pass);
dht.begin();
  timer.setInterval(30000L, sendSensor);

 //Wire.begin();
lcd.begin();

//    lcd.backlight();    //
lcd.clear();
lcd.setCursor(3,0);
lcd.print("Air     Quality");
lcd.setCursor(3,1);
lcd.print("Monitoring");
delay(2000);   lcd.clear();
  }  void
loop() {
  Blynk.run();   timer.run();
float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for
Fahrenheit     int gasValue = analogRead(gas);   lcd.setCursor(0,0);
lcd.print("Temperature ");    lcd.setCursor(0,1);   lcd.print(t);
lcd.setCursor(6,1);   lcd.write(1);   lcd.createChar(1, degree_symbol);
lcd.setCursor(7,1);   lcd.print("C");   delay(4000);   lcd.clear();
lcd.setCursor(0, 0);   lcd.print("Humidity ");   lcd.print(h);
lcd.print("%");   delay(4000);   lcd.clear();   //lcd.setCursor(0,0);
 // lcd.print(gasValue);  //
lcd.clear();
if(gasValue<600)
  {     lcd.setCursor(0,0);
lcd.print("Gas Value: ");
lcd.print(gasValue);
lcd.setCursor(0, 1);
lcd.print("Fresh Air");
Serial.println("Fresh Air");
delay(4000);     lcd.clear();
  }   else
if(gasValue>600)
  {
    lcd.setCursor(0,0);
lcd.print(gasValue);
lcd.setCursor(0, 1);
lcd.print("Bad Air");
Serial.println("Bad Air");
delay(4000);     lcd.clear();
  }
    if(gasValue > 600){
```

```
  //Blynk.email("mithunkumaran79@gmail.com", "Alert", "Bad Air!");
  Blynk.logEvent("pollution_alert","Bad Air");
}  }
```

Platform Development:

The monitoring system is developed for transmission and reception of the information received from various data-sources with the use of sensors integrated with microcontroller. The wireless sensing real-time data are transmitted into desired form across the network through internet connection.AQMS is able to monitor concentration of CO2, CO, NO2, temperature and relative humidity and stores the concentration values in the database.

- ✓ Data Collection: Establishing data collection systems to gather information from the sensors. This can involve IOT devices, drones, satellites, or stationary monitoring stations.

- ✓ Data Processing: Developing algorithms and software for data processing, quality assurance, and real-time analysis to generate accurate air quality indices and pollutant concentration measurements.

- ✓ Communication: Implementing methods to transmit data to a central server or database. This may include cellular networks, Wi-Fi, or otherwireless communication technologies.

- ✓ User Interface: Creating user-friendly interfaces, such as mobile apps or web platforms, to display air quality information to the public or relevantauthorities.

- ✓ Data Visualization: Developing graphical representations and maps topresent air quality data in an understandable and actionable way.

- ✓ Alerts and Warnings: Integrating alert systems that notify users when airquality reaches unhealthy levels, enabling them to take appropriate precautions.

- ✓ Air Quality Modeling: Utilizing advanced computational models tosimulate and predict air quality patterns based on historical data andmeteorological conditions. This helps in forecasting air quality and assessing the effectiveness of potential interventions.

- ✓ Mobile Monitoring Units: Developing mobile monitoring units that canbe deployed to specific locations or events to assess air quality in real time. These units are especially useful for tracking air quality during emergencies or large gatherings.

- ✓ Historical Data Analysis: Historical air quality data can provide valuable insights into long-term trends and the effectiveness of past interventions. Analysing this data can inform future policy decisions and environmentalplanning.
- ✓ SQLite: Data Base for Web Development is created by SQLite

Code Implementation:

```
import sensors # Import your sensor library

import sqlite3# Import your database library

import analysis # Import your data analysis library

import visualization # Import your data visualization library

# Initialize sensors (replace with actual sensor initialization code)

air_quality_sensor = sensors.AirQualitySensor()

temperature_sensor = sensors.TemperatureSensor()

humidity_sensor = sensors.HumiditySensor()


conn = sqlite3.connect('your_database.db')

cursor = conn.cursor()


# Create the sensor_data table if it doesn't exist

cursor.execute('''

    CREATE TABLE IF NOT EXISTS sensor_data (

        id INTEGER PRIMARY KEY,

        air_quality REAL,

        temperature REAL,

        humidity REAL

    )

''')
```

```
conn.commit()


# Main data collection loop

while True:

 # Read sensor data

  air_quality_data = air_quality_sensor.read()

  temperature_data = temperature_sensor.read()

  humidity_data = humidity_sensor.read()

