1. INTRODUCTION

Air pollution poses a significant threat to public health and the environment, making effective monitoring and management crucial. In response to this challenge, the integration of Internet of Things (IoT) technology has emerged as a powerful tool for real-time monitoring and data collection. This project report focuses on the development and implementation of an IoT-based air pollution monitoring system designed to provide accurate and timely information on air quality parameters.

Air pollution is a global issue with far-reaching consequences for human health, ecosystems, and climate change. Monitoring air quality is essential for understanding pollution levels, identifying sources, and implementing targeted interventions to improve air quality.

The primary objective of this project is to design, develop, and deploy an IoT-based air pollution monitoring system that can collect data on key pollutants such as particulate matter (PM), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), and ozone (O3). By leveraging IoT sensors and connectivity, this system aims to provide real-time data for informed decision-making and public awareness.

This project report is structured to provide a comprehensive overview of the IoT-based air pollution monitoring system, including the design and implementation process, sensor selection criteria, data collection methodology, data analysis techniques, and potential applications. Additionally, the report will discuss challenges faced during the project, lessons learned, and recommendations for future enhancements.

By combining IoT technology with air pollution monitoring, this project aims to contribute to a more sustainable and healthier environment for present and future generations.

2. LITERATURE REVIEW

The literature review for an air pollution monitoring is the significance of leveraging IoT technology for monitoring air quality parameters. Air pollution monitoring, air quality system, IoT, and smart cities. In particular, most of the literature are found by searching for "(("air pollution") OR ("air quality")) AND ("IoT") AND ("smart") AND ("system")" in ACM database where we chose the publication date to be no older than 2015. In our review, we focus on which pollutants each system can detect, where the sensors are deployed, which sensors are used, and how the communication is done between the different parts of the system.

- **Global Warming and Climate Change:** The rise in global warming and climate change has heightened the need for monitoring air pollution due to increased quantities of trace gases in the atmosphere.
- **IoT-Based Monitoring Systems:** IoT provides a versatile platform for monitoring various pollutants like CO2, Methane, NH3, CFC, Dichlorofluoromethane, NOx, and others in real-time.
- **Sensor Integration:** Various studies have utilized sensors like MQ135, MQ7, DHT11, and Node MCU to detect pollutants and monitor air quality levels.
- **Smart Cities Implementation:** IoT-based air pollution monitoring systems have been developed for smart cities to assess and improve urban air quality.
- **Air Quality Index (AQI):** The Air Quality Index is used to categorize air quality levels based on pollutant concentrations, with higher levels indicating increased health risks.
- **Sensor Calibration:** Sensors like MQ-135 are calibrated to detect specific gases such as alcohol, carbon dioxide, hydrogen, and methane to ensure accurate monitoring.
- **Programming and Connectivity:** Projects often involve programming in C++, utilizing Arduino IDE, and connecting to platforms like ThingSpeak cloud for data visualization and analysis.

3. EXISTING AND PROPOSED SYSTEM

Existing System:

- **Design:** The existing IoT-based air pollution monitoring systems are designed for outdoor air quality monitoring, utilizing sensors to measure parameters like CO2, CO, PM10, NO2, temperature, and humidity.
- **Data Collection:** These systems collect data using micro-sensors placed on an electric car, which is then transmitted to a server for real-time monitoring and dissemination to users.
- **Security:** An emphasis is placed on secure data transmission protocols to ensure system security, providing resilience against attacks and interception, a feature lacking in previous solutions.
- **Sensor Integration:** Sensors like MQ-135, DHT11, and Node MCU are commonly used to detect pollutants and monitor air quality levels in the existing systems.
- **Functionality:** The existing systems enable real-time localized air quality monitoring, allowing users to subscribe for local pollution information and receive alerts when air quality deteriorates.

Proposed System:

- **Aim:** The proposed IoT-based air pollution monitoring system aims to monitor air quality using the internet from anywhere, providing real-time data on air quality levels and sending alerts when pollution exceeds safe limits.
- **Components:** The proposed system utilizes sensors for detecting harmful gases like CO2, CO, smoke, temperature, and humidity, with Node MCU playing a central role in data collection and display.
- **Data Visualization:** Monitored data is displayed on the Thing Speak cloud, allowing for easy analysis of air quality in the area, enhancing decision-making for pollution control measures.
- **Working Mechanism:** The proposed system employs a flow diagram to illustrate its working mechanism, showcasing how sensors, Node MCU, and cloud connectivity interact to monitor and analyses air quality levels.

