SMART ENVIRONMENTAL HEALTH MONITORING SYSTEM WITH CLOUD INTEGRATION

EC5512 - SUMMER PROJECT REPORT

Submitted by

ABHIROOPA S 2022504008

VASUNTHIRA S 2022504054

SEAJANGINI S V 2022504513

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS ENGINEERING

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY: CHENNAI - 600 044

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project report **"SMART ENVIRONMENTAL HEALTH MONITORING SYSTEM WITH CLOUD INTEGRATION"** is the bonafide work of

ABHIROOPA S	**2022504008**
VASUNTHIRA S	**2022504054**
SEAJANGINI S V	**2022504513**

who carried out the project under my supervision.

SIGNATURE

Dr.C.Sridevi

PROJECT GUIDE

Associate Professor

Department of Electronics Engineering,

Madras Institute of Technology,

Anna University, Chennai-44.

ACKNOWLEDGEMENT

With profound gratitude and due regards, We sincerely acknowledge with thanks the opportunity provided to us by our respectful, Dean **Dr.K.RAVICHANDRAN**, Madras Institute of Technology, Anna University for providing a good environment and facilities.

We thank our respectful Head of the Department, **Dr.D.MEGANATHAN** ,Professor and Head of Electronics Engineering, Madras Institute of Technology, Anna University for his encouragement during the Project work.

We sincerely express my gratitude to our project supervisor, **Dr.C.SRIDEVI**, Associate Professor, Department of Electronics Engineering, for being an inspiration to us and for her valuable guidance throughout the course of this project.

Place: Chennai

Date: 22/11/2024

ABHIROOPA S 2022504008

VASUNTHIRA S 2022504054

SEAJANGINI S V 2022504513

ABSTRACT

In this project, an air quality monitoring system has been designed to measure environmental parameters such as temperature, humidity, and pollutant levels using multiple sensors, including the MQ135 and MQ7 sensors. The MQ135 sensor is used to detect a range of gases such as ammonia, carbon dioxide, and methane, while the MQ7 sensor is specifically used to measure carbon monoxide (CO) levels. A PM2.5 sensor is employed to measure particulate matter in the air, which plays a significant role in assessing the impact of fine particles on human health. The primary focus of the project is the calculation of the Air Quality Index (AQI), which provides a standardized measure of air pollution levels based on various pollutants. The AQI for each pollutant is calculated individually, with particular emphasis on the contributions of PM2.5 and CO, as these pollutants are considered to have the most significant health impacts. The overall AQI is determined by considering the concentrations of PM2.5 and CO, which are the most critical pollutants in the monitored environment. The calculated AQI values, along with corresponding environmental data, are displayed on an LCD. Additionally, ThingSpeak, a cloud-based IoT platform, is integrated to display gas concentration values and AQI trends in graphical form in real-time. The system aims to provide accurate and accessible data for air quality monitoring, enabling individuals and communities to assess their exposure to air pollutants and take necessary precautions. The importance of continuous monitoring and timely alerts for improved health and safety in environments with compromised air quality is emphasized throughout the project. The integration of real-time data collection and cloud-based analytics allows for the continuous tracking of air quality conditions in both indoor and outdoor environments. This project serves as a foundation for future developments in smart environmental monitoring, contributing to healthier living conditions and raising awareness of air pollution's impact on health.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER -1

1.1 INTRODUCTION

Air pollution is a growing concern worldwide, impacting public health and the environment. With the increase in industrialization, urbanization, and vehicular emissions, air quality has deteriorated, leading to health issues such as respiratory diseases, allergies, and cardiovascular problems. Monitoring air quality is essential for understanding pollution levels and mitigating health risks. The project focuses on creating an air quality monitoring system that measures various environmental parameters like temperature, humidity, and the concentration of hazardous gases, such as carbon monoxide (CO) and particulate matter (PM2.5). This data is then used to calculate the Air Quality Index (AQI), providing a real-time assessment of air quality conditions.

1.2 PROJECT AIM

The aim of the project is to design and implement an air quality monitoring system that accurately measures temperature, humidity, and pollutant levels using MQ135, MQ7, and PM2.5 sensors. The system calculates the AQI for each pollutant, with particular focus on CO and PM2.5 contributions, and displays the results in real-time. Additionally, the system aims to visualize this data through ThingSpeak.

1.3 PROJECT OBJECTIVE

- To display AQI values and environmental parameters on an LCD screen for local monitoring.
- To integrate the system with Thing Speak for real-time visualization of gas values and AQI trends.
- To provide a reliable and cost-effective solution for continuous air quality monitoring.

1.4 PROJECT SCOPE AND LIMITATION

The scope of this project includes the measurement of temperature, humidity, and specific gas concentrations (CO, ammonia, methane, and particulate matter) using commonly available sensors such as MQ135, MQ7, and PM2.5 sensors. The system is designed for indoor and outdoor environments where air quality is a concern. Data collection and visualization are achieved through a microcontroller (e.g., NodeMCU), which interfaces with the sensors and ThingSpeak. However, the system has limitations, including dependency on the accuracy of the sensors used and environmental conditions that may affect sensor performance. The project is also limited by the resolution and range of the sensors, which may not provide highly precise measurements under extreme conditions. The system may not account for all possible pollutants and does not provide advanced features like real-time alerts or automated adjustments based on detected pollution levels.

1.5 DESCRIPTION OF THE PROJECT

This project involves the development of a real-time air quality monitoring system that uses a combination of sensors to detect temperature, humidity, and harmful pollutants. The MQ135 sensor is used to measure gases like ammonia and carbon dioxide, the MQ7 sensor focuses on carbon monoxide detection, and the PM2.5 sensor measures fine particulate matter in the air. The readings from these sensors are processed to calculate AQI values for each pollutant, with special attention to CO and PM2.5, which are considered the most harmful to human health. These AQI values are then displayed on an LCD screen for local monitoring. The system is connected to ThingSpeak, where users can view real-time graphs of gas concentrations and AQI trends, providing an accessible and informative tool for monitoring air quality. The project serves as an important step toward creating a low-cost, highly efficient, and easily deployable solution for environmental monitoring

CHAPTER -2
LITERATURE SURVEY

Several studies have been conducted to develop effective air quality monitoring systems using low-cost sensors, microcontrollers, and IoT platforms. These works aim to provide real-time, cost-effective solutions for monitoring and mitigating the effects of air pollution.

Patel et al. (2018) presented a system for monitoring air quality using MQ135 and MQ7 sensors, interfaced with a microcontroller. Their research focused on detecting harmful gases like carbon monoxide (CO) and ammonia in urban environments. Similarly, Sharma et al. (2019) implemented a particulate matter monitoring system using a PM2.5 sensor, highlighting its role in tracking fine particulate pollutants that pose significant health risks. These studies demonstrated the effectiveness of combining multiple sensors to measure different air quality parameters.

In systems with multiple sensors, prior research by Gupta and Roy (2020) explored the use of analog multiplexers to optimize microcontroller GPIO usage. They found that multiplexers efficiently switch between sensors, allowing data acquisition from various sources with minimal hardware complexity. This approach is particularly useful in compact designs where GPIO pins are limited, such as in NodeMCU-based systems.

Several works, such as those by Zhang et al. (2020), have incorporated IoT platforms like ThingSpeak for real-time air quality data visualization and remote monitoring. These systems allow for historical data storage, trend analysis, and notifications, enhancing the utility of air quality monitoring devices. IoT integration has been shown to improve accessibility and decision-making, especially in urban and industrial settings.

Research by Banerjee et al. (2017) emphasized the importance of user-friendly alert mechanisms in air quality systems. Their work incorporated RGB LEDs to display air quality levels visually, with colors corresponding to predefined AQI ranges. They also included buzzers to provide audible warnings, ensuring immediate attention when air quality deteriorates. These systems were particularly effective in scenarios where immediate intervention was necessary.

CHAPTER -3

3.1 HARDWARE REQUIREMENT

- NodeMCU ESP8266
- DHT11 Sensor
- MQ135 Gas Sensor
- MQ7 Sensor
- PM2.5 Sensor
- Analog Multiplexer
- I2C Serial Interface Module
- 16X2 LCD Display
- Piezo Buzzer
- RGB LED
- B type Cable
- Jumper Wires

3.2 SOFTWARE REQUIREMENTS

- Arduino IDE
- ThingSpeak

CHAPTER-4

DESCRIPTION OF HARDWARE REQUIRED

4.1 NODEMCU ESP8266

The NodeMCU ESP8266 is a powerful and versatile open-source platform designed for Internet of Things (IoT) applications. Based on the ESP8266 Wi-Fi module, it combines a microcontroller with integrated Wi-Fi capabilities, making it a cost-effective solution for developing connected devices. The board is equipped with the ESP-12E module, which features a Tensilica L106 32-bit RISC processor, operating at clock speeds of 80 MHz to 160 MHz. With 4MB of flash memory, it supports a wide range of applications, from sensor data collection to smart home automation. Its compact design includes a USB-to-serial converter for easy programming and debugging, and it can be powered via USB or an external 5V supply.

The NodeMCU supports multiple communication protocols such as UART, SPI, and I2C, allowing seamless interaction with various peripherals and sensors. Additionally, it integrates a 10-bit ADC for analog input and provides PWM functionality for control applications. The board is fully compatible with the Arduino IDE and Lua scripting language, simplifying the development process for beginners and experienced developers alike. With its built-in Wi-Fi and low power consumption, the NodeMCU ESP8266 is ideal for creating innovative IoT solutions in areas like remote monitoring, automation, and smart devices. The NodeMCU ESP8266 has gained widespread popularity due to its ease of use, affordability, and flexibility. The integrated Wi-Fi transceiver enables it to connect to local networks or the internet, allowing devices to send and receive data wirelessly. This feature is particularly useful for IoT projects, where remote control and data collection are essential. Its compact size and built-in pin headers make it ideal for prototyping and embedding.

The ability to code in Lua offers an easy entry point for beginners, while the Arduino IDE provides access to an extensive library ecosystem for advanced development. This combination of hardware and software flexibility has made the NodeMCU a go-to solution for hobbyists, educators, and professionals working on IoT systems. Whether it's building a smart home system, a weather station, or an automated industrial application, the NodeMCU provides the necessary tools and features to bring ideas to life efficiently and effectively. Its versatility, combined with a strong community of developers, ensures that ample resources and support are available for tackling complex projects.

