An Internship Project Report

on

“IOT BASED AIR MONITORING SYSTEM”

Submitted in Partial Fulfilment of the Requirements For the award of the Degree of

**Bachelor of Technology**

**In**

**Electronics & Communication Engineering (ECE)**

By

**B. NAVEEN** -20311A04D9

UNDER THE GUIDANCE OF

Mrs. B. Krishnaveni

**Associate Professor Department of ECE**



# DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

**SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Autonomous)**

**Yamnampet(V), Ghatkesar(M), Hyderabad-501301.**

**August 2023-January 2024**

# SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad) Yamnampet (V), Ghatkesar (M), Hyderabad – 501 301

Department of Electronics and Communication Engineering

## CERTIFICATE

This is to certify that the project report entitled “IOT BASED AIR MONITORING SYSTEM ” is being submitted by B. NAVEEN (20311A04D9) in partial fulfilment of the requirements for the award of **Bachelor of Technology** degree in **Electronics and Communication Engineering** to **Sreenidhi Institute of Science and Technology** affiliated to **Jawaharlal Nehru Technological University, Hyderabad** (Telangana). This record is a bona fide work carried out by them under our guidance and supervision. The results embodied in the report have not been submitted to any other University or Institution for the award of any degree or diploma.

## Internal guide: Head of the Department

B. KRISHNAVENI Dr. S.P.V. SUBBA RAO

Associate Professor, Department of ECE Professor, Department of ECE

# DECLARATION

I hereby declare that the work described in this this titled “**IOT BASED AIR MONITORING SYSTEM** ” which is being submitted by us in partial fulfilment for the award of Bachelor of Technology in the Department of **Electronics and Communication Engineering**, Sreenidhi Institute of Science and Technology is the result of investigations carried out by us under the guidance of **Mrs. B. KRISHNAVENI, Associate Professor, Department of ECE, Sreenidhi Institute of Science and Technology, Hyderabad.**

No part of the this is copied from books/ journals/ internet and whenever the portion is taken, the same has been duly referred. The report is based on the project work done entirely by us and not copied from any other source. The work is original and has not been submitted for any Degree/Diploma of this or any other university.

Place: Hyderabad, Date: 14-12-2023.

**B. NAVEEN**

**20311A04D9**

# ACKNOWLEDGEMENT

I profoundly thank **Dr. SPV. Subba Rao**, Head of the Department of Electronics & Communication Engineering who has been an excellent guide and also a great source of inspiration to my work.

I would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

I would like to express my heart-felt gratitude to my parents without whom I would not have been privileged to achieve and fulfil my dreams. I am grateful to our principal**, Dr. T. Ch. Siva Reddy**, who most ably run the institution and has had the major hand in enabling me to do my project.

I would like to thank my Project coordinator **Mrs. B. Krishnaveni** for her technical guidance, constant encouragement and support in carrying out my project at college.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

**B. NAVEEN – 20311A04D9**

**CERTIFICATE**



**ABSTRACT**

Air pollution is one of the biggest threats to the present-day environment. Everyone is being affected by air pollution day by day including humans, animals, crops, cities, forests and aquatic ecosystems. Besides that, it should be controlled at a certain level to prevent the increasing rate of global warming. This project aims to design an IOT-based air pollution monitoring system using the internet from anywhere using a computer or mobile to monitor the air quality of the surroundings and environment. There are various methods and instruments available for the measurement and monitoring quality of air. The IoT-based air pollution monitoring system would not only help us to monitor the air quality but also be able to send alert signals whenever the air quality deteriorates and goes down beyond a certain level.

In this system, NodeMCU plays the main controlling role. It has been programmed in a manner, such that, it senses the sensory signals from the sensors and shows the quality level via led indicators. Besides the harmful gases (such as CO2, CO, smoke, etc) temperature and humidity can be monitored through the temperature and humidity sensor by this system. Sensor responses are fed to the NodeMCU which displays the monitored data in the Blynk IOT cloud which can be utilized for analyzing the air quality of that area. The following simple flow diagram (as shown in Fig. 1) indicates the working mechanism of the IoT-based Air Pollution Monitoring System.

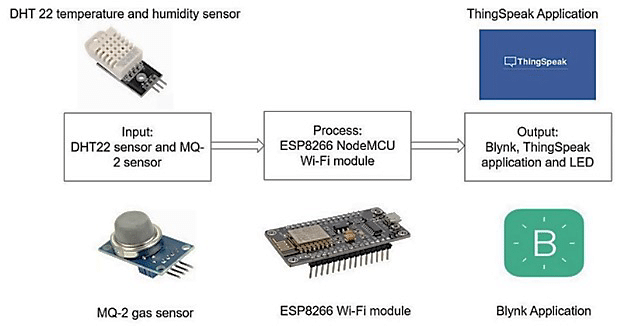


Table of Contents

**INTRODUCTION8**

**THEORY9**

**COMPONENTS11**

Hardware11

Software15

**CIRCUIT DIAGRAM17**

**CONNECTIONS18**

**SOFTWARE REQUIREMENTS19**

**SOURCE CODE20**

**CODE OVERVIEW24**

**WORKING25**

**RESULTS27**

LCD Outputs27

Blynk application outputs28

**CONCLUSION30**

**REFERENCES31**

**INTRODUCTION**

**Aim of the Project**

Air is getting polluted because of the release of toxic gases by industries, vehicle emissions and increased concentration of harmful gases and particulate matter in the atmosphere. The level of pollution is increasing rapidly due to factors like industries, urbanization, increase in population, vehicle use which can affect human health. Particulate matter is one of the most important parameters having a significant contribution to the increase in air pollution. This creates a need for measurement and analysis of real-time air quality monitoring so that appropriate decisions can be taken in a timely period. This paper presents real-time standalone air quality monitoring. Internet of Things (IoT)is nowadays finding profound use in each and every sector, plays a key role in our air quality monitoring system too. The setup will show the air quality in PPM on the webpage so that we can monitor it very easily. In this IoT project, we can monitor the pollution level from anywhere using your computer or mobile.