4. TOOLS AND TECHNOLOGY USED

- **Sensors:** Various sensors like MQ135, MQ7, DHT11, dust sensors, gas sensors, temperature, and humidity sensors are commonly employed to detect pollutants such as CO2, CO, smoke, alcohol, benzene, NH3 gas, particulate matter, gases, and volatile organic compounds (VOCs).
- **Connectivity:** IoT-based systems utilize connectivity technologies to link sensors to a central data management system through wireless or wired communication protocols for real-time data collection and analysis.
- **Data Visualization:** Data collected from sensors is processed and analysed in real-time using cloud-based analytics platforms. Users can access this information through web or mobile applications for visualizations, alerts, and historical data tracking.
- **IoT Platforms:** Platforms like ThingSpeak are utilized for data visualization and analysis in air pollution monitoring systems. These platforms enable easy access to monitored data and enhance decision-making for pollution control measures.
- **Power Supply:** IoT-based air pollution monitoring systems require a stable power supply for operation. External power supplies are provided for permanent installations, while batteries are used for portable devices.
- **Enclosures:** The components of these systems are protected by enclosures that shield them from environmental factors like dust, water, and temperature fluctuations.

Technology Used

The **Arduino** is an open-source platform that combines hardware and software to facilitate the development of electronic projects. It consists of a physical programmable circuit board (microcontroller) and an Integrated Development Environment (IDE) that runs on a computer, allowing users to write and upload code to the board.

The Arduino platform is popular among beginners in electronics due to its user-friendly nature and simplified programming process. Arduino boards do not require a separate programmer to load new code, making them accessible for novices. The Arduino IDE uses a simplified version of C++ for programming, making it easier to learn. Arduino boards come in various types, with the Arduino UNO being one of the most common options for beginners, featuring digital I/O pins, PWM pins, analog inputs, and a USB connection. Additionally, the Arduino platform provides a standard form factor that breaks out the functions of the microcontroller into a more accessible package. Overall, Arduino is a versatile tool that enables users to create a wide range of projects, from simple LED blinking to complex robotics, making it a valuable resource for both hobbyists and professionals in the electronics field.

Language Used

Embedded C is a set of language extensions for the C programming language developed by the C Standards Committee to cater to the specific requirements of embedded systems.

- **Purpose:** Embedded C addresses commonality issues that exist between C extensions for different embedded systems, requiring nonstandard extensions to support enhanced microprocessor features like fixed-point arithmetic, multiple memory banks, and basic I/O operations.
- **Standardization:** The C Standards Committee produced a Technical Report outlining a common standard for all implementations of Embedded C to adhere to, ensuring consistency and compatibility across different systems.
- **Features:** Embedded C includes features not found in standard C, such as fixed-point arithmetic, named address spaces, and hardware addressing for basic I/O operations, enhancing its suitability for embedded applications.
- **Syntax and Semantics:** Embedded C retains most of the syntax and semantics of standard C, including functions, variable definitions, data types, conditional statements, loops, arrays, structures, bit operations, macros, and more, making it familiar to C programmers.

5. HARDWARE AND SOFTWARE USED

Hardware Requirements:

Processor – i3 Core Processor

RAM - 8 GB Clock

speed - 2.5 GHz

Monitor - 1024 * 768 Resolution Colour

Keyboard - QWERTY

Mother board - 845c Intel Motherboard Backup

Media - Floppy/pen drive/Hard disk

Hard disk – 1 TB HDD I/O

Device - Standard input and output devices.

Software Requirements:

Text editor: Arduino

6. SYSTEM DESIGN

ESP8266:

The ESP8266 is a versatile and cost-effective microchip designed by Espressif Systems, offering built-in TCP/IP networking software and microcontroller capabilities. It is widely used in IoT applications due to its compact size, low cost, and powerful features. The NodeMCU ESP8266 development board is a popular choice for IoT projects, providing an easy-to-use platform for connecting devices to the internet.