4.1.1 HARDWARE

The NodeMCU ESP8266 hardware revolves around the ESP-12E module, which contains the ESP8266 microcontroller. This microcontroller features a Tensilica L106 32-bit RISC processor running at 80 MHz, with an option to boost to 160 MHz for enhanced performance. The module offers 4MB of onboard flash memory for firmware and program storage. It operates at 3.3V, with a built-in voltage regulator for stable operation. Power can also be supplied via a 5V input through the VIN pin or USB. The board includes a USB-to-serial converter (CP2102 or CH340G) for seamless programming and debugging through a micro-USB connection.

The NodeMCU provides multiple interfaces, including GPIO, analog input (A0), and communication protocols like UART, I2C, and SPI, enabling integration with various sensors and peripherals. PWM support on the GPIO pins makes it suitable for controlling LEDs and motors. The onboard reset (RST) and flash buttons simplify development and debugging, while the built-in LED on GPIO2 is useful for quick functionality tests. With its compact design and robust features, the NodeMCU ESP8266 serves as a powerful platform for prototyping and deploying IoT applications.

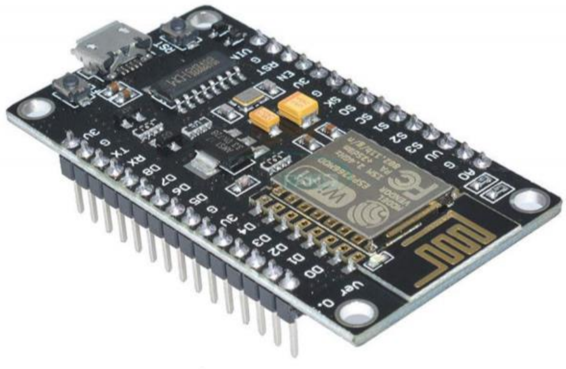

Figure 4.1 NodeMCU ESP8266

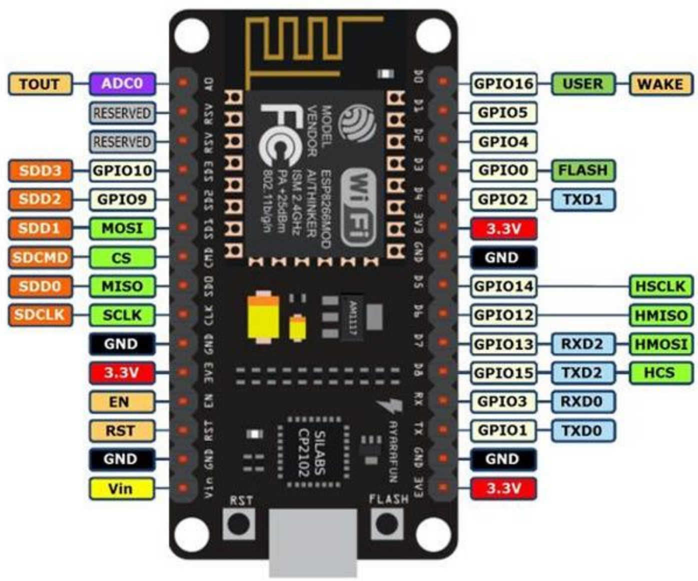

Figure 4.2 Pin Diagram of NodeMCU ESP8266

SOME TECHNICAL SPECIFICATIONS FOR NODEMCU ESP8266:-

- Microcontroller: ESP8266
- Processor:32-bit Tensilica Xtensa LX106 CPU
- Clock Speed: 80 MHz (default), can be overclocked to 160 MHz
- Operating Voltage: 3.3V
- Input Voltage: 4.5V to 9V (via the onboard voltage regulator)
- RAM: 160 KB of SRAM
- Flash Memory: 4 MB (varies with different models)

4.1.2 PIN DESCRIPTION

Pin Name	Details
VIN	Power input pin (5V). Used to power the NodeMCU when not using the USB connection.
3V3	Provides a regulated 3.3V output. Can power external components or circuits requiring 3.3V..
GND	Ground pins. Connect to the ground of the circuit.
D0	GPIO16; General-purpose I/O. Can be used as input or output. Also used to wake the board from deep sleep.
D1	GPIO5; General-purpose I/O. Supports PWM, I2C (SCL - Serial Clock Line), and SPI (limited use).
D2	GPIO4; General-purpose I/O. Supports PWM, I2C (SDA - Serial Data Line), and SPI (limited use).
D3	GPIO0; Boot mode selection pin. During boot, it determines whether the system boots normally (LOW) or loads custom firmware.
D4	GPIO2; General-purpose I/O. Onboard LED is typically connected to this pin.
D5	GPIO14; General-purpose I/O. SPI CLK (Serial Clock) for SPI communication. Capable of PWM and I2C SCL.
D6	GPIO12; General-purpose I/O. SPI MISO (Master In Slave Out) for SPI communication. Capable of PWM and I2C SCL.

D7	GPIO13; General-purpose I/O. SPI MOSI (Master Out Slave In) for SPI communication. Capable of PWM and I2C SDA.
D8	GPIO15; General-purpose I/O. SPI CS (Chip Select) for SPI communication. Must be LOW during boot to enable normal boot.
RX	GPIO3; UART RX (Serial data input). Used for receiving data via UART. Can also be used as a general-purpose I/O pin.
TX	GPIO1; UART TX (Serial data output). Used for transmitting data via UART. Can also be used as a general-purpose I/O pin.
A0	Analog input pin. Reads analog voltages between 0–1V with 10-bit resolution. Typically used for sensors like temperature or light sensors.
RST	Reset pin. Active LOW. Pulling this pin LOW resets the ESP8266 chip.
EN	Enable pin. Active HIGH. Must be pulled HIGH for normal operation; pulling it LOW disables the ESP8266

4.1.3 COMMUNICATION

The NodeMCU ESP8266 offers robust communication capabilities that make it ideal for IoT applications. At its core is an integrated Wi-Fi transceiver supporting IEEE 802.11 b/g/n standards, enabling seamless wireless connectivity. The board can connect to local networks, act as an access point, or operate in a hybrid mode, allowing devices to communicate with each other or the internet. Its TCP/IP stack supports standard internet protocols, ensuring compatibility with web servers, MQTT brokers, and cloud platforms. This makes it perfect for tasks like remote monitoring, data transfer, or controlling devices via web interfaces.

In addition to Wi-Fi, the NodeMCU supports multiple hardware communication protocols, including UART, I2C, and SPI, to interface with sensors, actuators, and other devices. The UART pins allow serial communication with devices like GPS modules or debugging tools. The I2C interface facilitates communication with multiple devices over just two pins, making it ideal for connecting sensors or displays. The SPI protocol is well-suited for high-speed communication with components like flash memory and LCDs. These versatile communication options, combined with the NodeMCU's Wi-Fi capabilities, provide a comprehensive solution for building connected systems.

4.2 DHT11 SENSOR

The DHT11 is a digital sensor widely used for measuring temperature and humidity in IoT and environmental monitoring applications. It features a capacitive humidity sensor and a thermistor for temperature measurement, providing accurate readings with a temperature range of 0–50°C (±2°C accuracy) and a humidity range of 20–90% RH (±5% accuracy). The sensor communicates using a single digital pin, making it simple to interface with microcontrollers like Arduino and NodeMCU. While it has a slow sampling rate of once per second, the DHT11 is reliable, cost-effective, and ideal for applications such as home automation, weather stations, and greenhouse monitoring.

TECHNICAL SPECIFICATIONS OF DHT11 SENSOR

- Temperature Range: 0°C to 50°C
- Temperature Accuracy: ±2°C
- Humidity Range: 20% to 90% RH
- Operating Voltage: 3.3V to 5.5V
- Communication Protocol: Single-wire digital communication
- Sampling Rate: 1 Hz (1 reading per second)
- Output: 16-bit digital signal
- Dimensions: 15.5mm x 12mm x 5.5mm

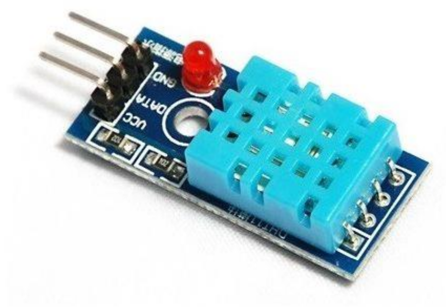

Figure 4.3 DHT11 Sensor

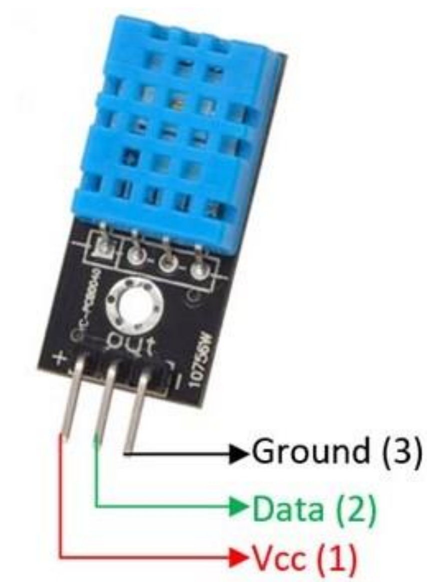

Figure 4.4 Pin Diagram of DHT11 Sensor

4.3 MQ 135 GAS SENSOR

The MQ135 is a widely used gas sensor designed for detecting a range of gases, including ammonia (NH3), carbon dioxide (CO2), benzene, alcohol, and smoke. It operates on a principle of resistive heating, where the resistance of the sensor changes based on the concentration of specific gases in the air. The sensor has a sensitivity that can be calibrated for different gases, making it suitable for air quality monitoring applications, particularly in environments like homes, factories, and hospitals. It is typically powered with a 5V DC supply and provides an analog output that can be read by a microcontroller to determine the concentration of gases. Due to its low power consumption and ease of use, the MQ135 is popular for air pollution detection and environmental monitoring projects. The sensor's response time and sensitivity can be adjusted by modifying the heating time and the load resistance in the circuit. While it is not highly specific to one gas, it offers broad detection for a range of airborne pollutants, providing a reliable estimate of overall air quality. To achieve accurate readings, the MQ135 typically requires a pre-heating period of several minutes when powered on, and its output should be calibrated based on environmental conditions.