**Literature Survey**

The explanation of the Air Quality Index (AQI) and its standard ranges are from 0-100 ppm the atmosphere is safe for living. If the ppm level increases above 100 then it moves out of the safety zone. If the ppm value rises above 200 then it becomes extremely dangerous for human life. The DHT11 sensor module is used to measure the temperature and the humidity of the surroundings. The MQ-135 gas sensor is used to measure the air quality of the surroundings. It can be calibrated with respect to fresh air, alcohol, carbon dioxide, hydrogen and methane. In this project, it has been calibrated with respect to fresh air In the controlling action of NodeMCU has been described. This research has shown the uses of C++ as the programming language for scripting the software code. It has an inbuilt Wi-Fi module which allows the project to implement IoT easily. Arduino IDE is used to implement the coding part of the project. Blynk IOT cloud is used for the cloud service. It has a free version which requires a delay of 15 seconds to upload an entry in the cloud. As this project uses two sensors, both of them have internal heater elements and withdraw more power(P=V\*I), so though both sensors are turned ON, their output voltage levels vary and show unpredictable values due to insufficient power drive. So, we used a separate power supply for the sensors as NodeMCU alone is not sufficient to drive two sensors.

**THEORY**

The Internet of Things (IoT) describes the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools.

The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, and machine learning.

Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems, cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT is also used in healthcare systems.

There are a number of concerns about the risks in the growth of IoT technologies and products, especially in the areas of privacy and security, and consequently, industry and governmental moves to address these concerns have begun, including the development of international and local standards, guidelines, and regulatory frameworks.

IoT devices are a part of the larger concept of home automation, which can include lighting, heating and air conditioning, media and security systems and camera systems. Long-term benefits could include energy savings by automatically ensuring lights and electronics are turned off or by making the residents in the home aware of usage.

A smart toilet seat that measures blood pressure, weight, pulse and oxygen levels. A smart home or automated home could be based on a platform or hubs that control smart devices and appliances. For instance, using Apple's HomeKit, manufacturers can have their home products and accessories controlled by an application in iOS devices such as the iPhone and the Apple Watch. This could be a dedicated app or iOS native applications such as Siri. This can be demonstrated in the case of Lenovo's Smart Home Essentials, which is a line of smart home devices that are controlled through Apple's Home app or Siri without the need for a Wi-Fi bridge. There are also dedicated smart home hubs that are offered as standalone platforms to connect different smart home products and these include the Amazon Echo, Google Home, Apple's HomePod, and Samsung's SmartThings Hub. In addition to the commercial systems, there are many non-proprietary, open-source ecosystems; including Home Assistant, OpenHAB and Domoticz.

Significant numbers of energy-consuming devices (e.g. lamps, household appliances, motors, pumps, etc.) already integrate Internet connectivity, which can allow them to communicate with utilities not only to balance power generation but also helps optimize the energy consumption as a whole.These devices allow for remote control by users, or central management via a cloud-based interface, and enable functions like scheduling (e.g., remotely powering on or off heating systems, controlling ovens, changing lighting conditions, etc.).The smart grid is a utility-side IoT application; systems gather and act on energy and powerrelated information to improve the efficiency of the production and distribution of electricity.Usingadvanced metering infrastructure (AMI) Internet-connected devices, electric utilities not only collect data from end-users but also manage distribution automation devices like transformers.

Another example of integrating the IoT is Living Lab which integrates and combines research and innovation processes, establishing a public-private-people-partnership.There are currently 320 Living Labs that use the IoT to collaborate and share knowledge between stakeholders to co-create innovative and technological products. For companies to implement and develop IoT services for smart cities, they need to have incentives. The governments play key roles in smart city projects as changes in policies will help cities to implement the IoT which provides effectiveness, efficiency, and accuracy of the resources that are being used. For instance, the government provides tax incentives and cheap rent, improves public transport, and offers an environment where start-up companies, creative industries, and multinationals may co-create, share a common infrastructure and labor markets, and take advantage of locally embedded technologies, production process, and transaction costs.The relationship between the technology developers and governments who manage the city's assets is key to providing open access to resources to users in an efficient way.

In this project, we have tried to implement the concept of IoT to monitor the temperature, humidity and air quality of the surroundings

**COMPONENTS**

**Hardware Components:**

1. NodeMCU V3
2. DHT11 Sensor Module
3. MQ-135 Gas Sensor Module
4. I2C Converter
5. LCD Display
6. Breadboard
7. Connecting Wires
8. AC-DC Adapters