 # Store data in a database (e.g., SQLite, MySQL, InfluxDB)

  cursor.execute("INSERT INTO sensor_data (air_quality, temperature, humidity)
VALUES (?, ?, ?)",

            (air_quality_data, temperature_data, humidity_data))

  conn.commit()

 # Analyze data for air quality index (AQI)

  aqi = analysis.calculate_aqi(air_quality_data)

 # Visualize data (e.g., on a dashboard)

  visualization.update_dashboard(air_quality_data,
temperature_data,humidity_data, aqi)
```

Modules:

Sensors Module:

```
import random

class AirQualitySensor:

    def read(self):

        # Simulate air quality data (replace with actual implementation)
```

```
        return random.randint(0, 100)
    class TemperatureSensor:
        def read(self):
            # Simulate temperature data (replace with actual implementation)
            return random.uniform(20.0, 30.0)
    class HumiditySensor:
        def read(self):
            # Simulate humidity data (replace with actual implementation)
            return random.uniform(30.0, 60.0)
```

Analysis Module:

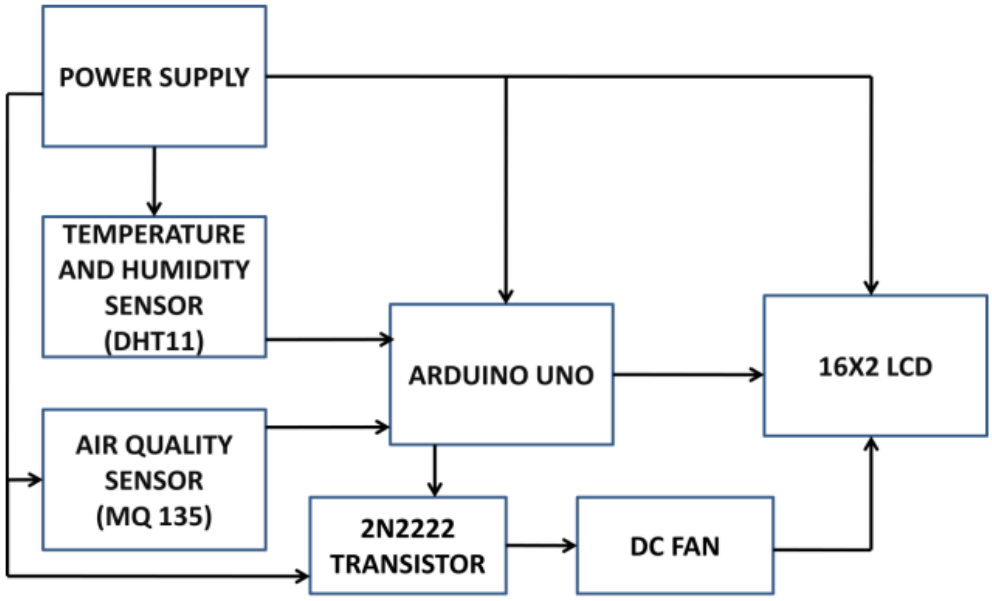
```
    def update_dashboard(air_quality_data, temperature_data, humidity_data, aqi):
        # Your visualization logic here
        print(f"Air Quality: {air_quality_data}")
        print(f"Temperature: {temperature_data}")
        print(f"Humidity: {humidity_data}")