Figure 4.5 MQ135 Gas Sensor

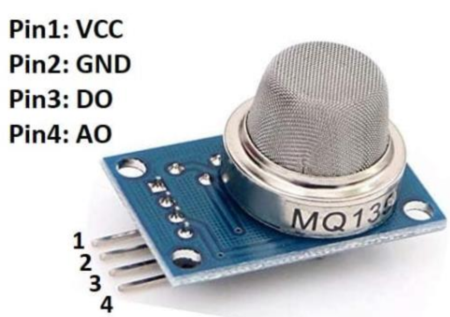

Figure 4.6 Pin Diagram of MQ135 Gas Sensor

TECHNICAL SPECIFICATIONS OF MQ135 SENSOR

- Gas Sensitivity: Sensitive to a variety of gases, including ammonia (NH3), carbon dioxide (CO2), benzene, alcohol, smoke, and other harmful gases.
- Operating Voltage: 5V DC (typically).
- Output Type: Analog voltage output (Vout) and a digital output for certain models.
- Operating Temperature: 0°C to 50°C.
- Operating Humidity: 30% to 90% RH.

- Preheating Time: Requires a preheating period of 24–48 hours for accurate readings after initial power-up.
- Response Time: Typically 30–60 seconds for gas detection.
- Sensitivity Range: Detection range for gases is typically from 10 to 1000 ppm for most gases.
- Power Consumption: Low power consumption, typically around 150mA during operation.
- Dimensions: Around 40mm x 30mm x 25mm

4.4 MQ7 SENSOR

The MQ7 is a gas sensor designed to detect carbon monoxide (CO) in the air. It operates based on a metal oxide semiconductor (MOS) sensing element, which changes resistance when exposed to CO. The sensor is commonly used in air quality monitoring systems, automotive CO detection, and industrial safety applications. The MQ7 requires a heating cycle for the sensor to stabilize, which typically takes a few minutes after powering on. It provides an analog output that is proportional to the concentration of carbon monoxide in the air, and it can be calibrated for different sensitivity levels. The sensor operates in a voltage range of 5V and has a typical sensitivity range of 10 to 1000 ppm for CO detection.

TECHNICAL SPECIFICATIONS OF MQ7 SENSOR

- Gas Sensitivity: Primarily detects carbon monoxide (CO), but can also respond to other gases such as methane (CH_4) and LPG.
- Operating Voltage: 5V DC (typically).
- Output Type: Analog voltage output.
- Sensor Type: Metal Oxide Semiconductor (MOS) type, where resistance changes in response to gas concentrations.
- Operating Temperature: -10°C to 50°C.
- Operating Humidity: 30% to 90% RH.

- Sensitivity Range: 10 to 1000 ppm for carbon monoxide.
- Response Time: Typically 30–60 seconds for gas detection.
- Power Consumption: Low power consumption, typically around 150 mA during operation.
- Dimensions: Approximately 40mm x 40mm x 18mm

Figure 4.7 MQ7 Sensor

Figure 4.8 Pin Diagram of MQ7 Sensor

4.5 PM2.5 SENSOR

A PM2.5 sensor is designed to detect fine particulate matter (PM2.5) in the air, which refers to particles with a diameter of 2.5 micrometers or smaller. These particles can include dust, dirt, soot, and liquid droplets that are small enough to be inhaled, posing significant health risks. The sensor uses a laser scattering or infrared light principle to detect these particles, providing real-time data on air quality. It typically outputs data in micrograms per cubic meter (µg/m³), and is commonly used in air quality monitoring systems, indoor air pollution detection, and environmental research. PM2.5 sensors are essential in assessing pollution levels in urban areas, factories, and homes, as prolonged exposure to fine particulate matter can lead to respiratory and cardiovascular problems.

Figure 4.9 PM2.5 Sensor

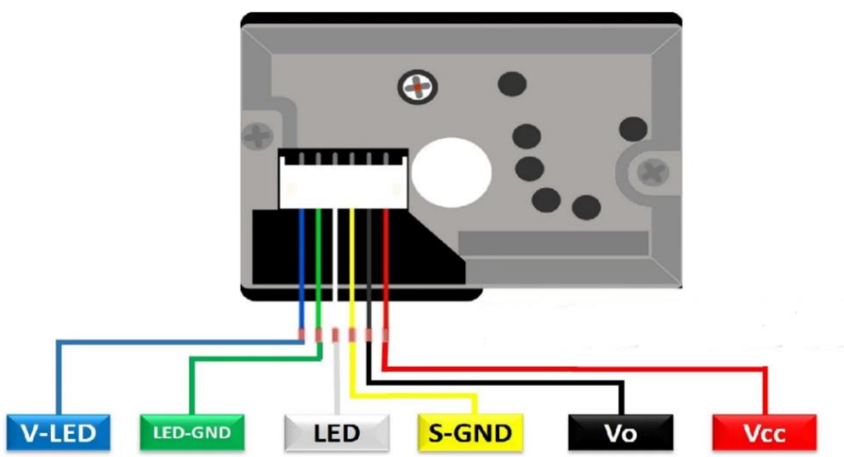

Figure 4.10 Pin Diagram of PM2.5 Sensor

TECHNICAL SPECIFICATIONS OF PM2.5 SENSOR

- Detection Range: Typically 0 to 500 µg/m³, but it can vary based on the specific model.
- Output Type: Analog or digital output, often with serial communication (e.g., UART or I2C).
- Particle Size Detection: Designed to detect particles with a diameter of 2.5 micrometers or smaller (PM2.5).

- Operating Voltage: 3.3V to 5V DC (depends on the model).
- Operating Temperature: 0°C to 50°C (varies by model).
- Operating Humidity: 10% to 90% RH (relative humidity).
- Response Time: Typically 1-3 seconds for detecting changes in particulate concentration.
- Power Consumption: Generally around 100-200mA during operation.
- Dimensions: 50mm x 50mm x 25mm.
- Accuracy: Typically ±10 μg/m³ or better, depending on the model.

4.6 16 CHANNEL ANALOG MULTIPLXER

An analog multiplexer is an electronic device used to route multiple analog signals to a single output line, effectively allowing one input to be selected at a time. It operates by using a set of control pins that select which input signal is connected to the output, while the other input lines remain disconnected. Analog multiplexers are often used in applications where multiple sensors or signal sources need to be monitored or processed by a single analog-to-digital converter (ADC) or processing unit, reducing the need for multiple channels. These multiplexers are widely used in data acquisition systems, signal routing, and communication systems, and typically come in various configurations, such as 4-channel, 8-channel, or even 16-channel versions, depending on the number of inputs required.

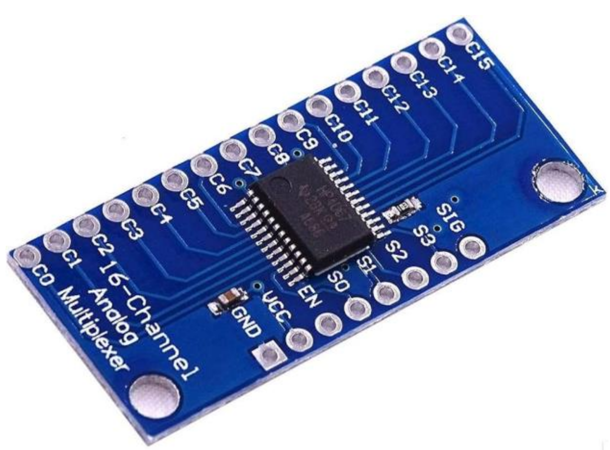

Figure 4.11 16 Channel Analog Multiplexer

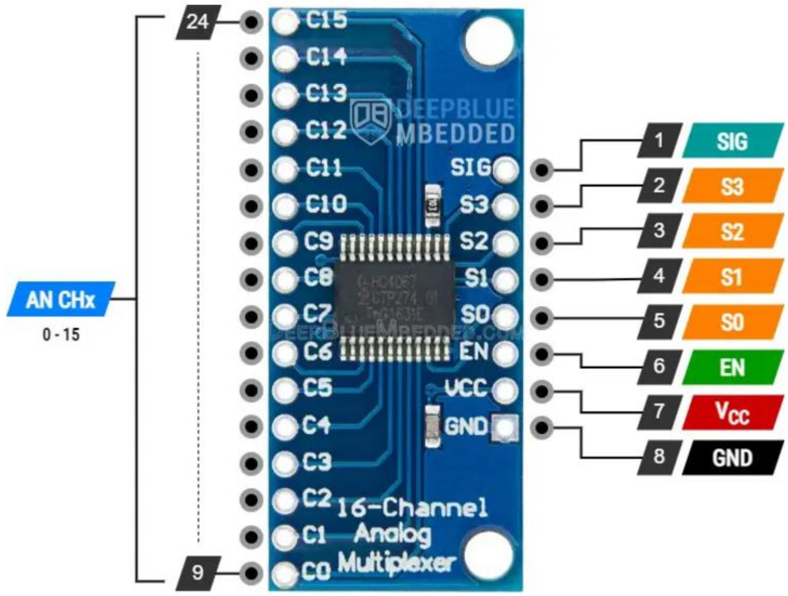

Figure 4.12 Pin Diagram of Analog Multiplexer

TECHNICAL SPECIFICATIONS OF ANALOG MULTIPLEXER.

- Operating Voltage: Typically 3.3V to 5V DC, depending on the model.
- Data Rate: Typically in the range of a few MHz
- Switch Resistance : Typically in the range of 50Ω to 200Ω
- Power Consumption: Low power consumption range of 1mA to 10mA
- Temperature Range: Typically -40°C to 85°C

4.7 I2C SERIAL INTERFACE MODULE

The I2C (Inter-Integrated Circuit) is a widely used, two-wire communication protocol designed for efficient data exchange between microcontrollers and peripheral devices. It uses a serial data line (SDA) and a clock line (SCL) to facilitate synchronized communication, supporting multiple devices on a single bus. Known for its simplicity and reduced wiring, I2C is ideal for applications requiring multiple sensors or modules. Each device on the I2C bus is assigned a unique address, allowing communication without signal interference. It supports bidirectional data transfer and operates in multiple speed modes, including standard, fast, and high-speed. Due to its scalability and reliability, I2C is commonly used in embedded systems, IoT devices, and sensor networks.

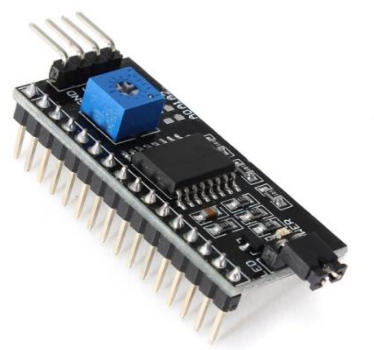

Figure 4.13 I2C Module

4.7.1 PRINCIPLE

I2C (Inter-Integrated Circuit) is a synchronous, multi-master, multi-slave serial communication protocol. It is commonly used for connecting low-speed peripheral devices like sensors, EEPROMs, ADCs, DACs, and display modules to microcontrollers.

