Project-1

IoT Architecture and Protocols

Air Quality Detection using ThingSpeak

By:
VU22CSEN0600047-Ganesh Koushik
VU22CSEN0600082-Kintali sai kiran

Table of Contents

Abstract

This project presents an IoT-based air quality detection system utilizing the NodeMCU ESP8266 microcontroller along with the MQ-135 and BME280 sensors.

The system measures temperature, humidity, atmospheric pressure, and air quality, sending real-time data to ThingSpeak for visualization and analysis.

By integrating affordable and easily accessible components, this system provides a cost-effective solution for monitoring air quality and environmental conditions, making it suitable for smart homes, industries, and pollution monitoring applications.

Objectives

- To design and implement an IoT-based air quality monitoring system.

- To integrate MQ-135 for detecting harmful gases and pollutants.

- To utilize BME280 for measuring temperature, humidity, and atmospheric pressure.

- To send real-time data to ThingSpeak for remote monitoring and analysis

- To develop a user-friendly dashboard for data visualization.

- To ensure the reliability and accuracy of the collected data.

Introduction

With increasing environmental concerns, monitoring air quality has become crucial.
Traditional air quality monitoring systems are expensive and complex.

This project aims to develop a low-cost, real-time monitoring system using IoT technology.

The NodeMCU ESP8266 acts as the central processing unit, collecting sensor data and transmitting it to the ThingSpeak cloud.
The MQ-135 sensor detects harmful gases like CO2, NH3, and benzene, while the BME280 sensor provides temperature, humidity, and pressure readings.

This information is crucial for environmental monitoring, smart homes, and industrial safety applications.

Methodology

System Design

The system consists of:

1. **Sensor Node:** NodeMCU ESP8266 connected with MQ-135 and BME280.
2. **Cloud Connectivity:** Data is sent to ThingSpeak for storage and visualization.
3. **User Interface:** The ThingSpeak dashboard allows real-time monitoring of air quality parameters.