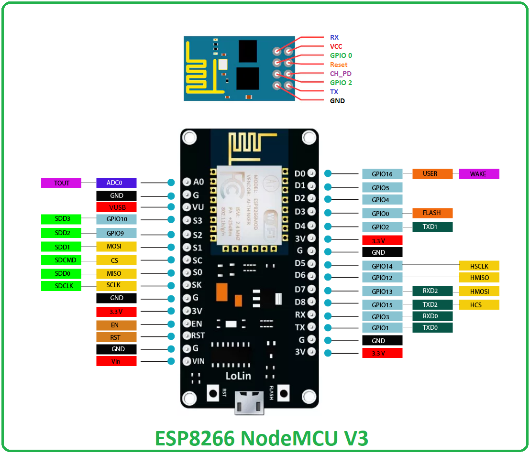
**Software Components**

1. Blynk IOT Cloud
2. Arduino IDE

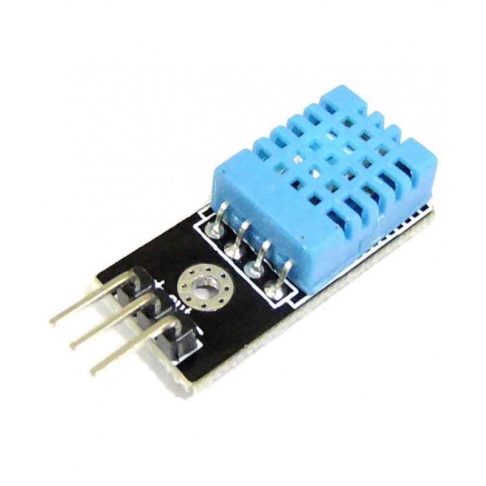
**Brief Description**

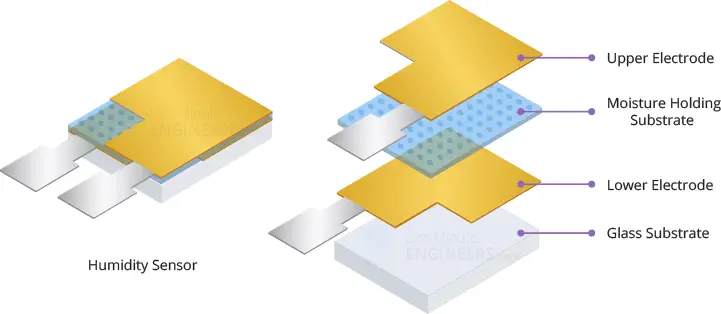
**NodeMCU V3**

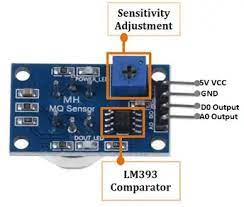
NodeMCU V3 is an open-source ESP8266 development kit, armed with the CH340G USBTTL Serial chip. It has firmware that runs on ESP8266 Wi-Fi SoC from Espressif Systems.Whilst cheaper, CH340 is super reliable even in industrial applications. It is tested to bestable on all supported platforms as well. It can be simply coded in Arduino IDE. It has avery low current consumption between 15 µA to 400 mA.



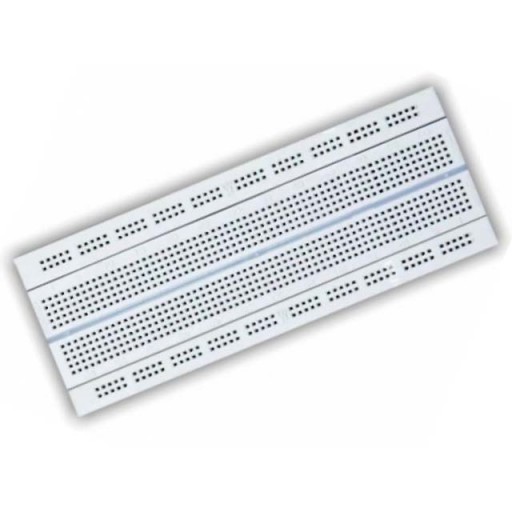
**DHT11 Sensor Module**

 The DHT11 is a temperature and humidity sensor that gives digital output in terms of voltage. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air. As shown in Fig. 2.2, we need to supply a voltage of 5V (DC) to the Vcc pin and ground it to the GND pin. The sensor output can be easily read from the Data pin in terms of voltage (in digital mode). Humidity Measurement: The humidity sensing capacitor has two electrodes with a moisture-holding substrate as a dielectric between them as shown in Fig 2.3. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process these changed resistance values and then converts them into digital form. Temperature Measurement: For measuring the temperature, the DHT11 sensor uses a negative temperature coefficient thermistor, which causes a decrease in its resistance value with an increase in temperature. To get a wide range of resistance values, the sensor is made up of semiconductor ceramics or polymers.

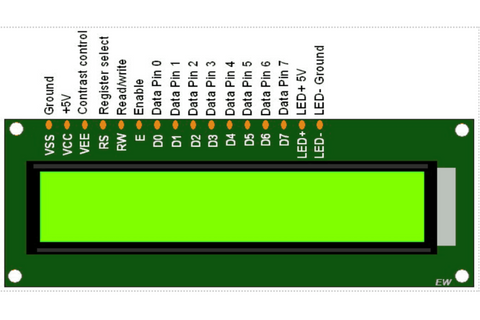
**MQ-135 Gas Sensor Module**

The material of MQ135 is SnO2, it is a special material: when exposed to clean air, it is hardly being conducted, however, when put in an environment with combustible gas, it has a 7 | P a g e pretty performance of conductivity. Just make a simple electronic circuit, and convert the change of conductivity to a corresponding output signal. MQ135 gas sensor is sensitive to Ammonia, Sulphide, Benzene steam, smoke and other harmful gases. Used for family, surrounding environment noxious gas detection device, apply to ammonia, aromatics, sulphur, benzene vapor, and other harmful gases/smoke, gas detection, tested concentration range: 10 to 1000ppm. In a normal environment, the environment which doesn’t have detected gas set the sensor’s output voltage as the reference voltage, the analog output voltage will be about 1V, when the sensor detects gas, harmful gas concentration increases by 20ppm per voltage increase by 0.1V.