        print(f"AQI: {aqi}")
        # Add your visualization code here (e.g., updating a dashboard)
```

Visualization Module:

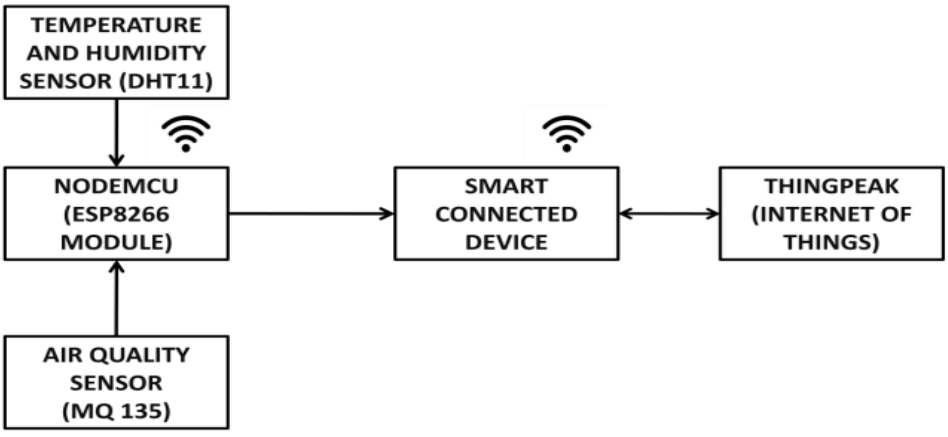
```
    def calculate_aqi(air_quality_data):
        # Your AQI calculation logic here
        aqi = air_quality_data * 2  # Replace with your actual AQI calculation
        return aqi
```

DIAGRAMS:

BLOCK DIAGRAMS:

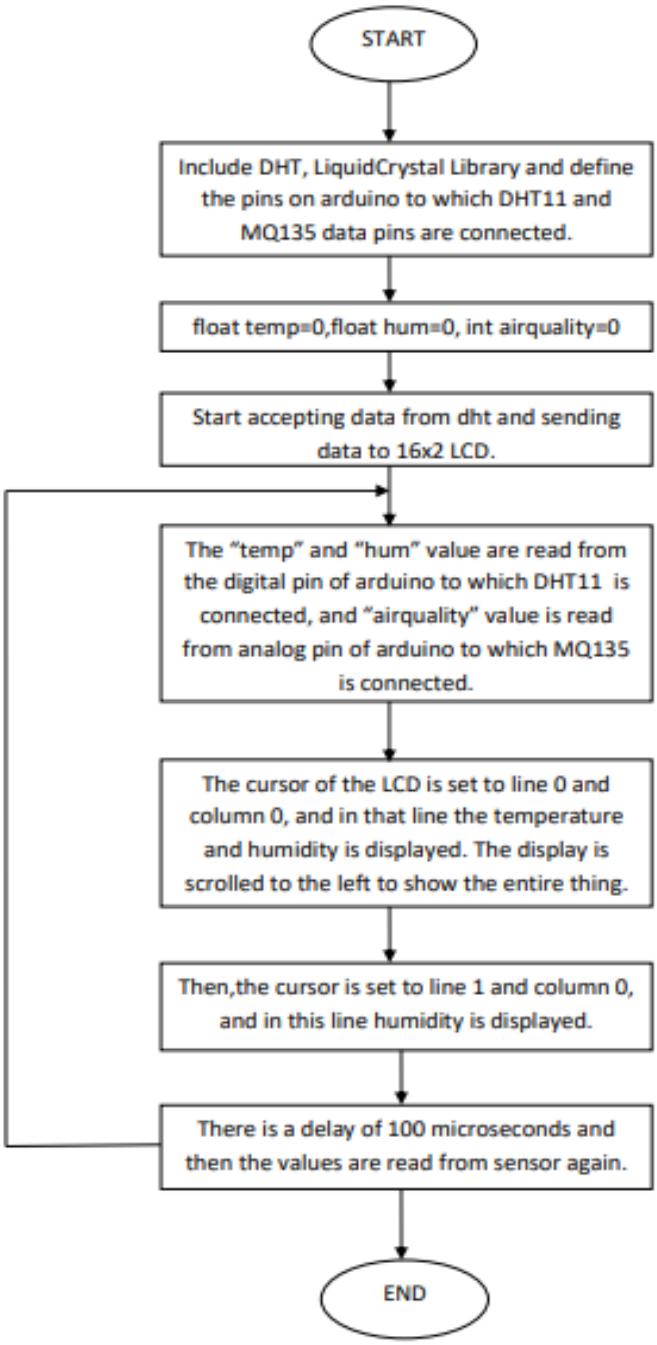


Block Diagram of Air Quality Monitoring and Sensing

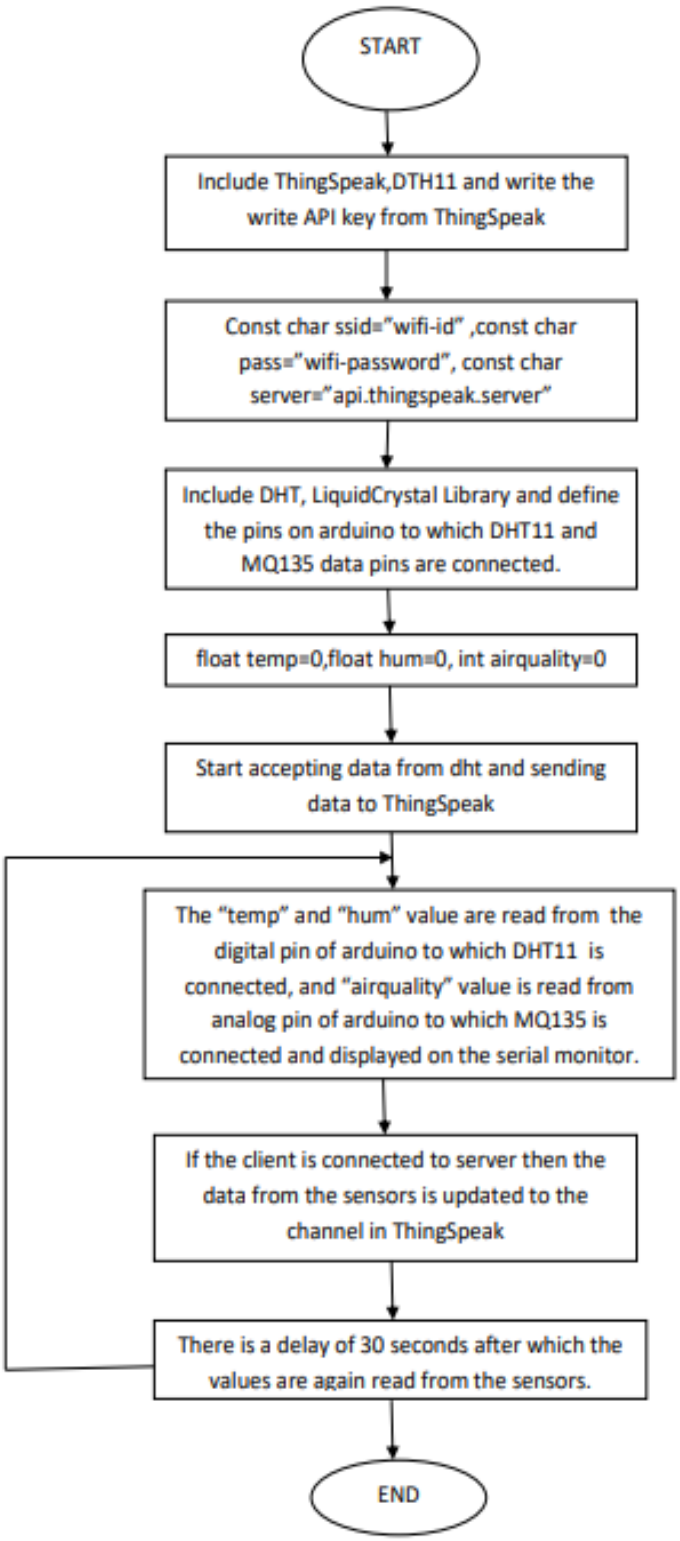


Block Diagram of Sending the data to PLATFORM using NodeMUC

Flow Chart:

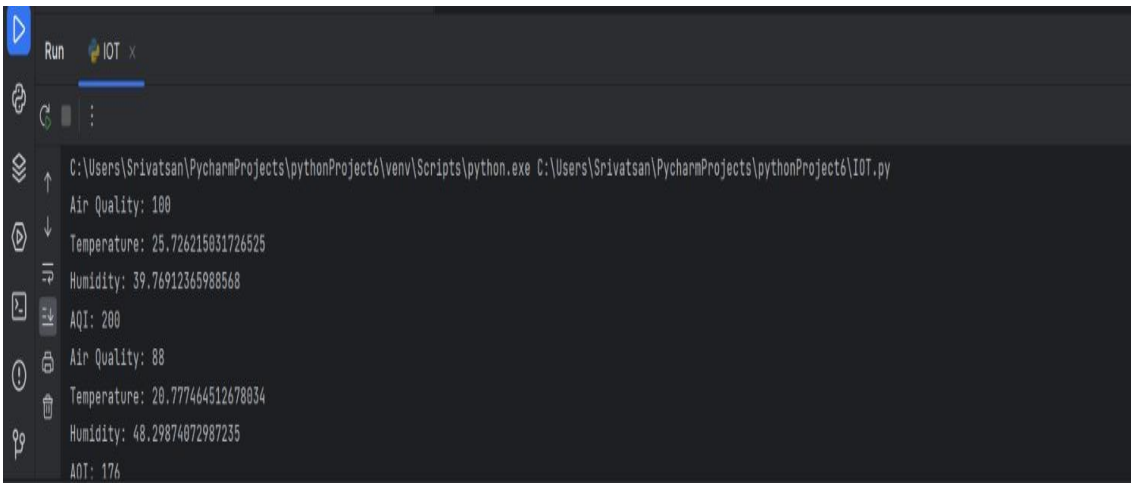


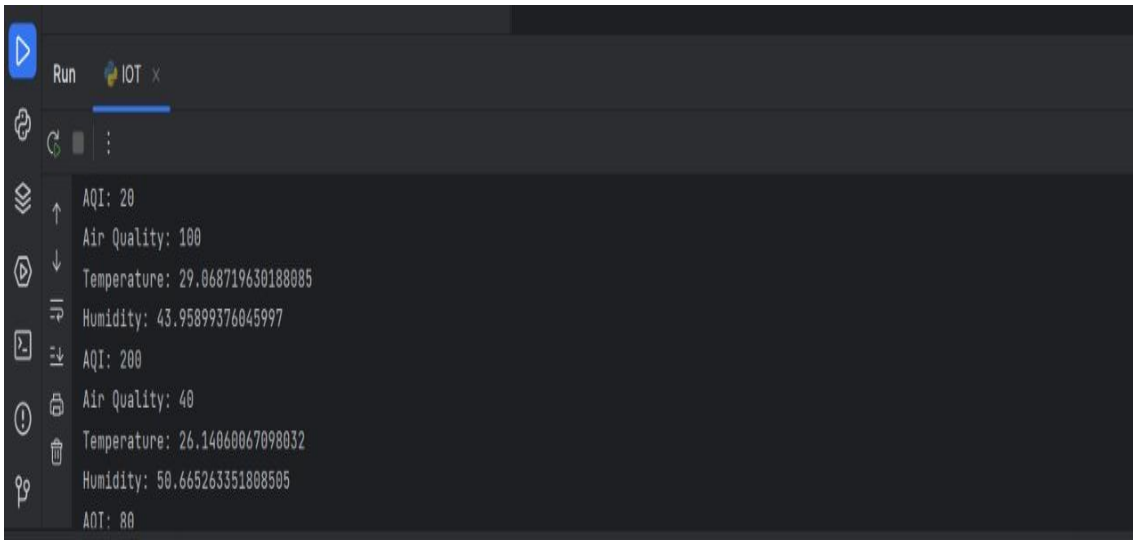