- SDA (Serial Data Line): This is the line used to carry the data between the master and the slave devices. The data is transmitted serially, meaning one bit is sent at a time.
- SCL (Serial Clock Line): This line carries the clock signal used to synchronize the data transfer between devices. The clock is generated by the master device.

4.7.2 FEATURES I2C MODULE

- Two-wire interface: The I2C protocol requires only two lines, making it simple and efficient for communication between devices.
- Multi-master and multi-slave capability: Multiple master and slave devices can be connected to the same I2C bus.

- Data transfer speed: I2C supports different communication speeds, including standard (100 kbps), fast (400 kbps), and high-speed mode (3. 4 Mbps).

4.7.3 PIN DESCRIPTION

Pin Name	Description
SDA (Serial Data Line)	This is the bidirectional data line that carries the data to and from the devices. It is used to send and receive data bits in a serial fashion. The line is open-drain, meaning it is pulled high through a resistor and can be driven low by devices.
SCL (Serial Clock Line)	This pin is used to provide the clock signal, which is generated by the master device. The clock signal synchronizes the data transfer between devices on the bus. It is also an open-drain line and is pulled high by an external resistor.
VCC	This pin is connected to the positive supply voltage, usually 3.3V or 5V, depending on the module and system.
GND	This is the ground pin that is connected to the system's ground.
ADD (Optional)	Some I2C modules have an address selection pin (ADD) that allows the user to configure the address of the device. This can typically be connected to GND or VCC to change the device's I2C address.

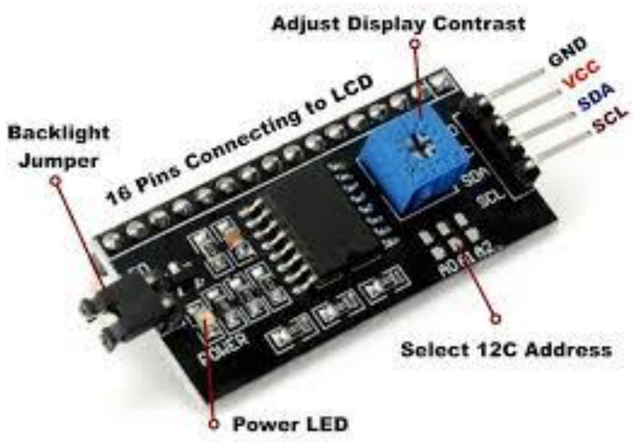

Figure 4.14 Pin Diagram of I2C Module

4.8 16X2 LCD DISPLAY

A 16x2 LCD display is a popular display module used in electronics projects. It consists of 16 columns and 2 rows, capable of displaying up to 32 characters. It uses a HD44780 driver to control the characters and is commonly used with microcontrollers like Arduino for displaying data. The display has a backlight for visibility in low light and features adjustable contrast. The LCD can be interfaced using parallel communication or I2C for fewer wiring connections. It is widely used in applications like sensor readings, status displays, and user interfaces in embedded systems.

Figure 4.15 16x2 LCD display

4.9 PIEZO BUZZER

A buzzer is an audio signaling device that emits sound when powered or activated by an electronic circuit. It operates by using an electromagnet or piezoelectric element that vibrates at a specific frequency when an electric current passes through it. Buzzers are widely used in a variety of applications, including alarms, timers, notifications, and indicators. They come in different types, such as active and passive buzzers, with active buzzers generating sound on their own when powered, while passive buzzers require an external signal or waveform to produce sound. Buzzers are commonly used to provide audible alerts or feedback in response to specific events, such as crossing a threshold value or triggering a sensor. For example, a buzzer can be used to alert users when a sensor detects an abnormal condition, like high gas levels, temperature, or motion. The sound produced by the buzzer can be continuous or intermittent, depending on the design, and it can be controlled using digital output pins from microcontrollers such as NodeMCU.

Figure 4.16 Piezo Buzzer

Figure 4.17 Terminals of Piezo Buzzer

4.10 RGB LED

An RGB LED (Red, Green, Blue Light Emitting Diode) is a type of LED that combines three separate LEDs (Red, Green, and Blue) into a single package, allowing it to produce a wide range of colors. By adjusting the intensity of each of the three primary colors, an RGB LED can create millions of color variations. This makes RGB LEDs highly versatile for use in displays, indicators, lighting systems, and decorative lighting. Commonly used in both commercial and consumer electronics, such as screens, lighting strips, and smart devices, they are ideal for applications that require customizable or dynamic lighting. RGB LEDs typically come in two varieties: common cathode and common anode. In common cathode RGB LEDs, the negative (cathode) terminal is shared by all three LEDs, and the positive terminals of each color are connected to separate control pins. In contrast, common anode RGB LEDs share a common positive (anode) terminal, with each color's negative terminal connected to a separate control pin. The color mixing is controlled by varying the voltage supplied to each LED channel, creating different shades based on the combination of red, green, and blue intensities.

Figure 4.18 RGB LED

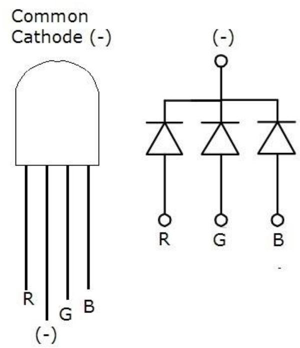

Figure 4.19 Pinout of Common Cathode RGB LED

TECHNICAL SPECIFICATIONS OF RGB LED

- Operating Voltage: Typically 3.0V to 3.5V for each individual LED color (Red, Green, and Blue).
- Forward Current: Typically 20mA per color channel.
- Total Power Consumption: Typically around 60mA when all colors are fully lit (at full brightness).
- Color Range: Can produce millions of colors by combining Red, Green, and Blue light in varying intensities.
- Luminous Intensity: Typically between 100-200 mcd per color at 20mA current, but this can vary by model.
- Viewing Angle: Usually between 120° to 160°, depending on the specific LED model.

4.11 B TYPE CABLE

A Type B cable typically refers to a USB Type B cable, commonly used for connecting larger devices, computers or other host devices. USB Type B cables are primarily designed for transmitting data and power over short to medium distances. Type B cables are often used for peripherals that require more power and a stable connection, distinguishing them from smaller cables like USB Type-A and Type-C used for mobile devices.

Figure 4.20 B Type Cable

4.12 JUMPER WIRES

Jumper wires are short electrical wires used to make connections between different points on a breadboard or between components in a circuit. They are typically used for prototyping and testing circuits, providing a flexible and easy way to create temporary connections without soldering. Jumper wires are often available in different lengths and come with either male or female connectors at both ends, depending on the required connection type. These wires are commonly used in Arduino and other microcontroller-based projects for quickly connecting sensors.

Figure 4.21 Jumper Wires

DESCRIPTION OF SOFTWARE REQUIREMENTS

4.13 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a powerful and user-friendly platform used for programming and controlling Arduino microcontrollers. Designed with simplicity in mind, the Arduino IDE is a cornerstone for hobbyists, students, and professionals looking to create innovative electronics projects. It provides an accessible gateway into embedded systems and hardware programming. Arduino IDE simplifies the process of uploading code to the microcontroller through a USB connection. It comes pre-configured with essential libraries that allow interaction with sensors, motors, displays, and other hardware. The IDE also supports additional libraries, which can be added to enhance project functionality. Key features of the Arduino IDE include serial communication for debugging, extensive community support, and compatibility with numerous Arduino boards, such as the Uno, Mega, and Nano. Its open-source nature encourages collaboration and sharing of ideas, making it a hub for innovation in electronics and programming.

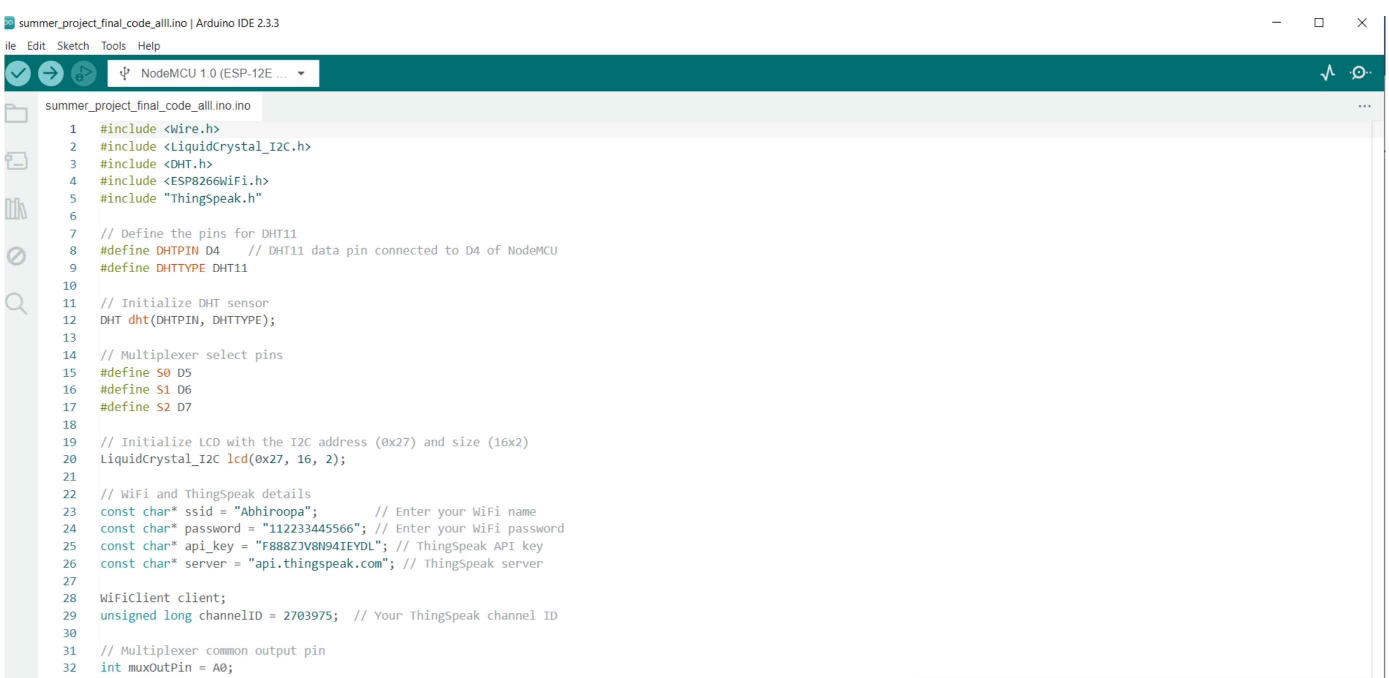

Figure 4.22 Arduino IDE Platform

4.14 THINGSPEAK

ThingSpeak is an Internet of Things (IoT) analytics platform and API service that enables the collection, analysis, and visualization of real-time data from IoT devices. It provides an easy and efficient way for users to connect sensors, microcontrollers, and other devices to the cloud for monitoring, control, and analysis. ThingSpeak is widely used in projects involving environmental monitoring. One of ThingSpeak's standout features is its data channels, where users can store sensor data. Each channel consists of up to eight fields for numeric or alphanumeric data, alongside metadata like timestamps and location coordinates. Data from devices such as NodeMCU can be sent to these channels using HTTP or MQTT protocols. ThingSpeak includes built-in tools for data analysis and visualization. Users can generate real-time graphs, charts, and heatmaps directly on the platform, offering instant insights into sensor performance or environmental conditions. Additionally, the integration with MATLAB provides powerful data analysis capabilities, enabling advanced computations, machine learning, and predictive modeling.