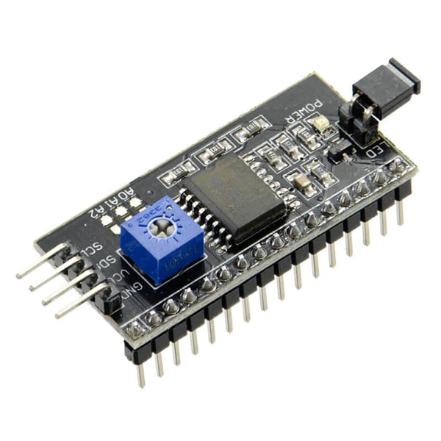
**Bread Board**

A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits. Unlike a perfboard or stripboard, breadboards do not require soldering or destruction of tracks and are hence reusable. For this reason, breadboards are also popular with students and in technological education.

**LCD Display**

A 16x2 LCD display is a common electronic component with the ability to show 16 characters per line across two rows. Utilized in various projects, such as those involving Arduino or Raspberry Pi, it requires connection to a microcontroller. By initializing and programming the display through code, users can showcase messages, sensor data, or other information on the screen. Pin configurations may vary, necessitating adherence to the LCD datasheet for proper connections.

**I2C Converter**

An I2C converter, often referred to as an I2C bus converter or I2C level shifter, facilitates communication between devices operating at different voltage levels on the I2C (Inter-Integrated Circuit) bus. It ensures compatibility between devices with varying voltage requirements, allowing them to exchange data seamlessly. This converter typically acts as a bridge, adapting voltage levels to enable proper communication between microcontrollers, sensors, or other I2C-enabled components.

**Blynk IOT Cloud**

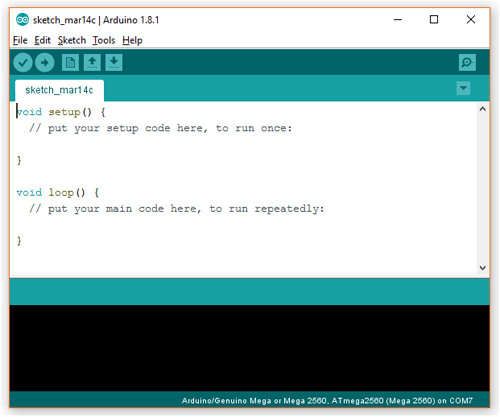
Blynk is an Internet of Things (IoT) platform that provides a cloud-based service for building and managing IoT applications. Blynk's IoT cloud platform enables users to connect hardware devices, such as microcontrollers (Arduino, Raspberry Pi, etc.) and IoT modules, to the cloud. Users can then remotely monitor and control these devices through a mobile app or web interface.

Key features of Blynk IoT cloud include:

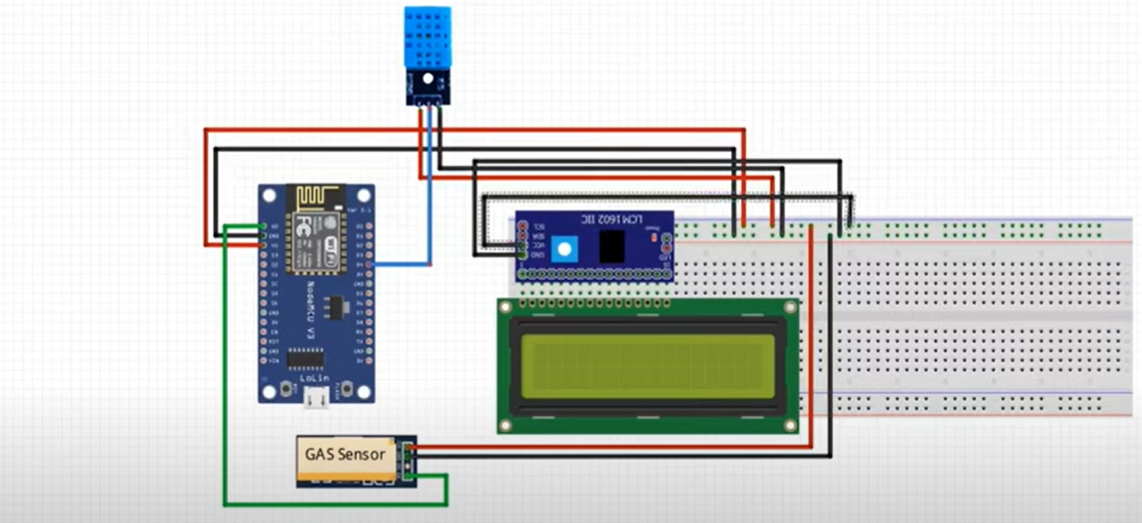
* **Drag-and-Drop Interface**: Blynk offers a user-friendly, drag-and-drop interface to design custom dashboards for controlling and monitoring IoT devices. Users can easily create buttons, sliders, graphs, and other widgets to interact with their connected devices.
* **Wide Hardware Compatibility**: Blynk supports a variety of hardware platforms, making it versatile for different IoT projects. This includes popular microcontrollers like Arduino, Raspberry Pi, ESP8266, ESP32, and others.
* **Mobile App Integration**: Blynk provides a mobile app for iOS and Android, allowing users to remotely access and control their IoT devices. The app facilitates real-time monitoring, data visualization, and interaction with connected devices.
* **Cloud Connectivity**: Blynk's cloud infrastructure enables seamless communication between IoT devices and the user interface. This cloud connectivity allows for data storage, synchronization, and access to devices from anywhere with an internet connection.
* **Widgets and Notifications**: Blynk supports a variety of widgets (UI elements) that can be added to dashboards, such as buttons, sliders, gauges, and more. Users can also set up notifications to receive alerts based on specific events or sensor readings.