Flowchart for displaying the Temperature and Humidity and Air Quality on LCD

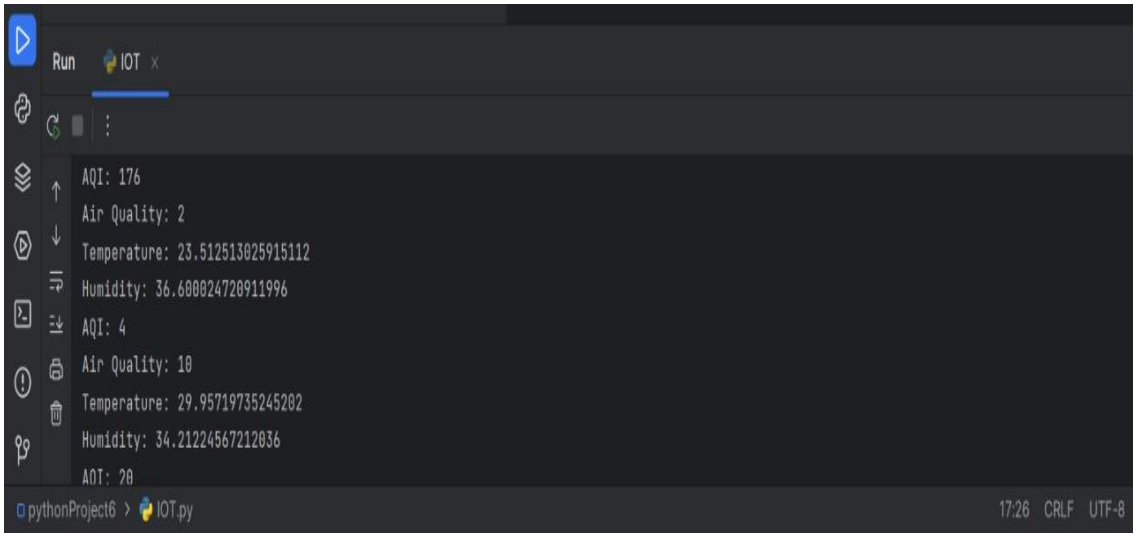


Flowchart for sending the Temperature, Humidity and Air Quality of THINGSPAD

Data Sharing Platform







Conclusion:

Monitoring air quality is crucial for ensuring public health and environmental protection. By regularly assessing pollutants such as particulate matter, ozone, sulfur dioxide, nitrogen dioxide, and carbon monoxide, authorities can implement necessary measures to mitigate harmful impacts. Continuous monitoring facilitates the formulation of effective policies, promotes public awareness, and fosters a cleaner and safer environment for all.