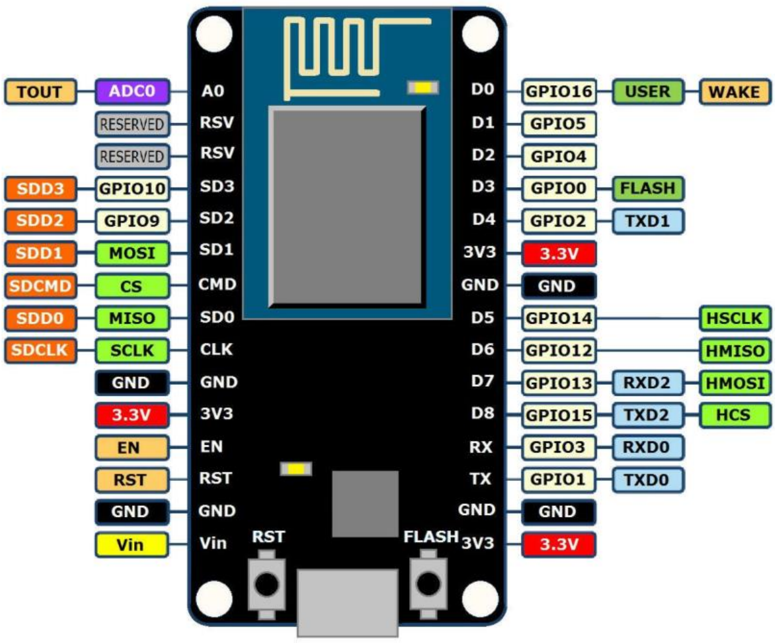


Fig 6.1: ESP8266

MQ – 135 Sensor:

The MQ-135 gas sensor is a crucial component for air quality monitoring, offering various features and applications. The MQ-135 sensor has a wide detection scope, high sensitivity, fast response time, stability, and a long lifespan. It operates at +5V and can detect various gases like NH3, NOx, alcohol, benzene, smoke, and CO2. The sensor consists of a steel exoskeleton housing a sensing element that responds to gas concentration changes by altering its resistance when ionized by gases. This change in resistance affects the current passing through it, providing an output signal proportional to gas concentration.

Fig 6.2: MQ-135

DHT11 Sensor:

The DHT11 sensor, also known as the AM2302, is a digital temperature and humidity sensor with a single-wire digital interface. The DHT11 sensor utilizes a capacitive humidity sensor and a thermistor to measure temperature and humidity in the surrounding air, providing accurate readings without the need for additional components. It outputs a digital signal on the data pin, eliminating the requirement for analog input pins. The DHT11 operates on a power supply of 3.3-6V DC and outputs a digital signal via a 1-wire bus. It has an operating range of 0-100% relative humidity and -40~80 degrees Celsius for temperature, with high accuracy and resolution for both parameters.

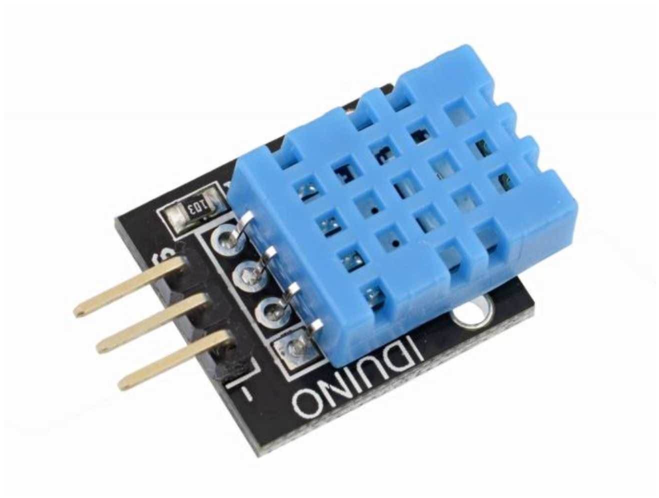


Fig 6.3: DHT11 Sensor

System Design for Air Pollution Monitoring:

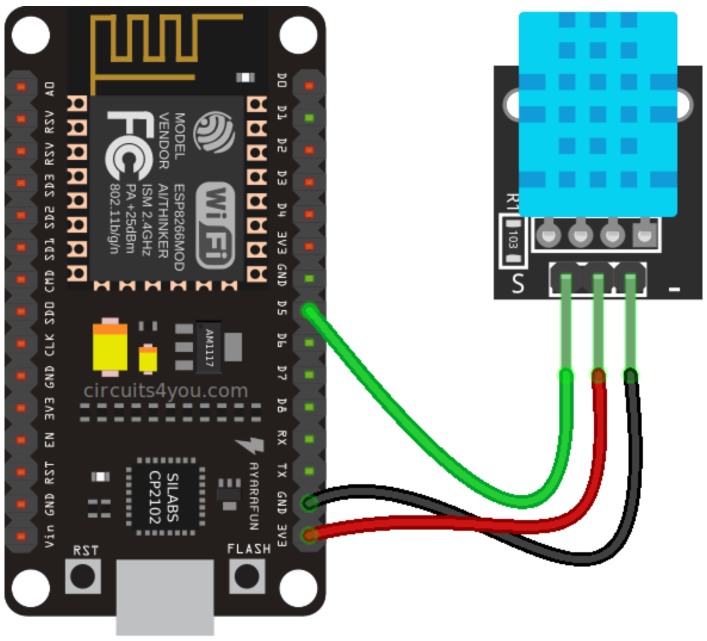


Fig 6.4 System Design 1

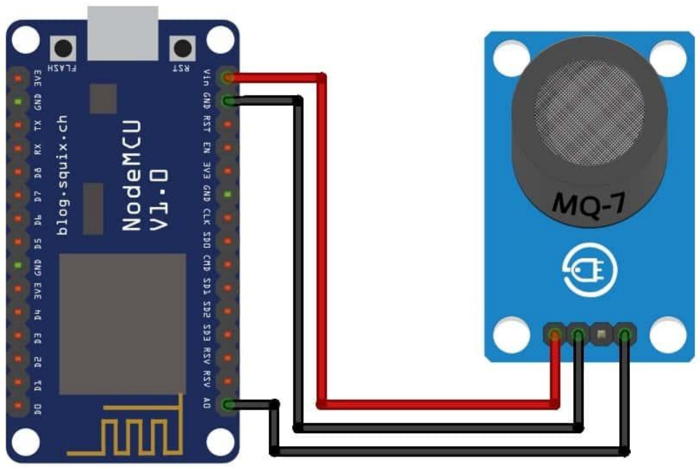


Fig 6.5 System Design 2

7. CODING

MQ135

```
#include <ESP8266WiFi.h>
String apiKey = "BLJ1ATL2PUROOSBI"; // Enter your Write API key from ThingSpeak
const char *ssid = "Shree"; // replace with your wifi ssid and wpa2 key
const char *pass = "12121212";
const char* server = "api.thingspeak.com";
WiFiClient client;
void setup()
{
Serial.begin(115200);
delay(10);
Serial.println("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED)
{
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
}
void loop()
{
float h = analogRead(A0);
if (isnan(h))
{
Serial.println("Failed to read from MQ-135 sensor!");
return;
}
if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com
{
String postStr = apiKey;
postStr += "&field3=";
postStr += String(h/1023*100);
postStr += "r\n";
client.print(postStr);
Serial.print("Gas Level: ");

Serial.println(h/1023*100);
Serial.println("Data Send to Thingspeak");
}
```