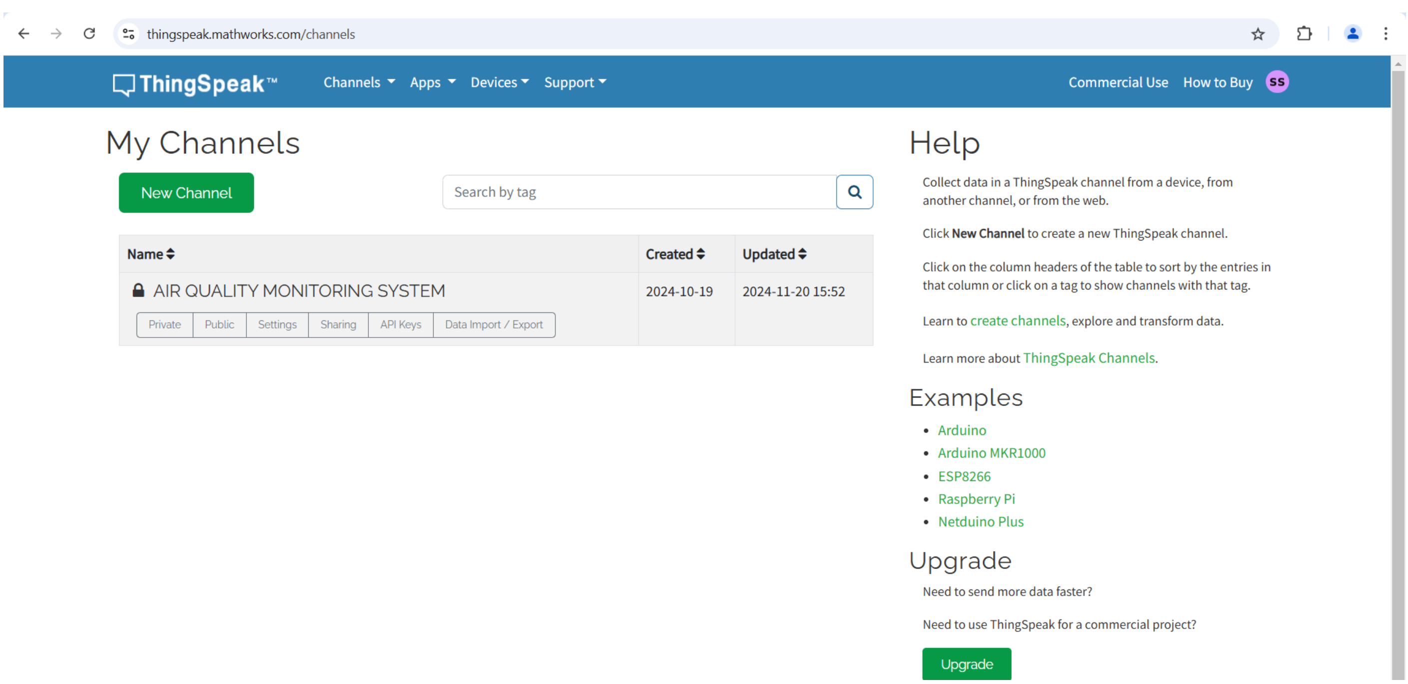

Figure 4.23 ThingSpeak Platform

CHAPTER-5

5.1 DESIGN AND IMPLEMENTATION

The air quality monitoring system integrates sensors, a microcontroller, and cloud-based visualization tools to measure temperature, humidity, and pollutant levels such as CO and PM2.5. The MQ135 sensor detects a range of gases like ammonia and carbon dioxide, MQ7 sensor is focused on carbon monoxide (CO), and the PM2.5 sensor measures fine particulate matter. These sensors are interfaced with the NodeMCU (ESP8266) microcontroller, which collects and processes data from the sensors. The NodeMCU is responsible for calculating the Air Quality Index (AQI) for each pollutant, with emphasis on CO and PM2.5 due to their higher health impact. The calculated AQI values are displayed on a 16x2 LCD and also uploaded to ThingSpeak, a cloud platform, for real-time data visualization.

The NodeMCU microcontroller, chosen for its built-in Wi-Fi capabilities, reads data from the sensors and processes it to calculate AQI values using standard formulas. The system calculates the AQI for CO and PM2.5, considering the sensor readings and applying predefined breakpoints to convert raw values into AQI values. These AQI values are displayed on the LCD for local monitoring, while the data is also sent to ThingSpeak for remote tracking. ThingSpeak visualizes the data in real-time graphs, showing trends in gas concentrations and AQI over time.

The design also incorporates key considerations for sensor integration and data accuracy. The MQ135 and MQ7 sensors provide analog output signals, which are read by the NodeMCU's analog-to-digital converter (ADC) to determine gas concentrations in the air. The PM2.5 sensor uses a digital signal to detect fine particles, offering an additional layer of data crucial for assessing air quality. The NodeMCU continuously collects data from these sensors, calculates the AQI for each pollutant, and determines the overall AQI by considering the highest value from the sensors. This value is then displayed on the 16x2 LCD in real time, providing immediate feedback to the user about the air quality in the environment.

For the cloud-based data visualization, the system leverages ThingSpeak, where data from the NodeMCU is uploaded via Wi-Fi. The integration of ThingSpeak allows for remote monitoring and provides users with graphical representations of the air quality data over time. This feature enables long-term tracking and comparison of air quality trends, which is especially useful in urban environments where air quality can fluctuate due to various external factors. Additionally, the system can be extended with more advanced features, such as sending alerts when AQI thresholds are exceeded, or integrating the system with smart home devices to optimize ventilation and air filtration. While the current setup is effective for monitoring CO and PM2.5 levels, future iterations may include additional sensors for a more comprehensive environmental analysis.

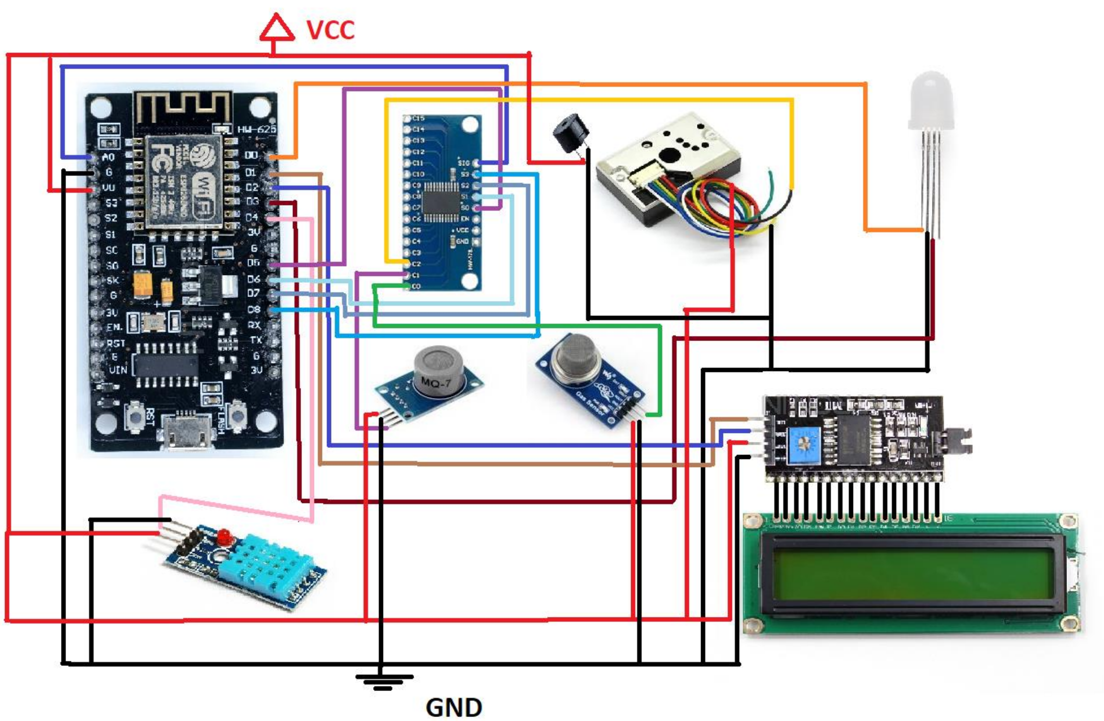

Figure 5.1 Circuit Diagram

5.2 WORKING METHODOLOGY

The working principle of the air quality monitoring system begins with the acquisition of environmental data through three primary sensors: MQ135, MQ7, and PM2.5. These sensors are interfaced with an analog multiplexer, which allows the NodeMCU to sequentially read data from each sensor using a single input pin, thus minimizing the number of GPIO pins required. The multiplexer switches between the sensors based on control signals from the NodeMCU, allowing the system to read and process the analog signals from the MQ135 and MQ7 sensors. The NodeMCU then converts these analog signals into corresponding gas concentrations (in ppm for CO and $\mu g/m^3$ for PM2.5) using its analog-to-digital converter (ADC). The data from these sensors is used to calculate the Air Quality Index (AQI) for CO and PM2.5 based on standard formulas, and the highest AQI value is chosen as the overall AQI.