**Arduino IDE**

The Arduino IDE (Integrated Development Environment) serves as a central hub for programming Arduino microcontrollers, offering a user-friendly interface and essential tools for software development. With a simple code editor, users can create and manage Arduino sketches, each comprising a setup() function for initialization and a loop() function for continuous execution. The IDE supports easy integration of external libraries, facilitating the addition of specialized functionalities to Arduino projects. A built-in Serial Monitor aids in debugging and monitoring data exchanges between the Arduino board and the computer. The Board Manager simplifies the configuration for various Arduino boards, ensuring compatibility and ease of use. The platform is cross-platform, catering to users on Windows, macOS, and Linux, making it accessible to a diverse audience. As open-source software, the Arduino IDE allows users to delve into its source code, fostering community collaboration and contributions.

In essence, the Arduino IDE streamlines the development process by providing a comprehensive environment that encompasses code creation, verification, and uploading to the Arduino microcontroller. Its user-friendly nature, coupled with cross-platform support and an open-source ethos, has solidified its status as a go-to tool for both beginners and experienced developers engaged in creating diverse electronic projects with Arduino.

**CIRCUIT DIAGRAM**

****

**CONNECTIONS**

**1. Connect DHT11 Sensor to ESP8266:**

* Connect the VCC pin of the DHT11 to the 3.3V pin on the ESP8266.
* Connect the GND pin of the DHT11 to the GND pin on the ESP8266.
* Connect the data pin (out) of the DHT11 to, for example, GPIO pin D4 on the ESP8266.

**2. Connect MQ-135 Gas Sensor to ESP8266:**

* Connect the VCC pin of the MQ-135 to the 5V pin on the ESP8266.
* Connect the GND pin of the MQ-135 to the GND pin on the ESP8266.
* Connect the analog output pin (AO) of the MQ-135 to, for example, analog pin A0 on the ESP8266.

**3. Connect I2C LCD 16x2 Display to ESP8266:**

* Connect the VCC pin of the LCD to the 5V pin on the ESP8266.
* Connect the GND pin of the LCD to the GND pin on the ESP8266.
* Connect the SDA pin of the LCD to, for example, GPIO pin D2 on the ESP8266.
* Connect the SCL pin of the LCD to, for example, GPIO pin D1 on the ESP8266.

**4. Uploading the Code:**

* Connect the ESP8266 to your computer via USB.
* Select the correct board and port in the Arduino IDE.
* Upload the code to the ESP8266.

Ensure that the power requirements of the sensors (DHT11 and MQ-135) and the LCD are met.

Double-check the pin configurations of your specific sensors, as they may vary.

Use jumper wires to establish connections between the components on the breadboard.

Verify that the I2C address of your LCD matches the address in the code (0x27 is a common default address).

Always power off the system before making or modifying connections.

**SOFTWARE REQUIREMENTS**

**Setting Up Arduino IDE:**

* Download and install the Arduino IDE from the official Arduino website.
* Open Arduino IDE, go to "File" -> "Preferences" and add the ESP8266 board support through the "Additional Boards Manager URLs" field.
* Install the ESP8266 board support using the Boards Manager.

**Installing Necessary Libraries:**

* Install the required libraries for DHT11 sensor, MQ-135 sensor, and the LiquidCrystal\_I2C library for the LCD.
* For DHT11: Install the "DHT Sensor Library" by Adafruit.
* For MQ-135: Install the "MQ135" library.
* For LCD: Install the "LiquidCrystal\_I2C" library.

**Blynk Integration:**

1. Install Blynk Library:

* In Arduino IDE, go to "Sketch" -> "Include Library" -> "Manage Libraries..."
* Search for "Blynk" and install the Blynk library.

2. Create a Blynk Project:

* Download the Blynk app on your smartphone and create a new project.
* In the Blynk app, add three Value Display widgets for moisture, temperature, and gas values. Each widget has its unique Virtual Pin (Vx) number.

3. Obtain Blynk Auth Token:

* In the Blynk app, click on the Nut icon to access Project Settings.
* Copy the "Authentication Token" (Auth Token).

**SOURCE CODE**

#define BLYNK\_TEMPLATE\_ID "TMPLgwKssgggsnFXp"

#define BLYNK\_DEVICE\_NAME "Air Quality Monitoring"

#define BLYNK\_AUTH\_TOKEN"k03gT6nJosdsfsffesrJV\_S5SXEAdgdsDfj"

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.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[] = "NAV"; // type your wifi name

char pass[] = "naveen123"; // 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 DHT11

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.

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("shameer50@gmail.com", "Alert", "Bad Air!");

Blynk.logEvent("pollution\_alert","Bad Air");

}

}

**Code Overview**

**1. Libraries and Definitions:**

* The code includes necessary libraries for ESP8266, Blynk, DHT (for the DHT11 sensor), and LiquidCrystal\_I2C (for the LCD).
* Definitions are set for Blynk template ID, device name, authentication token, and Wi-Fi credentials.