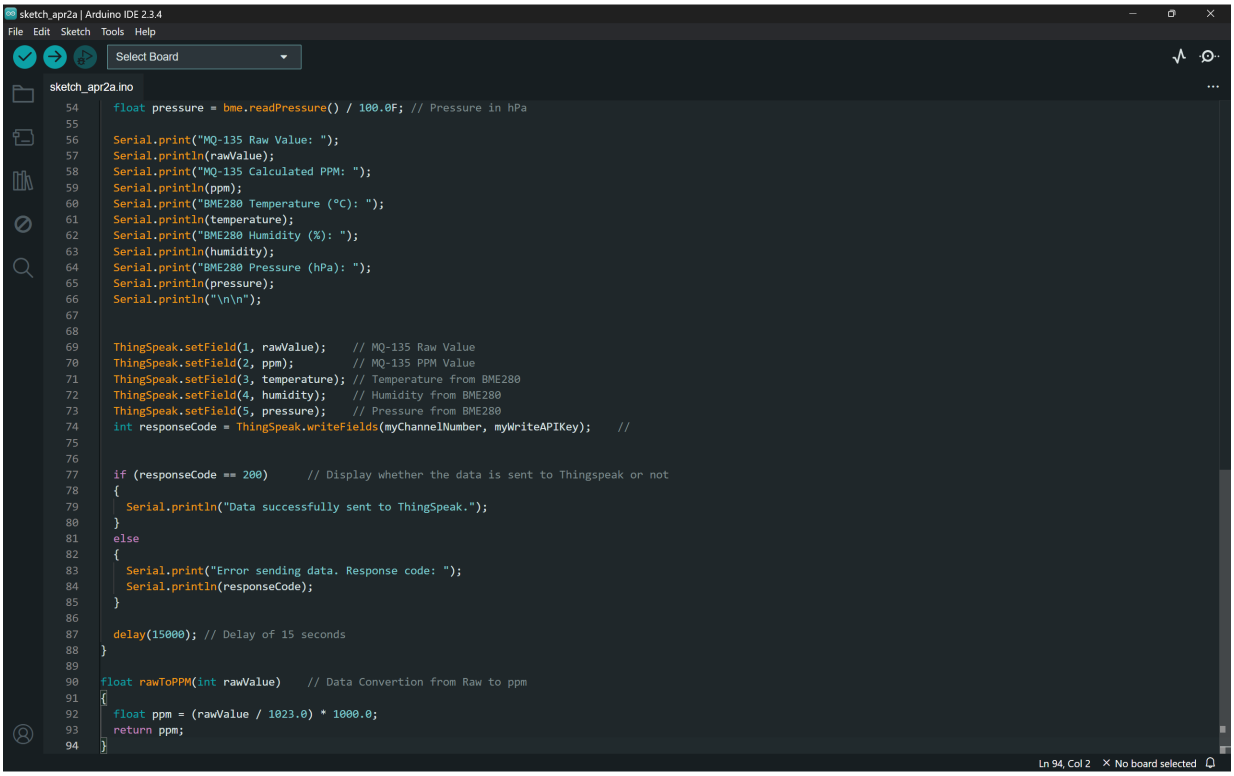
Software Development

- **Arduino IDE** is used for coding and uploading firmware to NodeMCU.
- **Libraries Used:**
  - Adafruit_Sensor.h
  - Adafruit_BME280.h
  - MQ135.h
  - ESP8266WiFi.h
  - ThingSpeak.h
- **Data Processing:** Sensor values are read every 10 seconds and sent to ThingSpeak.

Implementation Steps

1. Connect the MQ-135 and BME280 sensors to the NodeMCU ESP8266.
2. Write and upload the Arduino sketch.
3. Configure Wi-Fi credentials and ThingSpeak API key in the code.
4. Power the system and monitor data on the ThingSpeak dashboard.