Additionally, the system uses an RGB LED to visually indicate the air quality based on the AQI. The RGB LED is programmed to light up in different colors: green for good air quality (AQI < 250), blue for moderate air quality (AQI between 250-300), and red for poor air quality (AQI > 300). To alert the user of potentially harmful air quality levels, a piezo buzzer is included. The buzzer is programmed to sound when the AQI exceeds a certain threshold, providing an audible warning of poor air quality. The entire system is controlled by the NodeMCU, which also uploads the AQI data to the ThingSpeak cloud platform for remote monitoring and analysis. This integration ensures both local and cloud-based tracking of air quality in real-time.

5.3 TECHNICAL SPECIFICATION FOR THIS PROJECT

1. Microcontroller:

- NodeMCU (ESP8266)
- Built-in Wi-Fi for cloud connectivity (ThingSpeak)
- 1MB Flash memory
- 32-bit processor, 80 MHz clock speed

2. Sensors:

- MQ135:
- Gas detection for ammonia, $CO2$, ethanol, and other gases
- Voltage output (analog)
- Detection range: 10-1000 ppm ($CO2$)
- Operating voltage: 5V
- Response time: < 60 seconds

3. MQ7:

- Carbon monoxide (CO) gas detection
- Voltage output (analog)
- Detection range: 20-1000 ppm (CO)
- Operating voltage: 5V
- Preheating time: 60 seconds

4. PM2.5 Sensor:

- Measures particulate matter smaller than 2.5 micrometers
- Digital output
- Detection range: 0-500 $\mu g/m^3$
- Operating voltage: 5V

5. Display:

- 16x2 LCD Display (I2C):
- Displays temperature, humidity, AQI, and gas concentrations
- I2C interface for reduced pin usage

6. Analog Multiplexer:

- 74HC4051 8-channel analog multiplexer
- Allows sequential reading from multiple analog sensors (MQ135, MQ7)

7. RGB LED:

- Common anode RGB LED
- Controlled by NodeMCU to indicate AQI status
- Green (AQI < 250), Blue (AQI 250-300), Red (AQI > 300)

8. Buzzer:

- Provides audible alert when AQI exceeds a threshold
- 5V operating voltage
- Continuous sound or intermittent based on AQI level

9. Power Supply:

- 5V DC from USB or external power supply
- Low power consumption when idle

10. Cloud Integration:

- ThingSpeak cloud platform for real-time data monitoring
- Data upload every 1 minute via Wi-Fi
- Graphical visualization of gas concentrations and AQI

11.Environmental Conditions:

- Temperature range: 0°C to 50°C
- Humidity range: 20% to 90% RH

CHAPTER-6

6.1 PROS OF THIS PROJECT

- Real-time Monitoring: Provides continuous tracking of air quality, enabling users to make timely decisions.
- Cloud Integration: Uses ThingSpeak for remote monitoring and graphical visualization of air quality data.
- Affordable: Low-cost hardware components like NodeMCU, sensors, and LCD make it budget-friendly.
- Scalable: The system can be easily expanded to include more sensors or additional features (e.g., smart alerts, data logging).
- Energy Efficient: The system consumes low power, especially in idle mode, which makes it suitable for long-term deployment.
- Multiple Pollutant Detection: Measures a variety of pollutants (CO, PM2.5) for more comprehensive air quality monitoring.
- User-Friendly Interface: Local display via LCD and visual feedback through RGB LED makes the system easy to understand.
- Audible Alerts: The buzzer provides an immediate warning when air quality deteriorates.
- Wi-Fi Connectivity: Allows data to be shared and monitored remotely through cloud platforms like ThingSpeak.

6.2 CONS OF THIS PROJECT

- Sensor Limitations: Sensors may not be highly accurate under varying environmental conditions without proper calibration.
- Calibration Requirement: Sensors need periodic calibration to ensure accurate readings, which can be time-consuming.

- Dependence on Wi-Fi: The system relies on a stable Wi-Fi connection for cloud data upload and remote monitoring.
- Limited Range of Detection: Sensors like the MQ135 and MQ7 have limited detection ranges and may not cover all types of pollutants.
- Maintenance: Dust or environmental contamination can affect sensor performance over time, requiring occasional cleaning or recalibration.
- Temperature Sensitivity: Sensors may show varying results in different temperature and humidity conditions unless compensated for in software

6.3 APPLICATION OF THIS PROJECT

- Smart Homes: Used for monitoring indoor air quality and ensuring a healthy environment, especially in homes with vulnerable individuals.
- Urban Monitoring: Helps in measuring air pollution in cities, contributing to smart city initiatives by providing real-time pollution data.
- Industrial Monitoring: Suitable for factories and industries to monitor hazardous gases and particulate matter, ensuring a safe work environment.
- Environmental Research: Used by researchers to collect data on air quality and its impact on health in different geographical regions.
- Agriculture: Helps in monitoring the effects of air pollution on crops and soil, contributing to more sustainable agricultural practices.
- Public Health: Provides important data for public health agencies to assess and improve air quality standards, particularly in high-traffic areas or regions with significant pollution.
- Schools and Offices: Can be installed in schools and offices to ensure air quality remains within safe limits, especially during the cold months when ventilation is poor.
- Transport and Vehicles: Used in monitoring the air quality in vehicles, particularly for detecting harmful gases like CO, ensuring a healthier commute.

CHAPTER-7

RESULTS AND DISCUSSIONS

Here are the few results we have obtained:

7.1.1 HARDWARE OUTPUT

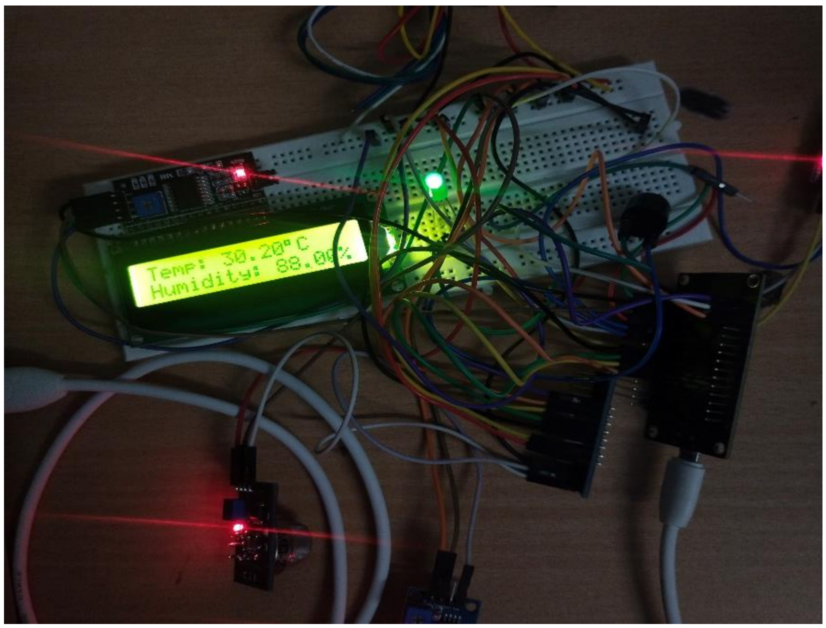

Figure 7.1 Temperature Humidity Measurement

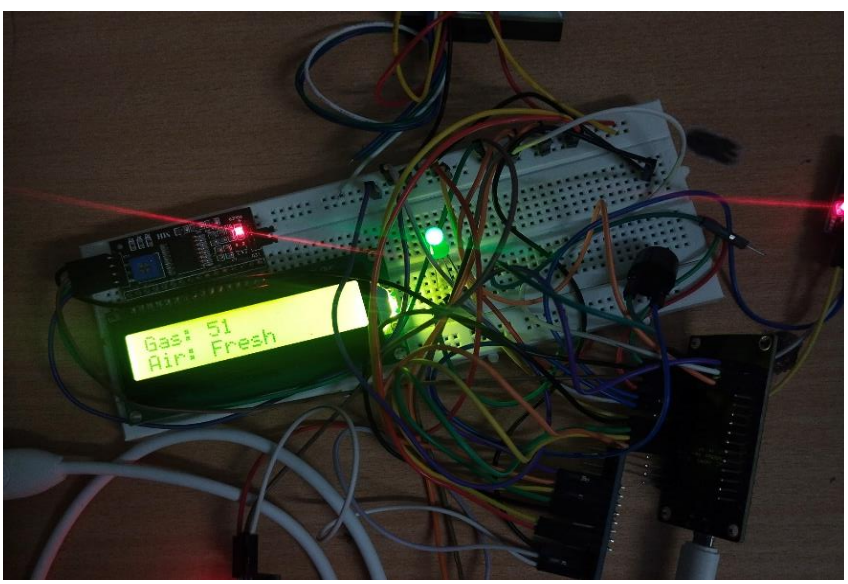

Figure 7.2 Fresh Air Detection

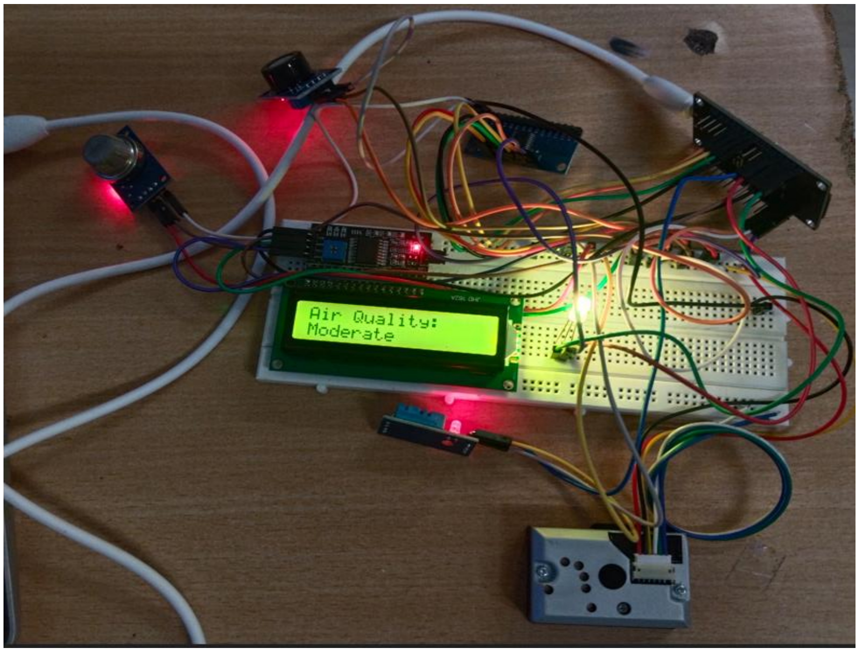

Figure 7.3 Moderate Air Detection

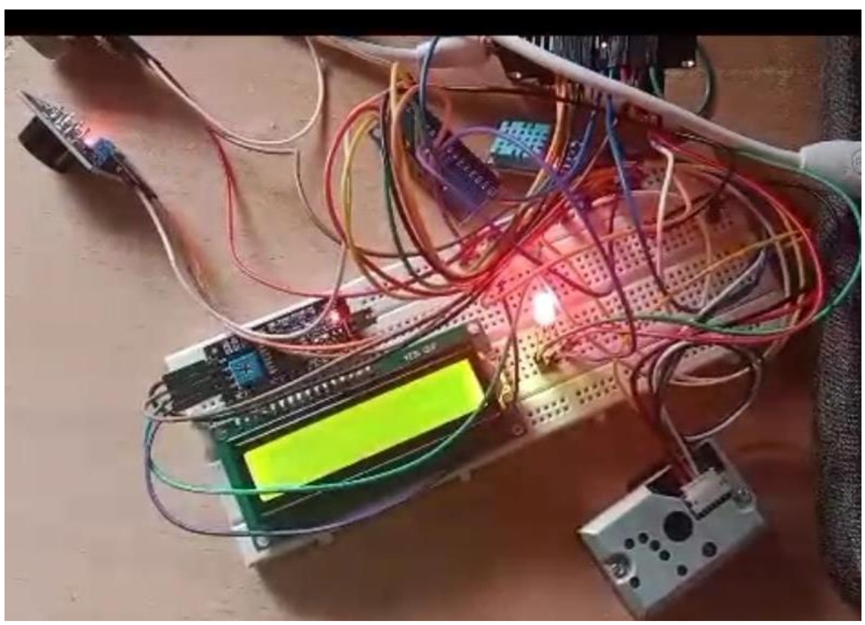

Figure 7.4 Hazardous Air Detection

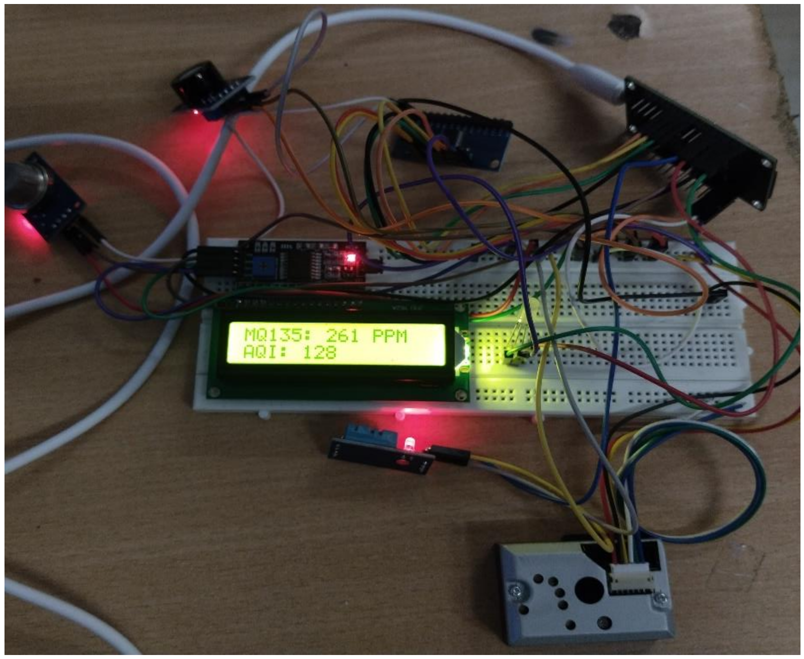

Figure 7.5 MQ135 AQI Indication

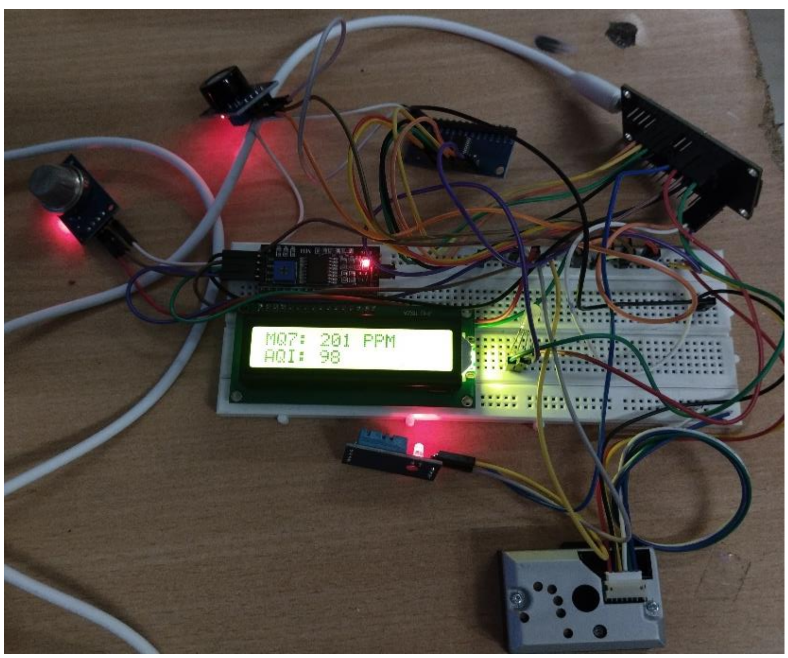

Figure 7.6 MQ7 AQI Indication

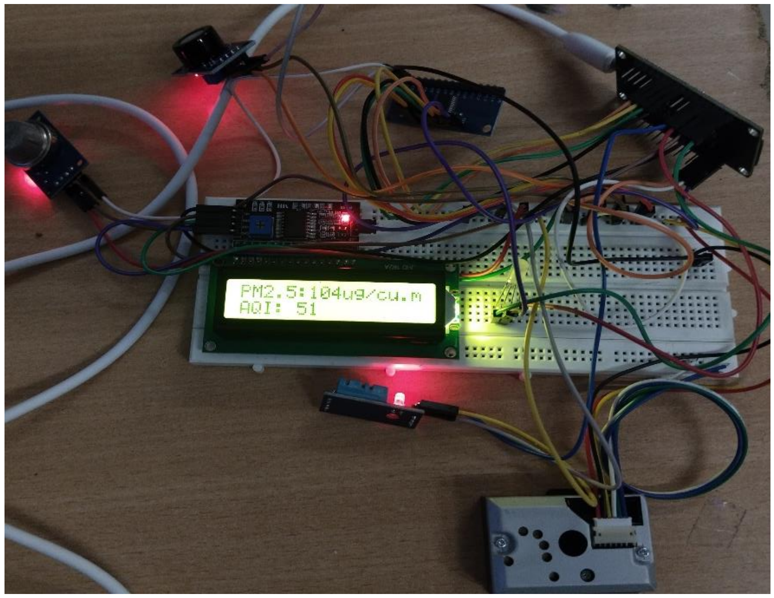

Figure 7.7 PM2.5 AQI Indication

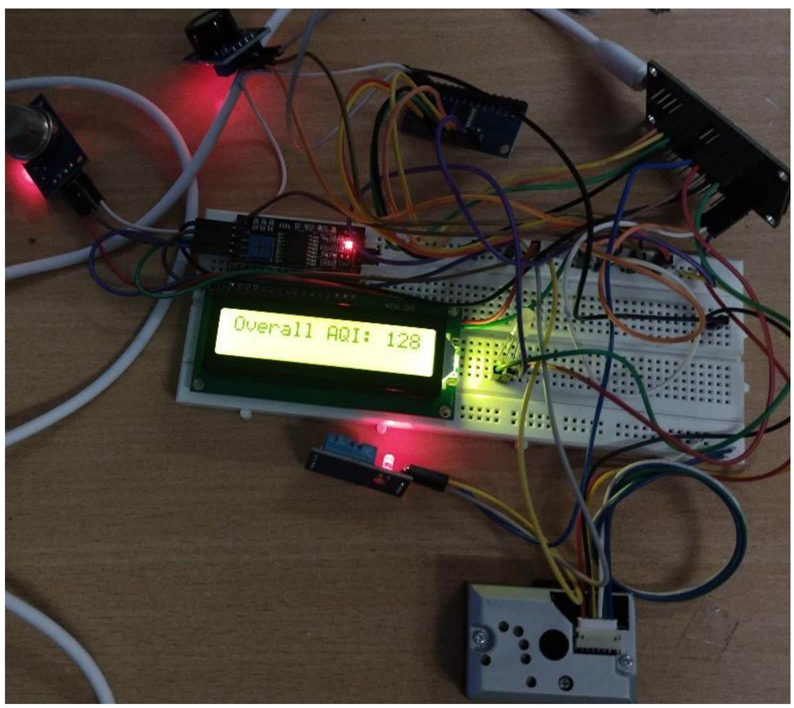

Figure 7.8 Overall AQI Indication

7.1.2 THINGSPEAK OUTPUT

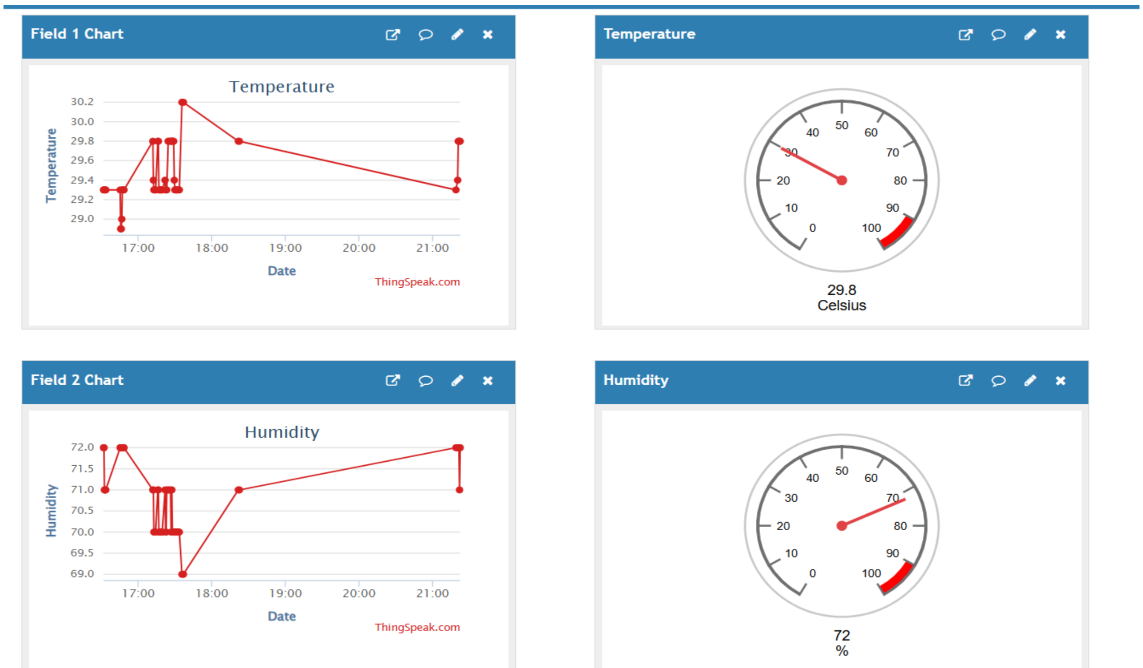

Figure 7.9 Temperature and humidity

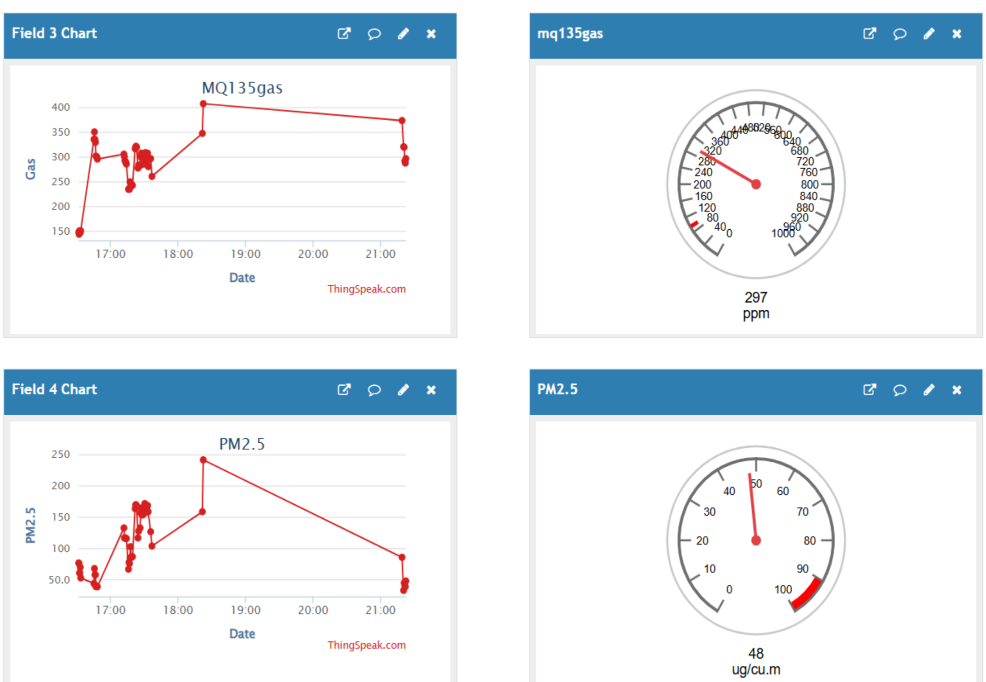

Figure 7.10 MQ135 and PM2.5

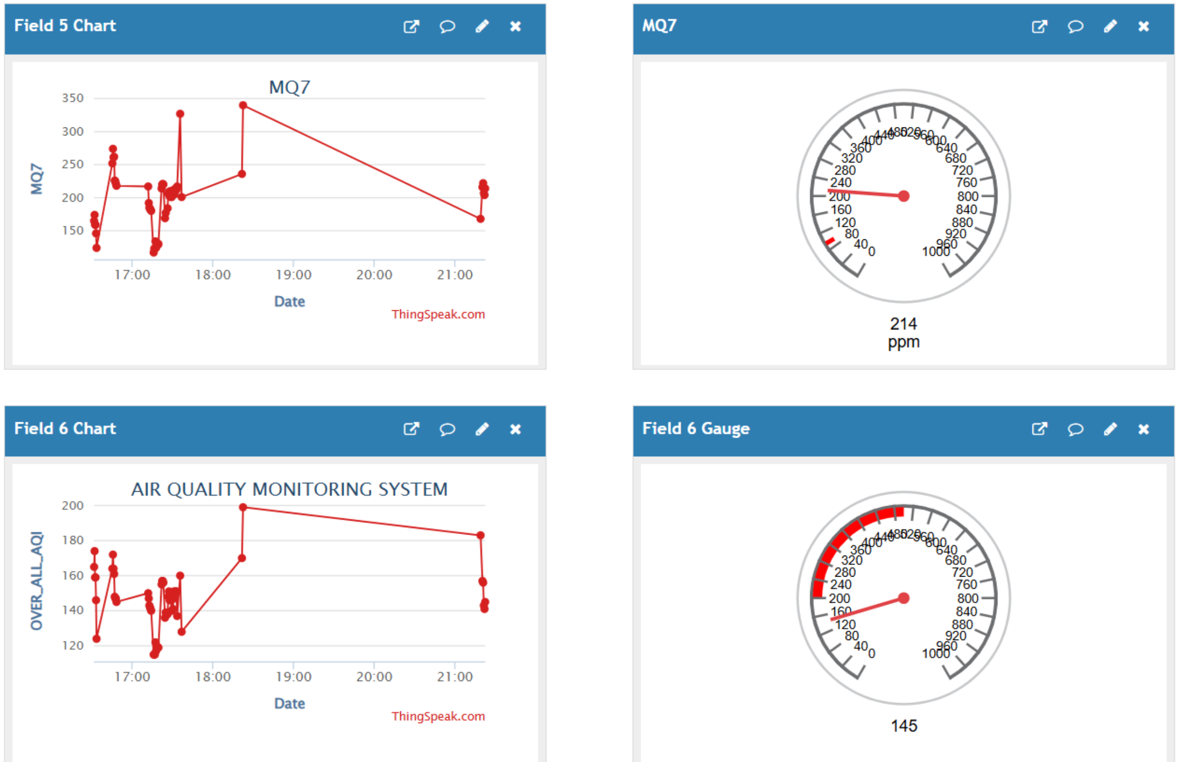

Figure 7.11 MQ7 and overall AQI

7.2 DISCUSSIONS

The air quality monitoring system offers an effective solution for real-time tracking of environmental pollutants, with a focus on CO and PM2.5 levels, which have significant health implications. By using sensors like the MQ135, MQ7, and PM2.5, along with an analog multiplexer to streamline sensor readings, the system provides accurate and continuous data on air quality. The integration of a NodeMCU for Wi-Fi connectivity allows for seamless data upload to ThingSpeak, enabling remote monitoring through cloud-based graphs. Despite its low cost and scalability, the system does have limitations, such as sensor accuracy, calibration needs, and Wi-Fi dependency. However, its real-time local display through an LCD, visual feedback via RGB LEDs, and audible alerts via a buzzer make it user-friendly. The system's applications range from smart homes and industrial environments to urban pollution monitoring, with potential to expand into smart city frameworks.

Calculate the Overall Air Quality Index (AQI) using raw sensor values:

- Gather Raw Sensor Data: We need to take the raw sensor readings from each of the three sensors: MQ135, MQ7, and PM2.5. These values are typically read using an analog-to-digital converter (ADC), and they range from 0 to 1023.

- Convert Raw Sensor Values to AQI: Now, you need to convert these raw values into AQI values (ranging from 0 to 500). The AQI is often calculated using a linear mapping function based on the sensor's range. In our case, we are assuming a simple mapping where:

  •Raw value of 0 corresponds to AQI 0 (best air quality).

  •Raw value of 1023 corresponds to AQI 500 (worst air quality).

- Calculate the AQI by using the formula: AQI=(Raw value/1023)×500

- Determine Overall AQI: To calculate the overall AQI, we simply select the highest AQI from the individual sensor AQI values because the worst air quality is represented by the highest AQI. The overall AQI is the maximum of these three values.