**2. LCD Configuration:**

* The I2C LCD is configured with its address (0x27) and dimensions (16x2).
* A custom character is defined for the degree symbol.

**3. Blynk and Wi-Fi Configuration:**

* Blynk authentication token, Wi-Fi SSID, and Wi-Fi password are configured.
* A BlynkTimer object is created for scheduling periodic tasks.

**4. Sensor and Display Initialization in setup():**

* Serial communication is initiated along with the Blynk connection, DHT sensor, and timer for periodic sensor readings.
* The LCD is initialized, and an introductory message is displayed.

**5. Sensor Reading Function:**

* A function (sendSensor()) reads temperature, humidity, and gas values.
* The values are sent to corresponding virtual pins on Blynk.

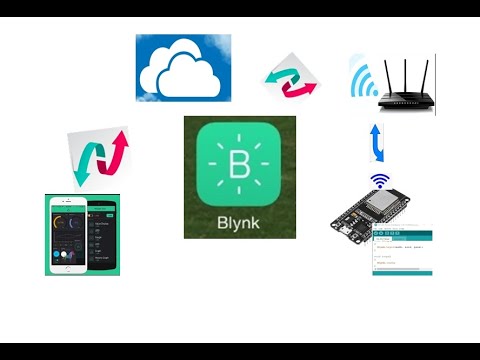
**6. Main Loop:**

* The main loop continuously runs Blynk and timer tasks.
* Temperature and humidity are displayed on the LCD.
* Gas value is checked, and whether the air is fresh or bad is displayed on the LCD.
* A Blynk event log is triggered if the gas value indicates bad air.

**WORKING**

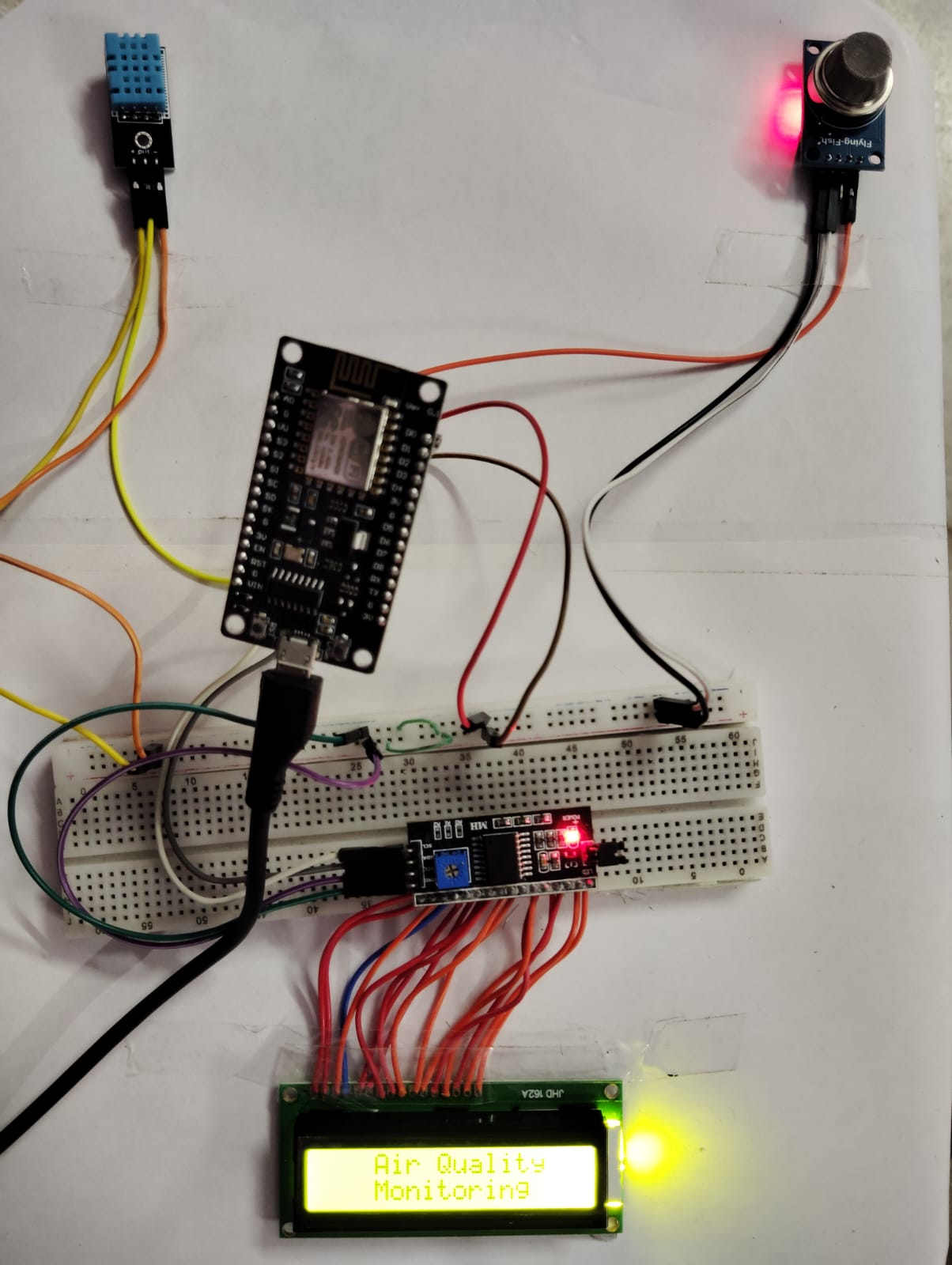
1. **Temperature and Humidity Monitoring:**
   * Utilizes a DHT11 sensor to measure temperature and humidity in the surrounding environment.
   * Displays the temperature in degrees Celsius on a 16x2 LCD with a custom degree symbol.
   * Shows the humidity percentage on the same LCD display.
2. **Air Quality Monitoring:**
   * Incorporates an MQ-135 gas sensor to detect air quality based on the concentration of certain gases in the environment.
   * The gas sensor readings are displayed on the LCD screen.
   * If the gas concentration exceeds a predefined threshold (600 in the provided code), the LCD indicates "Bad Air."
3. **Remote Monitoring with Blynk:**
   * Integrates with the Blynk IoT platform for remote monitoring and control.
   * Sends temperature, humidity, and gas values to the Blynk mobile app in real-time.
   * Blynk app widgets visualize the data, allowing users to monitor air quality from anywhere with an internet connection.
4. **Alerts and Logging:**
   * Generates an event log on the Blynk app in case of bad air quality, triggering a pollution alert.
   * This provides a historical record of pollution events, aiding in analysis and awareness.
5. **LCD Display for Local Visualization:**
   * The LCD screen serves as a local interface for immediate visual feedback on environmental conditions.
   * It displays key information such as temperature, humidity, and air quality status.
6. **Customizable Thresholds and Configurations:**
   * The code includes threshold values that can be adjusted based on specific air quality standards or project requirements.
   * Users can customize the code and configurations to suit their environmental monitoring needs.
7. **Continuous Monitoring and Timed Readings:**
   * The system continuously monitors environmental parameters using a BlynkTimer for timed sensor readings.
   * Regular updates ensure that users have access to up-to-date information on temperature, humidity, and air quality.