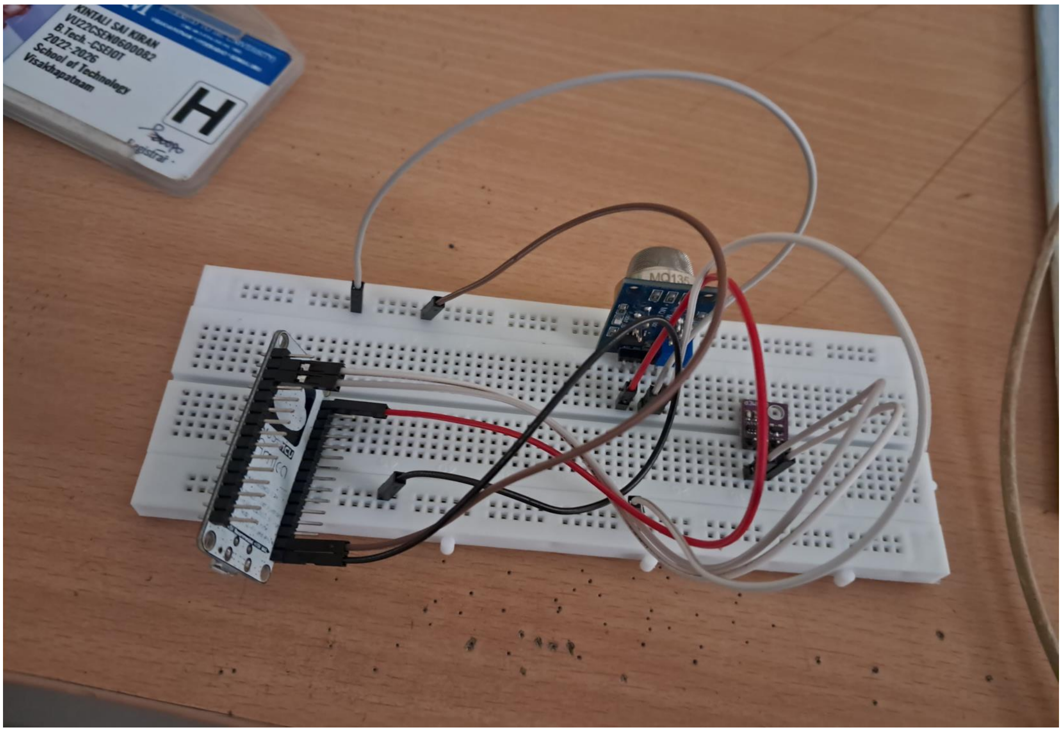
Arduino IDE Sketch

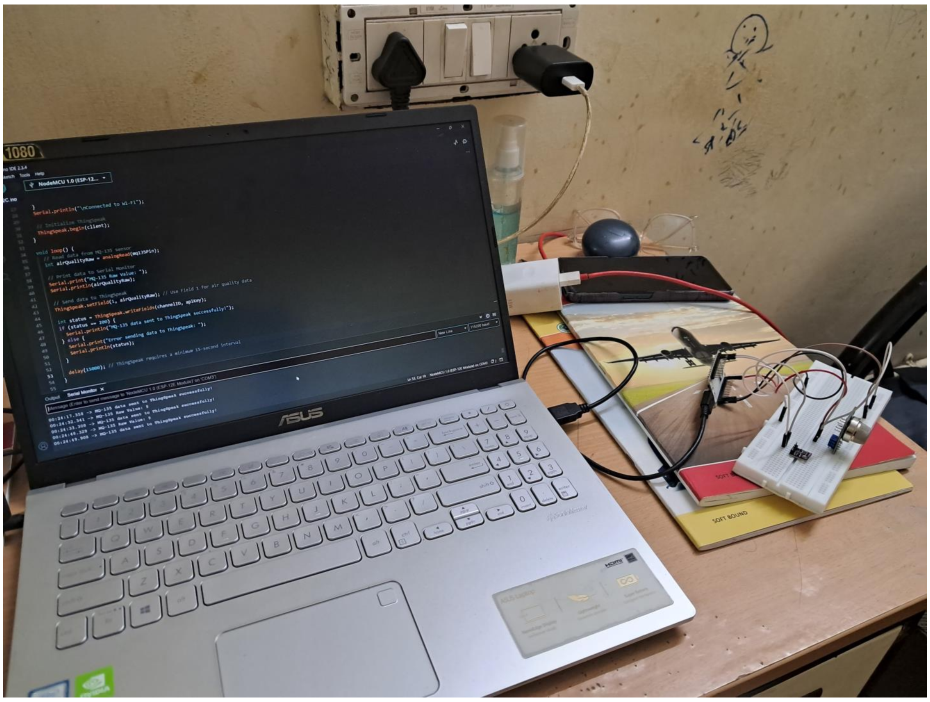


```
  float pressure = bme.readPressure() / 100.0F; // Pressure in hPa

  Serial.print("MQ-135 Raw Value: ");
  Serial.println(rawValue);
  Serial.print("MQ-135 Calculated PPM: ");
  Serial.println(ppm);
  Serial.print("BME280 Temperature (°C): ");
  Serial.println(temperature);
  Serial.print("BME280 Humidity (%): ");
  Serial.println(humidity);
  Serial.print("BME280 Pressure (hPa): ");
  Serial.println(pressure);
  Serial.println("\n\n");


  ThingSpeak.setField(1, rawValue);    // MQ-135 Raw Value
  ThingSpeak.setField(2, ppm);         // MQ-135 PPM Value
  ThingSpeak.setField(3, temperature); // Temperature from BME280
  ThingSpeak.setField(4, humidity);    // Humidity from BME280
  ThingSpeak.setField(5, pressure);    // Pressure from BME280
  int responseCode = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);    //


  if (responseCode == 200)      // Display whether the data is sent to Thingspeak or not
  {
    Serial.println("Data successfully sent to ThingSpeak.");
  }
  else
  {
    Serial.print("Error sending data. Response code: ");
    Serial.println(responseCode);
  }

  delay(15000); // Delay of 15 seconds
}

float rawToPPM(int rawValue)    // Data Convertion from Raw to ppm
{
  float ppm = (rawValue / 1023.0) * 1000.0;
  return ppm;
}
```

Hardware Requirements

Components:





Components Table:

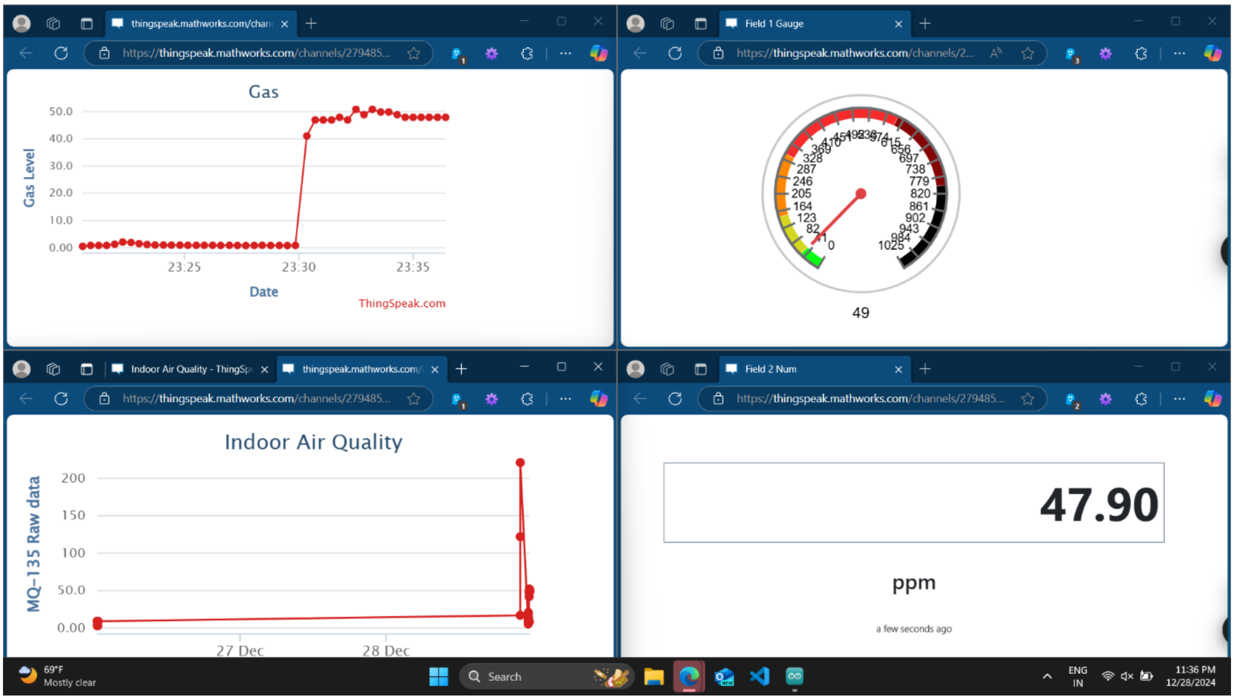
Component	Quantity	Description
NodeMCU ESP8266	1	Wi-Fi-enabled microcontroller
MQ-135 Sensor	1	Air quality sensor for detecting harmful gases
BME280 Sensor	1	Temperature, humidity, and pressure sensor
USB Cable	1	For power and programming
Jumper Wires	As required	To connect components

Connections Table

Component	NodeMCU Pin
MQ-135 VCC	3V3
MQ-135 GND	GND
MQ-135 AO	A0
BME280 VCC	3V3
BME280 GND	GND
BME280 SDA	D2
BME280 SCL	D1

Results

- The system successfully collected air quality, temperature, humidity, and pressure data.
- Data was transmitted to ThingSpeak in real-time and visualized through graphs.
- The system detected changes in air quality and environmental conditions effectively.
- Challenges faced included Wi-Fi connectivity issues, which were resolved by optimizing network settings.



Future Scope

This project lays a foundation for further development:

- **Multi-Sensor Integration:** Additional sensors like CO, NO2, and PM2.5 can be added for enhanced air quality detection.

- **Mobile Alerts**: Integration with a mobile app to notify users when air quality reaches hazardous levels.

- **Data Logging:** Store historical data for long-term analysis and pattern recognition.

- **AI-based Predictions:** Use machine learning to predict air quality trends and provide proactive alerts.

- **Solar Power Integration**: Implement solar panels to make the system energy-efficient and suitable for remote locations.