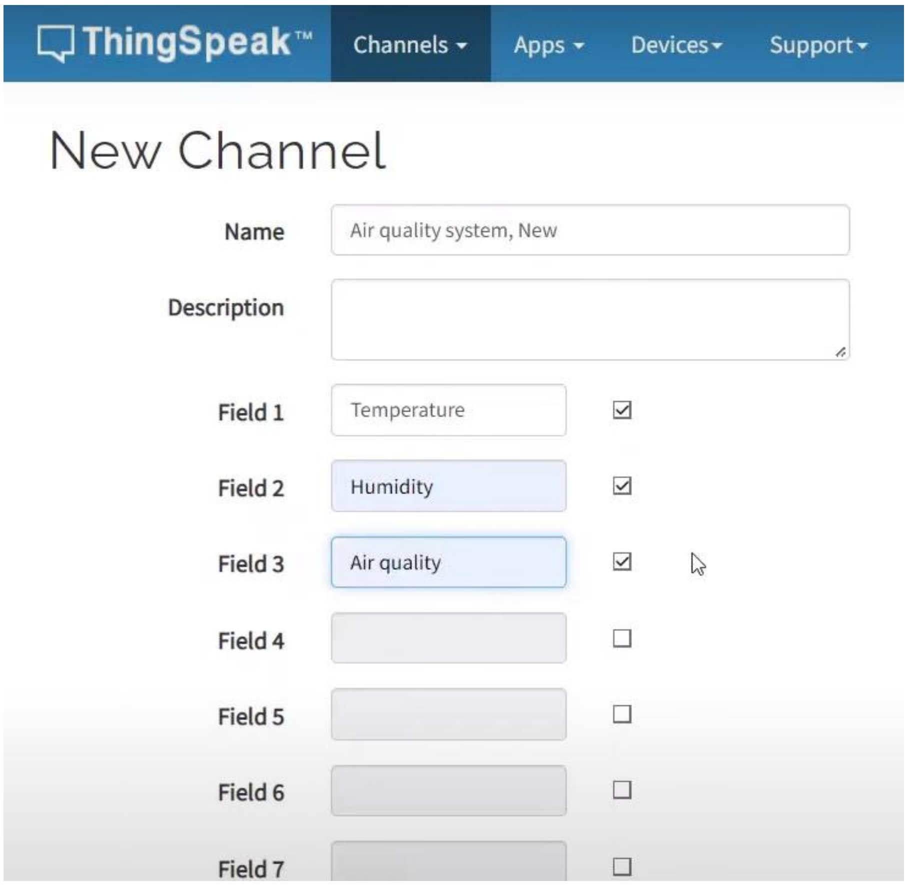
TEMPERATURE_AND_HUMADITY_NODEMCU

```
#include <dht.h> // Including library for dht
#include <ESP8266WiFi.h>

String apiKey = "BLJ1ATL2PUROOSBI"; // Enter your Write API key from ThingSpeak
const char *ssid = "Shree"; // replace with your wifi ssid and wpa2 key
const char *pass = "12121212";
const char* server = "api.thingspeak.com";

#define DHTPIN 0          //pin where the dht11 is connected, D3 in ESP8266
dht DHT;

WiFiClient client;

void setup()
{
       Serial.begin(115200);
       delay(10);
       pinMode(2, OUTPUT);
       digitalWrite(2, 0);
       Serial.println("Connecting to ");
       Serial.println(ssid);

       WiFi.begin(ssid, pass);

      while (WiFi.status() != WL_CONNECTED)
     {
            delay(500);
            Serial.print(".");
     }
      Serial.println("");
      Serial.println("WiFi connected");

}
void loop()
{
      int chk = DHT.read11(DHTPIN);
      float h = DHT.humidity;
      float t = DHT.temperature;
      if (isnan(h) || isnan(t))
                 {
                     Serial.println("Failed to read from DHT sensor!");
                      return;
                 }
}
```

TEMPHUMIDQUALITY

```
#include <dht.h> // Including library for dht
#include <ESP8266WiFi.h>
// #include <pthread.h>

String api_key = "BLJ1ATL2PUROOSBI"; // Enter your Write API key from ThingSpeak
const char *ssid = "Shree";           // replace with your wifi ssid and wpa2 key
const char *pass = "12121212";
const char *server = "api.thingspeak.com";

#define DHTPIN 0 // pin where the dht11 is connected, D3 in ESP8266
dht DHT;

WiFiClient client;
void AQ(double *);
void T(double *);
void H(double *);

void setup()
{
   Serial.begin(115200);
   delay(10);
   pinMode(2, OUTPUT);
   digitalWrite(2, 0);
   Serial.println("Connecting to ");
   Serial.println(ssid);

   WiFi.begin(ssid, pass);

   while (WiFi.status() != WL_CONNECTED)
   {
      delay(500);
      Serial.print(".");
   }
}
```