In summary, the project provides a comprehensive solution for monitoring and visualizing key environmental parameters, offering both local and remote access to real-time data. It is suitable for applications where awareness of indoor or outdoor air quality is essential, such as smart home systems, environmental monitoring, or workplace safety initiatives.



**RESULTS**

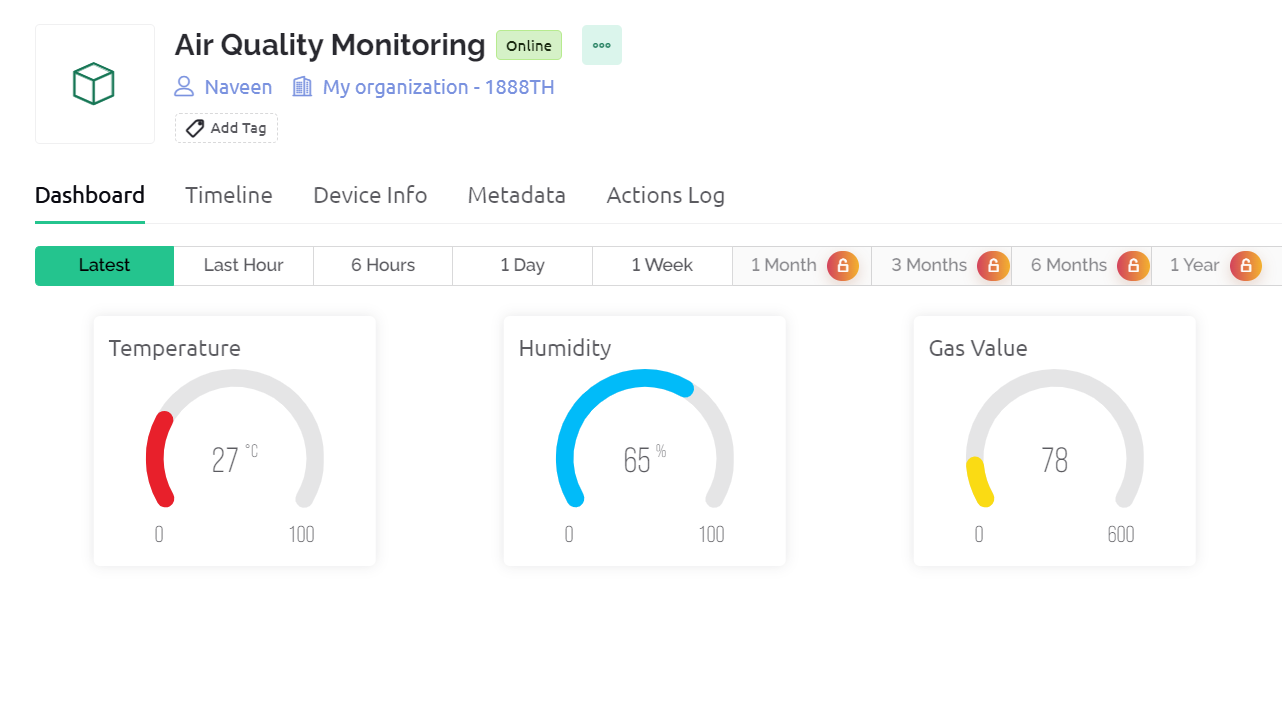
**LCD Outputs:**

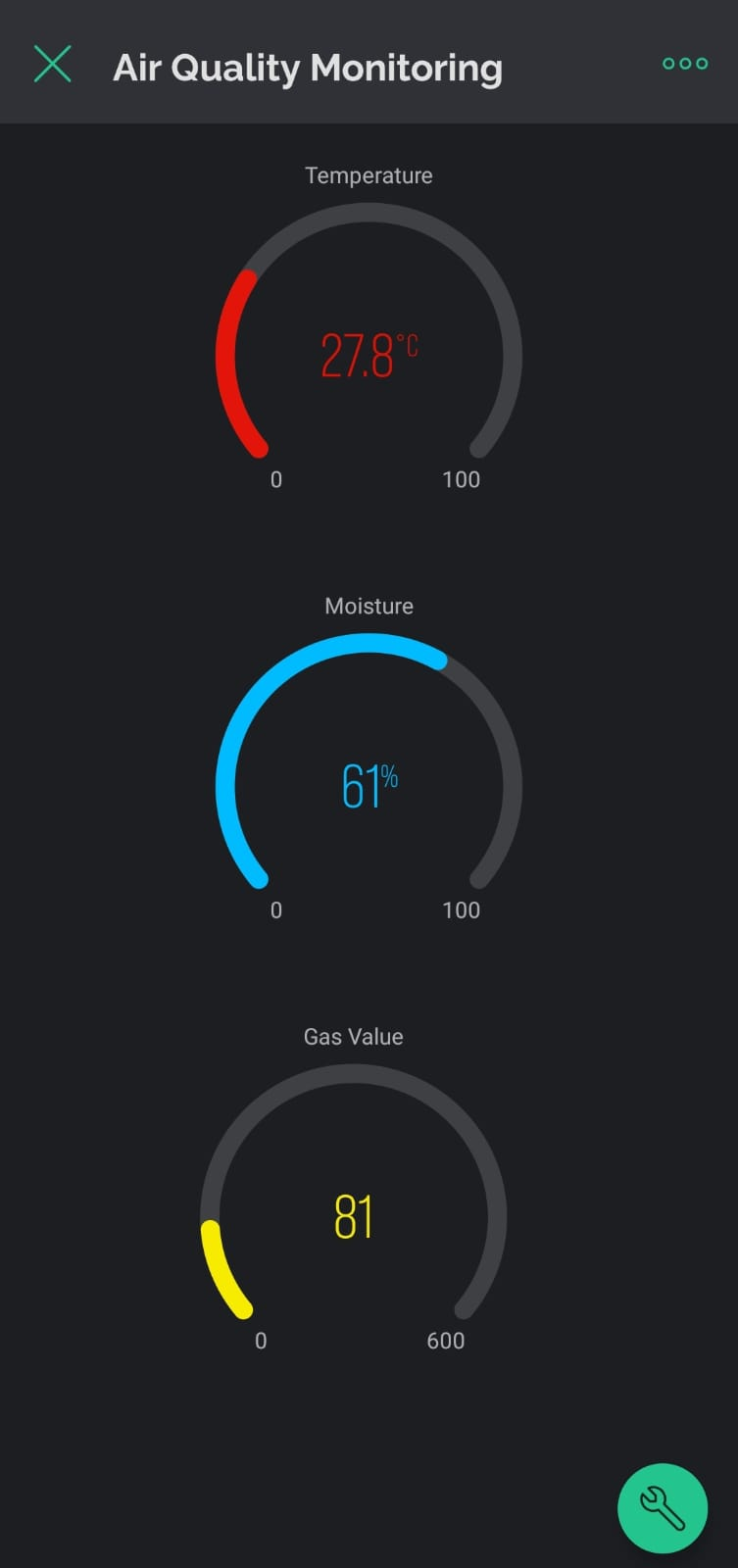
****

****

****

****

**Blynk Application Outputs:**

****

**CONCLUSION**

The IoT-based Air Quality Monitoring System successfully provides a comprehensive solution for real-time monitoring of crucial environmental parameters. By integrating sensors like the DHT11 for temperature and humidity and the MQ-135 for air quality, coupled with an ESP8266 microcontroller and an I2C LCD display, the system offers both local visualization and remote monitoring capabilities through the Blynk IoT platform.

The project's key features include continuous monitoring of temperature, humidity, and air quality, enabling users to stay informed about their surroundings. The local display on the 16x2 LCD offers immediate feedback, while the Blynk app provides a remote interface for accessing real-time data and receiving pollution alerts.