Air Quality Index (AQI) Values	Levels of Health Concern	Colors
When the AQI is in this range:	*...air quality conditions are:*	*...as symbolized by this color:*
0 to 50	Good	Green
51 to 100	Moderate	Yellow
101 to 150	Unhealthy for Sensitive Groups	Orange
151 to 200	Unhealthy	Red
201 to 300	Very Unhealthy	Purple
301 to 500	Hazardous	Maroon

Figure 7.12 Air Quality Index Chart

CHAPTER-8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

In conclusion, this air quality monitoring system successfully integrates multiple sensors to measure key pollutants such as CO, PM2.5, and other gases, providing real-time data for users to monitor air quality. The system's ability to calculate and display the Air Quality Index (AQI), along with visual indicators using an RGB LED and audible alerts through a buzzer, offers an effective way to raise awareness of environmental conditions. By leveraging NodeMCU's Wi-Fi capabilities, the system ensures continuous data upload to the ThingSpeak cloud platform, enabling remote monitoring and analysis. While the system has proven effective for basic air quality monitoring, future improvements in sensor calibration and power efficiency, along with the potential integration of additional sensors or control systems, will further enhance its accuracy and usability for broader applications in both domestic and industrial settings.

8.2FUTURE WORK

Future work for this air quality monitoring system can focus on enhancing sensor accuracy and calibration to improve the reliability of measurements across varying environmental conditions. Additionally, integrating more advanced sensors to detect a wider range of pollutants, such as nitrogen dioxide (NO2) or sulfur dioxide (SO2), could provide a more comprehensive assessment of air quality. Improving the power efficiency of the system would enable longer deployment durations, especially for outdoor use. Moreover, incorporating machine learning algorithms to predict air quality trends based on historical data could provide advanced insights, while adding automation features like controlling air purifiers or ventilation systems based on the AQI could further contribute to improving indoor air quality.

CHAPTER 9

REFERENCES

S. Pawar, S. Kelkar, N. Khire, T. Khairnar and M. Kharabe, "AQI Monitoring and Predicting System," 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2023

R. K. Jha, "Air Quality Sensing and Reporting System Using IoT," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020

A. N, A. C. S, S. P and T. V, "Air Quality Monitoring System," 2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS), Chennai, India, 2023

S. R. Enigella and H. Shahnasser, "Real Time Air Quality Monitoring," 2018 10th International Conference on Knowledge and Smart Technology (KST), Chiang Mai, Thailand, 2018.