8. SNAPSHOTS



New Channel

Name	Air quality system, New	
Description		
Field 1	Temperature	☑
Field 2	Humidity	☑
Field 3	Air quality	☑
Field 4		☐
Field 5		☐
Field 6		☐
Field 7		☐

Fig. 8.1 Creating fields in ThingSpeak Cloud

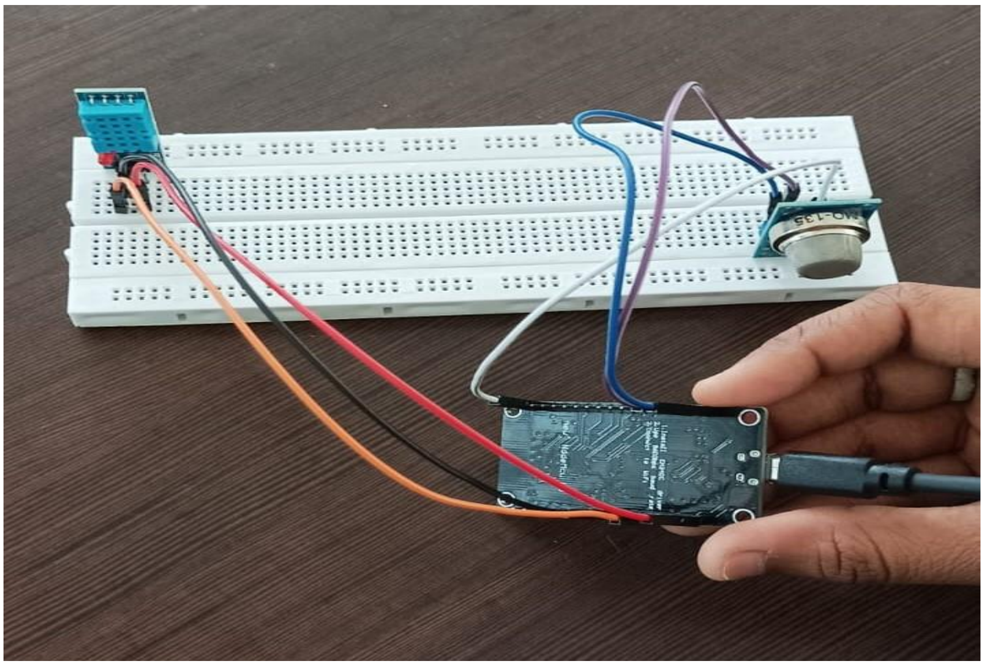


Fig: 8.2 Final Model

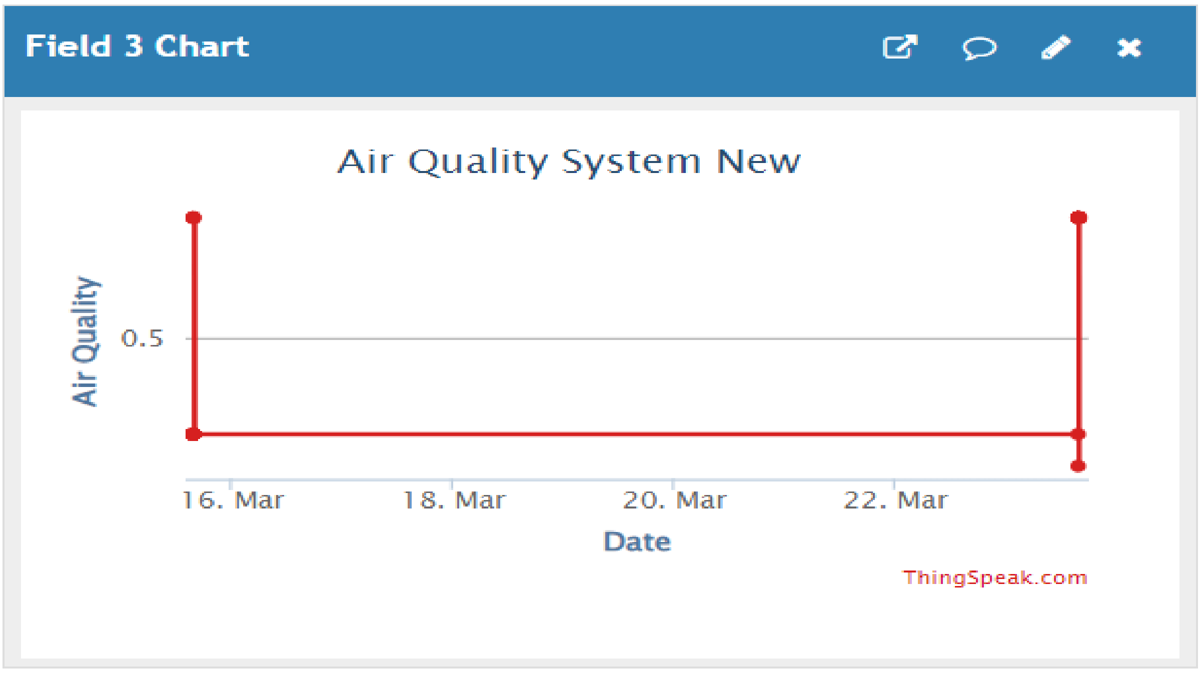


Fig. 8.3 Air Quality

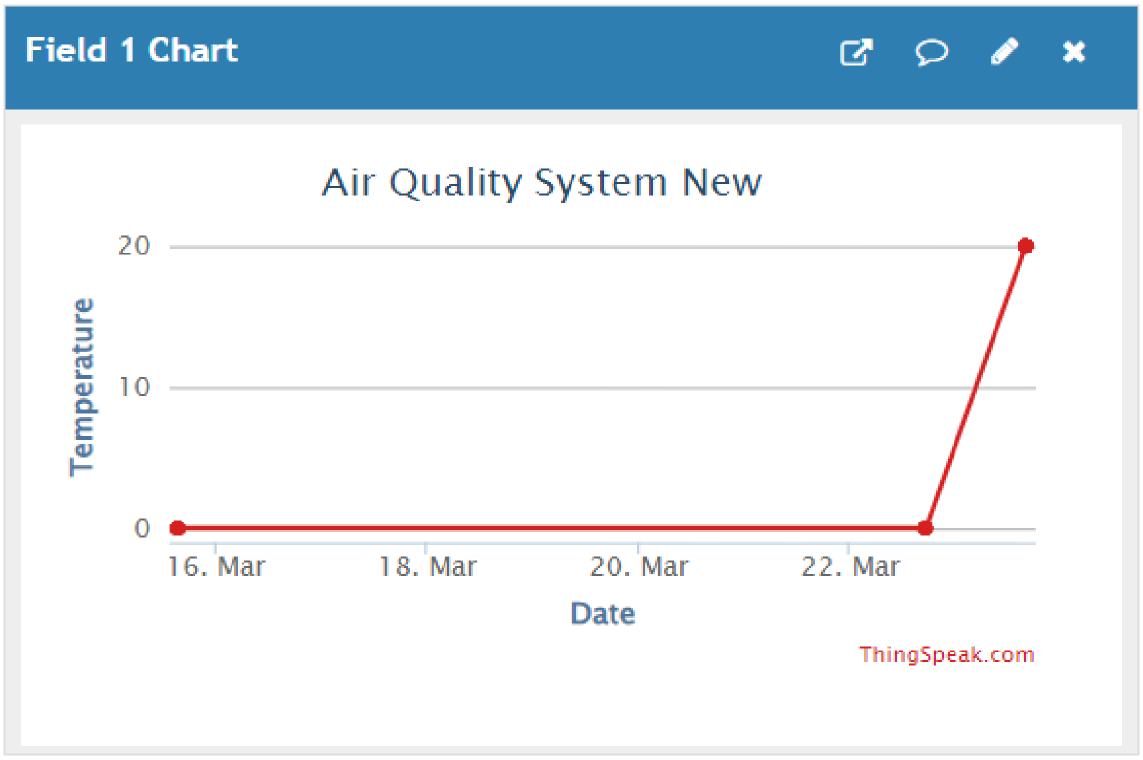


Fig. 8.4 Temperature

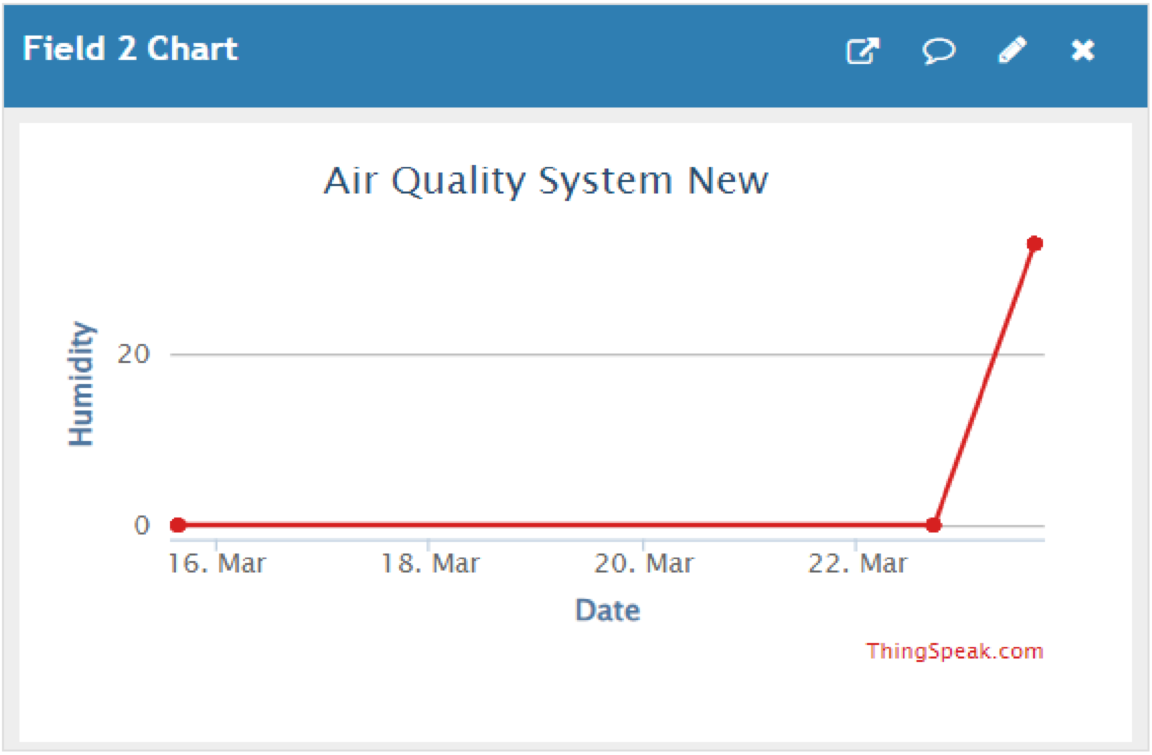


Fig. 8.5 Humidity

9. CONCLUSION

1. **Challenges Identified:** The project faces technical hurdles such as sensor accuracy and data transmission issues. Regulatory constraints, community engagement difficulties, and resource limitations also pose significant challenges.

2. **Recommendations for Improvement:** To address these challenges, we recommend enhancing sensor technology, strengthening data management capabilities, promoting stakeholder engagement, facilitating policy support, ensuring sustained funding, and promoting data transparency and open access.

3. **Impact Areas:** The project has the potential for significant impact across various dimensions:

- **Environmental Impact**: Improved air quality monitoring and pollution source identification.
- **Social Impact:** Increased public awareness and community empowerment.
- **Economic Impact:** Healthcare savings, productivity gains, and economic development.
- **Technological Impact:** Advancements in sensor technology and data analytics.
- **Policy and Governance Impact:** Influence on policy formulation and regulatory compliance.

10. BIBLIOGRAPHY

- https://www.arduino.cc/en/Guide/Libraries
- https://components101.com/sensors/dht11-temperature-sensor
- https://quartzcomponents.com/products/mq-135-air-quality-gas-sensor-module
- https://www.ppsthane.com/blog/iot-air-pollution-monitoring-system
- https://www.sciencedirect.com/science/article/abs/pii/S221478532104966X