Customization is a notable aspect, allowing users to adjust threshold values and configurations to meet specific project requirements or adhere to air quality standards. The alerting system further enhances user awareness by logging events on the Blynk app in case of poor air quality.

Ultimately, this Air Quality Monitoring System addresses the growing importance of environmental awareness, providing individuals, households, or organizations with a valuable tool for making informed decisions about their indoor or outdoor air quality. Whether for smart home applications, workplace safety, or general environmental monitoring, the project exemplifies the potential of IoT in enhancing our understanding and control over the environments we inhabit.

**REFERENCES**

[**https://blynk.cloud/dashboard/248784/global/devices/1688451**](https://blynk.cloud/dashboard/248784/global/devices/1688451)

[**https://docs.arduino.cc/software/ide-v1/tutorials/installing-libraries**](https://docs.arduino.cc/software/ide-v1/tutorials/installing-libraries)

[**https://randomnerdtutorials.com/getting-started-with-esp8266-wifi-transceiver-review/**](https://randomnerdtutorials.com/getting-started-with-esp8266-wifi-transceiver-review/)

[**https://lastminuteengineers.com/esp8266-i2c-lcd-tutorial/**](https://lastminuteengineers.com/esp8266-i2c-lcd-tutorial/)

[**https://www.electronicsforu.com/electronics-projects/iot-enabled-air-pollution-meter**](https://www.electronicsforu.com/electronics-projects/iot-enabled-air-pollution-meter)